IDA

Joint
Advanced
Warfighting
Program

INSTITUTE FOR DEFENSE ANALYSES

MS&G, When Worlds Collide: A Primer for Potential

Brian Williams
Bob Bates
Tara McGovern
Terry Heuring
Gordon Walton

AS AMENDED

July 2006
IDA Document D-3335
Log: H 06-001567/1

CLEARED
For Open Publication
DEC 27 2006 5
AS AMENDED
Office of Security Review
Department of Defense

07-S-0510

IDA

Joint
Advanced
Warfighting
Program

INSTITUTE FOR DEFENSE ANALYSES

MS&G, When Worlds Collide: A Primer for Potential

Brian Williams
Bob Bates
Tara McGovern
Terry Heuring
Gordon Walton

July 2006
IDA Document D-3335
Log: H 06-001567/1

This work was conducted under contracts DASW01 04 C 0003 and W74V8H 05 C 0042, Task AK-8-246201 for the Defense Modeling and Simulation Office. The publication of this IDA document does not indicate endorsement by the Department of Defense, nor should the contents be construed as reflecting the official position of that Agency.

© 2006 Institute for Defense Analyses, 4850 Mark Center Drive, Alexandria, Virginia 22311-1882 • (703) 845-2000.

This material may be reproduced by or for the U.S. Government pursuant to the copyright license under the clause at DFARS 252.227-7013 (NOV 95).

INSTITUTE FOR DEFENSE ANALYSES
Joint Advanced Warfighting Program

IDA Document D-3335

MS&G, When Worlds Collide: A Primer for Potential

Brian Williams
Bob Bates
Tara McGovern
Terry Heuring
Gordon Walton

Preface

Approximately two-and-a-half years ago, the Defense Modeling and Simulation Office (DMSO) sought to tap new thinking about problems facing the Department of Defense (DoD), especially modeling and simulation (M&S) tools and processes for addressing them. The dynamic nature of contemporary warfare and emerging roles for the military were redefining the M&S requirements. A diffuse adversary set, a dynamic pace of change, the complexity of human interaction, and the simple plethora of "bad guys" highlighted the need for new tools, processes, and thinking.

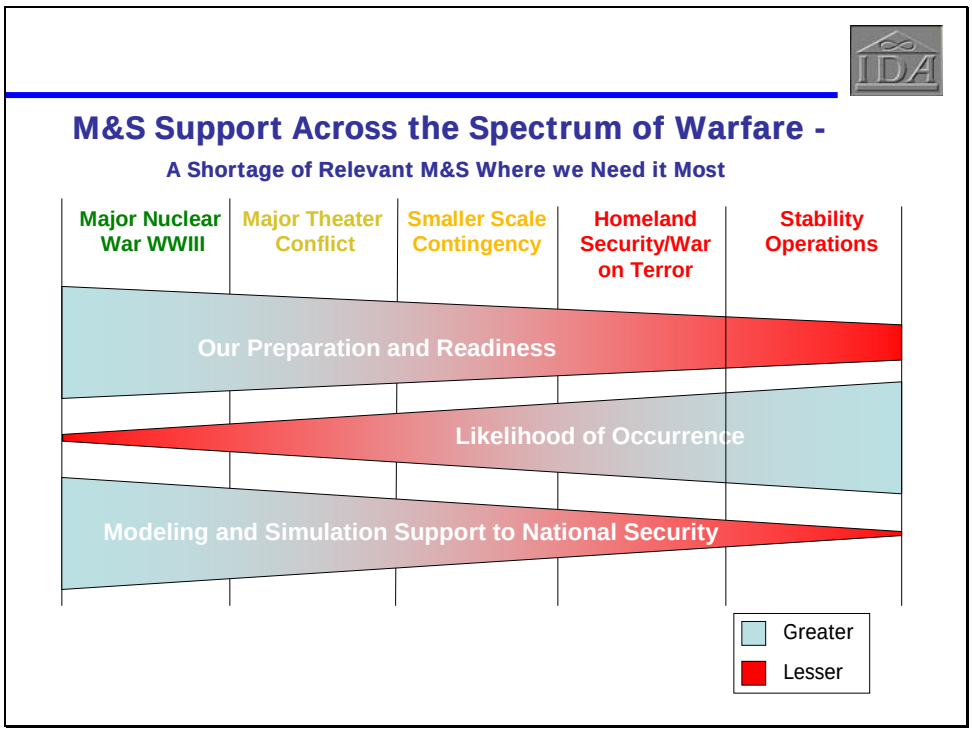


Figure P-1. M&S Support Across the Spectrum of Warfare

To that end, the Institute of Defense Analyses (IDA) convened on DMSO's behalf a workshop exploring the future of M&S. Participants came from across DoD and the government sector with approximately 35 participants generating over 700 concepts and ideas.

Those ideas were aggregated and "threaded," and an interesting and dominant thread emerged: both the problem and solution space converged on exploiting Massive Multiplayer Online Games (MMOGs) and commercial gaming in general. Gaming was seen as potentially providing insight into emergent behavior, new business models, and new collaborative and communication techniques in addition to the more "accepted" use as a training tool. Furthermore, gaming technology, through the simple power of its market size, had become a leading driver in the computer, graphics, and networking industries.

As a result, DMSO, through IDA, launched an effort to better understand and capitalize on the capabilities of the people, processes, and technology of the commercial gaming marketplace. The effort contributed to seriously considering commercial-based games as a tool in the DoD problem space.

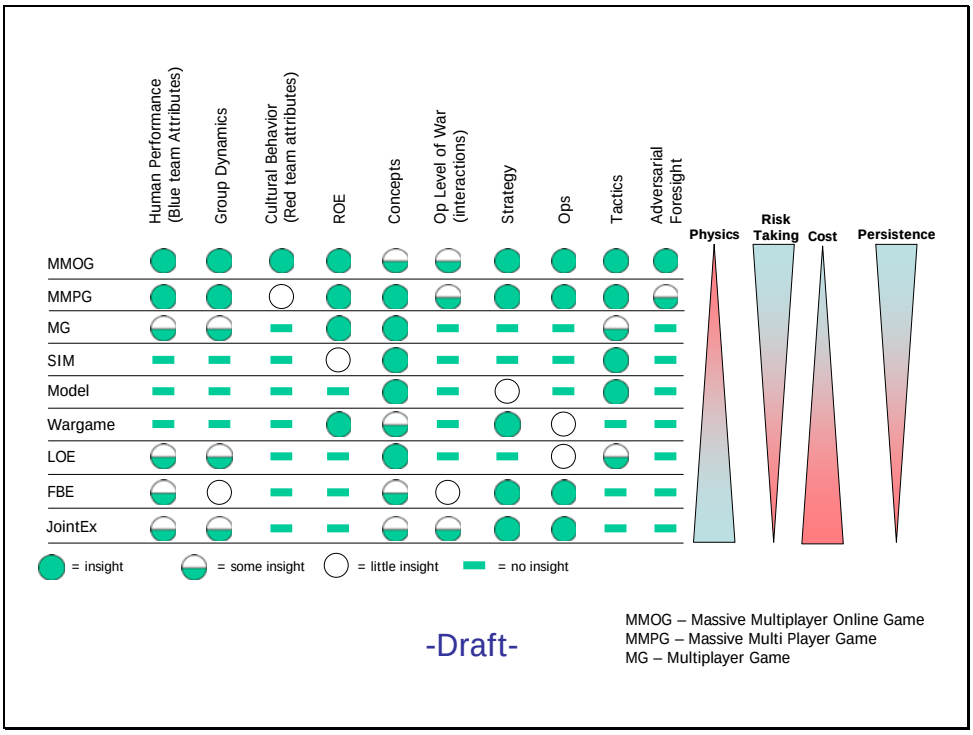


Figure P-2. Spectrum of Model and Game Application

During the two-year period that followed, much has been learned through engaging a broad range of DoD-sponsored gaming efforts, conferences, advisory groups, and teleconferences focused specifically on creating this paper.

What became apparent was that these two communities of interest (DoD M&S and Commercial Gaming) evolved very differently due to customers' needs

and market dynamics, resulting in philosophical and operational differences that must be understood and bridged before the communities can work effectively together. DMSO asked if we might capture some of that knowledge and pass it on to DoD and the gaming communities alike—a mixture of Primer and Roadmap—something to familiarize both parties with the issues and nature of the other's "business."

Acknowledgments

We can not completely acknowledge all of those who have directly and/or indirectly contributed to this work. Perhaps the most amazing part of this work has been to watch a community of interest begin to build around a vision—often diffused or scattered, but a community nonetheless—that appreciated the emergence of this interactive media onto the communication landscape. The people, through direct engagement or encouragement, helping to shape the interaction of commercial-based games and DoD include: Jim Blank, JFCOM/J9; Tony Cerri, JFCOM/J9; Ralph Chatham, DARPA; Pat Conway, USA CASCOM; Neale Cosby; Doug Failor, JFCOM/J9; Rob Gehorsam, Forterra; CAPT Kent Gritton, USN; Keith Halper, Kuma Games; Dan Kaufman, DARPA; Paul Kozemchak, DARPA; Del Lunceford, Total Immersion Software; Perry McDonough, Naval Post Graduate School; Alex Noric, IGDA; Benjamin Pecheux, Mitretek; Sue Numrich, IDA; Cory Ondrejka, Second Life; Peter Perla, Center for Naval Analysis; Ben Sawyer, Digitalmill; Steve Swenson, Ocean State Technology; Jack Thorpe; Doug Whatley, BreakAway; and Rich Vogel.

In addition, we recognize the individuals who persevered through reading this information and were kind enough to give us their editorial comments and recommendations: Vince Roske and Martin Stytz, both IDA. Furthermore, we want to acknowledge Kenn Atkinson and Charles White and the DMSO management for its patience as other demands in support of the warfighting effort continued to reprioritize the work of the IDA writers, extending timelines, and delaying the completion of this work.

Table of Contents

Executive Summary ES-1
Findings and Recommendations ES-1
Current Efforts and Research ES-1
Business Challenges ES-3
Product Development ES-4
Technical Challenges ES-6
The Future ES-8
A Government-Chartered Non-Profit "Marketplace" Entity ES-10

The Commercial World

I. Introduction 1

II. A Short History of Commercial Games 3
Introduction 3
Timeline 4

III. Commercial Game Genres 9
Action 9
Adventure 10
Role-Playing Games (RPGs) 10
Strategy 11
Simulations 12
Sports Games 13
Platform Games 13
Fighting Games 14
Casual Games 14
God Games 15
Educational Games 15
Puzzle Games 15
Online Games 15

IV. Inside the Game Studio 17
Vision 17
Production 18
The External Producer 19
The Internal Producer 20
Assistant and Associate Producers 21
Design 22
The Game Designer 22
The Level Designer 23
The Writer 23
Programming 24

The Tech Lead....24
Programmers....25
Art....26
The Art Lead....26
Artists....27
Testing....28
The Test Lead....28
Testers....30

V. Outside the Game Studio....31
Voice....31
Music....32
Sound Effects....33
Video....35
Motion Capture....35
Language Localization....36
The Manual....38
Artwork....39
Programming....40
Customer Service....40
Quality Assurance....41
Legal Issues (Getting the Rights)....41

VI. Commercial Game Project Management....43
Lifecycle Models....43
The Waterfall....43
The Modified Waterfall....44
Iterative Prototyping (Agile Development)....44
Understanding Design Goals....47
Planning and Scheduling....47
Scope....48
External Pressures....49
Padding....49
Altered Requirements....49
Developer Optimism....50
Waiting for the Prototype....50
Avoiding Problems....51
Classic Mistakes....51
Recovery....53
Ineffective Strategies....53
Effective Strategies....54

VII. How Commercial Games are Marketed, Distributed and Sold....57
Hidden Pressures on Commercial Teams....58
The Publishing Team....60
Public Relations (PR)....60
Marketing....60
Sales....61

The Community Team 63
The Networking Team 64
The Customer Service Team 65
Promotional Tools 66
Demos 66
Trial Versions 68
Interviews 68
Screenshots 68
Sell Sheets 69
The Contract 69
Advances 70
Royalties 71
Reserve Against Returns 71
Milestones and Deliverables 72
Intellectual Property Rights 72
Proprietary Technology 73
Term 73
Termination 73
Confidentiality 73
Ancillary Revenues 73
"Indie" Development 74

VIII. Single-Player Game Engines 75

IX. Massively Multiplayer Game Engines 79
What is an MMOG Engine? 79
Why acquire an engine to make an MMOG? 79
Open Source vs. Proprietary Engines 80
Commercially Available Engine Information 80

X. Standards and the Game Industry 81
Background 81
Current State 83
Middleware and Engines 87

XI. The Future of Mainstream Games 89
Important Trends 89
Worldwide Growth of the Internet 89
Increased Broadband Penetration and Voice Over Internet 91
Digital Distribution 92
Next Generation Console Development 93
Mobile connectivity 94
Globalization 96
Micro-Transactions 97
The Next Big Thing: Emotion 98

The Government World

I. Introduction 103

II. A Short History of DoD Modeling & Simulation 105
Early Modeling, Simulation and War Gaming 105
The Post-World War II World 106
Recent Trends in Modeling and Simulation 107

III. DoD M&S Genres 109
Training 109
Analysis 110
Acquisitions 111
Planning 113
Testing 113
Research and Experimentation 113

IV. Contracting with the DoD 115
DoD Organization 115
Potential MS&G Users 118
Training 118
Analysis 118
Acquisition 119
Budgeting and Acquisition Process 119
Authorization and Appropriation 119
Continuing Resolutions 120
Funding Models 120
Procurement Procedures 120
Request for Proposal (RFP) 120
Broad Area Announcement (BAA) 121
Small Business Innovative Research (SBIR) & Small Business Technology Transfer (STTR) programs 121
Multidisciplinary Research Program of the University Research Initiative (MURI) 122
Unsolicited proposals 122
Types of Contracts 123

V. Project Development in the DoD 125
The "Waterfall" Development Model and the Defense Acquisition Framework125
Recent Reforms 128
Challenges 129
Bureaucracy 129
Semantic differences 130
"Ilities," Standards and VV&A 130
Cultural Differences 131
Changes in Personnel 132

VI. Game Engines and the DoD 133
Government-sponsored Engines 133
Evaluating Commercial Engines 135

Conclusion 140

VII. DoD Standards and M&S 141
DSP and the "Perry Memorandum" 142
The Defense Modeling and Simulation Office (DMSO) 142
Simulation Interoperability Standards Organization (SISO) 143
The Advanced Distributed Learning (ADL) Initiative and SCORM® 144
Other Organizations 145
Speed of Technology vs. Speed of Standards 145
When NOT to Standardize 146

VIII. DoD Specific Requirements 149
Acquisition and Accounting Regulations 149
Intellectual Property Rights 149
Security 151
Clearances 151
Malicious software and aggregated information 152
Verification, Validation and Accreditation (VV&A) 153
The "Ilities" 155
Data 156
Conclusion 157

When Worlds Collide

I. Introduction 161

II. Current Efforts, Research & Roadblocks 163
Current Efforts 163
Areas of Interest 163
Education 164
Representative Sample of Companies in the Field 169
"Persuasion" Games 171
Research About the Effectiveness of Game Technologies 172
Roadblocks & Recommendations 176

III. Business Challenges 179
Finding Each Other 179
Contracts 180
Long-term Funding Commitments 181
Project Development 182
Approval and Payment Cycles 182
Compliance 183
Legacy of the Primes 183
Intellectual Property 184
Recommendations 185

IV. Technology 189
Rapidly Advancing Commercial Hardware 189
Console vs. PC Gaming 190
Leveraging Game Industry Technical Methods 191

Resolving Different Approaches to Optimization 192
MMOG Technologies and Network-Centric Warfare 193
Engines 194
Standards 196
Legacy Systems 197
Recommendations 198

V. The Future 199
Games Will Play a Large Role in the Military 199
Game Companies Will Become More Interested in Serious Games 199
DoD Will Need More Agile M&S 200
Commercial Games will Continue to Innovate, but 200
Government won't Continue to Drive Innovation and Standards 201
Offshoring Implications 202
Globalization & Open Sourcing 202
Games as Ambassadors 202
Democratization of Development 203
Keeping Pace with Foreign Governments 203
Testing Virtually 204
Evolution of Collaborative Virtual Spaces 205
An Innate Videogame Military Generation 206
Implementing Usability Research 207
New Opportunities 207
Collaborating Across Borders 208
Recommendations 209

Appendices

Appendix A: Acronyms & Abbreviations A-1

Appendix B: References B-1

Appendix C: Bibliography C-1

Appendix D: Glossary D-1

Appendix E: MMOG Engine Information E-1

List of Figures & Tables

Figure P-1. M&S Support Across the Spectrum of Warfare iii
Figure P-2. Spectrum of Model and Game Application iv
Figure 1. The Long Tail 93
Figure 2. Organization of the Department of Defense 116
Figure 3. Organization of the Office of the Secretary of Defense 117
Figure 4. Defense Acquisition Management Framework 126
Figure 5. Evolutionary Requirements and Acquisition Process 128
Figure 6. Relationship of Working Group Size to Standards Development Cost ... 146
Figure 7. Research and Development Funding by Source, 1953–2003 150

Table 1. Well-Known Commercially Available Game Engines 80
Table 2. World Internet Usage and Population Statistics 90
Table 3. Top 15 Countries in Broadband Subscribers (Year-End 2005) 91
Table 4. Category Share of Revenue and Gender Revenue Share Breakdown (US) 95
Table 5. Engine Capabilities 136
Table 6. Current DoD-Related Projects 170
Table 7. Other Games Relevant to the Military 171
Table 8. Single-precision Dens Matrix Multiplication (GEMM)
for various processors 190
Table E-1. MMOG Engines E-2
Table E-2. MMOG Features E-4

Executive Summary

Findings and Recommendations

Serious games already play an important role in the world of DoD M&S. They will become even more important in the future as battlespaces become more dynamic and adversaries more adaptive. But adopting serious games is hampered by a number of factors, including (1) the poor dissemination of academic research validating where games are appropriate and effective, (2) business and technical challenges that make it difficult for government and small game companies to work together, and most importantly, (3) the lack of a dynamic and sustainable serious games marketplace.

The most important and far-reaching recommendation of this report is that **DoD should create a government-chartered non-profit entity to act as an ombudsman, clearinghouse, and facilitator between government and industry; and that this entity be empowered to create a marketplace that:**

- **Helps customers find suppliers,**
- **Creates mechanisms to smooth financial transactions, and**
- **Encourages speedy delivery of creative products to DoD entities that need them.**

We discuss possible roles for this proposed organization more completely at the end of this Summary.

Current Efforts and Research

Soldiers entering today's military have grown up playing videogames. Their approach to group interaction and problem-solving is different from the generations preceding them. The principles shaping these soldiers' decision-making has also been transformed. Gamers have logged thousands of hours rapidly analyzing new situations, interacting with people they don't know, and learning to solve problems quickly and independently. **DoD must recognize the fundamental shift in the analytical and strategic problem-solving skills and techniques of the next generation of soldiers, and adapt its training and motivational methodologies accordingly.**

The field of serious games is expanding rapidly, with scores of games in development and already deployed. This is largely the result of a grass-roots movement within DoD. We believe **DoD should coordinate these efforts and help games emerge as a recognized component of M&S under the new name of MS&G**.

Despite hundreds, perhaps thousands of studies documenting the effectiveness of games as educational and training devices, there has not been an official statement from an authority within DoD that games are accepted alongside more traditional methods. The very word "game" has frivolous connotations, and career officers may perceive advocating games as risky to their advancement, especially when this light-hearted word is juxtaposed with the life-and-death consequences often associated with the development of military programs. **DoD should internally publicize existing research about the effectiveness of games, fund its own research into validation, verification and assessment (VV&A), and issue definitive guidelines stating when game solutions should be considered as options to meet program requirements**. Contracting officers should be able to arrange for acquiring and developing games without having to "creatively circumvent" current policies.

Games are not a panacea—they are not a solution to every problem. But game companies typically don't plan a "proof of effectiveness" phase into their development schedules (other than to ensure a game will meet its sales goals), and the reputation of games within the M&S community suffers as a result. With serious games, where effectiveness is defined as reaching the goal a project was funded to meet, **DoD should require each game project to include a formal evaluation phase that proves whether it will be effective**.

The cross-section of talent and expertise needed to create a serious game is rare. Games need to be fun, and serious games need to be effective. This requires a team of experienced game designers, subject matter experts, training/educational personnel, and qualified technical and artistic talent, all of whom may be motivated differently. The intersection of these capabilities does not often occur by chance. To be successful, **DoD must provide appropriate motivation for workers in diverse fields to come together and work on serious game projects**.

Business Challenges

The fundamental business challenge confronting game developers and the government is the misalignment of their working environments and the supporting infrastructures by which each can meet the other's needs. This lack of a common "way of doing business" affects almost every phase of the business relationship, including how organizations find each other, contracts bidding, long-term funding commitments, project development, payment cycles, profitability goals, compliance with regulations, accountability, and acceptance criteria.

Specific problems confronting government agencies and game companies setting up projects include:

- Government officials who want to work with game companies frequently don't know how to contact them, how to vet them, or what contracting vehicles they can use to work with them.
- Game companies typically aren't aware of government contracting opportunities. They don't know that these are posted on www.fedbizopps.gov, nor do they track or have the resources to monitor RFPs, BAAs, MURIs, SBIRs, STTRs, etc.
- Game companies find the maze of government contracting regulations bewildering, and the contracting cycle is too resource-intensive and lengthy for most small developers.
- Smaller game developers have neither the manpower nor the systems to deal with the accountability requirements of government cost-based contracting.
- The funding uncertainties associated with the governmental budget cycle are difficult for small game companies to work with.

Each of these problems can be solved or eased by creating the government "marketplace" organization outlined at the end of this Summary.

DoD should try to avoid consolidating suppliers that has occurred in other areas of acquisition. Instead of four or five massive suppliers, the Department should try to work with a wide range of small companies to take advantage of the agility, diversity, and creativity that the game industry has to offer.

The development culture within large, prime contractors is different from that of small game developers. The primes involved in CPO are typically not motivated to hold down costs, and the standard government contracting process encourages "feature creep"

because contractors rarely have an incentive to say no to a request. Game companies, by contrast, are accustomed to minimizing costs to eke out profits. **DoD should try to structure contracts that encourage the development methodologies that produce good games**.

Additionally, once these deals are in place, they will require different project management techniques from the government program directors who oversee them.

The uncertainties of managing software project schedules and budgets are magnified when working with games, because of the elusive "fun" factor. Neither partner in a game project wants to assume all the financial risk of a project overrun, so **DoD should also take care to structure deals emphasizing cooperation and shared risk**.

The government's legal community is under-prepared to deal with the complications of co-developed game software. Many governmental departments have small legal staffs who must deal with an extremely wide range of issues and who currently have little experience in game-related issues, such as intellectual property, licensing, contracts, privacy issues, freedom of speech, and ownership of "property" in a virtual world. **DoD should sponsor the equivalent of a Serious Games Conference for lawyers and contracting officers, where they can go (either online or in person) to learn about games and to discuss legal issues relating to game development**.

Product Development

The game industry has settled on agile development as the most effective way to create innovative and fun projects. This methodology relies on a process of creative discovery during development which makes rigid *a priori* specifications inherently impossible. While this phenomenon is understood and anticipated in funding government research projects, it is less familiar in the world of DoD development, acquisition, and operational programs, such as training. **Government and industry must recognize they have different definitions of agile development, and they must agree on the "best practices" to be integrated into all areas of their relationship**.

A key differentiator between conventional DoD military models and simulations and commercial game-based products is the reliance on the human element in the commercial side of things. The commercial game sector has *no product* unless humans are engaged and involved. M&S on the other hand has continually had a difficult time

embracing the human element in their models or sims. **Therefore, the more a program requires "human in the loop" features, the greater the reason to seek the skills and talents resident in the game community**.

Even with recent reforms, the DoD's arduous Acquisition Framework and its complex project development processes make game developers wary of working with the government. Additional tasks imposed by adhering to unfamiliar standards, the "ilities," VV&A, security procedures, and other government-specific requirements also make partnering with the government appear daunting. **Finding ways to streamline government processes—without compromising the goals they are designed to accomplish—is an essential step in attracting more game developers to DoD projects**.

Game developers who want to partner with the government must recognize its special needs. More time and money must be budgeted into individual projects for building in relevant data extraction, analytical tools, and VV&A procedures. Additionally, **both industry and government should participate in organized efforts to develop appropriate and useful standards in important areas**.

Nevertheless, governmental standards requirements still remain a roadblock to rapidly and effectively implementing serious games. The history and efficacy of standards are discussed extensively in this report, but it is appropriate to note that **whenever the government imposes interoperability requirements on a game, the development cycle for the project will be longer, more expensive, and less reactive to changing battlespace conditions**. Interoperability is useful only when it has been built into specific programs for a specific purpose, rather than enforced as a general principle.

With industry R&D spending now outstripping the government by a ratio of 2-to-1, the government is no longer the main driver in developing standards. In addition, the average time-to-completion for a new standard now exceeds the development cycles of many hardware platforms, which means that by the time a standard can be developed, the hardware and operating systems have moved on. As mentioned above, **there are many areas where DoD should participate with industry in standards development, but it should no longer try to drive the effort itself**.

Needless to say, the military environment is quite different than that of most game development companies. Our research suggests that this does not present significant problems for personnel on either side, and harmful "culture clashes" typically do not

emerge. However, in the area of contracting and project management, adjustments have to be made on the one hand by officers accustomed to issuing orders they expect to be followed, often without question or input, and on the other hand by game developers accustomed to holding a two-way dialogue with management to set project goals and discuss tradeoffs that may be necessary to reach those goals. **DoD should hire managers who have experience working with game developers, or train some of its managers in the different skills needed to manage game projects** (as opposed to typical prime-contractor cost-plus projects).

The traditional billet rotation cycle for military officers also presents an enormous problem for software development projects. It is well-established in the game industry that a change of management will always interrupt and slow development. A "dead time" often develops in a military project when a departing officer is reluctant to take action in deference to the incoming officer, and when the incoming officer does not take action while he becomes familiar with the project. This period of inaction can stretch to several months. Where possible, **DoD should work to maintain continuity of project management**, especially in cases where the military officer in charge might be rotated out during the course of development. This would also avoid the problem of leaving institutional knowledge of the project in the hands of the vendor, rather than of the government.

Technical Challenges

One reason DoD is interested in game technologies is its desire to deliver material to modern soldiers in a format they are already familiar and comfortable with: games. But a large part of the appeal of these soldiers' favorite commercial products depends on using hardware more advanced than is widely available in the military. If the government is committed to the idea of fully leveraging this opportunity, **DoD should implement a plan to continually upgrade existing and yet-to-be-purchased desktop and laptop computers, in order to create and maintain an installed base of media-capable machines**. **Additionally, DoD should sponsor projects for game consoles such as the Nintendo, Xbox, and Playstation,** the ones soldiers are most familiar with.

Just as hardware improvements have prompted re-evaluating long-held M&S assumptions, game industry advances in technical methodologies have also proven beneficial. **Future government MS&G efforts should aggressively seek to leverage game industry techniques and processes**.

Massively Multiplayer Online Games (MMOGs) may be the only current environment in which the principles of Network Centric Warfare can be tested and explored. Independent of many other reasons to study MMOGs (such as for their social and psychological interactions, effectiveness in education and training, and relevance in a military context), there are strong technical reasons for DoD to research this area through partnering with commercial developers. Many commercial service considerations—such as network attacks, information assurance, bandwidth optimization, server optimization, account accreditation, content creation, and production issues—are amazingly similar to the issues in a network centric environment and can therefore serve as a fertile field for informing DoD's understanding of persistent networks for warfare. **DoD should stay on top of commercial MMOG technology and should support academic research in this field**.

When evaluating engines for a game software project, the most important consideration should be the project's technical and operational requirements, rather than the engine's licensing fee. The initial acquisition cost of an engine is relatively small compared to the overall development costs of most products, and while a particular engine may be inexpensive to *acquire*, it may very expensive to *use*. **DoD should enter centrally-negotiated, enterprise-wide license agreements with several game-engine vendors**. Project managers will then have a broad array of engine choices enabling them to decide not based on license fees, but on capabilities, use, and total cost of ownership.

One potential barrier to DoD adopting new technology is its commitment to legacy systems. It is always difficult to know when to abandon maintaining and upgrading an existing system in favor of embracing a new one. However, as DoD evaluates future investment decisions, it must decide whether the original design specifications of these systems are too constraining to afford significantly expanding their scope to meet new requirements, or whether these expanded needs should be met by new systems, programs, or processes. Therefore, **as new MS&G requirements emerge, DoD should consider whether they are best met by expanding the capabilities of legacy systems or by developing new game-based programs**.

The globalization of the software industry in general and of the games industry in particular means that more projects will include code that is either open source or has been written overseas. Establishing that this code is secure will become increasingly difficult. Concerns include inappropriate messages or malicious code, Trojan horses, delib-

erately inaccurate algorithms, hijacked data, and other security breaches. To combat this, whether programs are written domestically or overseas, **DoD should develop or acquire code-checking programs that efficiently analyze code for security risks**.

The Future

This report concludes with a series of trends, summarized here, that will almost certainly influence the future of serious games:

- Games will play a large role in the future of the military.
- Game companies will be increasingly interested in serious games.
- DoD will need more than huge, monolithic M&S models.
- The consumer game industry will continue to spawn technological advancements which the government can take advantage of. However, there will always be governmental M&S needs that the industry will not fulfill.
- The government will no longer be the primary driver of innovation and standards.
- Offshoring will have capability and security implications for military projects.
- The globalization of the industry and open sourcing will also have security implications.
- Countries and organizations will continue to use games to explain and proselytize their cultural values to the world, which can conflict with Western cultural values.
- Lower barriers to entry will make it easier for individuals and poorly-funded organizations to create agenda-oriented games.
- Foreign governments will continue to fund their game-creating industries as an important part of maintaining their overall technical infrastructure.
- Massively Multiplayer Games and Net-based "sandbox" environments will become increasingly important test-beds.
- Collaborative, three-dimensional spaces (such as Croquet and *Second Life*) will also become useful extending gaming in multi-dimensional subject areas such as social, political, economic, and military PMESII games.
- Games will continue to profoundly influence the decision-making of tomorrow's soldiers.
- Innovative training ideas will come from "digital natives," rather than from the older generation of training personnel and M&S managers.

- The military will benefit from the game industry's usability research.
- The convergence of the Internet, mobile communications, and other advances in connectivity will create new gaming opportunities for the private sector and for DoD.
- The international flavor of the games community will increase contact and collaboration across borders.
- Some supra-national gaming communities will be sufficiently large to wield political and economic influence.

In the face of these trends, the current lack of interest in games by many in government circles is almost alarming, given that games are affecting fields as diverse as politics, economics, military training, general education, freedom of speech, constitutional law, propaganda, the spread of democracy, and national security.

We believe that the government has a strong interest in advancing the state of the art of games as an interactive medium, not only to stimulate the games industry to be a better partner to government, but also to make games more effective in areas such as education and training.

To this end, **we propose that DoD sponsor a "Grand Challenge" that encourages the industry to advance the medium of games as a practical art form**. More specifically, we propose a Grand Challenge that targets and rewards achieving "subtlety" within a game.

Currently, all the subtle information that people process in order to make decisions is incapable of being reproduced in our game environments, and is consequently lost. When one compares the crude representations of an in-game "agitated" crowd to the subtle indicators of a real-world gathering in Baghdad about to erupt, one quickly realizes how far the industry has yet to go, and how vital it is for that gap to be closed.

Until we can use subtlety to create deep emotions within players, we will be limited to emulating only the grossest human behaviors. But if we can achieve this goal, we will open the floodgates of the industry, enable it to evolve, and provide our soldiers with a truly immersive learning environment.

A Government-Chartered Non-Profit "Marketplace" Entity

A dynamic "ecosystem" significantly contributes to the success of any technical community. Good supply of talent, good demand from customers, and an efficient system to help them find each other, combine to create a healthy environment in which all participants benefit. Low barriers to entry create a fertile breeding ground for new artists, ideas, and technologies.

The emergence in the commercial game world of IGN, Gamespy, and similar news, distribution, and community sites has encouraged the development of just such an ecosystem where people talk about, compare, recommend, try out, and buy products. The result is an environment in which good, creative talent and superior products rise quickly to the top, while inferior talent and products rapidly disappear.

To understand the value of creating such a marketplace, one need only think of the accelerated dynamics that marketplace creators such as ebay and Gamespy have enabled by expanding upon the basic clearinghouse concept with their user community ratings, easy distribution, and facilitated financial transaction mechanisms.

The proposed marketplace entity could help:

- Game companies learn about and develop effective government contracting vehicles.
- Developers learn about game-related projects within government.
- Government project managers learn which contracting vehicles may be appropriate to apply to game-related projects.
- Government project managers share best practices and learn how to effectively manage game-related projects.
- Government project managers find developers with experience in specific areas of expertise.
- Government project managers understand what genres of games may be appropriate to achieve their individual goals.
- Government project managers learn how to effectively structure working relationships with game companies engaged in agile development.
- Each side understand the legal implications of working with the other to include ways of protecting intellectual property and evolving effective legal structures.

Additional capabilities of the proposed organization might include:

- Clearinghouse of information
- Product reviews
- Sample downloads
- User and supplier reviews
- Peer reviews
- Screen shots
- Updates
- Communities of interest
- Business forums
- Legal forums
- Project announcements area
- Transaction payments
- Participation in standards-setting communities and organizations, especially in the areas of evaluation capabilities and tools for data analysis.
- Representing DoD positions to other non-DoD gaming bodies and monitor areas of interest.

Some have suggested that such an organization could fund activities to help smooth the problems of year-on-year availability of government money, and to help set standards that address evaluation capabilities and tools for data analysis.

Perhaps the most important message of this report is that **DoD must recognize there has been a societal shift embracing new interactive and collaborative media**. This fundamental shift has added a new medium to our cultural repertoire; one which individuals and groups use to tell stories, learn, communicate, persuade, build friendships, and form communities. DoD and the government must recognize that games and other interactive/collaborative technologies have become so woven into the fabric of our lives that to ignore them is to be at a disadvantage. Instead of fearing this new media and restricting access and use of game products, DoD and the government in general must become a catalyst for accepting and adopting it—encouraging it, nurturing it, and taking a lead in advancing it. In other words, DoD must become a "digital native" rather than a digital immigrant. DoD and academia were the keepers of M&S in the past, but interactive media—products of the public—have stormed the gates, bringing with them innovation, creativity, and risk. When atoms collide, the result can be destruction or fusion. Similarly, DoD must decide whether it wants to approach this collision with potentially destructive delaying tactics, or to open the gates and accept the fusion of this media into the new world of MS&G.

1
THE COMMERCIAL WORLD

I. Introduction

Making games is serious business. Commercial game companies operate in an extremely competitive environment, often creating products on the bleeding edge of technology. Companies survive by evolving streamlined processes that allow them to efficiently develop these highly complex products and deliver them to market quickly. Understanding how games are built and the economics that drive the industry's business models is crucial to working successfully with a game developer.

This section of the report opens with a short history of commercial games and a discussion of the different game genres. It goes on to explain the anatomy of a game studio and the responsibilities of different members of the development team. Commercial development processes are treated in depth, especially insofar as they differ from government processes. Equally important is the discussion of how games are marketed and sold, which drives the business models underlying the entire industry. There are chapters on engines and standards, and the section closes with a look at important trends that will determine the direction that games will take over the next five years.

II. A Short History of Commercial Games

Introduction

The video and home computer game revolution has taken place within a remarkably short period of time—within the lifespan of most of you who are reading this.

Coleco's *Telstar*, the first successful home videogame console, was released in 1976. The first successful home computer, the Commodore 64, came out in 1982.

The main game genres were established quickly: by 1983, action, adventure, role playing, strategy, sports, fighting, and simulations had all debuted. Since then, only a few new genres have been introduced: "sandbox" or "god" games got their start with *Sim City* in 1989, and the first person shooter (FPS) genre was born in 1992.

Today, video and computer games play an enormous role in the lives of millions. According the Entertainment Software Association:

- More than 228 million games were sold in 2005, almost two games for every household in America;
- 69% of American heads of households play games; and
- US sales of game software in 2005 totaled $7 billion.

The first generation of "digital natives"—people who grew up with video games as an important part of their lives—has now matured and is moving into all areas of business and government. As the book *Got Game* so succinctly puts it, "All those hours immersed in game culture have created masses of employees with unique attributes: bold but measured risk taking, an amazing ability to multi-task, and unexpected leadership skills." [1]

It is difficult to draw a hard line identifying the age at which we would consider someone as a "digital native," rather than a digital immigrant. Some people say that in the year 2006, the dividing age is 40, noting that *Pong* came into homes in 1974, and

---

[1] Beck, John, C. and Mitchell Wade. *Got Game*. Boston, MA: Harvard Business School Press, 2004.

Coleco's *Telstar* was released in 1976. Other research reveals that above the age of 36, odds are 2-1 that a person had little or no videogame exposure as a teenager, whereas below the age of 36, odds are 4-1 that he or she had had substantial exposure to games.

The authors believe that 34 is a very reasonable estimate for this dividing line, believing that anyone who was 13 years old in 1985, when Nintendo launched *Super Mario Brothers* and the first Nintendo Entertainment System (NES), and 17 years old in 1989, when ARPANET became the Internet, has been profoundly influenced by the culture of games.

Timeline

1961 MIT student Steve Russell creates *Spacewar*, the first interactive computer game. It can only be played on a $120,000 Digital PDP-1, one of the first mainframes to have a monitor. All the "graphics" are ASCII text characters. The game is a duel between rocket ships that fire torpedoes at each other.

1969 DoD launches the Advanced Research Projects Agency Network (ARPANET) that later will evolve into the Internet.

1972 The first home videogame system—Magnavox's *Odyssey*—is released. It comes with swappable circuit boards that allow you to play only 12 different games. Interestingly, this first home console also ships with a light gun for shooting at the screen.

Pong invented by Nolan Bushnell. (Available at first only in arcades, Atari later ships 19,000 *Pong* machines.)

1975 Crowther & Woods create *Adventure*, a text-based game that can only be played on a mainframe computer. It is the first of many adventure games to encourage exploration in a fantasy setting populated by Tolkeinesque characters and creatures.

Pong becomes available for play in the home. Sears orders 150,000 copies of *Home Pong* for the Christmas season.

1976 Coleco releases *Telstar*, its first home videogame console, which grosses over $100 million in sales.

Steve Jobs and Steve Wozniak found Apple Computer.

Don Daglow writes *Dungeon*, playable only on a PDP-10 mainframe. It is an unlicensed version of the popular "live action" role playing game *Dungeons & Dragons*.

1977 The Atari 2600 is released. It is the first successful videogame console that allows consumers to plug in cartridges with new games on them. (Previously, consoles came with games hard-wired into their circuitry, and adding games was impossible). It is also the first consumer console machine with a joystick.

1978-82 The "Golden Age" of arcade video games. *Space Invaders* is followed by *Asteroids*, and then *PacMan*, *Donkey Kong*, *Centipede*, *Frogger*, *Missile Command*, *Tempest*, and *Ms. PacMan*. The business peaks when *PacMan* and *Ms. PacMan* become the only arcade machines to sell over 100,000 units.

1980 *Space Invaders* becomes the first arcade game to be licensed for a home console system.

Infocom publishes *Zork*, a PC-based descendant of *Adventure* and the prototypical adventure game for years to come.

1981 A group of military officers approach Atari to modify the game *Battlezone* for use as a military training tool.

1982 The Commodore 64 computer is released, marking the start of the home computer game revolution. Commodore is the first computer company to report a $1 billion sales year.

1983 Electronic Arts ships its first products.

The Great Videogame "Crash." The bottom suddenly drops out of the videogame industry. Consumers stop buying. Companies teeter on bankruptcy. Hundreds of thousands of games are marked down to $4.95, can't even sell at that price, and are eventually plowed into landfills. Large companies like Mattel, Magnavox, and Coleco abandon the industry.

1985 Launch of the NES (Nintendo Entertainment System). *Super Mario Brothers* marks the beginning of the comeback of videogames.

Broderbund publishes *Where in the World is Carmen San Diego*, one of the most popular educational games of all time.

1987 VGA and SVGA graphics cards are introduced, vastly improving the visual quality of games that can be played on home computers.

1989 ARPANET is replaced by the Internet.

Creative Labs releases the Sound Blaster, creating a new standard in sound cards.

1991 *Sonic The Hedgehog* is released by Sega in the United States.

1993 *Doom* is released. This is the first hit multiplayer game, and is the grandfather of the FPS (first person shooter) genre

Sid Meier publishes the first version of *Civilization*, the most influential strategy game ever.

Senator Joe Lieberman holds hearings on violence in videogames that lead to the establishment of the Interactive Digital Software Association (IDSA), now called the Entertainment Software Association (ESA) and the Entertainment Software Rating Board (ESRB).

1994 *Myst* becomes the first CD-ROM game to sell over one million units. *The 7th Guest* is also released.

1995 Sony launches the Playstation in the US.

Microsoft introduces the Windows 95 operating system, making the PC a much stronger gaming platform.

1996 Nintendo launches the N-64 with *Super Mario64*.

3dfx releases the Voodoo chipset, the first affordable 3D accelerator card for home computers.

1997 Origin releases *Ultima Online*, the first MMORPG (massively multiplayer online role playing game), and the ancestor of *Everquest*, *Asheron's Call*, and *World of Warcraft*.

2000 Sony launches the Playstation2 in the US—500,000 units sell out overnight.

Electronic Arts publishes *The Sims* which to-date holds the record as the top selling PC game of all time.

2001 Microsoft releases the Xbox, and Nintendo releases the GameCube.

2002 Launch of *America's Army*.

2005 Microsoft releases the Xbox 360

2006 Anticipated release of the Playstation 3 and the Nintendo Wii.

This short history demonstrates the youth of the games industry and the rapid innovation that characterizes its creative, technical, and business processes. Anyone wishing to partner with a game company should be prepared to deal with both the benefits and the drawbacks of working within this fast-paced "culture of change."

As technology advances and enables the creation of more engrossing and complex games, genres proliferate and are refined. The next section discusses what typifies each commercial game genre.

III. Commercial Game Genres

Almost any kind of game can be adapted for "serious" use, but it is important to select a game genre that fits the goals of the project at hand. This section contains a list of commercial game genres and their defining characteristics. They are arranged loosely in decreasing order of popularity.

Action

Action games are real-time games in which the player must react quickly to what is happening on the screen. The category is dominated by First-Person Shooters (often abbreviated to FPS) such as *Quake*, *Unreal*, and *Halo*, but it also includes the sub-genres of driving games such as *Need For Speed* and even pinball games.

The action-adventure hybrid is often a third-person game, such as *Tomb Raider*, in which you can see the hero or heroine as he or she moves through the environment. Typically, the gamer has much more to do than just shoot and kill enemies.

The difference between "first person" and "third person" is that first-person games put the camera in the character's head. The player sees what his character sees. In third person, the camera is outside the main character, usually floating just above and behind but sometimes moving to different positions to provide a better view of the action.

First-person games tend to be faster paced and more immersive. There is a greater sense of being "in the world" as the player sees and hears along with his character. Third-person games allow the player to see his character in action. They are less immersive but help the player build a stronger sense of identification with the character he is playing.

First-person games tend to have more beautiful game environments and higher-detail non-player characters (NPCs). This is because the game engine doesn't have to devote any of its resources to drawing the main character. Third-person games chew up a lot of resources drawing the main character and the animations that go along with it, leaving correspondingly fewer resources to render the game world and the creatures in it. (Project managers making the first-person/third-person decision would do well to

consider these differences—if a project demands ultra-realistic environments, first-person might be the best choice; but if it is important to see the movements of the player character or his relationship to the positions of other characters, then third-person might be the wiser option.)

In general, action games are far less cerebral than adventure, strategy, or puzzle games. Players are looking for the adrenaline rush of fast-paced action that calls for snap judgments and quick reflexes. Opponents can be computer-generated artificial intelligences (AIs) or other human players connected to the game over a local network or the Internet.

Adventure

Adventures are story-based games that usually rely on solving puzzles to move the action along. They can be text-based (such as the early adventures from Infocom, *Zork* and *Planetfall*) or graphical (Sierra's *King's Quest* and *Gabriel Knight* series). They can be told from the perspective of a first person (*The 7th Guest*), second person (most text games in which the hero is "you"), or third person (*Monkey Island*).

Adventures are generally not real-time games, unless they are an action-adventure hybrid. The player usually takes as much time as he wants between turns, and nothing happens in the game world until he enters a command.

The original adventures were parser based; that is, they accepted simple sentence commands from the keyboard. More modern adventures are point-and-click; the player indicates what he wants to do by moving the mouse around the screen. An active community of hobbyist developers is still making parser-based text adventures, but these are rarely published commercially.

Players generally expect an adventure to have a large, complex world to explore, along with interesting characters and a good story.

Role-Playing Games (RPGs)

In role-playing games, the gamer generally directs a group of heroes on a series of quests. Gameplay revolves around gradually increasing the abilities and strengths of his heroes. Classic RPGs include *Ultima*, *Might and Magic*, and *Final Fantasy*.

Like adventure games, RPGs feature a huge world with a gradually unfolding story. Players expect to be able to micromanage their characters, all the way down to the weapons they carry and the specific armor for each part of their bodies. Combat is an important element—it generally is the mechanism by which the heroes gain strength, experience, and money to buy new equipment. Early RPGs were called "hack-and-slash" games, and combat still plays an important role in the category.

Fantasy RPGs also feature complex magical systems, as well as diverse races of characters that make up the player's party.

RPGs have a slow build that starts the player's character as a weakling in a strange and dangerous world. Through carefully managed encounters and alliances, the hero and his party slowly grow in competence and power until they are able to take on the baddest of the bad guys.

Storytelling in RPGs is generally accomplished through a series of quests. As the player carries out the missions, he explores the world and learns more about its inhabitants and his place among them.

Strategy

Strategy games require players to manage a limited set of resources to achieve a predetermined goal. This resource management frequently calls for deciding which kinds of units to create and when to put them into action.

In the classic *Command & Conquer*, for example, the player has to continually balance which kind of unit to build, how much raw material to harvest, how many resources to allocate to offense and to defense, and so on. Other canonical strategy games include *Age of Empires*, *Warcraft*, *Starcraft*, and *Civilization*.

Some strategy games are turn-based. The player takes his time to make and implement a discrete set of decisions, and the computer acts only when the player indicates he is ready. Real-time strategy (RTS) games, on the other hand, set the computer AI in motion against the gamer, whether he's ready or not.

Multiplayer versions of RTS games substitute human opponents for the computer AIs. These games are enormously popular on the Internet.

Most strategy games have two teams opposing each other, but some have three "sides" or more. Regardless, each side in the game must have an equal chance to win. It was said of Robert E. Lee that he could take his soldiers and beat yours or take your soldiers and beat his. The good strategy player must feel the same about the teams available at the start of the game.

Another important feature of commercial strategy games is that while they try to evoke reality, they generally don't seek to model it. For example, even though designers often base their weapons on actual ordinance, they will almost always choose to make them fun to use, rather than have them correspond exactly to their real-world counterparts. The physics of a certain weapon can confine it to a restricted range, but if it would balance the game better to give it a longer range, the designer will do so.

How much ammunition does a particular gun hold? How long does it take to reload? The designer starts with the real world in answering these questions, but what survives in the game is generally what is the most fun for the player. Real combat is made up of hours and days of boredom, followed by 15-minute bursts of real terror. Strategy games try to capture and extend those 15 minutes.

Simulations

Simulations are games that seek to emulate the real-world operating conditions of complicated machinery, such as jet fighters, helicopters, tanks, and so on. While games in the "casual" genre are often described as "a mile wide but only a foot deep," a simulation (or "sim"), by contrast, is only about a yard wide but miles deep. It focuses on only one piece of equipment or activity and mines that experience for all it's worth. Popular examples include *Microsoft's Flight Simulator*, *Nascar Racing 2003 Season*, *Falcon 4.0*, and *Silent Hunter*.

The more serious the sim, the higher the premium placed on absolute accuracy, especially with equipment controls. Players expect to spend hours learning the intricacies of the machine and expect a thick manual to help them with the finer points.

Less serious sims, however, seek to let the player "get in and go." These are sometimes referred to as arcade or casual sims. Controls are simplified, the player has less to learn, and he is punished less often for making mistakes.

The differences between the serious sim and the casual sim usually cannot be bridged in a single product. For the hardcore, no detail is too small to get right. The physics model must be accurate. Measurements and tolerances must be precise. The controls must respond as they would in real life. The casual gamer, however, wants to "get in and go." He doesn't want to be bothered with learning a million controls before he can do something. Casual sims generally simplify the controls and simulate the fantasy the gamer has in his head, instead of the reality.

Sports Games

Sports games like *John Madden Football*, *NBA Live*, and *FIFA* let players vicariously participate in their favorite sport, either as a player or a coach. Ironically, prowess in the real world does not translate to success in the computer game, but that is sort of the point. One of the things people want from a computer game is wish fulfillment, the opportunity to do things they can't do in real life.

These games accurately reproduce the rules and strategies of the sport. One gameplay session can cover an individual match, a short series, or an entire season. Some titles focus on emulating the athlete's actions, on actually playing the game. Others approach the sport from the management side, allowing the user to be a coach, general manager, or owner who sends in plays, makes trades, or tries to build a franchise while worrying about the salary cap.

Platform Games

Platform games are characterized by the players controlling an onscreen character whose main activity is running and jumping through the game environment while avoiding obstacles and monsters.

Well-known examples are the *Mario* games from Nintendo, *Crash Bandicoot* from Sony, and the *Prince of Persia* games developed by Ubisoft.

The control scheme of platform games lend itself more to a handheld controller, rather than a keyboard and mouse, and consequently these games tend to be more popular on console systems than on the PC.

Platform games also incorporate a higher degree of trial-and-error learning than most other genres. It is expected that the player will fail often and replay levels several times before mastering the patterns necessary to succeed.

Fighting Games

Fighting games are two-person games in which each player controls a figure on the screen and uses a combination of moves to attack his opponent and to defend against his opponent's attacks.

These games are generally viewed from a side perspective, and each session lasts only a few minutes. Classic examples include *Mortal Kombat*, *Tekken*, *Virtua Fighter*, *Street Fighter*, and *Soul Calibur*.

Fighting games are simple and direct, yet they can be very engaging. They are one of the few genres to assume that the players are physically sitting side by side and can talk to (and taunt) each other. The game's goal is to create quick bursts of swift and intense action, followed by more of the same.

Because the focus is so tight, great graphics are standard. The only things players see are a confined fighting area, a relatively static backdrop, and the two characters. These characters are the most visually developed of all the genres because the processor can focus so much attention on them.

Casual Games

Casual games include adaptations of traditional games such as chess, bridge, hearts, and solitaire. They also include easy-to-play, short-session games on the Web, such as *Slingo*, poker, and *Concentration*.

Television game shows are also represented in this category, including the very popular *Jeopardy*, *Wheel of Fortune*, and *Who Wants to Be a Millionaire?*

Players generally want to drop into and out of these games quickly. They are already familiar with the rules of the real-world game and expect to find them emulated here. These games generally have an extremely simple user interface, with little or no learning curve.

God Games

God games (sometimes called software toys or sandbox games) are games that have no real goal, other than to encourage the player to fool around with them just to see what happens. Examples include *The Sims* and *RollerCoaster Tycoon*.

Designers in this genre try to create games in which the player can do no wrong. The games are very open-ended, with no "correct" way to play and no preset winning conditions.

Educational Games

The goal of an educational game is to teach a specific body of knowledge.

The design team must have a clear idea of what this knowledge is from the start. They cannot create a game first and then tack on some educational value at the end. This usually means working with a subject matter expert and adhering to strict guidelines.

Most educational or edutainment games have been aimed at a much younger audience than other commercial products. Examples include *Oregon Trail* and *Reader Rabbit*.

Puzzle Games

Puzzle games exist purely for the intellectual challenge of problem solving, such as *The Castle of Dr. Brain* and *The Incredible Machine*.

In these games, the puzzles are an end in themselves and are not integrated into a story, as is common in adventure games.

Online Games

Games from any genre can become online games when modified to be played over the Internet. Popular genres include casual games (poker, hearts, chess), action games (*Counter-Strike*, *Battlefield 2*), role playing games (*World of Warcraft*, *Everquest*), and strategy (*Civilization*, *Starcraft*).

Entire communities grow around the most successful of these games, and the designers of products like *World of Warcraft*, *Everquest*, and *Ultima Online* are constantly creating features that encourage those communities to flourish.

Anyone who is thinking of working with a private company on an online game must be aware of the particular business model that drives that company's products. Business models vary widely, and they affect not only a game's design and presentation, but also the amount of time the company wants the player to be online.

Some business models include:

- Free to the user, supported by advertisements
- Pay-per-play with an hourly connect fee
- Box-cost only, with free online play
- Subscription based, with a monthly fee
- Free to the user, supported by in-game purchases of "premium items."

The proliferation of genres has lead to a confusing collection of acronyms that people use in connection with online games. Here are a few of them:

- **MMOG.** Massively-Multiplayer Online Game. Any game that has a large number of simultaneous users. This can include anything from poker to Everquest. MMOGs are also frequently referred to either as MMOs or Massively Multiplayer Games.
- **MMORPG.** Massively Multiplayer Online Role Playing Game. A "persistent world" game in which each player adopts a persona within the game universe. Examples include Ultima Online, World of Warcraft, Star Wars Galaxies, and City of Heroes.
- **MUD.** Multi-User Dungeon. Usually a role-playing game that is text-only.
- **PSW.** Persistent State World. Another way to refer to MMORPG.
- **PvE.** Player versus Environment. Games in which players fight only AI-created characters.
- **PvP.** Player versus Player. Games in which players fight other live players' characters.

Regardless of the genre, a game requires a team of talented people to bring it from an idea to a finished product. For small games, that team may only be three people (each of whom wears several hats), but these days most commercial games are backed by a design studio of 15–100 people. We examine their varying roles and responsibilities in the next section, Anatomy of a Development Studio.

IV. Inside the Game Studio

Every development studio divides the subtasks of making a game in its own peculiar way. What each job is called and how it is done changes from company to company. The tasks, however, stay the same—each game must be managed, designed, and programmed; it needs art, sound, and music; and it must be tested.

Although the tasks seem to separate neatly into a one-job-per-person classification, in reality everything is much sloppier. On some teams, a job can be divided among several individuals. It's not uncommon, for example, for a game to have two co-producers, or for a pair of artists to share the duties of the art lead. On other teams, a single person can take on more than one project role. The tech lead, for example, can double as the producer, or the art lead can also be the game designer. The smaller the team, the more likely roles are to overlap—and team sizes vary widely. Some games are developed by just one or two people; most teams number between 15 and 50. However, it is not unusual for high-profile games to have teams of over 100 people working on them.

Most development groups allow their members wide latitude in the tasks they take on, rather than restricting individuals to tasks that fall within a specific job description. For each project, they tend to look at the talent they have access to and divide the tasks among the available individuals in whatever way makes the most sense.

Vision

Every project has one person who is the keeper of the vision. This isn't a job title that will be found on any organizational chart. It is a function that usually falls to the game designer, but the slot is sometimes filled by the producer, tech lead, or art director. In very rare cases, the vision can be shared by two individuals working closely together, but that is a tricky proposition and should be approached with caution.

The vision guy is the person who knows throughout the chaos of development how all the pieces will eventually come together and be experienced by the player. Although not an expert in any of the disciplines, he has a working understanding of all of

them. He may not be a programmer, but he understands how technical issues affect and constrain the project. He may not be an artist, but he understands the complex subdivision of tasks that go into creating images on the screen. He may not be a game designer, but he has a feel for what is fun. He may not be a psychologist, but he is usually a good "people person," someone who can smooth ruffled feathers and get people with different interests and agendas to work together towards a common goal.

The vision guy is the game's internal compass. He is the gatekeeper, through whom all new ideas must pass. He is the final arbiter of what stays in and what doesn't.

The vision guy must have a firm understanding of the core elements that will make the game successful, the irreducible feature set that must be in the game before it can be considered complete. During development, thousands of ideas will appear and beg to be included in the game. At the same time, schedule pressure and production problems will put enormous pressure on these and other features to be dropped. It is up to the vision guy to decide whether a new idea contributes to, is neutral to, or detracts from the core of the game.

Production

In the games industry, two separate jobs commonly bear the title of producer. The first is the external producer, who works for a publisher and oversees the efforts of an external development house. The second is the internal producer at a development studio, who manages the team itself and represents it to the outside world (including the publisher's management, as well as the marketing, PR, and sales departments). Some companies call this person the project manager, project lead, or director.

Whether internal or external, the producer is the game's champion to the company that has commissioned it. He explains the game's highlights and selling points to PR, marketing, and sales. He understands how the game aligns with the company's goals and can explain to "the suits" why it is a good idea to keep the product in development. He sticks up for the team. He demonstrates the game at project review meetings and explains where the team is, whether they're ahead or behind in their schedule, the problems that have cropped up, and what's being done about them.

The producer is also the client's champion back to the development team. He explains to them how the game fits into the company's plans. He keeps the team up-to-

date on the PR and marketing efforts being made on behalf of the game. He gets the team what they need to perform well, whether it's software, hardware, or other equipment. He keeps the team informed about the company's overall health.

The External Producer

The external producer is the person in a publisher's organization who is responsible for getting the developer to deliver the game on time, on budget, and with great features. He (and his assistant or associate producer) tracks milestones, approves payments, handles hardware requests, and generally ensures that the developer is well fed and cared for. In short, his job is to be the oil that reduces friction between the publisher and the developer.

Although he may not be the vision guy for the game itself, he is definitely responsible for making sure that the game meets the goals his company has set out for it. If the day comes when there has to be a trade-off of money against time or features, he has to be in the thick of the discussion. He needs to know which features are essential to the game's success and which are "nice-to-haves." He needs to know whether it's more important to his company to have the game be great, or to have it be on time.

The external producer acknowledges receipt of milestones, responds promptly to questions, takes quick action on requests, and above all, makes sure that the developer gets paid. He tries to keep the developer in a comfort zone, where they are focused on the work and not distracted by extraneous issues.

When changes are made to the project (and trade-offs are constant), it is the producer's job to track them. He follows up conversations with written summaries so there is always a detailed paper trail so both the developer and producer can say with confidence, "This is what we agreed to, and here is when it happened."

Email acknowledgements for small changes are routine, but larger issues are generally noted as amendments to the contract. In particular, any redefinition of what constitutes a milestone, a change in the delivery schedule, or an alteration to the payment schedule is important enough to update the contract. Turnover in the game industry is constant, and projects run for a long time. No one wants to be in a position where obligations are murky because of verbal agreements made between people who are no longer around.

The Internal Producer

The internal producer manages the development team directly and reports on their status to the funding organization, whether it's the company for which he works or an external organization such as a government agency or a commercial publisher.

One of his first tasks at the beginning of the project is to work with the art lead and tech lead to determine the right staffing for the game. If his studio does not have enough resources, he must either create a hiring plan to bring in new people or survey outside resources to see whether they can be effectively used on a contract basis (more easily done with art than with programming).

During development, his contribution comes mainly in the day-to-day running of the project. A game design is not a static document, and a project team is rarely stable. There is an old saying that no battle plan survives the firing of the first bullet, and that the genius of the field commander is how he responds to the chaos around him. The same is true of the producer. After development begins, his goal is to guide the team through the fog of war, keeping everyone together and moving toward the same goal so that when the smoke clears, the team has achieved its objective.

With each new idea that is generated, the internal producer helps determine its specific effect on the game's design but also the more global effect on the project: Whom does this affect? How many schedules will have to be altered? Does it bring the game closer to the publisher's goals? The answers to these questions will determine whether the idea is accepted or rejected. Either way, he will constantly be forging compromises among the working groups, and no matter what the game design looked like on paper, the game that winds up in the hands of the customer will be the result of those compromises.

At the end of the development cycle, he works with the test lead and lead programmer to evaluate the bug list. He considers the seriousness of each bug, determines the level of effort it would take to fix it, and assesses the risk of the "fix" itself creating other bugs.

Finally, he is the one who must decide whether the game is ready to ship. No game is ever bug-free. He needs input from all the departments to help decide whether any of the problems that remain should be considered showstoppers. In the end, though, he is the one who declares the release candidate "good enough."

Assistant and Associate Producers

The duties of the assistant or associate producer (an AP) vary, depending on the strengths and weaknesses of the producer and the development team. He might find himself in a pure management role, or in a hands-on position on some portion of the project, or handling a variety of general support tasks. In general, he will be hip-deep in the minutiae of the development process, and will have to be more detail-oriented than anyone ever thought possible.

Some common tasks for APs include:

- **Managing assets.** As projects and teams grow, the amount of data generated during development explodes. The typical high-end program has hundreds of thousands or even more than a million files to manage. Which is the latest version? Where is it on the server? Can someone overwrite it with a version from his or her local machine or otherwise accidentally delete it? What if someone updates it, and the change doesn't get recorded? If the file is no longer useful, is it included in the final build "just in case"? Most companies have rudimentary tools for addressing these problems, but ensuring the safety and usability of data will nevertheless be an almost full-time job, and the task usually falls to an AP.
- **Supervising the daily build and the backup.** When a project is well underway, it generally falls to an AP to ensure that a current playable version is always on the network. He is also generally responsible for ensuring that a solid daily, weekly, and monthly backup plan is in place and implemented.
- **Maintaining the design website.** Teams are rapidly moving away from the large paper "telephone book" design document, and towards designs that live solely on an internal Web site. This is a great advance because everyone can see what everyone else is working on. It doesn't appear magically, however, and someone (usually the AP) must collect, organize, and post the information while archiving old information as well.
- **Generating screenshots and supporting PR.** When a project is announced, an immediate, almost insatiable demand for screenshots springs up. Generating interesting shots is an art unto itself, and it takes time to produce pictures that show off the game at its best. In addition, the team always needs a knowledgeable "demo guy" who is available to sit down with visitors and take them through the latest version of the game.
- **Reviewing milestones.** When a milestone is submitted, the material must meet specific requirements before it can be accepted. It usually falls to the AP to actually ex-

amine the game or materials to ensure that all the requirements have been fulfilled. (We will discuss milestone requirements more completely in How Commercial Games Are Marketed, Distributed and Sold.)

- **Paperwork.** The AP often generates most of the management paperwork associated with the project, including submissions to console manufacturers to satisfy their approvals process.

Design

The formal design team is made up of the game designer, the level designers, and the writer. Although everyone on the project will influence the design before it is done, this is the group that establishes the game's original blueprint.

The Game Designer

The game designer creates the official design document and updates it throughout the course of development. He designs the basic gameplay mechanics and is also likely to be the vision guy who evaluates new ideas to determine whether they help or hurt the game.

The designer works with a storyboard artist to design the introduction, extro (end movie), and cut scenes (small, non-interactive movies). If he is doubling as the writer, he creates all the game dialog and also probably does the first draft of the manual. If he himself doesn't do the writing, he hires and directs a professional writer to do the job instead.

The designer usually directs the level design (LD) team, if there is one. He creates the flow of the game and then directs the LDs in creating the smaller units that fit into that flow.

The designer also collaborates with the PR department as it builds the Web site, the marketing department as it creates advertisements and the packaging, and the sales group as it generates promotional material for the trade. He designs demos for all three groups so that they can promote the game, and he makes himself available for press interviews as well.

The game designer is the one who must figure out what the player will actually do. He is the source of the fun. He is responsible for entertaining the player from moment to moment.

Usually (but not always), he is the vision guy, the one who can play the movie in his head. It is rare that a game design can be written down and then simply implemented. In the course of development, thousands of decisions are made by all the team members. The designer is the filter through which those decisions must pass. He compares each idea against the vision of the game and decides whether the two match.

During this process, he must stay flexible. His vision cannot be some unassailable monolith that must be implemented no matter what. Game design always involves compromise: each hardware platform has limitations, no budget is bottomless, no schedule stretches to infinity. He will constantly be asked whether something can be done this way rather than that. He must always be practical and ready to adapt the vision so that it can be implemented by the rest of the team.

The Level Designer

Level design is one of the newest fields in the industry. A few years ago, the position didn't exist. Then it became the province of the talented amateur. Now it is a key position on many teams.

When the design document is done, teams of specialists swarm over it to bring the words to life. Engine programmers figure out graphics pipelines and how to detect when objects in the world collide. Modelers build complex creatures and hand them off to the animators, who give them movement. AI programmers tell the creatures how to behave. Texture artists clothe the creatures and the world in which they live. Composers dream up atmospheric music. Audio technicians twist everyday sounds until they emerge anew from the speakers as echoes from a wholly imagined world.

It is the level designer who takes all these pieces and stitches them together to make a game. He is part-artist, part-architect, part-programmer, and part designer. How well he does his "stitching" determines whether he is a Dr. Frankenstein creating a monster or a Pygmalion breathing life into a beautiful Galatea.

The Writer

A freelancer or a staff writer who is not the designer will probably not be on the project full-time. Instead, he will be brought in from time to time to work on various parts of the game. This work can include character dialog, sports commentary, cut-

scene narratives, journals, an instruction manual, a hint system, or any other portion of the game where words are needed.

When he first comes on board, he sits down with the designer and gets a comprehensive overview of the game. He reads all the design documents, and talks with various team members to ensure that he understands what the team is trying to accomplish. Later, when he re-enters the project after having been away for some time, he repeats this process to see what has changed, because something always does.

Programming

The programming team is not only responsible for developing the game code, but also for maintaining the technical infrastructure that will be needed to build the game. This usually includes building or buying a game engine, creating additional software tools, managing the team's internal network, and keeping development on schedule.

The Tech Lead

The technical lead is on the project from the very start, alongside the producer, designer, and art lead.

One of his earliest tasks is to inform and inspire this group as to what is technically achievable—holding down unrealistic expectations on the one hand while identifying exciting areas of innovation on the other. Generally, he will try to pick no more than two areas of major technical risk per project. He could shoot for the moon and attempt more, but he will probably end up missing the moon (and his schedule, budget, and market window as well).

The tech lead evaluates the hardware delivery platform (PC, console, mobile phone, etc.), and creates an architecture that will maximize its strengths and compensate for its weaknesses.

During preproduction, he creates a technical plan that enumerates all the knowable tasks on the project and estimates the manpower and schedule required to complete them. When he delivers these estimates, they are often identified as a bell curve of probabilities, which should help manage expectations about their accuracy.

Another task during preproduction is evaluating the technology necessary for the game and to recommend whether it be built internally or be acquired from outside the company. This applies not only to the game engine but also to the suite of tools the team will use during production. The tech lead is always wary of the "not-invented-here" syndrome, and is amenable to purchasing ready-made tools that will speed development.

It is the tech lead's job to set coding standards, encourage "best practices," establish version control procedures, and implement a regular data back-up plan in case of catastrophic failures. During production, he manages the programmers' tasks and schedules.

Because the technical world is such foreign territory to nonprogrammers, the lead must become adept at explaining technical issues, de-mystifying them as much as possible. In particular, he should be able to explain technical trade-offs to the producer. A project can be optimized for schedule, for cost, for quality (lack of bugs), or for user satisfaction (great gameplay)—but not all four at once. There are choices that must be made every day of the project's life, and the tech lead is the one who must explain the choices well enough for everyone to understand them.

Programmers

Programming is where game design is accomplished. Programmers are the ones who must make real what a designer can only imagine.

Game designers and producers frequently don't understand the technical implications of the features they request. When programmers come across problematic areas, they work with the designer to explain what will have to be done to implement the design. In particular, they point out places where minor changes can result in major savings. Sometimes these improvements are internal to the code (for example, reduced complexity and increased efficiency), while other changes directly affect what the player sees (better game speed and loading times or fewer potential bugs).

On any given project, a programmer may find himself working on one of the following subcomponents:

- Rendering engine
- AI
- Physics
- Tools
- Database
- Network and multiplayer code
- Graphics effects
- Sound effects
- Scripting languages
- Weapons
- Game logic
- Interface and I/O (input/output)

- Server programming
- Web programming
- Asset integration

On a large team, separate individuals can work on each of these modules. However, the smaller the team is, the more likely one individual will be asked to take on several of them.

Art

Great art has become one of the benchmarks by which games are judged.

It's been said that one can't judge a book by its cover, yet millions of people do it every day. The same is true of games. People make their purchasing decisions based on what they see—after all, they can't evaluate gameplay while they're standing in the store. However, a quick visual demo and some socks-knocking screenshots on the box can propel the game from the shelf to the shopping cart.

Artists now affect every aspect of game design—from the user interface to the representation of the gameworld on the screen, to the special effects. Creating art has become increasingly complex through the years, as have the tools used to create it. Many companies who previously farmed out their art needs have now come to recognize great art as a competitive advantage and have built art departments of their own.

The Art Lead

The art lead is responsible for the "look" of the game. Frequently, he will be the production designer or concept artist, but if not, he will direct the people in those positions to create art that represents his vision.

The art lead lives at the crossroads of design, programming, and management. He needs to analyze what the designer wants, work with the tech lead to establish the production path, and then determine the scope of the art tasks, how many people he will need, what kind of skills they need, what tools they need, and how long it will take to bring it all together. He also decides which art should be developed internally and which should be shopped out to specialists.

When he works with the designer, one of art lead's goals is to develop a consistent style for the game. This style extends through all elements of the game, from the splash screen to the characters and environments, and even to the menu interfaces. After he

has established the look, he codifies it in a style guide (frequently called the Bible), the visual resource to which all the artists refer when creating new art.

Artists

No area of game development is evolving more rapidly than art. Artists must constantly keep up with their craft and be ready to adapt. Artists cannot afford to be technophobes. Not only do they need high-end computers and sophisticated software to create images, but they also must have a working understanding of the limitations of their target hardware platform so they can tailor their work to its strengths and avoid its weaknesses.

Most artists working on a game fall into one of the following specialties.

- **Concept artists** work with the designer to create the look of the game. They make multiple sketches of characters and settings, trying to bring the designer's vision to life. The final versions of these sketches become part of the game's Bible and guide other members of the art team so that the game has a cohesive feel instead of a jumble of conflicting styles. The concept artist also works with the designer to storyboard cut scenes so everyone is on the same page when actual production begins, and no time is wasted creating unnecessary material.
- **Character artists** design and create people, creatures, and objects. Working from the concept art, they make a 3D wire mesh and then apply textures to that mesh (although this "skinning" is sometimes a separate subspecialty). They might start entirely from scratch, or perhaps create a real-world 3D model in clay, scan it in, and work from there.
- **Animators** give life to the creatures by making them move. They receive a list of all the activities the creature will perform in the game, and then they create a series of movements for each. When deciding how to animate human characters, project managers must choose between artist-generated key-frame animation, and the more expensive—but more realistic—motion capture technique. With key-frame animation, an artist draws a character in key positions during the course of a movement, and then uses software tools to generate the character's motion between those positions. Motion capture, on the other hand, is generated by recording the movements of a live human using sensors on an actor's body. This technique is now widely used in sports and action games.

- **Background modelers** build the worlds through which the player moves. They generally start with basic geometric shapes (called primitives) and combine and deform them to create the rooms and objects that make up the game's environment. After the wireframe mesh is completed, they add flat shading, then textures, and finally lights, all of which are needed to bring the world to life.
- **Texture artists** create the "skins" that fit over the modeler's wire meshes. For background textures, they generally work in 2D, painting surfaces that are then stretched over the geometry so that a wall looks like brick, stone, or metal. Sometimes they create these textures from scratch, building them up layer by layer. However, sometimes it's more effective to photograph an existing surface, scan it, and touch it up. For character textures, 3D painting packages are gaining in popularity. These allow artists to paint in a WYSIWYG (What you see is what you get) environment, instead of constantly shuttling between 2D and 3D, tweaking as they go.

Testing

Testing is not just a quick check for bugs before the game goes out the door. Testers begin to play a vital role in the development team as soon as the first code is written. At the very end, it is the testers, as much as anyone, who determine when the game will actually ship.

The Test Lead

The role of the test lead is to ensure that the game not only works, but is also fun to play.

Early in the project, the team will be small, and its goal will be to provide a tight feedback loop to the developers. On a day-to-day basis, the programmers will implement small pieces of code, take a quick look to see whether they fulfill their main purpose, and then ask the testers to bang on them to find any unintended side-effects.

As the project approaches alpha (the stage where the game is more or less playable from start to finish), the test lead brings on the rest of his team and writes his test plan. At this stage, his job goes far beyond merely reporting game crashes. He must be in sync with the designer and programmers because he must know from moment to moment whether the game is behaving as they intended.

He uses the design document as a starting point for his test plan, but a written document can never capture all the thousands of small choices made in the course of development. To supplement the document, he must have as complete an understanding as possible of what the vision guy has in his head.

A good bug-reporting software package is essential. If a bug report has an automatic version stamp, a programmer can quickly check to see whether it was reported before or after a given fix. A system that lets the test lead sort bugs by developer is also helpful because it is far less intimidating to a programmer if he is handed a few pages that contain only his bugs instead of a large sheaf that contains the bugs for the entire team. (This also saves time because he doesn't have to weed through the larger document to ferret out which bugs are his.)

When the project hits beta (the stage at which the game is content-complete), the test team will be larger, and the lead will establish daily tasks, telling each tester specifically what to look for on that day. The feedback to the development team at this point is vital because everyone is trying to make the final decisions concerning feature trade-offs.

If the project is a console game, the lead will send builds to the console company and coordinate with their testers. If the game is multiplayer, he might also set up an external beta test involving hundreds of volunteers.

As the project nears its end, most of the test lead's time will be spent managing the bug list that results from all these activities.

It is important for government project managers to understand that games are not a "zero-defect" world, and that it is recognized and accepted that every game ships with bugs in it. Most companies request the test lead to rate the severity of the bugs to help determine whether a game is ready to ship. Usually, "A" bugs are crash bugs or other bugs serious enough to prevent the game from being shipped. "B" bugs are quality problems that should be fixed, but if the game absolutely has to go out the door for other reasons, they won't be considered showstoppers. (Enough B bugs, however, generally equal an A bug.) "C" bugs are usually nice-to-haves or obscure problems that arise only on rare occasions.

At the end of development, the test lead will meet daily with the producer and the department heads to discuss outstanding problems and decide which will be addressed and which will be left by the wayside.

Testers

It's no accident that many game designers get their start in the testing department. Here is where one sees firsthand all the mistakes that can be made and how they can be fixed. There is probably no better training a designer could get than to spend a year in testing.

Testers are on the lookout for several things at the same time:

- **Is it fun?** Early in a game's life, this is a question to ask again and again. Are the basic gameplay mechanisms enjoyable? Even though the game isn't balanced or tuned yet, can the tester see where the fun is going to come from? His feedback to the designers and programmers during this pre-alpha development will have an enormous influence on game design.
- **Is it easy to use?** Are the controls awkward? Is the interface well laid out? Is the manual accurate? Has the designer helped orient the player to the gameplay and the game's goals?
- **Does it make sense?** If the player follows along, will he get the experience the designer is looking for?
- **Is it fun (part 2)?** As the game approaches alpha, this question emerges again. Earlier, the tester was examining the basic ideas to see whether they are enjoyable. Now, he is testing the game itself to see whether the fun has survived the implementation. Is it too difficult? Too easy? Are there places where the player will be lost or not understand what he is supposed to do?
- **Does it work?** This is the task most people think of when they think of game testing. If the tester plays through the game doing what the player is supposed to do, can he get to the end? If he does things the player isn't supposed to do, does it work anyway? Does it perform according to spec? Can he make it crash?

These are all tasks that are generally handled in-house at a development studio. Other tasks, such as voice recording, music composition, sound effects, and manual writing are often out-sourced, which is discussed next in Outside the Game Studio.

V. Outside the Game Studio

Few game developers have enough talent in-house to create everything that goes into a game.

Voice acting, music, sound effects, and video are all important game elements that have routinely been handled by outside professionals. Localization (translating the manual and adapting the game for foreign markets) is another task that has traditionally been done by external teams. With the advent of globalization, companies are now outsourcing even more work, including elements of art, programming, customer service, and quality assurance.

Voice

In the early days of the industry, voiceovers were often performed by members of the development team or their friends, using inexpensive recording equipment. No more. Modern games generally use professional talent, and AAA games often feature well-known Hollywood actors and actresses. ("Triple-A" titles are those with the biggest budgets, greatest visibility, and highest expected sales.)

The key person in this process is the voice director, usually an independent contractor, who is familiar with the talent pool of actors and the myriad requirements of working with unions such as SAG (Screen Actor's Guild) and AFTRA (American Federation of Television and Radio Artists). The voice director usually conducts auditions and helps with the casting, makes the arrangements with the recording studio, directs the recording sessions themselves, and handles the mountain of paperwork that accompanies these tasks.

While some designers and writers leave the recording sessions completely in the hands of the voice director, most prefer to be present. Because the actors have little context for the lines they are given, it's helpful to have someone at the session who knows the big picture, someone who can explain the nuances of a line and, for example, say whether it should be read with sarcasm, irony, or despair.

Music

Music has become an essential part of game-making. Music can heighten the thrill of action, tell the player when danger lurks around the corner, or set a lighter tone for comedic moments.

Some developers have in-house musical talent, but most turn to outside composers to create the music that will go into their games. Some of these composers perform and record the music themselves, using special synthesizers that put a wide range of instruments at their fingertips. Others bring in bands or even full orchestras to perform the score.

As with many external contractors, there are two sides to working with a composer/musician—the creative side and the administrative side.

The creative side begins with technical direction. The composer needs to know the game's target hardware platform and what kind of music it will support. PCs and all next-generation consoles allow the use of full RedBook audio (regular CD audio), but handhelds and Internet-delivered games probably still require MIDI music, which doesn't sound as good but which uses very small files.

Once the platform is established, the composer will collaborate with the development team to decide how much music will be in the game (usually measured in minutes), and whether it will be a continuous soundtrack or discrete pieces of music that will play at key moments.

On the administrative side, composers will be concerned about payment and rights. These can be tricky to negotiate because although there are established norms in other areas of their business (TV and movies), the game industry is groping towards a different set of standards.

In Hollywood, composers are generally paid by the song (or score), and they retain a host of rights that provide an income stream from their music for years after they do the work.

In the game business, companies want to pay by the finished minute, they want to acquire all the rights, and they want to avoid entangling the project in royalty accounting and down-the-road payments. They also want to avoid restrictions that can prevent them from repackaging the game and managing it through its normal lifecycle, from

front line, to marked down, to budget, to compilation, with manufacturer bundles and other OEM deals thrown in along the way.

A common compromise is for the company to make a one-time payment to acquire all the rights for the life of the game, while the musician retains the rights for "non-interactive" use so that he can sell the music again in other arenas.

Some game producers favor using already-existing songs or music from popular musical groups to increase the marquee value of their games. Not only is it questionable whether this is effective, but it can also be prohibitively expensive and create nightmare back-end problems with royalties and rights.

The whole area of licensing music is a complicated minefield that is best negotiated with the help of a lawyer. Fortunately, a new kind of agreement—a new media license—is evolving that should one day make it much easier to license music for Internet and computer game use.

Sound Effects

Sounds can be used to immerse the player in the gameworld, provide feedback for his actions, and give clues that help along the way.

Not long ago, the only sounds a computer could make were the beeps and boops that came from its tiny internal speaker. Then came several years when some gamers had sound cards and some didn't. During that time, PC game designers couldn't make sound an integral part of the game because they were never sure whether the player would be able to hear it. Now, every gaming computer and videogame console ships with sophisticated sound capability, and sound design has taken an important role in overall game design. (It should be noted, however, that not all government computers are "gaming" computers, and their sound reproduction capabilities may be limited.)

In the real world, background noises are the soundtrack of our lives. No matter where you go, there is a constant hum of background noise. (One encounters pure silence only in artificial situations, such as anechoic chambers.) In games, background sound effects can be used to establish ambiance and atmosphere. They become part of the stream of concrete details that help make the fictional world seem real.

These ambient sounds give life to a scene. The player doesn't focus on them, but would notice if they weren't there, just as you would notice if all the background noise around you were to suddenly stop. Some games, of course, won't have ambient sounds—board games, card games, and trivia games are rarely set against an audio background, although they can have event-based sound effects and perhaps some background music as well.

Event-based sounds serve as feedback to the player's actions. These can be realistic, such as a golf club swishing through the air and hitting a ball, or artificial, such as the *ka-ching* of an arcade-style game that lets the player know when he's racking up points.

Sounds can also provide gameplay hints to the player. In an action game, if the gamer comes to a door and hears a monster roaring on the other side, he should know to expect danger when he opens that door. In a driving game, if the player is out in front and suddenly hears the sound of another car behind him, it's a pretty solid clue that he'll soon be challenged for the lead.

In movies, two people add sound effects, the sound FX editor and the Foley artist. The sound FX editor creates the big noises—jets taking off, bomb explosions, subway trains screaming through a station. The Foley artist creates the small noises—footsteps walking down a gravel path, the click of a key in a lock, the tinkle of ice cubes as they fall into the glass. (The job is named for Jack Foley, a Hollywood pioneer who helped studios make the transition from the silent era to talkies. Using a bunch of old props and a lot of ingenuity, he re-created noises in a sound studio that were not easily captured on the movie set and then synced them to the action on the screen. Many techniques he invented are still being used in studios around the world today.)

In games, both the big and small sound effects are selected or created by the sound designer. His job is easier than Jack Foley's because now entire sound libraries can easily be bought. Every sound designer has one, but he generally uses it only as a starting point. It is not uncommon for a sound guy to go out into the world armed with a microphone and digital recorder to capture the natural sounds that occur around us and then return to his studio to twist them electronically into the screaming dive of a wounded jet fighter or the pulsing heartbeat of an alien monster.

External sound designers are generally paid based on the number of sounds they are asked to provide. They should work out a file-naming convention with the tech lead so

that the files are easy to identify and track. They should also know as much as possible about the game so that they can bring their own experience and creativity to the task.

Video

Shooting Full Motion Video (FMV) had a wave of popularity that has now receded. Most action games deliver their cut scenes with in-engine graphics. These graphics are less complicated to produce, leave control in the hands of the game creators, cost less, don't interrupt the suspension of disbelief with a jarring visual style, and involve fewer legal hassles.

If the plan is to include FMV anyway, the very first step should be to hire a producer. The complicated business of hiring actors, finding or designing costumes, booking a studio, renting equipment, finding props, hiring a crew, doing the shoot, and overseeing postproduction is no place for an amateur. A producer is especially needed to handle the regulations of unions such as the Screen Actors Guild (SAG), the Directors Guild (DGA), and the Teamsters.

There are plenty of independent directors and producers in almost every city, and they can handle of all these details. They hire the studio and equipment, find the talent and crew, and handle the morass of paperwork. They need to know the budget they must work within, and the game producer must understand the trade-offs the video director will have to make to live within that budget.

Most live-action filming is done in a studio against a solid-color background, with the actors composited over a pre-rendered background later. This process is called chromakeying. The first part of the word comes from the single-color background: this color, or chroma, is selected to provide the most contrast to the subject, and when filming people, this color is usually green or blue because there are no green or blue tints in human flesh. During postproduction, the chroma is deleted, and the actor is composited, or keyed, against a pre-rendered background.

Motion Capture

The most realistic human animation is generated through motion capture. This is done in a special studio where a technician places optical sensors on key spots on an ac-

tor's body and then makes a digital recording of his movements. The recording is then imported into a graphics package and manipulated by an animator.

When this technology was young, it often yielded an overwhelming amount of data that would have taken longer to sort through than to do the animation by hand. Now, however, most houses provide cleaned up data that is much easier to use.

Each studio has its own fee structure, but one can expect to pay a flat day-rate for the facility, labor, and equipment, plus a price per second for hand-tweaking the data afterwards. Separate arrangements must be made with the actors for their fees and releases.

Some companies also sell stock data that can be purchased and applied to models the art team has already created.

Language Localization

These days, more than half of a game's revenue is likely to come from outside the United States. Publishing has become a worldwide business, and making multiple-language versions of games is standard practice.

Furthermore, many publishers insist on releasing localized versions on the same day worldwide, which means that the localization process must be included in the original production schedule rather than tacked on afterwards.

The worst way to localize is to wait until development is almost over, go back through the code, strip out all the language-related elements, and ship them off to a translator. Pasting in these translations line by line when they come back is time-consuming and prone to error. Dealing with other problems as they pop up can also destroy the schedule. The best way to localize is to plan for it from the start.

While the interface is being designed, it is important to remember that words in foreign languages (especially German) can be up to three times longer than their English counterparts. This is important when allocating space for things like menus. Status bars in particular can be a problem because their width is limited to the width of the screen. The producer should consult with a translator ahead of time to see whether suitable abbreviations can be used to save space. (Localization firms are certain to have encountered this problem before and might have a ready solution at hand.)

When the game displays text messages, the size of the text boxes should never be hard-coded. English is a compact language, and different grammatical structures can cause translated sentences to be considerably longer than the original. Instead, the box should be dynamically sized so that it can automatically expand to fit whatever it contains. Likewise, if the game requires the player to type in information, make sure that the entry fields are much larger than would be necessary for English.

Another trap created by the compactness of English is that animations linked to dialog are likely to be too short. This is not just a problem of inferior lip syncing. If a cut scene contains a dialog between two characters and the camera cuts back and forth between them as they speak, the timing of the cuts is almost certain to be wrong for the foreign language versions. The localization team should receive the cut scene animations *before* they do the translations so that they can trim the dialog to fit.

It is a bad practice to sprinkle text strings throughout the source code. Instead, one file should be created that contains all the text that is to appear on the screen. This saves countless hours when it comes time to extract text for translation and countless days when it's time to integrate the translations back into the game. If what goes to the localizers is one big file, with each line properly numbered, the localization may be completed within days, instead of weeks or months.

If the game is to be localized into Asian languages, the programmers need to assign two bytes for each text character, instead of one.

Text or speech should not be algorithmically generated—in other words, sentences should not be constructed "on the fly." In the early days of game development, when space was excruciatingly tight, it was not uncommon to cobble together words and sentences based on a set of rules, because it was much more efficient to store the rules than it was to store all the variations of the words themselves.

Consider the following: *You search the man but don't find his key.*

Depending on the game situation, one might need to change the subject of the sentence (you) to he or she and to change the objects (man and key) from singular to plural. In code, it wouldn't be too difficult to store all the variations. The changes, if plugged into the sentence, would look something like this:

You/(s)he search(es) the man/men but do(es)n't find his/their key(s).

The code could very efficiently check the situation for the details and spit out the right sentence. However, if one were to store all these sentences separately, they would be

You search the man but don't find his key.
He searches the man but doesn't find his key.
She searches the man but doesn't find his key.
You search the men but don't find their keys.
He searches the men but doesn't find their keys.
She searches the men but doesn't find their keys.

Even though the latter method requires more storage, it is the correct one to use if the game is to be localized. Space is cheap, and the last thing a programmer should be doing is trying to develop algorithms for generating text using multiple foreign languages and grammar.

Drive letters and paths for filenames should not be hard-coded. Not all countries use C to designate the main hard drive.

Text should never be embedded in graphics, especially text that is critical to playing the game. The cost of generating and storing multiple versions of the art is likely to be prohibitive. Similarly, icons on buttons should be used instead of words wherever possible. When translated, the words are likely to take up more space than is available on the button.

Finally, it is important to remember that localization applies to more than just words. Different countries have different standards regarding what they will allow to appear in games. The most notorious example is that Germany will not permit the image of a swastika to be published, causing problems for games staged in the World War II era. Other countries, such as Brazil, have tough standards concerning violence and bloodshed.

The Manual

Writing the manual is another task many companies turn over to external resources, and it's more difficult than most people realize. By the time the final line of code is written, all the in-box materials are already sitting in a warehouse waiting for the game discs to show up. Game features change right up to the moment the master disc goes into duplication, yet by then the manual is already printed. (This is why so many games come with readme files.)

The first mission of the manual is to ensure that the gamer has enough information to install the game and get it running, and its second mission is to tell him how to play it. As the size of game boxes has decreased, manuals have shrunk as well—most are now designed to fit inside a jewel case—and there might not be room for the elaborate backstory that used to come with most games.

The way many companies handle the job is to have someone (usually the designer or producer) create a rough draft during the pre-alpha phase of production. This draft covers the major features of the game and explains the controls but has many portions marked TBD (To Be Done).

At this point the draft is turned over to the manual writer, who generally works with an assistant producer or lead tester to track the game while it's in progress, filling in all the TBDs as the information becomes available. The writer might request screen shots or other graphics to help explain the features or just to make the manual look more interesting. The writer also works with the legal department to make sure that all the notices required by licensors are present, as well as the trademark and copyright information for the game itself.

The writer continues to play-test the manual against the game right up until his deadline. After the manual is in production, the producer assumes the responsibility of tracking last-minute changes to the game and creating a readme file with anything that didn't make it into the manual.

Artwork

Modern games have significant amounts of artwork, including animation, 3D models, textures and 2D art. Similar to Hollywood, which outsources much of its special effects work overseas, game companies are starting to do likewise to allow for more rapid and less costly development. Outsourcing can also occur in-country to augment internal staff productivity, although this rarely saves much money.

Outsourcing artwork requires clarity from the team on exactly what assets must be created, what style is required, and what the tool pipeline will be. In addition, the developer must be prepared to send a representative to spend significant time onsite with the outsource team to train and evaluate its people.

The way many companies outsource artwork is to have the art leads for each major area create reference art, usually at the game prototype stage. These leads, under the direction of the art director, then create complete asset lists and requirements for the art to be outsourced, ensure the tools and pipelines are as well-documented and efficient as possible, train the outsource artists directly, and then review every incoming asset before it is added to the game build.

Programming

Traditionally, game programming has been a highly integrated and iterative process. As games move to larger productions with more pre-production planning, it has now become common to outsource some programming tasks.

As with all outsourcing, it is essential to clearly understand the deliverable requirements. These requirements include duplicating the development environments between the core team and the outsource team to reduce potential errors and misunderstandings. Ideally the outsource team should be able to build the entire product and submit their work to the QA department in the same way as the core team. It is critical to specify what code needs to be written, how it will be acceptance-tested, and how it will interface with related game systems.

The technical leadership of the core team must "own" this specification process, and they must also manage the relationship with the extended outsourced team. All code deliveries should be validated and reviewed prior to acceptance. As with artwork, it is critical to the process to allow the outsource group to fix any problems to insure the best results from that group over the long term.

Customer Service

Customer service comprises several activities. Technical support, which helps people install and get their game running, is common to both stand-alone and massively multiplayer games. Many companies outsource technical support, particularly email and online chat. Telephone-based technical support is less frequently outsourced due to customer backlash to overseas accents. Game support for email and online chat is similarly outsourced. Billing support is typically only needed for massively multiplayer online games. Billing is the task least often outsourced due to requirements concerning the

confidentiality of credit card information. However, more and more companies are building the relationships that will soon allow them to outsource billing as well.

Quality Assurance

Quality assurance, the testing of game software, is also commonly outsourced. Companies use quality assurance firms both in and out of country to augment their internal staff. The most effective use of these external teams is to create detailed test plans that they use to ensure that every element of the game matches the specifications. This allows the core team to offload the validation of new builds, which is a time-consuming and repetitive task.

Legal Issues (Getting the Rights)

An important part of working with external resources is acquiring the rights to use the work they have been paid to do.

Under current US copyright law, a piece of art, music, writing, or code belongs to its originator from the moment it is created. This is true whether or not the individual takes any formal steps to register his work.

The only way a company can acquire the rights to his work is if he explicitly assigns them to the company, and this must be done in writing—a verbal agreement will not do.

Acquiring these rights is tedious but necessary. Failure to obtain signed agreements from every contractor can result in the game not being published, or its failing to become an asset should the company be sold. (An acquiring company will want to ensure that they have complete rights to the title so that they can republish it without fear of legal problems.)

The best way to go about this is to be relentlessly methodical. First, the company needs a rights-transfer agreement that has been approved by its legal advisors. Usually, this is a work-for-hire agreement that broadly assigns all rights to the company. Then, an assistant producer is usually assigned the task of getting that agreement signed by every single contractor the company works with, at the moment they first work together. This includes every voice actor, composer, and artist, no matter how small each person's contribution.

A good rights-transfer agreement allows the game company to do demos, run ads, and use the material in other games or sequels, all without having to go back to the creator for additional permissions.

Similarly, if a game company wants to incorporate a piece of already-existing material into a game, they must acquire the rights to it (unless it is in the public domain). Whether it's a snippet of a song, a clip from an old TV show, or even an old movie poster—if it has already been created, someone owns it, and the company must acquire the proper rights before they can use it.

Small developers have a tendency to overlook these legal issues, figuring that no one will ever pay attention. This is a mistake that must be guarded against.

In all these matters, the only way to be certain that all the bases are covered is to work with a lawyer, preferably one who is experienced in intellectual property law in general and entertainment products in particular.

Just as important as assembling the development team and having a vision are the techniques that are used to coordinate the team's efforts and keep the project on schedule. What development model suits the job best? How will the changes that inevitably visit every game project be handled? We discuss these considerations in the next section about Project Management.

VI. Commercial Game Project Management

Before it is anything else, building a game is a software development project, and the keys to managing it successfully are selecting the right lifecycle model, understanding the design goals, creating a reasonable tech plan and schedule, anticipating and avoiding problems during development, and dealing effectively with problems that arise anyway.

Lifecycle Models

The discipline of software engineering recognizes several formal approaches to design and development. Some of them are best suited to very small projects, with only one programmer and just a few thousand lines of code. Others are suited to very large projects, where there can be 50 or more developers and millions of lines of code.

Most game projects fall solidly into a medium category, with teams of four-to-eight programmers and a few hundred thousand lines of code.

Given the size and nature of these projects, the most appropriate lifecycle models are generally the waterfall, the modified waterfall, and iterative prototyping (or "agile" development).

The Waterfall

In a perfect world, software development would follow the classic waterfall model, which includes the following steps:

1. Concept development
2. Requirements analysis
3. Architectural design
4. Detailed design
5. Coding and debugging
6. Testing

This orderly progression through a defined set of processes works best when everything is well defined up front and will not change. As we have already seen, the game

world is usually more chaotic than that, so this model is best suited to projects like sequels to established games, where the basic gameplay mechanics are already known, and other major subsystems—such as the game engine and interface—are also already set. If the task is merely to give a product a facelift, add a few new features, or build some new levels, the classic waterfall is a good choice.

The Modified Waterfall

The modified waterfall allows for overlapping steps. For example, coding on some sections can begin before detailed design is complete, especially if those subsections are understood and relatively independent modules.

As an example, the user interface for a trivia game can be designed and coded early in the project, while the questions themselves may be developed anywhere along the way.

Another genre that lends itself to the modified waterfall approach is the adventure game. In these games, meaningful interaction with the environment is important to the gameplay, but the designer usually will not know exactly what those environments look like until the artists create them. Even though the basic story and puzzles are known at the time development starts, specific interactions might have to wait until the designer can see the world the artists have created.

Iterative Prototyping (Agile Development)

Rapid iterative prototyping is the best development model for most new games (i.e., non-sequels or games with innovative features.) Whether one calls it Extreme Programming, *Crystal*, Adaptive Software Development, Dynamic Systems Development, SCRUM, or simply Agile Development, the idea is to get a rough prototype up and running quickly, play it, keep what you like, throw out what you don't like, then go back and build another one. Repeat as needed.

Sid Meier, designer of the classic games *Civilization*, *Railroad Tycoon*, *Pirates!*, and many more, is famous in game designer circles for saying that he always gets the kernel of a new game up and running as quickly as possible, even with just stick figure graphics, so that he can be playing the actual game throughout the development process. Will Wright, who designed *Sim City* and *The Sims*, routinely does hundreds of mini-prototypes on the way to building his products.

The core of Agile Development is to deliver "customer satisfaction through early and continuous delivery of useful software components." (In context of a game development project, the "customer" is the game publisher or the government client.)

The key to this lifecycle model is to refine the design continually, based on what has been built so far. At the beginning of each cycle, the team sits down to consider what needs to be done next. The customer writes out "use cases" or "user stories"—succinct statements in plain English of things he needs the game to do. The development team attaches time estimates to these, and then everyone sits down in a horse-trading session. If the customer learns that a requested feature will take two weeks, but that 10 others could be done in one day apiece, he might decide he'd rather see those 10 features implemented first. Or not. The goal is to have the customer intimately involved in the process, taking responsibility along with the development team for the decisions made as the game is built.

Real-time strategy games are perfect candidates for this lifecycle model. As units are created and features added, unforeseen gameplay dynamics and strategies emerge, some of them good and some bad. Because flexibility is built in to the process, the bad dynamics can be weeded out and the good ones allowed to flourish.

Action games might do well to mix the modified waterfall and the iterative prototype for different portions of the development process. The asset-creation portion of the task lends itself to the waterfall model. Creatures, weapons, and environments can all be designed, specified, and built in an orderly process. The gameplay portion of the task lends itself to iterative prototyping, where the AI for the creatures and the operation of the weapons are designed, coded, tested, and refined in a series of prototypes.

Agile Development is most useful in projects where the requirements are not well understood at the beginning, where new technologies must be created, and where the customer is willing to be deeply involved in the development process. This perfectly describes most major games in development today. It is fair to say that the Agile Development model has "won the race" against more traditional models such as the Waterfall, and that to get the most out of working with game companies, government clients should be prepared to embrace its principles.

In their famous "Manifesto for Agile Development," the proponents of this model say they value:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan[2]

"That is, while there is value in the items on the right, we value the items on the left more."

The principles behind this manifesto are:

- Satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development.
- Deliver working software frequently, as often as every few weeks.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- Face-to-face conversation is the most efficient and effective method of conveying information.
- Working software is the primary measure of progress.
- Promote sustainable development. The sponsors, developers, and user should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design.
- Simplicity—the art of maximizing the amount of work not done—is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

In summary, selecting the right lifecycle model for a game is an important first step to getting the project off on the right foot. Project managers must take a close look at the feature-set they hope to implement, and as a general rule of thumb, the more unknowns, risks, and innovations they plan to include, the more they should move away from the traditional Waterfall and embrace the principles of Agile Development.

---

[2] Beck, Kent, et al. "Manifesto for Agile Development." 2001. *agilemanifesto.org*

Understanding Design Goals

The game design document is the equivalent of a requirements analysis in the world of formal software development. It lists the features the game should have, describes the user interfaces, defines the art requirements, and so on.

Starting to code a waterfall project without a design and tech plan is a mistake many teams make, especially teams under schedule pressure from the very start. However, study after study has shown that short-changing the critical "upstream" activities—such as effective planning, design, and establishing the scope of the project—results in cascading problems down the road.

When using Agile Development, it's vital that a team understands the game's design goals at the beginning, even if they don't yet know the particulars of how they are going to accomplish them. These goals should be easy to articulate, and it's a good idea to display them where the team can often see them. Establishing them at the beginning gives the team a benchmark against which they can evaluate suggested features. Although individual elements of an initial design often change during development, the overall goals will most likely remain stable.

Planning and Scheduling

Building software is one of the most complicated tasks known to man.[3] At the risk of brutally simplifying the job of creating a software project schedule, the best method is to follow three steps:

1. estimate the size of the project.
2. estimate the effort in man-weeks or man-months that it will take to build something of that size.
3. apply the man-month estimate to the number of people on the team, and spread it out over a calendar schedule (while making sure to allow for overhead activities such as meetings, holidays, vacations, and so on).

---

[3] Irlam, Gordon and Ross Williams. "Software Patents: An Industry at Risk." Cambridge, MA: League for Programming Freedom, 1994. *http://lpf.ai.mit.edu/Patents/industry-at-risk.html*

"In most other industries, a product will contain perhaps twenty parts. In the case of sophisticated consumer goods, such as video cameras, we could raise this to 1000 parts. Nevertheless, the constraints of the real world ensure that the complexity of the product cannot become too great. Software, however, is essentially free from these constraints. A major computer program can comprise anywhere from 100,000 to 10 million lines of code."

If this three-step process is so simple, why doesn't it always work? Why do studies show that two out of every three software projects significantly overrun their schedules?

Here are some of the most common problems that cause projects to be late.

Scope

Scope is the single largest factor in determining the schedule of a project.

In the world of formal software design, there are many approaches to estimating the size of a project, but it remains an extraordinarily difficult task. One can buy estimation software, bring in outside experts, estimate the number of function points or the total lines of code, eyeball the different modules and add up the components to get a total, and so on.

All these methods are reasonable, however, in the game industry, the most reliable method of estimating a game's size is by comparing the features to those of a similar game the team has worked on in the past.

Naturally, this means that the more experience the team has, the more accurate their estimates will be. Nevertheless, because the business is driven by innovation, parts of the project are sure to be new, whether they are the individual features, genre, hardware platform, point of view, art style, gameplay modes, or whatever. For each of these, estimates are likely to be off because the team either overestimates or underestimates the complexity of the unknown tasks.

This means that it is important for everyone to recognize that estimates made early in a project have a high margin of error. As the project progresses, the unknowns become known, and the margin of error decreases.

Whatever estimation method is used, it is important to be aware that scope is the single biggest factor determining a project's schedule. Big projects take longer than small projects. The more features a game has, the longer it will take to design, build, debug, integrate, and test them. Because of this, increasing the size of the game results in a greater than one-to-one increase in the effort. If the team has a schedule problem, one of the first places to look for relief is features and scope. "Cutting the size of a medium-size program by one-half will typically cut the effort required by almost *two-thirds*."[4]

---

[4] McConnell, Steve. *Rapid Development Taming Wild Software Schedules*. Redmond, WA: Microsoft Press, 1996.

External Pressures

Too often, schedules and budgets are set by executives who aren't intimately involved in the project, and who haven't researched the effort necessary to deliver the features.

There are many good business reasons to target a particular date for releasing a product. However, creating an inflexible schedule and budget without regard for the feature-set is a recipe for disaster.

Before a team signs up for a schedule, it is absolutely critical that everyone up and down the line agree that it be based on a reasonable level of effort for the features that will be delivered, rather than on an arbitrary date. If the date is fixed (as it sometimes must be), the team must adjust the feature set and task lists to fall within the schedule.

Padding

Project managers often pad schedules because they figure that something's bound to go wrong that hasn't been anticipated. They hide the padding from the team because they think that an aggressive schedule will motivate the team to work harder, whereas a "relaxed" schedule will fall prey to Parkinson's Law: Work expands to fill the time available for its completion.

This is wrong. It creates credibility and trust problems, and when a team is faced with a short schedule, the first thing they do is abandon the practices that enable them to be efficient and fast. They grab whoever is available for the project rather than find the people best suited to the tasks. They jump into coding right away, without taking the time to understand the requirements or how the various pieces will fit. Design is short-changed. People start working 80-hour weeks right off, and burn out within a month.

The more rational technique is to create a bottom-up schedule, working with the developers from the start to achieve a reasonable, efficient schedule, modifying the project's features, scope, and goals to be achievable within the time available.

Altered Requirements

Another torpedo in the hull of a good schedule is a sudden or drastic change in scope. An executive may decide that a new gameplay feature is required, or the order may come down that the game must contain data-analysis capabilities or interoperability

with another product. Regardless of the reasonableness of the request, it's unrealistic to think that a major feature can be added to a game without changing the schedule.

While agile teams specifically welcome altered requirements as the project is in process, the customer must participate with the team to assess the impact on the schedule and to make the necessary trade-offs to keep the overall project in line with the company's goals.

Developer Optimism

The problem of creating overly optimistic schedules cannot be laid entirely in the hands of management. Sometimes developers shoot themselves in the foot with their own scheduling practices.

The most common developer problems are (1) they don't take enough time to plan properly, (2) they forget or overlook important tasks that belong in the schedule, and (3) they're just plain optimistic. Studies show that developers generally underestimate their own tasks by 20–30%.[5]

Waiting for the Prototype

No matter how complete the game design is at the start of development, it still provides an incomplete picture of what the final game will be like. Often, the only way to get a clearer picture is to build what has been specified and then examine it to see whether it's *fun*.

This is the most intractable problem of game development. Fortunately, it can be anticipated and managed. In practice, building a game is a process of constantly adjusting the design. If the game isn't fun, the design must change. If the design changes, the requirements change, and if the requirements change, the schedule changes, too.

This dilemma remains the core reason why so many games are late and over budget.

It is important to understand that features, schedule, and cost are *always* imprecise until the day a game ships. This does not mean that one cannot lock down the schedule

---

[5] van Genuchten, Michiel. "Why is Software Late? An Empirical Study of Reasons for Delay in Software Development." Los Alamitos, CA: IEEE Computer Society, 1991, pp. 143–152.

and cost: one can—if the features are flexible. Neither does it mean one cannot have all the features one wants: if the schedule and cost are flexible.

The unfortunate truth is that everyone involved with a project must be willing to participate in the horse-trading around the fundamental trade-offs of schedule, features, and cost. Otherwise, the project is doomed.

Avoiding Problems

Certain problems routinely arise over the course of development. Postmortems often look startlingly familiar: overly optimistic schedules, disruptive team members, feature creep, and more.

Classic Mistakes

In his seminal book *Rapid Development*, Steve McConnell identifies and discusses several classic mistakes that managers make on software projects.

- **Undermined motivation.** Game developers are not usually motivated by the same things that inspire the general workforce, and not all developers are motivated the same way. Project members need to have reasons to do a good job that make sense to *them*.
- **Weak personnel.** Skill levels vary among programmers by as much as a 10-to-1 margin. Waiting to get better people can be the smartest decision one can make on a project.
- **Uncontrolled problem personnel.** When a person doesn't work well with the team, bad things happen. The worst of them is that the rest of the team's morale suffers as they see unacceptable behaviors tolerated or even rewarded.
- **Noisy, crowded offices.** Programmers who have their own offices with doors they can close and phones they can turn off to avoid interruptions are up to 2.6 times more productive than those working in "bullpen" or cubicle environments (*Peopleware*).
- **Contractor failure.** Maintaining a schedule can be a nightmare when the team is relying on an external group to deliver a vital piece of technology or art, and that group doesn't deliver.
- **Requirements gold plating.** Sometimes a project is too ambitious in too many areas simultaneously. Successful projects often try for just a few innovations, and are content to rely on tried-and-true features for the rest, embracing them for their predictability and low risk.

- **Developer gold plating.** Managers aren't the only ones who ask for extra features. Very often in the course of development, team members themselves think up cool new features they want to add to the game. Often, they don't realize the effect these unplanned additions have on the schedule because few game elements stand completely on their own. The extra feature might have to be supported by additional art, AI, programming, or even changes in the design.
- **Insufficient management controls.** Team leaders often have no meaningful way to track progress to determine if the project is on schedule.
- **Omitting necessary tasks from estimates.** Some tasks never seem to make it onto the tasklist but do, nevertheless, take up valuable time. Commonly overlooked activities include new-hire interviews, project meetings, code reviews, creating screenshots for marketing, and days missed because of tradeshows or industry events.
- **Misunderstood tasks.** Sometimes tasks make it onto the schedule, but there is a misunderstanding about what the person who requested the task really wants. The longer these misunderstandings persist, the greater the impact on the schedule is likely to be.
- **Distributed development teams.** If the team is geographically dispersed, communication often breaks down.
- **The Not-Invented-Here syndrome.** Many developers are mistrustful of software that was not developed in their own shop. Yet, often it is better to buy than build technology.
- **Pop-up tasks.** Throughout the course of development, the team may be asked by management to provide people for unplanned tasks that just seem to pop up. It could be anything from sending someone to a VIP's office for a demo, to asking artists prepare high-resolution screenshots for the PR team.
- **Waiting too long to fix bugs.** Some developers claim that there is no point in fixing bugs until near the end of the development process. This is wrong. Fixing bugs not only uncovers other, hidden bugs but also creates more bugs! "Fixing a defect has a substantial (20-50 percent) chance of introducing another." (*The Mythical Man-Month*) Bugs should be fixed as they are found. Doing so significantly reduces the unknowns in a project and gives the schedule greater accuracy as the end approaches.[6]

---

[6] McConnell, Steve. 1996.

Recovery

The deadliest problem in a project is that the team often doesn't know what to do once it recognizes that it is behind. When projects start to slip, managers typically have one of several reactions, all of which are ineffective.

Ineffective Strategies

- **Plan to catch up later.** "They're two days behind, but we're only two weeks into the schedule, so there's plenty of time to make it up." Most often this is coupled with some sort of denial that a slip has actually occurred. A slip of two days in two work-weeks is actually a twenty-percent slip!
- **Require mandatory overtime.** "Everyone will just have to work harder." Although asking for small amounts of voluntary overtime can sometimes be effective for a short time just prior to major milestones, studies show that extended periods of mandatory overtime result in a significant drop in productivity. Prolonged overtime causes several things to happen: as developers tire, they make more mistakes, which results in more testing and reworking. People required to spend more time in the office start to attend to personal tasks, reducing their effective work time. Most importantly, their motivation plummets, and motivation is the single greatest predictor of productivity.
- **Add people to the project.** "If it's a question of man-months, let's throw on more men." Adding people to an understaffed project at the beginning is effective. However, adding people to a late project will almost never help it catch up. New people must come up to speed on the project, which means that other team members have to take time from their tasks to train them. The volume of email increases, team meetings take longer, communication becomes more difficult. "…when schedule slippage is recognized, the natural (and traditional) response is to add manpower. Like dousing a fire with gasoline, this makes matters worse, much worse." (*The Mythical Man-Month*)
- **Hold more meetings.** The more extraneous tasks people have, the less work they accomplish. The problem actually goes deeper than that. Intellectual work requires entering a state of flow, where the mind becomes fully immersed in the problem. Studies show that this process takes about fifteen minutes. Every time someone is interrupted, all the mental balls he was juggling fall to the ground, and he has to begin the process

all over again later. Creating a workday full of reports, meetings, and interruptions almost guarantees that creative work will be delayed.

- **Close their eyes and make a wish.** "It's probably not as bad as it looks, and everything bad that could happen probably already has." When managers engage in wishful thinking, they're hoping against hope that things will turn out right without their having to act. This never works.

Effective Strategies

If these strategies for recovery aren't effective, what strategies are?

Reduce scope

As we have seen, the biggest factor affecting a project's schedule is its scope and therefore *the single most effective way to cut a project's schedule is to decrease its scope.*

The fastest way to get a project back on track is to reduce the number of features it must deliver. Every eliminated feature creates savings in design, asset creation, implementation, integration, testing, and debugging. This still requires careful planning, however. Pulling features out of the game at the last minute can leave holes, create bugs, and demoralize the team. The solution is to prioritize, identifying the difference between core features and nice-to-haves from the very start of the project. This has enormous benefits:

- If development is geared towards finishing the high-priority items first, the team will be motivated to work efficiently so they'll have time to implement the features on their wish list.
- If the most important assets are created first, the team won't get caught short at the end. This also means they will be less likely to waste money creating assets that are never used.
- If everyone knows ahead of time that a feature might be cut, they are less emotional if it has to be killed. (This is especially true if it is someone's pet feature that he has been working on feverishly, only to learn at the last minute that it won't make it into the game.)
- Killing a feature won't create a gaping hole in the design, because it was known all along that it might get cut, so everyone has made sure the game would be complete without it.

In short, if schedule and cost are the wolves that are chasing the team, it helps to have a few features ready to push off the back of the wagon to lighten the load and keep the wolves at bay.

Effective Motivation

After scope, the second biggest variable in the schedule is the team's level of productivity. The largest influence on productivity is motivation, so it is important to keep everyone invested in the project. Piling on the pressure only leads to a vicious circle of increased stress, more mistakes, decreased motivation, and more schedule slips.

Other Strategies

When a project is in serious trouble, recovery is likely to come from a combination of strategies. The prudent manager does the following:

- Identifies short, discrete tasks that can be easily tracked so that people get used to feeling successful.
- Eliminates problem personnel.
- Changes the workplace to create a more developer-friendly atmosphere.
- Identifies core features and eliminates any activities that aren't geared toward delivering them.

Whether any or all of these strategies are pursued, the most important thing is to regain the team's confidence by creating a plan everyone believes in.

The process of putting a game in front of potential users—before and after it is finished—is almost as arduous as creating it, and it is done almost concurrently with development. We examine how mainstream games are sold in the following section.

VII. How Commercial Games are Marketed, Distributed and Sold

Once a game has been built, it must be marketed and distributed. Whether a game is meant for entertainment or more serious purposes, government organizations wishing to partner with commercial game companies must understand how these tasks are accomplished because they drive a company's metrics, motivation, and decision-making process. Plus, this *drive to deliver* will serve the government well—it does no good to develop a product whose intended users don't know it about or can't acquire.

It used to be that a new game had four-to-six months after its release to find an audience. Consumer advertisements were timed to appear a few weeks after the initial shelf date because marketers didn't see any sense in promoting something the public couldn't yet buy. Gaming magazines focused on *re*-views instead of *pre*-views because they wanted to see the final product before passing judgment. Television ads were unheard of.

No more.

Today, a game has as little as two weeks to prove that it belongs on retailers' shelves. Chain stores have specific targets for "turning" their shelves and are strict about enforcing them. If a game is not meeting those goals soon after it comes out, the publisher's sales group will come under immediate pressure for markdowns (sometimes called price protection) or returns.

Many developers don't realize that when their game goes onto a store shelf, it has not yet been truly sold. The retailer has the right to return it to the publisher for a 100% credit against future product. This creates an interesting struggle between the retailer, who wants to carry only games that sell well, and the publisher, who wants all his games to stay on store shelves as long as possible.

This balance of power is usually even: the retailer wants a supply of games to sell, and the publisher wants a place to sell his games.

Sometimes, however, the equation becomes unbalanced. If the retailer has a large backlog of the publisher's games languishing on his shelves, he can refuse to bring in more until the deadwood is cleared out. On the other hand, if a publisher has a hot new game coming out, he can try to muscle other games in the door on the coattails of the hit. This dynamic has made it difficult to survive for small publishers who bring out only one or two games per year. (A situation that may change as online distribution becomes more popular.)

Before a retail buyer will bring in a game, he has to be convinced that it will sell. He uses various guidelines to help with this decision: the pitch from the sales force, the sell sheet, the box art, the industry buzz, and above all, pre-orders. This means that the marketing, PR, and sales campaigns all must be geared to create maximum consumer demand for the game *before it even ships!*

Games almost never start slowly and then ramp up. Generally, their largest sales come within four weeks of their release, and their ultimate life is determined by how slowly their weekly sell-through numbers degrade thereafter. Exceptions occur, but they are rare and usually caused by chance factors, such as a popular movie suddenly sparking interest in a particular subject.

This pressure to create advance demand is why we see advertisements for games that won't be out for six months. It is why publishers push magazines to run previews of their games. It is why websites devoted to games-in-development have sprung up. More than anything else, it is why there is such pressure on the development team to deliver the game *on time*.

Hidden Pressures on Commercial Teams

Project managers who are considering partnering with developers on projects that have both a commercial and a government component should be aware of the behind-the-scenes activities that contribute to the pressures for the game to being completed on its scheduled date.

- The operations group must book a time slot at the disc duplication plant months in advance, especially before the busy Christmas season. For console games, a slot is reserved in the console company's manufacturing queue. If the game is late, another factory slot might not be available, and the game could miss its selling season altogether.

- The marketing group must commit to magazine ad space months ahead of time. The ads are going to run whether or not the product is ready. If the game is late, it could hit the stores with no current advertising push because the ads ran months previously and no money is left in the budget for new ones.
- The sales group buys premium shelf space in stores for the game. End caps and other special promotions are slotted out well ahead of time by the channel. If the publisher signs up for one and the game isn't there, the company pays for it anyway, and someone else gets the benefit of the premium placement. Then, when the game does come out, it will sell-in only a few copies per store.
- The PR department pushes, prods, begs, and pleads for premium magazine editorial coverage timed to hit the streets just before the game is due out. If the game misses the date, the "buzz" fades away, no magazine will give that coverage again, and the game comes out in obscurity.
- The marketing group can also decide to advertise the game on television, but they don't just want to spread 30-second spots randomly across the viewing day. Ad slots on programs with the right demographics for the game could be in heavy demand and must be booked well in advance. Furthermore, the sales bump from a television ad is immediate and short-lived. This means that it's pure folly to advertise on TV for a game that is not available on the shelves, and marketing directors will only commit TV dollars to a game they are absolutely convinced will be delivered on time.
- If the game is a Massively Multiplayer Online Game, even more must be in place on the day the game launches, including:
- The billing and login infrastructure to handle the influx of new buyers. Typically this part of the infrastructure will never see more simultaneous traffic than during the first week of sales.
- Sufficient servers set up to handle at least the first three months of projected sales. (Of course if the sales projections are wrong, the publisher will have either too much or too little equipment in place. To use a recent example, *World of Warcraft* launched with what they thought would be enough equipment for six months of sales, but due to the title's unprecedented popularity, after two months they had to stop selling retail units to give them time to set up new servers.)

- Staffs for networking, community, and customer service must be hired, trained, and in place for launch. If the game launches late, that means the publisher is paying dozens—if not hundreds—of people to do nothing while they wait for the game to be released.

The Publishing Team

Here is the structure of the team that creates this carefully orchestrated campaign.

Public Relations (PR)

Well before a game comes out, the people in the PR group will start beating the drum for it.

Their targets are trade magazines, websites, general interest magazines, newspapers, even radio and television. Their success is measured in previews, reviews, feature coverage, and the highest prize—magazine covers. Their tools are anything they can lay their hands on: demos, videos, concept art, interviews with the development team, and above all, screenshots.

Marketing

The industry sometimes makes an artificial distinction between marketing and public relations. In reality, PR is usually part of the marketing department, and the two groups work hand-in-hand to achieve maximum exposure for the game.

The marketing team has two goals: to help the developer target the game for a particular market and to persuade everyone in that market to buy the game.

To do the first, they study demographics. They advise the business group on the appropriateness of particular licenses. They try to determine whether certain features in the game will help or hurt with the target group. They say which of these features will affect the game's rating and how that rating will affect the sales. They help the developer understand cultural differences that can determine how the game is received in foreign countries.

To reach their second goal—persuading people to buy the game—they create an image for the game and try to let everyone in the target market know about it. For this, their primary tool is advertising.

When the marketing group is developing an image for a game, they are likely to fall back on the original high concept and to incorporate that message into all the various materials they develop, from magazine ads and sell-sheets to Web banners and TV spots.

Their decisions will be affected by the game's positioning strategy. Does the game target the hard-core or casual gamer? This decision, in turn, affects the game design itself. If the company is targeting the hard-core gamer, they must design in competitive features and depth of gameplay. If the game is going after the casual gamer, it must be easily accessible and not have any of the common barriers to purchase, such as excessive blood, sex, or foul language.

The marketing group can also suggest implementing specific features that will help the game find a broader audience and give it longer legs. Creating chat capabilities and lobby areas in online games, for example, helps build a stronger community and enables the game to survive longer and attract more people.

Sales

The sales force maintains personal relationships with the retail channel buyers who order a publisher's game in bulk. These buyers work for chain stores, national or regional distributors, video rental chains, and even hardware manufacturers (OEMs, or original equipment manufacturers).

Each of these customers requires a different approach, and each approach requires different game material to support it. Project managers involved in hybrid public/private games must be aware of how the sales team operates and what materials they will need from the developers during the course of the project.

Chain Stores

Chain stores make money only if they maintain a constant turnover of their inventory. Customers rarely think about this, but the continuing need to *move* product to pay the rent is the driving economic force behind every retail business. At some level, retail executives are not concerned with selling a game at all—what they care about is getting enough money to pay for the space on which the game sits.

In suburban malls and other prime retail locations, rents are high. Games that don't sell quickly aren't paying for the space they're sitting on, so they are sent back for

others that do. This is why the game industry has become such a "hits" business. Retailers don't care about providing a broad selection if most of that selection doesn't move. If they could get away with stocking only one product that sells enough to cover their costs, that's the only product they'd carry. There is no romance in retail.

As we have seen, this leads buyers in the chain stores to want a continuing relationship with large publishers who have a lot of product to move. They also want the publisher to provide money to promote the game at the retail level. This market development fund (MDF) money is used to buy special positioning in the store and ads in the store's flyers and newspaper circulars. (When consumers receive a flyer from a store crammed with small pictures of software products, it's not because the store has singled out those products as the ones it recommends—each of those square inches is actually an advertisement bought and paid for by a publisher.)

The amount of MDF a store asks the publisher for is generally a percentage of the anticipated sales for the year. At the beginning of the year, the chain sends around a booklet containing various combinations of end caps, shelf talkers, circulars, stand-ups, mobiles, storefront pyramids, and so on. The more business a publisher does with the chain, the more programs he is expected to commit to. (The money that pays for this is frequently deducted from a developer's royalty basis.) One of the most effective of these in-store promotions is a rolling demo of the game or, better still, an interactive demo that hooks customers on the spot.

Chain retailers also generally have an annual meeting where they gather all their managers for training and inspirational sessions. A portion of this meeting is usually reserved for a miniature tradeshow where publishers push their latest products from 8-by-10-foot booths. Less fancy than the big shows (such as E3, Comdex, and CES), these events generally feature lightning-fast game demos, and T-shirt, tchotchke, and product giveaways. The goal of the sales group is to let store managers know about upcoming products so that they will recommend the game to their clerks and customers.

Distributors

Not every retailer buys directly from the publisher. Many go through national or regional wholesalers. This enables "mom and pop" stores to carry less inventory but still access a wide range of products. This arrangement benefits the publisher as well because it has fewer relationships to manage.

The sales group's interaction with the distributor usually centers on encouraging the wholesaler's representatives to be aware of and promote the game. A distributor usually has a telephone boiler-room filled with salespeople, each of whom has a list of small retail stores with whom he or she keeps in constant touch. It's not unusual for a publisher to organize a special day at the wholesaler's place of business to promote his products. He comes in with promotional material such as T-shirts and caps and spends the day with the group. He sets up a running video of his products that the reps can look at during their coffee breaks, and leaves behind short demo versions of the game that the reps can explore on their own time.

Original Equipment Manufacturers

An OEM is a company that makes anything from a computer, to a graphics or sound card, to a joystick or mouse. The reason companies want to bundle software with their hardware is to add extra value to their product and to distinguish themselves from their competition.

OEM deals are large-volume, low unit-price deals that get a game out to a general audience that might not otherwise see it. The profit per unit is usually quite small, but it is guaranteed and usually paid up front. An OEM deal can require the development group to prepare a special version of the game that is optimized to work with that particular piece of hardware and show off its features.

The Community Team

Massively multiplayer online games require more publishing support than stand-alone, single-player games.

The community team is typically responsible for all communications to the current players and for the virtual environments in which players can gather to communicate with each other. The team maintains a website for current players and moderates the player bulletin boards. As soon as a game is announced, it is wise to have a community presence so the developer can actively participate in the building of the community around the game. When the game is ready for launch, the entire community team should be in place and fully trained. For most MMOGs, this normally occurs in the "open beta" period preceding launch.

Another part of the community team's job is to present customer issues to the development team. They interpret the problems, determine their severity and impact, and impart this information to the development and service teams. This is a key function, because most team members cannot review the large volume of customer feedback received on the game bulletin boards.

The community team is also responsible for communicating to customers what the development team is doing with the game. This includes changes to the game that are made in updates, policy changes, and how conflicts have been resolved. However, the community representatives cannot simply mouth the company position. They must be honest with the players, because their role requires them to be advocates for both the players and the company. This means they must pass player feedback to the development team, and also be responsive to player concerns, instead of staying mindlessly "on message." Community representatives should be fluent players of the game, they should have uncommon empathy, and they need superb communication skills.

The Networking Team

All online games—and MMOGs in particular—require a dedicated staff of networking professionals who install and keep the servers up and running at various collocation facilities. This staff must be on-board and trained by the launch of the open beta, by which time they must also have installed all the equipment necessary to support the launch.

Part of this team will run the Network Operations Center, which monitors the status of all the servers to ensure the systems are running correctly. Most companies automate the vast majority of this monitoring. Nevertheless, a small staff of network or operating system engineers will always be on hand to serve as first responders to emergencies.

Another part of the networking team are the engineers who set up and program the routers, switches, and computer systems. They also select collocation facilities (remote locations where the servers are housed to ensure best connectivity to the Internet), and they work with the personnel at the collocation facilities to maintain the equipment.

Whether the servers run on Linux, Microsoft, or some other operating system, the network group also includes specialists who create customized software, tune the operating system for the specific game applications, and diagnose and correct emerging problems.

The Customer Service Team

While all game publishers have customer service staffs, support for stand-alone games is typically limited to weekdays and regular business hours. MMOGs, on the other hand, usually provide customer service seven days a week and 16–24 hours a day. A key customer benefit of the massively multiplayer medium is that the game is available 24 hours a day, and customers expect service to be available whenever the game is. Unhappy MMOG customers tend to quit the game, which means less money for the publisher, so MMOG companies pay much more attention to customer support than stand-alone game publishers do.

Customer service is typically one of the largest ongoing expenses for a massively multiplayer game, running 5–15% of the monthly subscription fee. Customer service groups must be fully staffed and trained when the game launches, and this usually happens during the open beta period immediately before the commercial launch.

The Customer service groups include:

- Billing and Account Administration. Massively multiplayer games that have charges beyond the initial purchase need personnel who handle billing and account questions. This group works with customers who lose access to their accounts for whatever reason, although the most common problems revolve around credit card charges. Much of this work happens via phone calls and emails, due to the confidential nature of billing and other personal information. Very few companies outsource this area of customer service due to cultural issues and the sensitivity of the information being handled.
- Technical Support. All games require technical support. This group is tasked with ensuring that customers can install and play the software, and it is also responsible for resolving compatibility issues that may arise with the customers' hardware and software setup. The only difference between stand-alone game technical support and MMOG support is that the latter tends to be available 24 hours day. Most technical support can be done via email and live Internet chat, although phone support is also available from most publishers. The Internet chat and email support is often outsourced to reduce costs.
- Game Support. While all publishers offer some form of game support (answering questions about gameplay), stand-alone games limit this, both because it is expensive and because "discovery" is often part of the gaming experience. By contrast, MMOGs generally offer support within the game environment itself. The need to

help players while they are in the game has been driven by a combination of game complexity, bugs, and the necessity to moderate the social play environment.

- Game complexity can frustrate customers and impede their gameplay. While most publishers provide knowledge bases and websites with significant information about the game, players typically want the answer to their question right away without researching it themselves.
- Bugs can cause players to get stuck in game or to lose things of significant perceptual value. Players expect customer support to help them right away to get them unstuck or to compensate them for losses due to game errors.
- MMOGs are filled with people interacting together, and their social conduct within the game environment sometimes needs moderating. Players may abuse each other with text or with game mechanics, and the game publisher is expected to adjudicate these disputes and to punish customers that violate the rules of the game. All of these are significant and time-consuming activities that require a sizeable staff to handle.

Promotional Tools

The PR, marketing, and sales groups need information and materials about the game so that they can promote it to the outside world. To do their job, they need screenshots, concept art, videos, sell sheets, interviews with the development team, demos, and so on. All of these have an impact on the development team's task list and schedule.

Demos

Why is creating a demo such a big deal? Isn't the game up and running in some form every day? Why can't the development team just dump everything onto a CD and send it out?

First of all, the everyday version of a game in development is nothing like a finished product. It is loaded up with debug code—perhaps as much as 50% of the code in the build at any given time is mere "scaffolding" to help the developers build the game itself. This code typically makes the game run slowly and interferes with gameplay. Often, to show the product properly, the team would have to strip out the debug code, which can involve a lot of work. Imagine if every time the Pope wanted to show off the progress on the Sistine Chapel, Michelangelo had to remove all the scaffolding from the room and then put it back again afterwards.

In addition, game demos come in several flavors, depending on where in development the game is, the purpose of the demo, and its intended audience. As was mentioned earlier, a demo can last anywhere from just a few seconds (for example, when it is part of a video) to several hours (when journalists come on-site for an in-depth look at a game while it is in development). The length of a demo depends on the use to which it will be put.

Five seconds or less. Clips or screenshots for a video.

- **Thirty seconds to two minutes.** A highly distilled presentation of the high concept, usually something a buyer can "get" in less than two minutes.
- **Five to 10 minutes.** A narrated, looping video that shows screenshots and gameplay, useful for trade shows, PR kits, website downloads, and in-store kiosks.
- **Ten minutes and longer.** An interactive demo used on magazine demo disks, tradeshow stations, website downloads, and in-store kiosks.
- **Hours.** An in-depth presentation of many game facets during the course of a day.

Typical audiences for demos include:

- **Journalists.** Gaming fans who are always looking for something new. They're likely to be impressed by technological advances.
- **Buyers.** Harried people with not much time. They're looking for a quick presentation of the high points that will make the game a hit.
- **Tradeshow audiences.** Industry insiders looking for the flashy stuff that signals "the next big thing."
- **The general public.** Gamers who want to play a demo and figure out whether the game is fun.
- **Internal company groups.** Coworkers who need to learn what the game is about and what its features are so that they can promote it to the outside world.

Demos aren't easy to make. They are not "free" spin-offs of the regular development process. They have a different purpose than the overall game, a different rhythm, design, and execution. They should be designed with input from PR, marketing, and sales to maximize effectiveness with their intended audience. They have to be programmed and tested. They are mini-projects unto themselves.

Also, it is wise to assume that anything left in the hands of someone outside the company will eventually find its way into the hands of the public. Each demo should represent the game at its very best. No one should ever ask the development team to "throw together a quick demo so that I can drop it off with a buyer next week." The company might be embarrassed by what shows up on the Web the following week.

The wise producer will sit down at the beginning of a project with the tech lead and people from PR, marketing, and sales to discuss at what points in the development cycle it will be feasible to create various demos. The tech lead will be happy to do this because it gives him specific targets to shoot for, and he wants everyone to have the best promotional tools possible.

Trial Versions

Massively multiplayer games often go beyond demo versions, offering full trial versions of the game. A trial version is typically a complete copy of the game that the player can only use for a limited number of days. Free trials are a common way to gain subscribers, especially non-traditional customers who may not typically buy games at retail. They are also useful if the game loses access to retail distribution. Trial versions can be used in conjunction with OEM partners to introduce large numbers of people to the game.

Interviews

Developers tend to think that the benefits of their projects are self-evident, so they are often not the best ambassadors of their own games. Nevertheless, journalists like to have access to the people who create the games. While many journalists focus on the "name" behind the game (usually the designer), others dig deeper to try and cover more members of the team, especially in this era of collaborative development where the tech lead, AI programmer, and art director often have as much or more influence on the final product than the designer.

Screenshots

In the PR wars, the game with the best screenshots wins.

Magazines or websites should never be allowed to take screenshots of a game. They can use the wrong settings or pick a boring location. The screenshots should be carefully chosen by someone on the team to show off the game to its maximum advan-

tage. A picture is worth a thousand words only if the person creating it knows what he wants to say.

The demand for screenshots in advance of a game's release can be insatiable. Web sites are always clamoring for new and exclusive shots. The PR group will be pressured for literally hundreds of unique shots before the game is released.

The problems with this are, first, that it takes time and expertise to create a great screenshot, and second, if the team is not careful, they will overexpose the game, and people will feel that they've seen it all by the time it's released. Some producers solve the first problem by assigning one individual (usually an AP) to create all the screenshots. They solve the second by sequestering certain portions of the game and never releasing screen grabs from those areas before the game comes out.

Sell Sheets

Sell sheets are usually one-page flyers that the sales force gives out to the retail trade. They contain an unusual mix of information. One part resembles the back of a box, with a feature summary and attractive graphics or screenshots. Another section contains information of interest to the retail buyer, such as the size and nature of the marketing campaign (TV, radio, magazine), the anticipated release date, how many boxes are in a case, and so on.

Consumers never see these sheets. They are printed only to use within the trade. However, they can be graphically linked to the advertisements or other marketing materials.

The Contract

The main deal points of a contract are usually worked out between a publisher and developer without the help of a lawyer. Once these points are agreed upon, the lawyers are brought in to capture in unambiguous language what's been agreed on, to ensure that the contract has the right form and language to commit each side to its obligations, and to make provisions for the failure of either side to deliver on those commitments.

Each side should make all its intentions explicit in the contract. Unspoken agreements and unwritten understandings all go up in smoke when the individual who made them leaves the company and all that's left behind is the written word.

What follows is a discussion of the deal points that must be negotiated in most development contracts. Lesser issues, such as boilerplate language, are not addressed.

Advances

An advance is a royalty that is paid before it is earned. Generally, for every unit a publisher sells, he pays the developer a percentage of the money received (see "Royalties" next). An advance is money paid by the publisher to the developer *before* the product has sold those units.

Advances are generally not recoupable. That is to say, after a publisher makes a payment to the developer, he usually can't get it back.

When a game has sold enough units to cover these advances, it is said to have "earned out." Money paid to the developer after this point is commonly called "the back end."

Publishers want the developer on the lowest advances possible, not just for their own cash flow but also to give incentive to the developer to make a great game so that he'll get to the back end. If a developer has huge advances, he knows that the odds of earning out are slim, and there is correspondingly less incentive to extend himself for the game. Ideally, what the publishers want is a partnership, in which the developer is participating in the risk. The more the developer shoulders the up-front costs, the higher his royalty will be.

The developer, on the other hand, wants the advances to be high enough to cover the costs of running his business while the game is in development. It is in neither party's best interests for the developer to go belly-up halfway through the project. On top of that, the developer wants to build in a little profit up front because the publisher always retains the option to kill the game at any time. This can leave a developer, who has been counting on reaching the back end, high and dry. Sometimes this particular contingency is addressed by establishing a kill fee.

The negotiation over advances usually centers on each side finding the other's "point of pain" and determining whether an accommodation can be reached whereby both sides are only a little uncomfortable.

Royalties

Royalties vary greatly, and the percentage changes with how much risk the developer is willing to take on. Development houses that fund all their own development and don't take advances from the publisher are entitled to a higher royalty because they are taking on the risk of development.

The basis of the royalty is also important. Usually, it is the wholesale (not the retail) price paid for the game, less the cost of goods (COGs), marketing, and shipping. Other items that the publisher will request and the developer will resist are MDF, license fees, and the publisher's distributed overhead costs.

The royalty percentage is likely to decrease as the game is managed through its lifecycle. When the wholesale price drops below a certain percentage of its original price, the royalty can disappear altogether. Then a publisher can get rid of old product without being hampered by royalty accounting.

If a publisher and developer are working on more than one game at a time, the publisher can seek to cross-collateralize the games. This links the finances of the two. The practical result is that the publisher can withhold royalty payments on one product if the other product has not earned out its advances. This is favorable to the publisher and will be resisted by the developer.

If a publisher and developer are working on more than one game at a time, the publisher can seek to link the finances of the two through a procedure called cross-collateralizing. The practical result is that the publisher can withhold royalty payments on one product if the other product has not earned out its advances. This is also favorable to the publisher and will be resisted by the developer.

Reserve Against Returns

Because the retail stores can return 100% of their stock to the publisher at any point, a game isn't truly sold until it has sold through. (Even then, things can get dodgy because some stores allow limited return rights to their customers. Even a unit that has sold through can sometimes find its way back to the publisher.)

Because publishers hate to go back to a developer and try to extract money they've already paid, they build in a reserve against returns. This means that whenever a royalty is due, the publisher holds back a certain percentage of it, just in case the game eventu-

ally comes back from the retailer. This reserve ranges from 15–30%. (The higher number favors the publisher, and the lower number favors the developer.) The reserve is generally liquidated over a 12–18-month period. (Here again, the higher number favors the publisher; the lower number favors the developer.)

Milestones and Deliverables

Advances are generally hooked to milestones, which are significant points in development marked by completing a certain amount of work (a deliverable). An initial advance is usually paid upon signing the contract, but the rest of the payments hinge on completing the deliverables. The milestones should be broken out in an appendix to the contract, and their deliverables should be precisely defined. This protects both sides. The developer who has done the work wants to be paid. The publisher who hasn't received the work, doesn't want to pay. The best way to keep this hassle-free is with concrete deliverables. The best deliverables are binary—they're either complete or they're not, with no room for argument in between.

It is entirely reasonable for the milestone dates and deliverables to change in the course of development. This happens all the time, but each side must make sure that the new deliverables are as specific as the original ones, and that the contract is appropriately amended. (That's why it's easier to put milestones in an appendix in the first place rather than sprinkle references to them throughout the document.)

The mechanism for accepting milestones should be defined in the contract. Generally, the publisher wants a certain amount of time to review the work and declare whether it satisfies the milestone. It is to the developer's advantage that this period be as short as possible.

Intellectual Property Rights

Generally, each side of the table wants to retain as many rights as possible. Developers with an original intellectual property (IP) think long and hard before assigning it to a publisher. Publishers, on the other hand, think equally long and hard about plowing millions into developing and promoting an IP that the developer can walk away with after two years. (For a more complete discussion of IP rights, see the previous section on Outside the Game Studio).

Proprietary Technology

The publisher needs access to all the code necessary to publish and maintain the game. The developer should have the right to hold on to his own engine and tools. Some negotiation must take place here, but the issue should be explicitly addressed within the contract so that no disputes arise over who owns what after the companies go their separate ways.

Term

Most agreements have a fixed length of time or term. Neither side wants to create obligations that go on forever.

Termination

The effects of termination of the deal are usually spelled out in the deal. If either side needs to terminate for breach, the method is spelled out, as well as the consequences.

Confidentiality

This is usually not a sticking point in contract negotiations, and it is often covered in a separate non-disclosure agreement (NDA). At issue is what is considered a trade secret or confidential information, the methods used to safeguard the information, and the conditions under which the information can be revealed. Typically, each side agrees to guard the other party's confidential information in the same way it guards its own. If the information is published or otherwise becomes available through a third party, they agree that the restrictions against discussing it are removed.

Ancillary Revenues

Usually, there is a provision in the agreement for royalty revenues from clue books, action figures, and so on. Developers usually won't get rich from these merchandising deals, but they do help elevate the game's image (and therefore sales) to core customers. Massively multiplayer games have several opportunities to create ancillary revenues from specialized services such as transferring characters between servers, character name changes, and charges for special items or character enhancements.

"Indie" Development

Not all developers are concerned with creating games that follow the mainstream model. There is a flourishing community of independent game makers who develop games "on their own," without the benefit of publisher advances (or the hindrance of publisher input!).

Indie developers are typically one-or-two-man shops who self-fund the development of relatively small games that they market through non-traditional means, usually a shareware "try-before-you-buy" model. The chief attractions to the developer are that he has complete control over the development process, the creative content of his game, and the ways in which it will be promoted and sold.

Indie games are generally sold online, either directly to customers through the developer's website, or through a license arrangement with an Internet portal site. Almost all allow the customer to try out the game before making a purchase decision, either by granting access to the full game for a short period of time, or by letting him play a subset of the larger game that will whet his appetite for the full set of levels or features.

Whether working with an independent developer or a larger development studio, one of the most important issues associated with building a game is selecting the game engine. The two following sections discuss single-player and MMOG engines, and will be useful to anyone involved with game development projects.

VIII. Single-Player Game Engines

A game engine is the layer of code that interacts directly with the hardware a game runs on. It is usually a collection of modules, each of which performs a specific function, that together comprise the platform upon which developers build an individual game.

A typical game engine may include modules to handle one or more of the following functions:

- Graphics: rendering to the screen
- Audio: generating sound
- AI: Artificial Intelligence that guides the actions of non-player characters
- Collision detection: determining when objects in the game world come in contact with each other
- Basic Physics: controlling how objects interact when they do collide
- Graphic User Interface (GUI): the player's window into the game world
- Scripting and Editing tools: allowing game designers to control sequences of events
- Networking: connecting computers to each other
- Various Graphical Features

Lighting	Shadows	Texturing
Shaders	Rendering	Scene Management
Animation	Meshes	Surfaces and Curves
Special Effects	Terrain	

Using a game engine (or "middleware") means a game developer doesn't have to write every line of code from scratch. The engine takes care of many of the "housekeeping" tasks common to all games, freeing the developer to concentrate on writing new code that is specific to the game.

Game engines became popular in the mid 1990s, when some companies began to separate their core software from their game-specific content, and made that core software available for licensing by other companies. These other companies developed content (or "game assets") that differentiated their products in the marketplace, creating new characters, weapons, and game environments.

Licensing engines has proved to be quite lucrative, with middleware companies charging anywhere from $10,000 to $750,000, depending on how many modules the engine includes, how advanced it is, and how much support is offered.

It is interesting to note that the basis for a developer's decision whether or not to license an engine has changed over time. Initially, it was a purely technological decision: will using this engine help us get to market faster with a better game? Increasingly, however, it is becoming a marketing question: will using this engine create the public perception of increased value for our game? This shift is generally not welcomed by technical teams, who often find themselves ordered to use a well-known and expensive engine that doesn't fit their technological needs.

An alternative to commercially available engines are Open Source engines. These are software modules that are collaboratively developed and are typically available at no cost (or relatively low cost) to the user. (See the box below for a strict and thorough definition of open source.)

Open source engines vary widely in their feature-sets, reliability, and level of support. Some efforts are well-coordinated and have hundreds or thousands of contributors, others are fly-by-night projects that are best avoided.

Keeping track of the available engines and libraries is difficult. Two websites that list hundreds of the available choices are www.devmaster.net/engines/ and abattoir.wolfpaw.net/personal/gamelibs.php

Among the most commonly used commercial engines are:

- Unreal
- Quake
- Source (*Half Life*)
- Torque
- Lithtek
- Renderware
- Gamebryo
- Virtools
- 3DGameStudio

Among the most commonly used open source engines are:

- OGRE
- Crystal Space
- Irrlicht
- Panda3D
- jME
- Reality Factory
- RealmForge GDK
- OpenSceneGraph
- Axiom

For government purposes, the decision whether or not to use an engine, whether it is commercially available or open source, is an issue discussed under the Government Engines section of this report.

Open Source Initiative—www.opensource.org

The website states the Open Source philosophy as follows: “The basic idea behind open source is very simple: when programmers can read, redistribute, and modify the source code for a piece of software, the software evolves. People improve it, people adapt it, people fix bugs. And this can happen at a speed that, if one is used to the slow pace of conventional software development, seems astonishing.” Here are the requirements that must be met for software to be considered open source.

The Open Source Definition

Introduction

Open source doesn’t just mean access to the source code. The distribution terms of open source software must comply with the following criteria:

1. Free Redistribution

   The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.

2. Source Code

   The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed.

3. Derived Works

   The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

4. Integrity of The Author’s Source Code

   The license may restrict source code from being distributed in modified form only if the license allows the distribution of “patch files” with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.

5. No Discrimination Against Persons or Groups

   The license must not discriminate against any person or group of persons.

6. No Discrimination Against Fields of Endeavor

   The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.

7. Distribution of License

   The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

8. License Must Not Be Specific to a Product

   The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

9. License Must Not Restrict Other Software

   The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.

10. License Must Be Technology-Neutral

    No provision of the license may be predicated on any individual technology or style of interface.

Source: Open Source Initiative. "The Open Source Definition." Version 1.9, 2006.
www.opensource.org/docs/definition.php

IX. Massively Multiplayer Game Engines

What is an MMOG Engine?

Typically a game engine includes most, if not all, of the elements needed to make a particular type of game. In the case of an MMOG engine, this is not always the case given the number of elements required to make a game and the service within which it will run.

Some of the major game components are:

- All the elements of single player engines (e.g., graphics, audio, AI, collision detection, physics, GUI, scripting and editing, networking, various graphics features, and art tools)
- Client/server networking system
- Server process framework
- Persistence system
- Object/NPC management system
- Character creation/clothing layering system (sometimes)
- Some major service components include:
- Patching system
- Login/validation
- Registration
- Billing
- Customer service tools
- Software/data deployment tools
- Metrics gathering/presentation tools
- Software and hardware monitoring systems

Why acquire an engine to make an MMOG?

Saving time in development is the primary reason for purchasing an MMOG engine; the other option being building an engine from scratch. Most MMOG engines already have 10+ man-years of development invested in them. More importantly, a commercially available engine has been thoroughly tested. Code that has been used suc-

cessfully by multiple products will have significantly fewer errors than a new engine, making it a more solid base upon which to build a new game.

Open Source vs. Proprietary Engines

As of this writing, the only viable fully open source MMOG engine solution is Worldforge (www.worldforge.org). Several other efforts are in progress. One of them, NEL, is available under the GNU license, a modified open source agreement that permits commercial use if one pays a license fee. True open source engines cannot be used in commercial applications.

Commercially Available Engine Information

This list is not intended to be exhaustive but it does include most of the well-known engines at the time of this report.

Table 1. Well-Known Commercially Available Game Engines

Engine Name	Company Name	URL
Bigworld	Bigworld Pty Ltd	www.bigworldtech.com/
Emergent	Emergent Game Technology	www.emergentgametech.com/
Forterra	Forterra Systems, Inc.	www.forterrainc.com/
Horizons	Tulga Games Llc	www.tulgagames.com
Kaneva	Klaus Entertainment, Inc.	www.klausentertainment.com/
Mm-Suite	Community Engine, Inc.	www.ce-lab.net/en/products.html#middleware
Multiverse	The Multiverse Network, Inc.	www.multiverse.net
Nel	Nevrax	www.nevrax.org/tiki-index.php
Nice-Tech	Nice-Tech Ltd.	www.nicetech.co.uk
Sun Darkstar	Sun Microsystems, Inc.	www.sun.com/products-n-solutions/gametech

These and other engines offer different feature sets. It is highly advised that any group planning an MMOG define its game and service requirements before analyzing the various engines. (See Appendix E for a feature-by-feature comparison of most of the available MMOG engines.)

X. Standards and the Game Industry

The word "standards" as it applies to software development is used with two meanings. The first refers to widely-agreed-upon methods of building software; the second to interoperability—the power to import elements of one game into another.

The game industry has avoided both, and efforts to introduce either have generally encountered stiff resistance. The reasons are both historical and financial.

Background

When the electronic entertainment industry was young, most games were developed by small teams of fewer than 25 individuals. These teams were isolated groups, even within their own companies, and generally had a strong prejudice against NIH or "Not-Invented-Here" technology. Almost every game was new "from the ground up," which is to say that the team used very little code from previous games, and virtually no code from outside sources.

The only need for standards in that environment was at the operating level, where the team needed to know that the game would run properly on different computers and operating systems, that the graphics would display correctly on different monitors, that the sounds would be audible through different brands of speakers, and so forth.

So the standards that emerged from that period tended to be oriented towards allowing teams to write one set of code that would work on many machines, rather than having to write separate code for every different chipset and brand of computer that was on the market.

One such standard was the Windows API (Application Programming Interface), which is a collection of libraries and instructions that forms a layer between the game's code and the hardware. This permits a developer to write code that interacts with that layer, and the API takes care of translating the instructions into something that will work on any computer that runs Windows.

In the area of graphics, OpenGL and Direct3D are cross-language, cross-platform interfaces that consist of function calls that can be used to create complex three-dimensional scenes and objects from simple primitives. As an article from the *MSDN Magazine* states:

> A fundamental challenge of a hardware-accelerated graphics API is to enable application developers to take advantage of the rapid technology advances occurring in the 3D hardware space while allowing a certain amount of compatibility and uniformity across graphics hardware solutions. One way to do this is to define a standard by committee and then have each vendor support that standard. Graphics hardware vendors can innovate and create proprietary extensions through an agreed upon extension mechanism. Over time, the hardware vendor can lobby the standards body to accept their proprietary extension as part of the standard. OpenGL version 1.1 is an example of this approach to hard-ware interoperability. One limitation is that it can take a long time to get vendor-specific innovations incorporated into a multi-vendor standard, thereby taking the risk that the standard will become obsolete.
>
> In DirectX® 9.0, the features of DirectDraw® and Direct3D® are combined into a single API called DirectX Graphics.[7]

In the area of sound, MIDI and Redbook Audio also emerged as standards. MIDI (Musical Instrument Digital Interface) converts any note a musician plays on an instrument into a digital message that can be read and played back by any MIDI-compatible device, such as the sound cards in computers. Redbook Audio is the set of specifications underlying all CD-ROM audio discs. Not only does it specify the physical properties of the disc itself (for example, how thick the CD layer is), but it also specifies the digital encoding format (2-channel, 16 bit PCM, clocked at 44100 Hz).

These and other standards allowed game developers to be reasonably confident that their games could be played on a wide variety of home computers, although the open architecture of PC devices created an environment in which unforeseen hardware combinations were inevitable, and incompatibility bugs still occurred. (Just a few of the components that users can "mix and match" include the monitor, keyboard, mouse, CPU, hard drive, memory, sound card, speakers, graphics card, and operating system. Advance testing of all combinations of these and other components is impossible. By contrast, the "closed architecture" of console game systems such as Playstation and

---

[7] Mirza, Yahya, H. and Henry da Costa. "Introducing the New Managed Direct3D Graphics API in the .NET Framework." *MSDN Magazine*, vol. 18, no. 7, 2003.

GameCube, ensure a stable hardware platform against which a game can be rigorously tested, which is why console games rarely crash.)

In those early days of game development, interoperability—even if it had been desired—was pragmatically impossible. With each team "rolling their own" engines, architectures, and data structures, there was no chance that games would work together.

Even within a single company, the business pressures on a team to get a full-featured game to market in the fastest and most economical way possible ensured that project managers would not allocate any resources to non-essential elements, and interoperability did not appear on anyone's list of "must-haves." Thus, although a company may have published many games in the course of the year, none of them could be made to work with each other.

But the disincentive to make games interoperable between companies ran even deeper. If a third party created and sold add-on or sequel products to an original franchise owned by a given company, not only did the company not derive income from the IP they had created, but the value of the franchise itself might be damaged, especially if the add-on was inferior or did not work well with the original title.

Current State

Today, making games is big business. Teams have grown to the point that it is standard to have more than 100 people working on a AAA title. Intellectual Property is licensed from such diverse areas as books (*Harry Potter*, *Lord of the Rings*); movies (*The Matrix*, *GoldenEye*, *Star Wars*); sports (NFL, NBA, NHL, FIFA); and comics (*Spider-Man*, *Batman*, *The Hulk*), and licensors are more vigilant than ever in protecting the value of their franchises.

Games have become ubiquitous, appearing on desktop computers, consoles, handheld gaming devices, PDAs, mobile phones, airport kiosks, hotel room televisions, and even key chains (there is a keychain version of *Tetris*). Interconnectivity has become mandatory for many game types, and most games today have an Internet or other multiplayer component.

So revenues are up, competition is fierce, and company executives everywhere are being driven to find the most efficient methods to create games.

In this environment, the standards that are emerging continue to be those that reduce development time and effort and that help developers guarantee their games will operate in these diverse environments. These standards tend to be ad-hoc and consensus-driven, rather than specifications that have been imposed by a regulatory or formal standards-setting organization.

One area of de-facto standardization is in the tool-sets that developers use to create their games. For example, the industry has settled on Adobe Photoshop as the application of choice to create 2D art such as textures, and any artist hoping to work in the industry must demonstrate mastery of this program. Similarly, 3DS Max and Maya have become the industry standards for creating 3D art and animation. Programmers looking for jobs must be proficient in C/C++, or Java.

Another area where standards have emerged are in gameplay and interface mechanics. For example, in most PC-based first person shooters, the movement keys are WASD, the left mouse button fires the primary weapon, and the Escape key pauses the game and brings up the options menu. (It is also "standard" to allow players to re-map or reconfigure these keys.) On most console controllers, movement is controlled by a joystick or a D-Pad, and the "action" button is conveniently located under the right thumb. These standards allow people to pick up a new game and start playing right away.

The larger world in which games appear has driven developers to be aware of and adhere to more formal standardization efforts such as the ones below, but their motive is still primarily to ensure their games are playable on a broad spectrum of platforms.

The Open Mobile Alliance (OMA) (www.openmobilealliance.org) was created by nearly 200 companies to ensure interoperability between mobile devices.

> OMA was formed in June 2002 by nearly 200 companies including the world's leading mobile operators, device and network suppliers, information technology companies and content and service providers. The fact that the whole value chain is represented in OMA marks a change in the way specifications for mobile services are done. Rather than keeping the traditional approach of organizing activities around "technology silos", with different standards and specifications bodies representing different mobile technologies, working independently, OMA is aiming to consolidate into one organization all specification activities in the service enabler space.
>
> OMA is the focal point for the development of mobile service enabler specifications, which support the creation of interoperable end-to-end mobile services. OMA drives service enabler architectures and open enabler interfaces

> that are independent of the underlying wireless networks and platforms. OMA creates interoperable mobile data service enablers that work across devices, service providers, operators, networks, and geographies. Toward that end, OMA will develop test specifications, encourage third party tool development, and conduct test activities that allow vendors to test their implementations.[8]

Similarly, the World Wide Web Consortium (W3C) develops interoperable technologies (specifications, guidelines, software, and tools) for the Web. Although not focused on games, W3C (www.w3.org) is the formal body that creates the Web standards and guidelines that make Internet-based games possible. Since 1994, W3C has published more than 90 such standards. The organization's goal is to allow any hardware and software used to access the Web to work together, which they refer to as "Web interoperability."

Within the game world, other standardization efforts are beginning to emerge. Some, like the IGDA's Artificial Intelligence Interface Standards Committee (AIISC), are the result of individuals from different companies coming together to avoid the massive "re-invention of the wheel" that characterizes many game development projects. Others, such as the efforts listed below, are the result of a single company taking the lead in a certain category, and picking up other organizations along the way.

Collada is a digital asset exchange schema for interactive 3D products that began at Sony and is now being embraced by many graphics-oriented companies.

> COLLADA ("COLLAborative Design Activity") is an open standard for the interactive entertainment industry that defines an XML-based schema for 3D authoring applications to freely exchange digital assets without loss of information. This enables multiple software packages to be combined into extremely powerful tool chains. Collada support programmable shaders authored and packaged using OpenGL ES Shading Language so that leading 3D authoring tools can work effectively together to create OpenGL ES applications and assets.[9]

Microsoft's "XNA" effort is another effort at standardization.

> XNA is Microsoft's next-generation software development platform, focused on enabling developers to make better games faster.
>
> The X in XNA represents a cross-section of powerful software tools and technologies from Windows, from Xbox and from our partners. The N stands for "next-generation" and A is for "architecture."

---

[8] Open Mobile Alliance, About Open Mobile Alliance, 2006, *www.openmobilealliance.org*.

[9] COLLADA, About COLLADA, 2006, *collada.org/public_forum/welcome.php*.

> XNA Studio is the visual studio for game development. It is an integrated team-based development environment tailored for game production. Today's game teams are wrestling with the challenges of growing content requirements, larger and more specialized teams, increasingly complex workflow and increased outsourcing. XNA Studio will address these workflow challenges by delivering an advanced build framework driven by a unified file format. The build framework is partnered with an integrated tool suite to optimize the game production process for all team members.
>
> XNA will allow developers to easily reuse code and tools between PC and Xbox titles, speeding the creation of games for both systems. Many games such as sports titles and shoot 'em-up games are released for both consoles and PCs, but porting between the two is usually a time-consuming and costly process. Microsoft began simplifying the process with the Xbox, which uses common PC components and the DirectX graphics library developed for PC games, but XNA will accelerate the process.[10]

Similarly, Sun is working to create open-source APIs for Java.

> As a group of game-oriented technologists within Sun, we believe we can drive an industry movement around development standards," says Melissinos. "So we set out to define a clear, concise stack of APIs that a game developer will be able to leverage across many different devices. Start with a standard, then differentiate!
>
> In essence, these APIs give you an unlimited but standardized way to develop rich visuals, sounds, and controls (from steering wheels and game pads to tools and weapons).[11]

Despite these efforts, however, interoperability is still not on most companies' planning horizon. Even when companies try to standardize on one engine for their internal projects, their goals are twofold: to reduce "from the ground up" development time, and to give their employees a recognizable tool set with which to build their games so they can more easily move people from one project to another. Both of these goals are aimed at efficiency, not interoperability.

As a presentation from MAK technologies points out, "[The] Producer/Studio model makes each project a separate business, focusing on the success of one title. [There is] very weak high-level corporate optimization." [12]

---

[10] Microsoft, XNA FAQ, *www.microsoft.com/xna/faq.aspx*.

[11] Melissinos, Chris. DataCenter News, Article 12480, vol. 73, no. 3, Mar 15, 2000.

[12] Katz, Warren. "Industry Perspectives on M&S Reuse." Simulation Interoperability Standards Organization, MAK Technologies, Inc. Jan 26, 2000. *www.amso.army.mil/smart/conf/2000/day-1/katz.ppt*

Or, as Gordon Walton, one of the authors of this report, has written, "No good deed will go unpunished! Project managers are ultimately penalized for any effort whatsoever that is at all extraneous to maximum efficiency for their own particular game. They will eventually be punished for any time they spend on interoperability."

Middleware and Engines

The increased use of middleware and "off the shelf" game engines may look like a drive towards interoperability, but it is not. Games built using the same engine are not necessarily interoperable, and will not work together unless they are specifically designed to do so. Teams take the engine as a starting point and innovate dramatically, thinking only about what they can do to make their game better and differentiate themselves from the crowd. Companies still do not have a compelling financial reason to make their games interoperable, and until such a reason emerges, they simply won't expend resources in that direction. Increasingly, each company wants to "own" its customers, and creating opportunities for them to go and play in someone else's world is something the company wants desperately to avoid.

Likewise, the now-common practice of making engine source code available to gamers along with published products will still not lead either to standardization or interoperability. When companies make game code public, teams of amateur game makers often alter some superficial elements such as the graphics to make a new game—a modification, or "mod"—that appears different from the original. Publishers of these engines encourage this activity for two reasons. First, it creates additional content for a product at no cost to them (and with no obligation from them to support it); and second, everyone who builds or plays such a mod has to purchase the original game, so the paid user base is expanded. The more vibrant a game's mod community is, the more financially successful it becomes.

However, these publishers ensure that the modder's efforts do not compete with their own products in the commercial world. Most engine End User License Agreements (EULAs) specifically prohibit commercial use of an engine, and the very companies who are making the engines free to users are simultaneously charging up to hundreds of thousands of dollars for the use of those same engines in commercial products.

With the rise of "serious games" and the entry of the government into the marketplace, discussions about interoperability are becoming more common, which is discussed in the "When Worlds Collide" section of this report.

XI. The Future of Mainstream Games

It is notoriously difficult to peer into a crystal ball and predict what the future will bring, especially in the field of technology. Innovation piles on top of innovation, technologies converge, and suddenly something entirely new emerges.

It is probably fruitless to predict which new technologies will take hold in the next five years and which will die. It is much easier to anticipate changes resulting from recent innovations that are just now being delivered, but whose impacts are already beginning to be felt.

Important Trends

Worldwide Growth of the Internet

The big story of the past decade has been the wildfire spread of the Internet, an enabling technology that affects the lives of nearly everyone on the planet—even in the remotest regions of the earth.

But as can be seen from Table 2, what has been a largely US-led phenomenon is being overtaken by the rest of world. With nearly 70% of the US population already connected, versus only 10% of Asia and 36% of Europe, the rest of the world is set to see explosive growth.

Table 2. World Internet Usage and Population Statistics

World Regions	Total Pop. (2005 Est.)	Pop. % Of World	Internet Users	Pop. % Penetration	Usage % Of World	Usage % Growth 2000–05
Africa	915,210,928	14.1	23,649,000	2.6	2.2	403.7
Asia	3,667,774,066	56.4	380,400,713	10.4	35.7	218.7
Europe	807,289,020	12.4	294,101,844	36.4	28.5	176.1
Middle East	190,084,161	2.9	18,203,500	9.6	1.8	454.2
North America	331,473,276	5.1	227,470,713	68.6	22.2	108.9
Latin American/ Caribbean	553,908,632	8.5	79,962,809	14.7	7.8	337.4
Oceania/ Australia	33,956,977	0.5	17,872,707	52.6	1.8	132.2
Totals	6,499,697,060	100.0	1,043,104,886	16.0	100.0	189.0

Note: Dynamic table at www.internetworldstats.com/stats.htm, last accessed Aug 7, 2006.

Source: Miniwatts Marketing Group

One important result of this growth will be that games will have the potential to reach far more people than ever before, and most experts predict there will be a rapid increase in the percentage of gaming revenue that comes from outside North America. According to a 2005 report from Price-Waterhouse, the global videogame market will double in revenue over the next five years, growing from $25.4 billion in 2004, to $54.6 billion in 2009, a 16.5% compound annual growth rate, "driven largely by Asia/Pacific." [13]

The top game companies have already begun to adapt to and accelerate this coming change. In late 2005, Electronic Arts opened a new office in Singapore to localize its games into five Asian languages. Ubisoft, a French company which employs over 1,000 people in their Montreal office, already has 600 people working in their studio in Shanghai, and its goal is to increase to 1,000 by the end of 2006. Even small- to mid-size publishers now translate its products into many languages.

The genres of games most effected by the growth of the Internet are MMOGs and multiplayer first person shooters. Both take advantage of the increased connectivity to move gaming from a solitary activity to one that is enjoyed in groups.

[13] PricewaterhouseCoopers. "Global Entertainment and Media Outlook 2005-2009." New York, NY: Pricewaterhouse Coopers, 2005.

Increased Broadband Penetration and Voice Over Internet

Hand-in-hand with the growth of the Internet is the increased penetration of broadband into homes around the world. Games that were too slow to play over a dial-up connection, suddenly become lightning fast over broadband. The elimination of lag time has been crucial to the success of games such as *Unreal Tournament*, *Counter Strike*, and *America's Army*.

According to Computer Industry Almanac, "The worldwide number of Internet broadband subscribers will surpass 215M in 2005—up from less than 5M in 1999 and 67M in 2002. USA is the leader in broadband subscribers and will reach nearly 47M at year-end 2005. China is in second place and will challenge for the lead in a few years. Worldwide broadband subscribers are forecasted to top 500M by the end of 2010." [14]

Table 3. Top 15 Countries in Broadband Subscribers (Year-End 2005)[15]

	Broadband Subscribers (#M)	Share %
1 USA	46.9	21.6
2 China	35.9	16.5
3 Japan	26.4	12.2
4 South Korea	13.1	6.04
5 France	9.6	4.42
6 Germany	9.5	4.4
7 UK	8.9	4.35
8 Canada	6.7	4.09
9 Italy	6.6	3.05
10 Spain	4.6	2.12
11 Netherlands	4.4	2.0
12 Taiwan	4.3	1.97
13 Brazil	3.0	1.39
14 Australia	2.6	1.21
15 Belgium	2.1	.97
Total for Top 15 Countries	185.2	85.25
Worldwide Total	212.2	100.0

[14] Computer Industry Almanac, press release, Nov 14, 2005. *www.c-i-a.com/pr1105.htm*

[15] South Korea is the leader in broadband subscribers per capita, followed by the Netherlands, Hong Kong, and the Scandinavian countries. The USA is ranked 15th in broadband subscribers per capita. Japan also has higher broadband penetration than the USA and is ranked 8th. All broadband technologies are included in the figures. Source: Computer Industry Almanac, Inc.

Until recently, it has only been PC games that have benefited from broadband Internet access. However, the next generation of consoles has connectivity built in, and a large part of publishers' strategies is to "own" their customers by continually feeding them new content down that pipeline, giving them enough new gameplay to keep them from turning to competitors' games.

The increasing availability and popularity of Voice Over Internet technologies (sometimes referred to as VOIP) is also worth noting as it intensifies group gaming experiences, motivates people to game together in "clans," and encourages friends to stay together as they migrate from one game to the next. The members of these groups may never meet in person, although it is common for clans in Asia to play in the same physical location.[16]

Digital Distribution

With the advent of broadband comes the possibility of digital distribution, doing away with boxed games altogether and delivering the software directly to gamers' desktops over their Internet connections. Services like Steam are already doing this with games like *Half-Life2* and *CounterStrike*.

However, the demise of bricks-and-mortar game stores is unlikely to occur in the next few years. Even with a high-speed connection, most of today's top games are so large that it would take over 24 hours to download them. The place where digital distribution will come into its own is in delivering updates, bug fixes, and additional content that complements the original games.

In addition, digital distribution will encourage innovation in PC game design because of the Long Tail phenomenon. In the current marketplace, there is a bottleneck at retail distribution, with most retailers only wanting to stock hit games because the cost of keeping slow-moving games on store shelves is too high. With digital distribution, however, there is no cost of inventory and suddenly the curve that shows game sales

---

[16] Schiesel, Seth. "An Online Game, Made in America, Seizes the Globe." *NY Times,* Sept 5, 2006. "In Asia…online players…often want to meet in the flesh to put a real face on the digital characters they have been having fun with. Even in the United States, more and more players are coming to see online games as a way to preserve and build human connections, even if it is mostly through a keyboard or microphone."

picks up a long tail as more games become economically feasible to create and sell to a public who can look beyond store shelves to a larger "virtual" inventory.

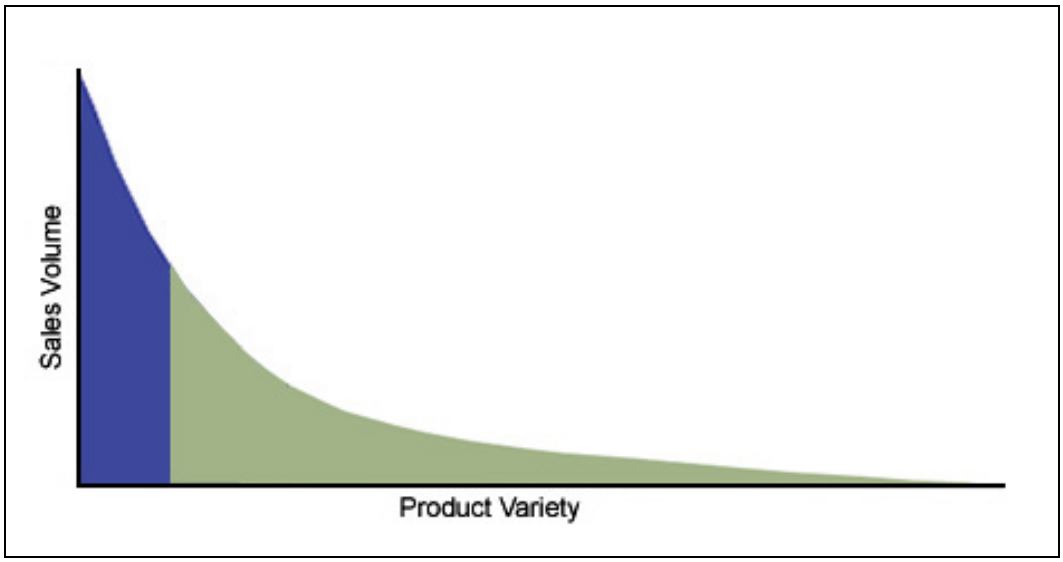


Figure 1. The Long Tail

This long tail will encourage independent developers to take a chance on smaller, less expensive, more innovative games that would never survive the approvals process at a major publisher. (See "How Commercial Games Are Marketed, Distributed and Sold" previously in this report for a more complete discussion of the retail channel).

Next Generation Console Development

The reason this experimentation and innovation will be confined to PC games is that the cost of developing console games will increase dramatically with the introduction of the next generation machines.

The year 2006 will see the debut of Sony's PS3 and Nintendo's Wii, joining the already-arrived Microsoft Xbox 360. The industry will settle into another five-to-seven-year cycle of hardware stability that will reverse this year's lagging sales and resume the years of growth, with console revenues predicted to rise to $21.9 billion by 2008.

The new machines have more power and are more complicated than ever. The PS3 has eight onboard processors, each with its own functions and memory. The programming teams required to take advantage of this power are correspondingly larger.

The increased power also means the machines can deliver better graphics, but that requires larger teams of artists—modelers, riggers, animators, texture creators—to create the stunningly beautiful games that audiences have come to expect.

With the size of the teams skyrocketing—most AAA games now have over 100 developers working on them—the size of the "bet" that a publisher has to make is larger than ever. The average development cost of a next-gen game is predicted to be over $20 million. This raises a game's break-even point to over one million units sold, and many current publishers cannot afford such large risks.

The result will be consolidation and a shake-out of console publishers with only the larger publishers surviving. Fewer games will reach the market, and those that do will be more strongly linked to existing franchises, such as movies or already-successful game series. The ranks of console developers will likewise be thinner.

Mobile connectivity

Many of these developers will find new homes in mobile gaming.

The convergence of PDAs and cell phones is creating an important new market for games. The new devices have increased functionality, more memory, QWERTY keyboards, and connectivity to other such devices and to the Web.

A *Business Wire* article states, "the mobile games market in 2004 was just above $3 billion, with a sharp rise forecast for the next five years. We believe the market will approach $18.5 billion by 2009." [17]

Part of the predicted increase comes from the concept of "gaming everywhere." Currently, when gamers turn off their consoles or PCs, they leave the game world behind. When they can reconnect to games over their cell phones, suddenly the games become available throughout the day, and otherwise "wasted" chunks of time can be filled by gaming.

Much of the innovation and experimentation that is predicted for the PC market is already occurring in the mobile space, where games are typically small, inexpensive to

---

[17] Juniper Research. "Research and Markets: Research Has Lead To Estimates That The Mobile Games Market In 2004 Was Just Above $3bn, With A Sharp Rise Forecast For The Next Five Years." As reported in *Business Wire*, Research and Markets, Feb 15, 2005.

produce, and quick to bring to market. In that environment, it is much easier to take a chance on a new idea and to discard it if it doesn't work.

Another driving force behind the growth of mobile games is the expansion of the market to include women.

Women have long been known to make up more than 50% of the "casual" PC gaming market, but only now are mobile games starting to be targeted at women. One reason for the delay is that women historically have not bought gaming "gadgets" (at least not for themselves), but now the games are coming to the devices they do own—mobile phones.

By the end of 2005, about 70.3% of Americans owned cell phones.[18] Of those 209 million mobile phone users, most studies suggest that slightly more than half are women. Furthermore, women now represent 59% of all consumers who play games on mobile phones, according to a study from Parks Associates.[19] The research firm Telephia reports that women also account for almost two-thirds of the total mobile gaming revenue in the United States (see Table 4).

Table 4. Category Share of Revenue and Gender Revenue Share Breakdown (US)

		Share of Revenue (%)	
Category	Share of Revenue (%)	Male Share	Female Share
Puzzle/Strategy	33.8	28	72
Card/Casino	18.3	34	66
Sports/Racing	12.9	39	61
Action/Adventure	12.8	60	40
Trivia/Word	11.4	26	74
Classic/Arcade	10.8	38	62

Source: Telephia[20]

The result of these shifts in demographics and market penetration is an ongoing explosion in casual game development, with more and more developers focusing on the mobile gaming market.

---

[18] Bridge Ratings LLC, press release, Feb 20, 2006, *www.bridgeratings.com/*.

[19] Parks Associates, press release, Jun 29, 2006, *www.parksassociates.com/press/press_releases/2006*.

[20] Telephia, "Mobile Game Report Q1 2006," *www.telephia.com*.

Globalization

Many of the above factors are driving the move to the globalization of the games industry. The increased cost of development in North America (along with the drop-off in computer science graduates who remain in the US)[21] causes publishers to look to overseas teams for less-expensive programmers, artists, and quality assurance personnel. The Internet makes communicating with these teams easier. Broadband connections make the transfer of huge amounts of data possible. The growth of foreign markets invites speculation that games designed by developers native to those countries may fare better than games designed by North Americans.

As mentioned above, many major game publishers now have overseas offices and have made other commitments to globalization. For example, Electronic Arts' most recent addition to its Board of Directors is Mr. Vivek Paul, vice chairman of Indian tech services company Wipro. EA announced the appointment with the statement, "The addition of Vivek Paul to EA's Board provides us with unique insight and experience related to several of our strategic priorities. His deep understanding of technology, management and global markets will be extraordinarily valuable in helping us set and define objectives for EA's future." [22]

Helping the trend towards globalization are the investments that foreign governments are making in support of their domestic gaming industry. China has announced it will invest $1.8 billion over the next five years to develop "100 kinds of online games with independent property rights,"[23] and Singapore's Economic Development Board will spend $1 billion Singapore (~US$614 million) over the next 10 years to encourage developers and publishers to locate offices there.[24] (By contrast, the United States has no Federal funding earmarked to support our domestic gaming industry).

Geographically-distributed teams are also now commonplace, and they, too, contribute to the globalization of the industry. One of the authors of this report recently designed a game while working in Virginia, in cooperation with a programming team in

---

[21] Zweben, Stuart. "2004-2005 Taulbee Survey: Ph.D. Production at an All-Time High with More New Graduates Going Abroad..." *Computing Research News*, vol. 18, no. 3, 2006.

[22] Electronic Arts, press release, Jun 17, 2005, *www.info.ea.com/news/pr/pr645.pdf.*

[23] People's Daily Online. "China to invest US$1.8 billion to develop online games." Jul 31, 2005. *http://english.peopledaily.com.cn/200507/31/eng20050731_199405.html*

[24] Singapore Economic Development Board, press release, Dec 5, 2005, *www.sedb.com/edb/sg/en_uk/ index/news_room/news/2005/edb_will_earmark_s.html.*

Slovenia, a graphics team in Florida, another graphics group in Hungary, voice recording artists in Los Angeles and Germany, and a publisher who was financing it from Austria.

Micro-Transactions

The ability to easily pay small amounts of money over the Internet has already had a large impact on gaming, and the effect of these micro-transactions will be broadly felt by game publishers and game players alike.

In Asia, micro-transactions have completely changed the business model of the MMO world. When a company publishes a new online game, it is now economic suicide to actually charge money for the game. People won't play it unless it is free. The main way the companies make their money are from selling virtual items within the game to the usually small—but avid—group of players who want to equip their characters with special items. Another method of recouping costs is to charge very minimal amounts for server time.

According to an article in *Fortune*, in China, a company named Shanda—

> ...gives the software away free and get players to buy time on the company's servers. For as little as 3 cents an hour, they could interact and compete. "They cracked the piracy problem," says Duncan Clark, chairman and co-managing director of BDA China, a Beijing technology consulting firm. "In China shrink-wrapped products don't sell."
>
> [Shanda] now has one of the largest market capitalizations ($1.8 billion) of any Internet company in China…
>
> Teenage boys and young men streamed into Internet cafes to log on to Shanda's games and assume the identities of warriors, monks, and magicians in order to kill monsters and one another. Online gaming became a national obsession, with as many as 2.5 million players logging on to Shanda's games at once. Revenues doubled every year, on average, reaching $61.7 million in the third quarter of 2005, up 41% from the previous year. Net income grew too, jumping 58%, to $32.3 million, in the same period. And Shanda's stock—listed on Nasdaq—nearly tripled after the IPO before cooling off in recent months....
>
> Industry analysts expect China's online gaming industry will continue to expand by 35% a year for the next five years.[25]

---

[25] Faris, Stephan. "Meet the Next Disney." *Fortune Magazine*, Nov 28, 2005. *http://money.cnn.com/magazines/fortune/fortune_archive/2005/11/28/8361976/index.htm*

The Next Big Thing: Emotion

Many of the technological improvements of the past few years—and many of those that lie just around the corner—are being welcomed by game developers as tools that may help gaming establish itself as a legitimate art form alongside the traditional media of books, plays, music, movies, etc. Vastly improved graphics, more realistic animation, lip-synching, advanced physics engines—all these have enabled game makers to develop more accurate simulations of the real world, and they hope these increasingly immersive environments will one day enable them to create audience emotions similar to those generated by other media.

It is already widely accepted that games are capable of eliciting certain emotions: the tension of battling an enemy, the frustration of repeated failure, the triumph at solving a puzzle, and so forth. But the so-called higher emotions—such as love, compassion, and generosity—have been more elusive. Some critics, including Roger Ebert of the *Chicago Sun Times*, have claimed games will never attain the status of art: "I… consider video games inherently inferior to film and literature…. I believe the nature of the medium prevents it from moving beyond craftsmanship to the stature of art. To my knowledge, no one in or out of the field has ever been able to cite a game worthy of comparison with the great dramatists, poets, filmmakers, novelists and composers." [26]

Many game makers take issue with this belief, pointing out that video games, barely 30 years old, are still in their infancy, and that the industry's stage of development is roughly comparable to the movie industry in days after the invention of the celluloid strip (1891) and Edison's Kinetograph (1894), but before DW Griffith's revolutionary *Birth of a Nation* (1915), considered by some to be the most important film in the development of cinema as art. Defenders of games' artistic potential believe that developers are still awaiting the invention of basic tools of the trade, comparable to Griffith's innovations of the close-up and cross-cutting, that will allow them to elevate games to an art form.

Additional technological advances that developers still await are flawless speech recognition and generation; dynamically generated sound; AI chips (to accompany the existing video, sound, graphics, and physics chips); and improved animation systems

---

[26] Ebert, Roger. "Movie Answer Man." *Chicago Sun Times*, Nov 27, 2005. *http://rogerebert.suntimes.com/apps/pbcs.dll/article?AID=/20051127/answerman/511270304/1023*

that reflect the internal feelings of a character. All of these technologies are being worked on, but none have been perfected.

When these tools become available, it will still be left to game designers to solve the basic conflict between the interactivity that is at the heart of a game, and the authorial control that is at the heart of storytelling. If they are successful, we will begin to see games that generate the full range of human emotion, games that illuminate as well as entertain, games that are art.

2
THE GOVERNMENT WORLD

I. Introduction

This section of the report serves as a sort of DoD 101 for game companies interested in working with the government. Because of its special position, the government has developed unique and highly-regulated contracting and acquisition procedures. This section provides an overview of those procedures, including the government's funding and procurement models, and it also guides the reader to important additional sources of information on these topics. The government's product development processes and the evolving role of standards are also explored.

II. A Short History of DoD Modeling & Simulation

For more than 2,000 years, militaries have been using war games in one form or another. The earliest games were strategy pieces designed to teach commanders how to think. Later developments allowed military planners to organize miniaturized forces on the battlefield and conduct mock engagements. With increasing sophistication, analysts and decision-makers were able to fix a course of action with the knowledge that different alternatives had been explored. War games have also allowed soldiers on the battlefield to think about missions before they occur and have helped train them to accomplish specific tasks. In recent decades, war-gaming has benefited from the advances in computer technologies that allow for complicated models and simulations.

Early Modeling, Simulation and War Gaming

While the German and the British governments actively began war-gaming in the later part of the 19th century, the United States only conducted limited gaming and experimentation activities. Under the supervision of Major W. R. Livermore of the US Army Corps of Engineers, some experimentation with an American version of the German war game *Kriegspiel* occurred in the mid 1880s, but the game did not receive widespread attention.

Several decades later during the period immediately before World War II, parts of the US military started to take a more serious look at war games to augment current practices. The Naval War College conducted many of the approximately 136 strategic-level war games that took place during this time.[27] The remaining games were carried out by the Marines, primarily in the area of amphibious operations. Results from these activities directed the planning of Marine operations during the war.

---

[27] Caffery, Matthew. "Chapter 2: Wargaming and the World Wars." Unpublished work, matthew.caffrew@wpafb.af.mil.

Other types of modeling and simulations also were popular during World War II. For example, the Link Trainer, the first electronic aircraft simulator, was used effectively to train pilots on the instruments in their airplanes. Created by organ builder Edwin Link, the trainer taught flying rules to new pilots.

After the war, an emphasis on the peace that would be generated by nuclear weapons threw war-gaming into decline. In 1949, after languishing for several years under the supervision of the Army Advanced Studies Group (AASG), there was a renewed interest in war games and the AASG became the Joint Advanced Studies Group, which began conducting new experiments.

Even though DoD interest in war-gaming was minimal throughout this period, the RAND Corporation, a government think-tank, actively pursued games and game theory as a way to conduct research. Its games and simulations covered tactical and strategic military wargames, as well as a variety of political games (*Goldhammer* and *Backstop*) and games on public opinion. These games helped to provide the impetus for the modeling and simulation effort that would slowly develop during the next 50 years.

The Post-World War II World

Each branch of the military evolved different programs and simulations to meet its needs for training and planning. Several programs significantly affected the evolution of modeling and simulation. RAND built the Air Battle Model (ABM) in the mid-1950s and the Air Force used it to model global war. With 32 kinds of aircraft, 31 different types of bombs, and 3,500 targets, ABM typified some of the problems still facing the military today: how to best gather and verify the data, how to input the information into the model, and how to understand the output from the simulation.[28]

During the 1960s and 1970s, Research Analysis Corporation's CARMONETTE (Computerized Monte Carlo Simulation) was used to simulate small-unit ground combat. CARMONETTE I, from the early 1960s, had a fairly basic set-up and concentrated mostly on tank and anti-tank operations. By the mid 1970s, after several years of improvements, CARMONETTE VI modeled individual soldiers, armed helicopters, platoons and, battalions.[29]

---

[28] Brewer, Garry. *The War Game*. Cambridge, MA: Harvard University Press, 1979, pp. 130–131.

[29] Brewer, pp. 135–136.

Beginning in the late 1970s, advances in computer technology started to have a greater impact on DoD war games. Atari, under the direction of DoD, modified its arcade game *Battlezone*, to help train soldiers on the Bradley fighting vehicle. The Control Data Corporation, an early computer firm and government contractor, created another tank simulator, Panzer Plato, for the US Army Armor School at Fort Knox, Kentucky. Throughout this period of advancement by the Army, the Navy continued to excel in the area of war-gaming and simulations. In the mid-1980s, its Naval War Game System (NWGS) evolved into the Enhanced Naval War Game System (ENWGS). While used primarily as a tool for training rather than experimentation, ENWGS doubled the computing power available to users and demonstrated another important advance in the field of modeling and simulation.

Recent Trends in Modeling and Simulation

In the 1990s, DoD picked up two popular commercial games: *Doom* and *Microsoft's Flight Simulator 98*. *Doom*, a well-known first-person shooter, was adapted by the Marines in the mid-1990s as a tool for building teamwork skills and the ability to follow orders. It was the Navy that began to use *Flight Simulator* to train undergraduates. Realistic input controls and large high-resolution screens helped *Flight Simulator* gain acceptance from those involved (the students at the Naval Academy and the Administration). Even as some of these commercial games became popular, military models and simulations continued to play a huge role.

The 1992 *Catalog of Wargaming and Military Simulation Models* listed approximately 500 "simulations, war games, exercises and models in general use throughout the Department of Defense and in the defense establishments of Australia, Canada, England, and Germany." [30] While some sims clearly were more commonly used, all of them had to be supported and supervised. Since then the list of models and simulations has continued to grow.

By early 2006, the Defense Modeling and Simulation Office's (DMSO) online Modeling and Simulation Resource Repository (MSRR) contained several thousand models and sims from the Army, Navy, Air Force, Missile Defense Agency, DIA, and other groups. Even with all of these programs ostensibly arranged in a searchable

---

[30] Joint Chiefs of Staff. "Catalog of Wargaming and Military Simulation Models." 12th ed. Washington, DC: The Joint Staff, DoD, 1992, p. ii.

online database rather than in hundreds of pages of text, the sheer number of models and simulations available makes identifying the correct one difficult. Simulations are available from simple targeting programs, and medical evaluation sims, to complex federated systems that link and interact with many smaller programs.

Models and simulations have come a long way from the early Link Trainers and table-top war games of the 1930s, and even further from the sand boxes and maps of the 1880s. Now a tool for both DoD and even many commercial interests, M&S allows for testing, training, experimentation, and a number of other activities that would be far more costly, time-consuming, and perhaps dangerous otherwise.

III. DoD M&S Genres

Models and simulations used by the government do not readily divide into categories like those used in commercial games. Whereas, the integral "human in the loop" nature of commercial games has tended to define the genre by the *role* the human plays in the game (i.e., first person shooter, third person, role playing, etc.), DoD M&S has defined genres by the *purpose* of the model or sim (training, acquisition, experimentation, etc.). Although it may be argued that some of the genres in the commercial industry are themselves artificial or that a game may fall into multiple categories, genres in the M&S community tend to be even more nebulous. The most readily accepted categories include training, analysis, and acquisitions. Other frequently-used genres include planning, testing, research, and experimentation. Sometimes, testing is included with, and research is separated from, acquisition. The same is true for experimentation, which can be listed separately from analysis. In this section, training, analysis, acquisitions, and planning will be examined with a brief discussion of testing, research, and experimentation.

Training

Models and simulations may train on any number of levels from the tactical/individual to operational.[31] Few trainers focus on the strategic level, as that tends not to have wide applicability among the services.[32] Training simulations may include tanks, planes, and other critical equipment such as computer systems. They may also emphasize teaching cooperation among individuals in the same squad or teaching basic skills regarding the military or a particular mission.

---

[31] US Department of Defense. *DoD Dictionary of Military Terms*. Washington, DC: DoD Joint Doctrine Division, 2006. *www.dtic.mil/doctrine/jel/doddict/*. The DoD dictionary defines the tactical level as the level at which "battles and engagements are planned and executed to accomplish military objectives assigned to tactical units or task forces." The operational level is defined as "the level of war at which campaigns and major operations are planned, conducted, and sustained to accomplish strategic objectives within theaters or other operational areas."

[32] US Department of Defense, 2006. The strategic level is defined as "the level of war at which a nation… determines national…security objectives and guidance, and develops and uses national resources to accomplish these objectives."

Two well-known trainers are the *Close Combat Tactical Trainer* (CCTT) and *Distributed Mission Operations* (DMO). CCTT is used as a "virtual, distributed interactive simulation for collective training." [33] Primarily a tool of the US Army, CCTT allows platoons and squadrons to crew tanks, armored personnel carriers, and HMMWVs (high mobility multipurpose wheeled vehicles), as well as to perform infantry maneuvers to accomplish training tasks. The Air Force operates DMO, which is used for training *and* mission rehearsal. Similar to the CCTT, DMO simulates all of the necessary components of actual missions, including sensors (for information gathering) and clutter (to distract and confuse the user as often occurs in the real world). Training stations may be linked together locally in groups of four or across networks to simulate larger training exercises.

Besides these types of trainers, traditional simulators still play a role in teaching basic skills from flying an Apache helicopter to learning the operation of a naval ship. These types of systems have existed for many years and have even used commercial games (such as *Microsoft's Flight Simulator 98*).

Other types of trainers focus on skills not directly related to combat. For example, some simulations focus on skills needed to interact with a population or to conduct an investigation. The Defense Advanced Research Projects Agency's (DARPA) *Tactical Language Trainer* supports "the rapid development of mission-oriented communication skills." [34] The simulation focuses on teaching and reinforcing basic language and cultural skills that soldiers would need on a mission.

Training comes in many different forms and is critical to the success of every mission. From the more advanced trainers to the basic tank simulator, DoD requires models and simulations across the entire spectrum in order to meet the needs of the armed forces.

Analysis

Analytic tools, like training ones, also cover the tactical and operational levels. They may also be used on the strategic level for certain kinds of tasks, including intelligence work. Most of the focus, though, is on the operational and strategic levels. Analy-

---

[33] US Army. "Close Combat Tactical Trainer," *www.fas.org/man/dod-101/sys/land/wsh/70.pdf.*

[34] Center for Advanced Research in Technology for Education. "The DARPA Tactical Language Training Project." Defense Advanced Research Projects Agency, *www.isi.edu/isd/carte/proj_tactlang/description.html.*

sis M&S focuses on helping users understand the functionality of systems, the reasons for particular outcomes, and other data pertaining to relevant missions for DoD. Three well-known analytical tools include SLAMEM, EADSim, and FFPAS.

Toyon Research Corporation created the currently used version of SLAMEM (Simulation of the Locations and Attack of Mobile Enemy Missiles) in 1998, to fulfill needs by both the analytical and experimentation communities. As an analytical tool, SLAMEM works in a stand-alone constructive analysis mode. It may be used to examine increases in effectiveness by sensors given particular parameters or to determine what is required from a certain technology to meet a particular mission goal.[35]

EADSim or Extended Air Defense Simulation (first used in the late 1980s) is a simulation used to analyze air defense systems and examine the effectiveness of theater missile defense. The simulation can model surface-to-air engagements, air-to-air engagements, as well as communications and attack operations.

A third commonly used analysis tool is FFPAS. FFPAS (FireFinder Position Analysis System) analyzes locations for the positioning of FireFinder radars. Developed by Technology Service Corporation in the mid-1990s, FFPAS continues to be used by the Army to evaluate the success of the radar at detecting the launch of particular projectiles given a specific terrain.

Modeling and simulation for analytical purposes continue to play a critical role in DoD as new technologies are created to meet the continuing challenges to national security.

Acquisitions

Acquisitions are another important area where modeling and simulation are frequently used. Broadly defined, it may include "the processes of developing concepts for new systems, assessing effectiveness in the field, designing and manufacturing, and training in use." [36] In the interest of dividing out all possible genres of modeling and simulation, only the process of developing new concepts and their design will be considered under the heading of Acquisitions.

---

[35] Toyon Research Corp., "Introducing the SLAMEM/GVS User Group," 2006. *www.toyon.com/slamem_gvs.asp.*

[36] National Research Council. *Modeling and Simulation in Manufacturing and Defense Acquisition: Pathways to Success*. Washington, DC: The National Academies Press, 2002, p. 12.

Because the models and simulations used in acquisitions tend to be more specific, we will discuss two defense programs that have used modeling and simulation as part of the acquisitions process, rather than looking at particular models and simulations for the general acquisitions process. The Joint Strike Fighter (JSF) program and the Future Combat Systems (FCS) program have both used and benefited from SBA or Simulation Based Acquisitions. SBA embodies the ability to provide low-cost, high-quality systems over a shorter timeframe by using simulations.

In the case of the JSF program, which was designated as an SBA pilot program, simulations and models were incorporated throughout the entire process to attain insights and results more quickly. Using constructive simulations—interactive digital simulations as well as the Delphi process (a technique pioneered by the RAND Corporation to generate consensus and new knowledge)—and Quality Function Deployment (a group decision-making technique), the JSF teams brought together all elements of the acquisitions process to implement this new strategy.[37] The team also incorporated various other simulations, such as EADSim, to investigate all of the required activities of the JSF.

The FCS project proposed to use simulations throughout the entire process, but especially for supporting design as part of the acquisitions process. Intended to "develop the capability to rapidly project a dominant ground force anywhere in the world within days," FCS required and continues to require large numbers of models and simulations to support all aspects of the project.[38] Plans for FCS included using the TRADOC Analysis Center, the Army Materiel Systems Analysis Activity and the Army Research Laboratory, among others, to meet all the deadlines and requirements.

Using models and simulations for acquisition is still an evolving effort as the merits of SBA are evaluated and as projects such as the JSF and FCS mature. There is some concern in the community that a lack of validation and verification of some of the models and simulations that might be used will cause more problems than the possible expense of continuing to build full-size models of the systems under development.

---

[37] Zittel, Randy C. "The Reality of Simulation-Based Acquisition." *Acquisition Review Quarterly*, Spring-Summer 2001. *www.findarticles.com/p/articles/mi_m0JZX/is_2_8/ai_81763216/*

[38] Federation of American Scientists. "Future Combat Systems (FCS)." Washington, DC: Federation of American Scientists, May 17, 2000. *www.fas.org/man/dod-101/sys/land/fcs.htm*

Planning

Determining the size and composition of a military force and learning how to plan the missions of military forces can be done using a variety of models and simulations. Models and simulations of entire strategic military plans as well as those of tactical engagements exist. While most of these plans remain classified (and some highly classified), it is known that nuclear exchanges as well as certain conventional operations (such as those in Korea) have been modeled.

One program used for planning is the Accelerated Combat Timeline (ACT) developed by the Air Force Wargaming Institute at Maxwell Air Force Base, Alabama. ACT is a "two-sided simulated war game" that teaches campaign planning.[39] It allows users to enter an operational campaign plan and iterate it over many days (without the loss of fidelity). It also allows the user to halt the iterations and enter new information or to return to a previous event and enter new or updated information, such as troop orders. Using a combination of Excel, the Air Command Exercise System (ACES) engine, and a variety of display tools, ACT is a typical new military planning simulation for campaign planning.

Testing

Models and simulations used for testing and evaluation purposes have a varied history. In some parts of DoD, M&S has been well-supported and found useful for several different types of testing. However, there are still many who believe that the only way to really test a new weapon or vehicle is to build a prototype and run tests with it. Unfortunately, time and cost have begun to limit the practicality of building a new prototype for each stage in the testing process. Furthermore, as the complexity of systems and systems of systems increases, models and simulations may be the only way to meet the rapid testing cycle required. Modeling and simulation may also help to better plan the live tests that are conducted and can also represent systems that may be unsafe to test (e.g., countermeasures to attacks using weapons of mass destruction).

Research and Experimentation

This final group encompasses those genres that neither fit with the others nor merit in-depth consideration on their own. Using modeling and simulation for research

---

[39] Hammack, Lonnie and E. L. Perry. "The Accelerated Combat Timeline." Conference paper, I/ITSEC 2001.

or experimentation sometimes overlaps the genres above, but generally it reflects a small niche within the entire system.

In the case of research, existing models and sims may be used to consider new problems and gather more information about old problems. Research on the strategic level may have implications for the entire field of conducting war, while at the tactical level it may have a focused impact on a particular subject.

Experimentation within DoD as typified by Joint Forces Command falls into two basic categories: experiments that are "classical" in their methodological approach involving well-stated hypotheses and testing; and those that are discovery events or "expressions of curiosity."[40] While the first is relatively straightforward in its VV&A requirements, the latter is more problematic and may require a more innovative approach to VV&A. In addition, experimentation at this level may involve a series or campaign of experiments blending constructive, virtual, and live environments and as such represents an added level of complexity in control and observation.

Furthermore, as adversaries of the United States have changed the "curiosity" to understand a broader range of human action, collaboration and thinking have emerged pushing the experimental community to look past the large scale force-on-force models of the past and as a result has driven an interest in gaming amongst some of the experimental community.

---

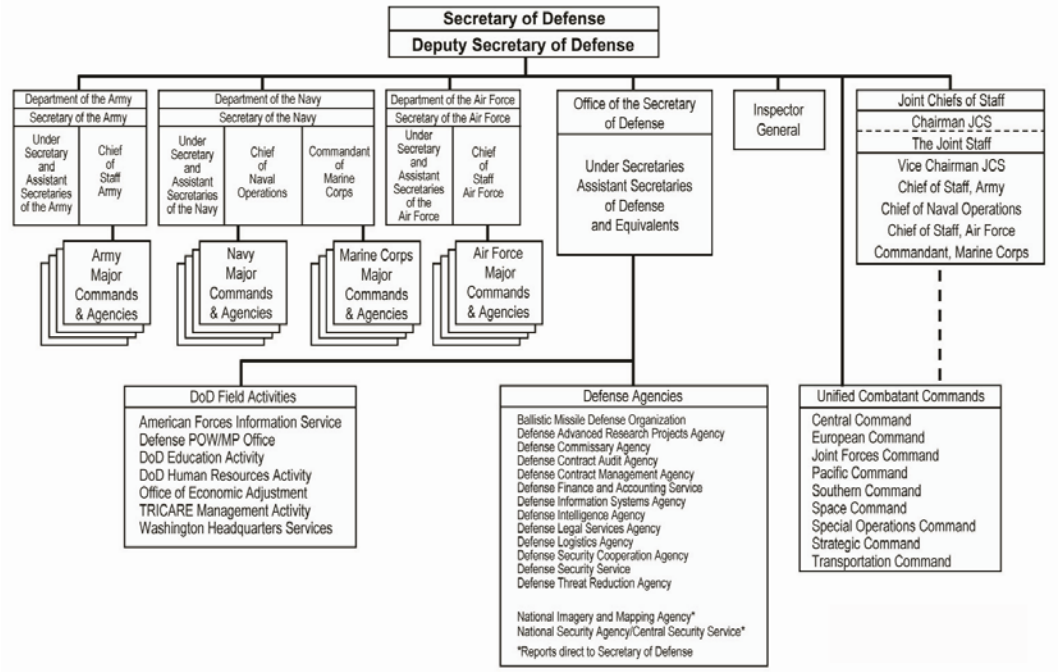
[40] Alberts, David S. and Richard E Hayes. *Code of Best Practice, Campaigns of Experimentation*. Washington, DC: CCRP Publications, Mar 2005, p. 15. *www.dodccrp.org/files/Alberts_Campaigns.pdf*

IV. Contracting with the DoD

Working with the government can be a trying experience—understanding who you're talking to, learning the process, and dealing with the idiosyncrasies of government contracting are critical to successfully landing a project. This section is intended to educate those who want to do business with the government and more specifically with the Department of Defense. It will also describe how the federal budget process influences many business interactions with the government. More valuable information about contracting with the government is available from www.firstgov.gov, a General Services Administration-based website that provides assistance to potential contractors.

DoD Organization

The US Secretary of Defense serves as the principal defense policy advisor to the President. He exercises control over the entire Department of Defense, which consists of the three military departments (the Army, Navy, and Air Force), the Office of the Secretary of Defense (OSD), the independent defense agencies, the Joint Chiefs of Staff, and the Unified Combatant Commands (Figure 2). The Secretary is responsible for developing and executing defense policy for the President and the nation.

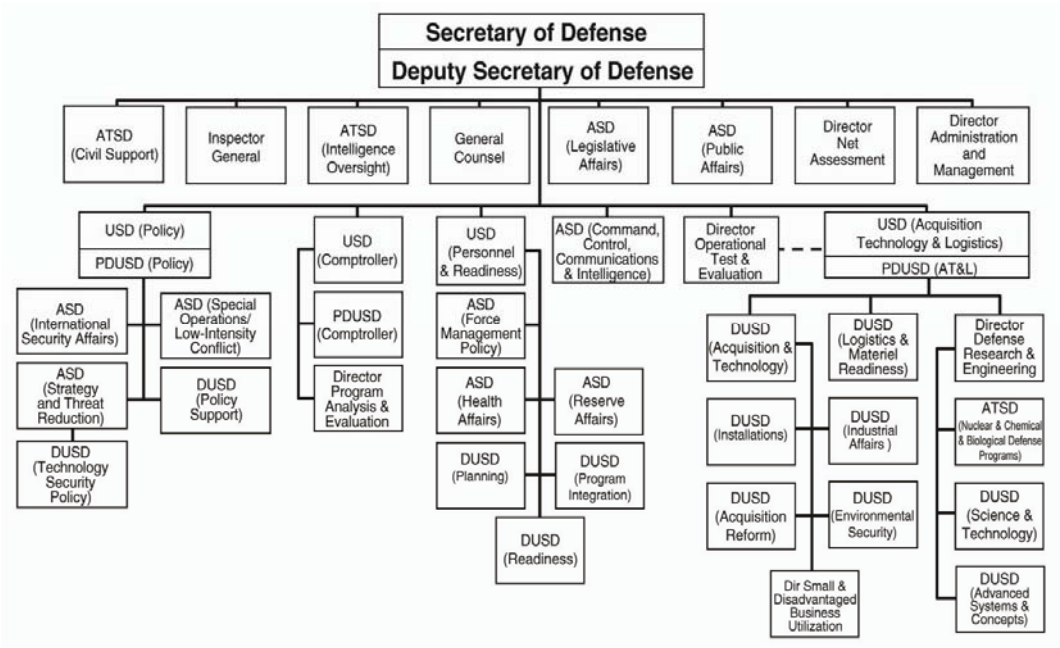


Source: Office of the Secretary of Defense

Figure 2. Organization of the Department of Defense

The three military departments, the Army, the Navy, and the Air Force, provide the military forces to the Secretary that allow him to carry out the President's instructions. Tanks, ships, and airplanes have to be designed and built. Soldiers, sailors, and airmen need to be recruited and trained. Each military department is headed by its own secretary (e.g., the Secretary of the Army) who reports to the Secretary of Defense, and each maintains a bureaucracy that enables it to provide these manned and equipped military forces. Day after day, month after month, year after year, recruiting, training, and acquisition activities support the military departments' mission—to provide competent military forces on a moment's notice.

The Office of the Secretary of Defense is the bureaucratic arm of the Secretary and helps develop and implement defense policy. It consists of several undersecretaries and assistant secretaries who handle personnel, research and development, procurement, and other support activities (Figure 3).



Source: Office of the Secretary of Defense

Figure 3. Organization of the Office of the Secretary of Defense

The Joint Chiefs of Staff serve as the military advisors to the President and the Secretary of Defense. The senior military officer from each of the services sits on the Joint Chiefs along with the Chief and the Vice Chief of Staff. The Joint Staff, under the direction of the Chief of Staff, supports the operations of the Joint Chiefs. The directorates of this staff support activities including personnel (J1), intelligence (J2), operations (J3), and logistics (J4), to strategic plans and policy (J5), command, control, communications, and computer systems (J6), operational plans and interoperability (J7), and force structure, resources and assessment (J8).

The defense agencies, as shown in Figure 2, provide department-wide support and typically report up through OSD or, in two cases, directly to the Secretary. These agencies include the Defense Advance Research Projects Agency (DARPA), which performs cutting-edge research for the entire department; the Defense Threat Reduction Agency (DTRA), which works to reduce the threat of weapons of mass destruction; and the National Security Agency (NSA), which collects and analyzes intelligence.

The Unified Combatant Commands (COCOMs) are the functional component of the Department; they're the ones that do the actual fighting. The COCOMs are assigned resources from the military departments and tasked to carry out missions by the President through the Secretary. Some COCOMs have geographical areas of responsibility,

such as Central Command that is responsible for military actions in the Middle East and parts of Africa. Some COCOMs have functional responsibilities such as Strategic Command that worries about nuclear deterrence, space, and information operations around the globe.

Potential MS&G Users

The OSD and each of the military departments maintains an MS&G policy office. These are the Defense Modeling and Simulation Office, the Army Modeling and Simulation Office, the Navy Modeling and Simulation Office, the Marine Corps Modeling and Simulation Management Office, and the Air Force Agency for Modeling and Simulation. While it is impossible to identify all potential MS&G users, the most obvious candidates are the fields of training, analysis, and acquisition. Historically, most projects have been training-related, but analysis and acquisition may be fertile ground for commercial game developers in the future.

Training

The military is an early adopter of models and sims for training. It was the Defense Advanced Research Projects Agency (DARPA) that developed the Internet and the early networked simulation called SIMNET.[41] The goal of these technologies was to deliver realistic and safe training environments to the military more cheaply and efficiently than other alternatives. The high cost of flight time, the dangers of live-fire exercises, and the continuing need for a well-trained fighting force make it easy to justify continuing large investments in MS&G. Both the Marine Corps and DARPA devote substantial fractions of their research, development, and acquisition budgets to exploring how commercial game technologies can be used to improve training.

Analysis

Analysis—more commonly referred to as "war gaming"—encompasses such areas as force structure development, experimentation, course-of-action analysis, and logistics planning. This work often involves repeating a battle simulation several times to see what works best and what doesn't. These exercises can answer such questions as: What

---

[41] Fuilford, Deb. "Distributed Interactive Simulation: Its Past Present and Future." Proceedings of the 1996 Winter Simulation Conference, J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain, eds., p. 179.

is the best mix of weapons to accomplish a particular task? What is the best way to complete the task with a given set of forces? or How should units be supplied so they don't run out of fuel or ammunition before the task is done? While the field of analysis holds great promise for commercial game developers, the current use of game technology in this area is limited.

Acquisition

The acquisition process is how DoD makes purchasing decisions. High fidelity models and sims are often used in studies to examine what combination of technologies will result in the best purchases. Applying these studies to the development and acquisition of weapons systems, for example, helps reduce the risk that the system won't perform as needed or will cost substantially more than anticipated. Also, once prototypes have been produced, simulations are frequently used to design the real-world tests that will evaluate whether or not a system meets its requirements. To date, the lack of fidelity, especially in physics, has limited the attractiveness of commercial games in this area.

Budgeting and Acquisition Process

The Planning, Programming, Budget Execution (PPBE) process is how DoD assembles its spending requests and sets long-term investment goals.

Each fiscal year begins the first day of the previous October (for example, FY06 began on October 1, 2005). All programs—and more importantly all *funding* for programs—is tied to the fiscal year and to Congressional approval of the authorization and appropriation bills.

Authorization and Appropriation

Constitutionally, Congress is responsible for providing funds to the executive branch of the government, including the Department of Defense. The DoD budget falls under the jurisdiction of four separate committees. The House Armed Services Committee and the Senate Armed Services Committee oversee DoD in matters of policy, and they pass the authorization bills that determine which programs will be approved. However, while authorization is important for any program, no money will flow without an appropriation. Actual funding decisions are made by the House Appropriations Committee and the Senate Appropriations Committee. These committees make

funding decisions for each of the program elements that the President submits in his budget. Any discrepancies between the committees are resolved in conference, and then the entire House and Senate vote on the final appropriations bill.

Continuing Resolutions

As soon as the President signs the appropriation and authorization bills, the department begins to disburse the money. Sometimes, when approval is delayed beyond the end of the fiscal year, Congress passes a "continuing resolution" that continues to fund the department at current levels. This allows existing programs to keep doing their work, but does not permit money for newly proposed programs to be disbursed. When this happens, contractors may find themselves in the difficult position of having projects that have been approved, but for which they cannot yet be paid.

Funding Models

Funding in the government comes in two forms: one-year and two-year money. The time frame refers to how long a contractor has to expend the money. Research, development, testing and evaluation (RDT&E) money is two-year money while almost all other funding is one-year. Before entering into a contractual agreement, it is very important to understand which kind of money is involved, as this can affect the project's timeframe. Non-research projects that extend beyond one year may suddenly find themselves without funds if their work is not approved and funded for the new fiscal year.

Procurement Procedures

To buy goods and services, the government contracting officials follow the procedures set forth in the Federal Acquisition Regulation (FAR), which can be found online at www.arnet.gov/far. All federal agencies are required to post procurement opportunities exceeding $25,000 at the government's "single point of entry" website for industry contractors, www.fedbizopps.gov. These opportunities can take many forms, the most relevant of which are outlined below.

Request for Proposal (RFP)

Most project contracts are the eventual result of a Request For Proposal. When a program manager is planning a project and identifies the need for a highly technical

product or service, he may publish an RFP that invites industry contractors to explain how they would fulfill that request and at what price.[42] The RFP is published publicly in order to attract the broadest level of participation by industry. The goal is not necessarily to identify the lowest price, but instead to find the contractor who will deliver the "best value" in satisfying the requirements.

Broad Area Announcement (BAA)

Broad Area Announcements are a convenient way for science and technology developers within the government to convey their needs to industry. BAAs are general statements about a technology requirement that are published in order to solicit competitive bids for "basic and applied research and that part of development not related to the development of a specific system or hardware."[43] Each BAA:

- describes the agency's research interest, either for an individual program or for broadly defined areas of interest covering the full range of the agency's requirements;
- describes the criteria for selecting the proposals, their relative importance, and the method of evaluation;
- specifies the period of time during which proposals will be accepted; and
- contains instructions for preparing and submitting proposals.
- Anyone can submit a proposal, provided they meet the filing instructions.

Small Business Innovative Research (SBIR) & Small Business Technology Transfer (STTR) programs

DoD participates in two major programs to facilitate commercial interactions with non-traditional contractors: the Small Business Innovative Research (SBIR) program, and the Small Business Technology Transfer (STTR) program. SBIR is a Congressionally-mandated program allocating 2.5% of federal research and development funding to foster small business innovation and the development of dual-use technologies. In FY2004, the program awarded just under $1.9B in grants to small businesses. The STTR program is similar to the SBIR program, except it requires participation of a university or non-profit partner.

---

[42] US Small Business Administration, "How the Government Buys." *www.sba.gov/businessop/basics/buys.html.*

[43] Federal Acquisition Regulation (FAR) 6.102(d)(2), General Services Administration FAR Secretariat, *www.arnet.gov/far/current/html/Subpart%206_1.html#wp1087648*.

SBIR/STTR solicitations that designate R&D topics are published twice each year. The most recent and upcoming DoD solicitations can be found online at www.dodsbir.net/. Submissions are evaluated by the individual agencies that submitted the research topics.

Small businesses that receive awards begin a three-phase program:

- **Phase 1** is a feasibility study to evaluate the merit of an idea. The award can be up to $100,000 over approximately one year.
- **Phase 2** expands on the work of Phase 1. Up to $750,000 can be awarded over two years to perform R&D work and to consider commercial potential. Only Phase 1 award winners are considered for Phase 2.
- **Phase 3** allows for the commercialization of the concept following Phase 2, but this phase carries no federal funding.

The government has an excellent online tutorial about these programs at www.dodsbir.net/tutorial/.[44]

Multidisciplinary Research Program of the University Research Initiative (MURI)

The MURI program supports research that spans more than one science and engineering discipline. Research topics are proposed annually by the participating defense entity. Awards are typically made for a three-year period with a two-year option bringing the total project length to as much as five years. Funding levels range from approximately $500,000–$1 million per year. Proposals must be submitted by US institutions of higher education with degree-granting programs in science or engineering. The program is currently administered by the Office of Naval Research.[45]

Unsolicited proposals

Unsolicited proposals are another way to gain funding. An unsolicited proposal is a written proposal for a new or innovative idea submitted to a government agency with the intent of gaining a government contract that is not made in response to a public request or announcement for proposals. Such a proposal should include details of the idea or technology in sufficient detail to allow evaluation, although the submitter would be wise

---

[44] Help is also available at *www.acq.osd.mil/osbp/sbir/solicitations/index.htm and www.dodsbir.net/*.

[45] Information about the program is online at *www.onr.navy.mil/sci_tech/3t/corporate/muri.asp*.

to identify proprietary information included in the proposal in order to protect his intellectual property. Unsolicited proposals have the disadvantage that funding may not be available, even for good ideas. Each agency has different policies and procedures for receiving unsolicited proposals, and these are usually published on the agency's website.

Types of Contracts

Government contracts can be divided into two broad categories: "fixed-price" and "cost reimbursement." Specific contract types range from firm-fixed-price, in which the contractor has full responsibility for the performance costs and resulting profit (or loss), to cost-plus-fixed-fee, in which the contractor has minimal responsibility for the performance costs and the negotiated fee (profit) is fixed. In between are the various incentive contracts, in which the contractor's responsibility for the performance costs and the profit or fee incentives offered are tailored to the uncertainties involved in contract performance.[46]

[46] Federal Acquisition Regulation (FAR) 16.101(b), General Services Administration FAR Secretariat, *www.arnet.gov/far/current/html/Subpart%2016_1.html#wp1085495*.

V. Project Development in the DoD

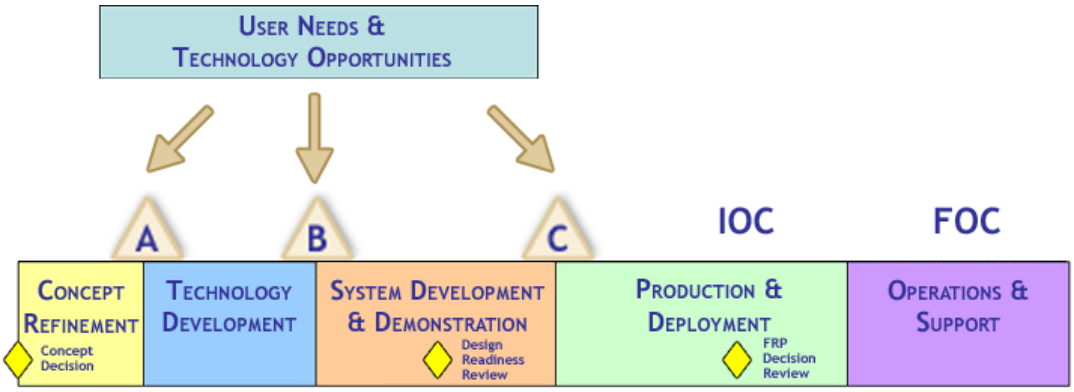
Government software products have traditionally taken longer to develop than commercial games. Although this is partially due to the higher standard of quality required of government software because of the serious consequences of failure or malfunction, it is also due in large measure to the close link between the government's project development models and its complex approvals and funding process. Recent reforms have been instituted that are designed to speed the overall process, but differences between the government and commercial worlds remain. This section will examine how government projects have traditionally been developed, how the acquisition process affects the development lifecycle, the intended effect of recent reforms, and issues that are likely to arise when game developers and the government work together.

The "Waterfall" Development Model and the Defense Acquisition Framework

Historically, the government has embraced the classic "waterfall" model of project development, where all the end requirements are specified at the beginning, and rigid boundaries are between each stage of development. These stages are:

1. Concept development
2. Requirements analysis
3. Architectural design
4. Detailed design
5. Coding and debugging
6. Testing

The steps in this process fit well into The Defense Acquisition Framework (See Figure 4.)



Source: US Department of Defense

Figure 4. Defense Acquisition Management Framework

The acquisition process comprises five steps:

1. Concept Refinement
2. Technology Development
3. System Development and Demonstration
4. Production and Deployment
5. Operations and Support

The process begins when someone within DoD identifies a military requirement and develops an Initial Capabilities Document (ICD) describing the capability needed to satisfy that requirement. The ICD assesses competing approaches; considers risk, cost, and responsiveness; and recommends one approach based on this assessment.

If the ICD finds a funding sponsor, most projects go into the Concept Refinement phase (although it is possible for a project to enter the framework at any of the first three phases). This phase explores the originally-considered approaches through an Analysis of Alternatives (AoA) and a Technology Development Strategy (TDS). The TDS identifies the development model (evolutionary or single-step-to-full-capability); the cost, schedule and performance of both the total program and the first spiral; and the test plan for the first spiral. At Milestone A (in Figure 1), Concept Refinement ends with the approval of the TDS by the Milestone Decision Authority (MDA), whereupon the program enters Technology Development.

The goal of this second phase is to reduce risk and to decide which technologies will be developed. It is an iterative process involving close interaction between the science and technology community, the user, and the developer, and it is designed to "as-

sess the viability of technologies while simultaneously refining user requirements." [47] The project team creates a Capabilities Development Document (CDD) that describes how the technology will lead to joint warfighting capability. The Technology Development phase ends when an affordable and relevant capability has been developed and the supporting technology demonstrated. At Milestone B, the MDA approves the CDD and the program enters the System Development and Demonstration phase.

During System Development and Demonstration, the project team concentrates on system integration and demonstration. It focuses on reducing manufacturing risk, minimizing the logistical footprint, and implementing the human-system integration. In addition, the program must demonstrate system integration, interoperability, and utility. In the middle of the phase, the program must pass a Design Readiness Review in order to continue. The phase ends when the team can demonstrate that the program works (at least in prototype) and that the team has developed an adequate test and evaluation plan. When the MDA approves Milestone C, the program moves on to the next phase.

The Production and Deployment phase often consists of two subphases: low rate initial production (LRIP), and full rate production and deployment (FRPD). However, software systems typically do not go through LRIP. The goal of this PD phase is to bring the system to operational readiness. In the case of software, the program must demonstrate the level of maturity specified in the Capability Production Document (CPD), and the program manager may decide to test the software with a limited deployment before moving it into an operational environment. Moving into full production and development requires completing the Initial Operational Test and Evaluation (IOT&E) and the approval of the MDA.

In the final phase, Operations and Support, the team develops and executes a plan to sustain the system in the most cost-effective way over its total life cycle, and to dispose of it when it has reached the end of its useful life. While required for all systems, disposal for software systems is relatively trivial compared to weapons systems that may contain hazardous materials requiring specialized handling.

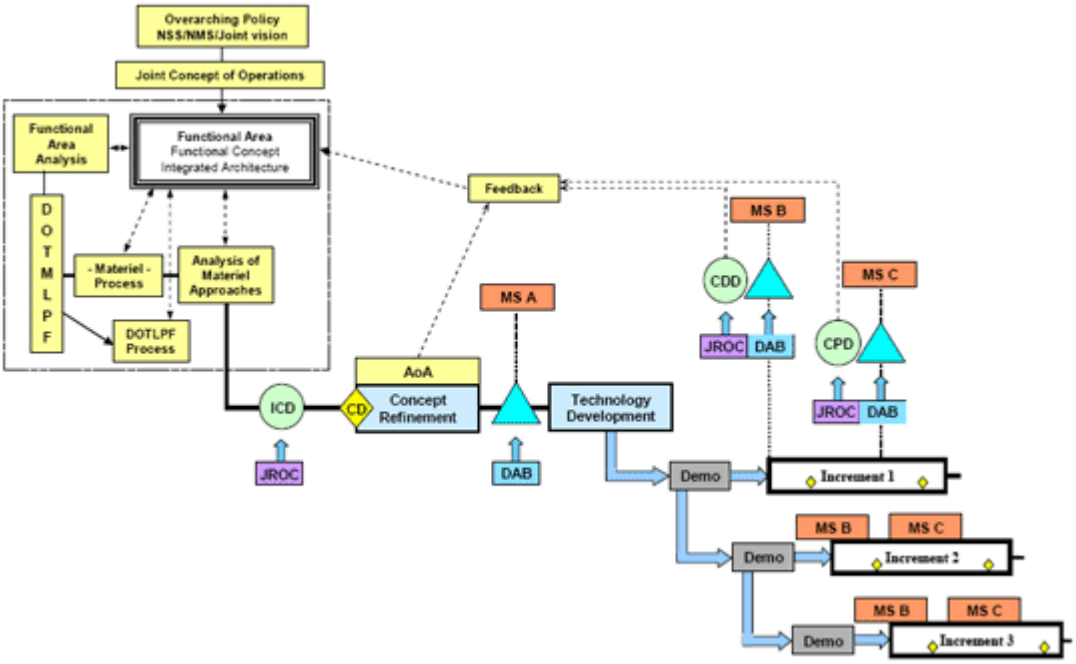
---

[47] US Department of Defense. DoD Instruction 5000.2 "Operation of the Defense Acquisition System" Section 3. May 12, 2003.

Recent Reforms

The government has realized that its traditional acquisition paradigm has led to unacceptably long procurement cycles, especially in the rapidly moving area of technology development. In response, DoD has moved from a requirements-based process to a capabilities-driven one known as JCIDS (Joint Capabilities Integration Development System) where evolutionary development is embraced (at least at a policy level), as opposed to the previous "specify-all-end-requirements-before-you-start" style.

"Evolutionary acquisition is the preferred DoD strategy for rapid acquisition of mature technology for the user. An evolutionary approach delivers capability in increments, recognizing, up front, the need for future capability improvements. The objective is to balance needs and available capability with resources, and to put capability into the hands of the user quickly. The success of the strategy depends on consistent and continuous definition of requirements, and the maturation of technologies that lead to disciplined development and production of systems that provide increasing capability towards a materiel concept." [48] (See Figure 5.)



Source: US Department of Defense

Figure 5. Evolutionary Requirements and Acquisition Process

[48] US Department of Defense, 2003.

In addition to adopting evolutionary processes and spiral development, the government has adopted other reforms to meet its goal of acquiring "quality products that satisfy user needs with measurable improvements to mission capability and operational support, in a timely manner, and at a fair and reasonable price." [49] These changes include:

- Integrating test and evaluation into development
- Including the total cost of the system in the acquisition decision
- Addressing some of the key shortcomings of the old system, such as the way in which excessive requirements for interoperability and sustainability adversely affect development speed and overall affordability
- Allowing programs to enter the process at the most appropriate phase (subject to legal and entrance criteria requirements).

Overall, the government's goal is to deemphasize process and encourage innovation, while maintaining proper accountability and oversight.

Challenges

Though the spirit of this new process may appear familiar to program managers in the game and software development business, problems remain.

Bureaucracy

Even with reforms in the acquisition system, the government is still a very bureaucratic environment. Thousands of pages of federal regulations (Federal Acquisition Regulation and the Defense Federal Acquisition Regulations), and detailed policy instructions (Department of Defense Directive 5000.1, Instruction 5000.2, the Defense Acquisition Guidebook, and the Chairman of the Joint Chiefs of Staff Instruction and Manual 3170.01)[50] engender substantial process and reporting requirements for program managers and technology developers. Some knowledge of these documents and the processes they support and implement will help avoid unforeseen and undesired difficulties.

---

[49] US Department of Defense. DoD Instruction 5000.1 "Operation of the Defense Acquisition System" Paragraph 4.2. May 12, 2003.

[50] Defense Acquisition University, Defense Acquisition Policy Center, CJCSI 3170.01 E and B, May 11, 2005, *http://akss.dau.mil/dapc/index.html.*

Semantic differences

Even though government and industry are using the same words, misunderstandings often arise because some words have different meanings in each environment. For example, "spiral development" has a substantially different meaning in the game development community than in government circles.

In the game community, spiral development is used to discover what a game "wants to be" while it is being built. The specific requirements are usually not known when the contractual agreements are established, which requires a significant amount of trust between developer and customer. To a game developer, spiral development is useful precisely because it leads to the discovery of the game's requirements.

From the government's point of view, spiral development is a way of incrementally delivering already-known requirements. While the requirements for the project's overall final capabilities may not necessarily be known at the kickoff of the first spiral, the requirements of this first spiral are explicitly identified in the contractual agreement at project initiation.

In other words, game developers use spiral development for experimentation and discovery, while government developers use spirals to divide a project into pieces that they can incrementally deliver. The difference is enormous.

Both government project managers and industry contractors should be aware of semantic differences such as these and work hard to understand each other's "language" at an early stage of discussion.

"Ilities," Standards and VV&A

Considering the billions of dollars DoD spends each year on modeling and simulation, it is not surprising that it tries to get the most bang for its acquisition buck. This desire has given rise to a group of requirements known informally as the "Ilities," which includes such concepts as reusability, scalability, composibility, reliability, affordability, usability maintainability, manageability, adaptability, survivability, etc.

The formal statement of this desire is DODD 5000.59 which states:

> "investments shall promote the enhancements of DoD M&S technologies in support of operational needs and the acquisition process; develop common tools, methodologies, and databases; and establish standards and protocols

> promoting the internetting, data exchange, open system architecture, and software reusability of M&S applications…" and "foster programs to develop and, where applicable, implement DoD M&S interoperability standards and protocols." [51]

A more complete discussion of the Ilities, standards, and Validation, Verification, and Accreditation (VV&A) can be found in the Government Standards and Government Specific Requirements sections of this report.

For purposes of this discussion, it is important that while these concepts are central to the government development community, they are unfamiliar (if not totally foreign) to game developers. Business pressures on commercial developers influence them to design and code only the minimum feature-set necessary to bring a game to market. "Extraneous" considerations, such as reusability, interoperability, or conforming to a standard—even for games developed within the same company—are typically regarded as "nice-to-haves" for the company, but disadvantageous for individual projects, and therefore ignored because, as one project manager put it, "No good deed goes unpunished."

Cultural Differences

The DoD work environment and that of the typical game development company could hardly be more dissimilar.

Game developers often start the day late, sometimes as late as noon. They work in visually and aurally chaotic environments. Their dress is sometimes beyond casual. They play games periodically throughout the day, and often stay in the office through the night. It is sometimes hard to the outside observer (or manager) to believe that any work is being done.

By contrast, the military day often begins at dawn, the workplace is regimented, a rigid dress code is second nature, and game-playing is regarded as frivolous, at best. When these cultures collide, it is difficult to predict the results. Managers on each side must make allowances for the working conventions of the other. Game developers have been known to "clean up their act" on days when their government clients visit. Government project directors often turn a blind eye to sights and sounds that make them question the efficiency of the gamers' workplace. When a project is going well, these accommodations

---

[51] US Department of Defense. DoD Directive 5000.59 "DoD Modeling and Simulation (M&S) Management," Jan 4, 1994. *www.dtic.mil/whs/directives/corres/html/500059.htm*

are generally not difficult to make. When a project is going poorly, however, it is not uncommon for each side to lay the blame at the cultural doorstep of the other.

Changes in Personnel

In the commercial sector, enormous efforts are made to keep a project team together throughout a product's development. Changes in personnel are regarded as disastrous. Pressures from both management and their peers often influence unhappy employees to stick it out until a game ships "for the good of the project."

The military, on the other hand, typically rotate program managers in and out of billets every two-to-three years, with the goal of developing well-rounded officers possessing a wide range of experience.

This policy can cause severe disruption to a software development project, both before and after the change in command. Six months before the current program manager is scheduled to leave, a project may begin to stagnate in anticipation of his departure and the arrival of a replacement. Once the new program manager arrives and tries to establish control and ownership of the project, he may revisit significant project and design details that had previously been agreed upon earlier. This transition can easily lead to wasted months and confusion as the program adjusts to new leadership.

Game companies that work with military clients should be aware of this rotational policy from the outset, and the project team should develop plans that minimize the disruption of the transition. DoD, for its part, should be aware of the impact this policy has on game projects, and consider timing its personnel moves to cause as little damage as possible.

VI. Game Engines and the DoD

To date, the Department of Defense has not taken a service-wide approach to developing or licensing game engines. Instead, evaluating and selecting engines has been made at the individual project level. This is appropriate, given the varied demands of the users. The most effective action the government can take in this area is to improve information-sharing among project managers, encouraging them to discuss the features of the widely-disparate available engines, and to compare notes on the different rationales for selecting them.

Engines are important to DoD because the government must make a "Build vs. Buy" decision with each of the games it develops. On one hand, there is a natural urge for the government to build an engine to its own specifications. On the other hand, it might be more cost effective to take advantage of work that has already been done in the private sector. This may especially be true because, unlike in years gone by when the government was the driving force behind the development of high-end technology, the commercial market now leads the charge, spurred in no small part by demands for ever-increasing capabilities from the gaming community. As a 2003 IEEE article has claimed, "By piggy-backing on game engines…military simulations can incorporate technological advances in lockstep with the commercial market without bearing development expenses their low volume could not support." [52]

Government-sponsored Engines

Historically, DoD has sponsored two efforts in the area of engine building: the DIS/HLA architectures, which are discussed more thoroughly under the "Government Standards" section of this report; and the Delta3D "Open Source" engine being developed at the Naval Post Graduate School, discussed here.

---

[52] Prasithsangaree, P. et al. "UTSAF: A Simulation Bridge between OneSAF and the Unreal Game Engine." Pittsburgh, PA: University of Pittsburgh, copyright IEEE, 2003.

The DIS/HLA architectures have their roots in SIMNET, a real-time distributed combat simulator that DoD created through the Defense Advanced Research Projects Agency (DARPA) in 1983. SIMNET became the basis for protocols for Distributed Interactive Simulation (DIS) which, along with its HLA successor, is the architecture underlying the current generation of networked simulations, including ModSAF and its descendents JSAF and OneSAF OOS, the embedded simulation engine for the Future Combat System.

Any commercial entity wishing to discuss engines with DoD personnel would be well-served to understand the mindset and the assumptions underlying ModSAF, JSAF, and OneSAF, as these and their associated products have permeated and defined DoD's thinking about the simulation space for some time.[53] Conversations about engines and standards are likely to begin with these products as a reference point. A discussion on the evolution of these standards appears in Steinman and Hardy's "Evolution of the Standard Simulation Architecture." [54]

The latest in this line of products, OneSAF OOS, deserves special evaluation and scrutiny. In development since 1998, the product is still in beta as of this writing. Its origins lie in Army—rather than Joint Forces—requirements, and it is unclear what set of features will be in the final version and what kinds of products it will be capable of delivering. It may be that OneSAF OOS will allow commercial game companies and their related engines to interface with large-scale DoD simulations, but it is beyond the scope of this report to determine whether that will be feasible.

The Delta3D Engine is more specifically targeted at video game development. Developed by the MOVES Institute at the Naval Post Graduate School, the engine is actually a federation of open-source sub-modules that integrate "well-known Open Source projects such as Open Scene Graph (OSG), Open Dynamics Engine (ODE), Character Animation Library (CAL3D), and OpenAL as well as projects such as Trolltech's Qt, Crazy Eddie's GUI (CEGUI), Xerces-C, Producer, InterSense Tracker Drivers, HawkNL, and the Game Networking Engine (GNE). Rather than bury the underlying modules,

---

[53] For more information. ModSAF *www.aiai.ed.ac.uk/~arpi/SUO/MODULES/modsaf.html,*
JSAF *www.cs.odu.edu/~tang_j/dtic/pdf/ADA417477.pdf,*
OneSAF *www.onesaf.org/public1saf.html; Delta3D www.delta3d.org/*.

[54] Steinman, Jeffrey S. and Douglas R. Hardy. "Evolution of the Standard Simulation Architecture." 2004 CCRT Symposium, US Department of Defense Command and Control Research Program. *www.dodccrp.org/events/2004/CCRTS_San_Diego/CD/papers/067.pdf*

Delta3D pulls them together in an easy-to-use API which allows direct access to important underlying behavior. Delta3D does its rendering using OpenSceneGraph and OpenGL. According to its developers, their goal is to provide a license-free engine for "the roughly 90% of DoD gaming applications that don't require a huge commercial engine." [55] More information on the engine can be found at www.delta3d.org.

Evaluating Commercial Engines

Evaluating the multitude of available engines is a daunting but necessary task. According to the website www.devmaster.net, there are over 200 commercial and open-source engines available. Meanwhile, the number of video games used or sponsored by the DoD is large and growing rapidly. The DoD gaming community website, www.dodgamecommunity.com, lists 65 games that are already being used by the military, and many more are in development. As this number rises, it becomes increasingly important for DoD personnel to understand the functionality that an engine provides and how to evaluate it against a project's needs. Likewise, game engine companies wanting to do business with DoD should be aware of the military's increasing interest in games and adapt their engines accordingly.

The following are important considerations when evaluating an engine.

Features and "fit"

- Does the engine have the necessary features to create the game?
- Are these features optimized for the style of game to be created? For example, all engines have a rendering component that draws images on the screen, but an action-game engine may be optimized for line-of-sight calculations (to determine what the player can see), while a strategy-game engine may be optimized to display as many units as possible at any one time.
- Is the engine "overkill"? In other words, does it have components that a particular game may not need, components that make the engine expensive to license and overly complicated to use?

Table 5 compares eight commercially available game engines presented by Mike McShaffrey of BreakAway, Ltd in his I/ITSEC 2005 paper, "Capabilities of Today's

[55] McDowell, Perry, (executive director, MOVES Institute, Naval Post Graduate School), private conversation with authors, Sept 19, 2005.

Engines." While this is not an exhaustive list of engines or capabilities, it does effectively demonstrate how one might classify capabilities and needs.

Table 5. Engine Capabilities

	Game Engines								
	Gamebryo	Gamebryo 2	Glacier	Half Life 2 (Source)	Reality Engine	Renderware Platform	Torque Game Engine	Unreal 2	Unreal 3
Platform & OS									
Windows	✓	✓	✓	✓	✓	✓	✓	✓	✓
Xbox	✓		✓		✓	✓		✓	✓
PS2	✓		✓			✓		✓	✓
Gamecube	✓					✓		✓	✓
Xbox360		✓							✓
PS3		✓							✓
Other (Linux & MacOS)							✓	✓	
Handheld (PSP, N-Gage)						✓			
Scripting	○	○	◒	○	◒	◒	◒	◒	●
Editors/Tools	◒	○	◒	●	◒	◒	◒	◒	●
Physics	◒	○	●	◒	●	◒	◒	◒	●
Lighting	◒	○	◒	◒	●	◒	◒	◒	●
Shadows	●	○	◒	◒	◒	◒	◒	◒	●
Texturing	●	●	◒	◒	●	●	◒	◒	●
Scene Management									
BSP				✓			✓	✓	✓
Portals	✓	✓		✓		✓	✓	✓	✓
Occlusion Culling				✓	✓	✓	✓		
Spatial Database							✓		
Scene Graph	✓	✓	✓						
Potentially Visible Set (PVS)						✓			
Animation	●	●	◒	◒	○	◒	●	◒	●
Meshes	◒	◒	◒	◒	◒	◒	●	◒	◒
Surfaces & Curves	○	○	●	◒	○	◒	○	○	●
Special Effects	◒	◒	●	◒	●	◒	◒	◒	●
Terrain	○	○	◒	◒	○	◒	●	◒	◒
Networking	○	○	◒	◒	◒	○	●	◒	◒
Sound/Video	○	○	●	○	●	◒	●	◒	●
Artificial Intelligence	○	○	●	●	◒	●	○	●	●
Rendering	◒	◒	◒	○	◒	●	◒	◒	●

○ = No capability ● = 100% capability

Source: "Capabilities of Today's Engines" Mike McShaffrey, I/ITSEC 2005.

Level of Support

- Some commercial games ship with their engines included to encourage fans to create "mods" that extend the game's life. Support for these engines varies. Some companies ship them "as-is," with little documentation and no technical support. Other companies employ mod community managers who act as a link to the technical team to provide help to game-builders using the engine. In either case, the company's financial goal in releasing the engine is to encourage more people to buy the game, rather than to make money by charging for the engine itself, and the company is under no obligation to respond to specific requests for engine fixes or changes. This style of engine is best suited to projects where the requirements line up well with what the engine already provides, rather than projects that will require engine modifications to make them run.
- Other commercial engines are offered by companies in business to make money, in whole or in part, by licensing the engine itself. These companies provide ongoing improvements, bug fixes, and technical support for licensees. The more complex a game is, the more likely it is to require such an engine.
- Some open-source engines are available with no formal support, but with robust communities that provide help to their users. This is a hit-or-miss proposition for a project team that has a technical problem, depending on whether the problem is unique to their game or one that has been encountered before. (The DoD may also have security issues with open-source engines, which are dealt with in the "When Worlds Collide" section of this report.)
- DoD's JSAF users may find difficulty finding technical support. According to a 2003 Northrop Grumman study, "Although JFCOM occasionally sponsors JSAF users conferences, there does not appear to be an active JSAF user/developer community, with mailing lists, news groups, and other such resources, where new users can go to find answers to their questions. There does not appear to be a JSAF help desk, or other similar type of support mechanisms. If these facilities exist, they are not well advertised. The lack of such facilities makes installation and use of the JSAF software more difficult." [56]

---

[56] Trott, Kevin. "Command, Control, Communications, Computer, Intelligence, Surveillance and Reconnaissance (C4ISR) Modeling and Simulation Using Joint Semi-Automated Forces (JSAF)." Rome, NY: Northrop Grumman Inc., 2003.

Stability

- The performance requirements of both DoD and commercial applications vary widely. Some applications have almost no room for error or failure, while others are more tolerant. Designers should evaluate the "error tolerance" of an engine when selecting it for a particular application. Additionally, project teams who have access to help from an engine's licensor (or a support community) have a better chance of reducing errors than teams working with unsupported engines.
- The older an engine is, the more likely it is to be stable. Unfortunately, an engine's age may also indicate that it lacks up-to-date features. Project managers should determine whether a game needs to be "on the bleeding edge" of technology, or whether they can meet project goals with an engine that may lag behind the latest and greatest.

Scope of the License

- Engine licenses vary widely with regard to how many "seats" the license conveys. Some licenses require a separate purchase for each user of the end product. Others have a one-time fee that allows unlimited users once the initial license has been acquired. Still others fall somewhere in-between, with prices established for different ranges of users.
- Another scope-related problem is the issue of which entities are allowed to create products using the engine. For example, the license that *America's Army* negotiated for the *Unreal* Engine allows any Army entity to create games using the engine, but not the Navy, Air Force, or other armed services. (An improved community, as called for above, might help DoD reduce overall costs by negotiating licenses that are appropriate to their potential use. For example, a cross-service license may reduce the acquisition cost of an engine that has widespread application; whereas avoiding the higher cost of a cross-service license may be more appropriate for an engine that is being acquired for a project than only one service will use). That said, once a service has acquired a license, it must still take care to match each of its projects with the most suitable engine. All too often, a license holder is constrained in its thinking because of a strong desire to use a license it has already acquired, rather than negotiate a new one. For example, someone with an unlimited *Unreal* license may try to develop all his projects using that engine, even though a different engine may be a better fit.
- A third dimension of the scope of the license is what subcomponent rights (and obligations) come with the engine. Many modern engines are actually a collection of

"sub-engines," and even the most sophisticated commercially-available engines often contain sub-components which are themselves licensed. (For example, the *Unreal* engine uses Ogg Vorbis for audio encoding and Havoc for physics. Ogg Vorbis is an open-source codec,[57] and it requires the display of a separate copyright notice on any product that uses it. Other codecs or components require users to publish their source code, or to make improvements available to the entire community, or they may require separate licensing agreements if the application falls outside the terms of service granted in their original license. In all cases, the commercial provider and the government should consult legal counsel with particular knowledge in this area before proceeding with development.)

Price

- The "Build vs. Buy" decision is often a difficult one. As with many technology-driven organizations, many game developers have a "Not Invented Here" aversion to software that they have not themselves developed; this often leads them outside their field of expertise, trying to develop engines that they would be better off purchasing. Before allowing their contractors to undertake engine development work, government project managers should analyze what features their game will need, and review available engines to see if it would be less expensive to license than to build.
- Total cost awareness, a holistic view of cost across programs and projects, is a difficult—yet critical—part of establishing the appropriate license and fees for a given engine. (This is discussed further in the Government Project Development and Contracting sections of this report.) Both the licensor and the government licensee should consider the breadth of the license. Acquisition officers would be well advised to invite as many potential government users as possible into the process, so that a correct scoping can be made and the potential for cost-sharing can be examined.
- New business models are also emerging that may help reduce price. Increasingly, commercial game developers are exploring alternative business models that include in-game advertising, product placement, and product-specific financing. While this is generally a product (rather than an engine) issue, some engines are being created with

[57] "Codec" is short for "coder-decoder." Most audio and video formats use some sort of compression so that they don't take up a ridiculous amount of disk space. Files are compressed with a certain codec when they are saved and then decompressed by the codec when played back. Common codecs for video files include MPEG and AVI, and WAV and AIFF for audio files. Codecs can also be used to compress streaming media.

these capabilities in mind, and the government may be able to derive some cost-savings from using engines whose development has already been subsidized by other funding resources.

Conclusion

This section has described the history of government-sponsored game-engines and set forth some guidelines for evaluating engines against a product's needs. Commercial game developers should be aware of the government's mind-set, and particularly the ways it has been shaped by past experiences. Government decision-makers should understand that there is no single "best" engine, and that a project's intended use and scope are the biggest factors affecting the engine acquisition decision.

VII. DoD Standards and M&S

The nice thing about standards is that there are so many of them to choose from.

—Andrew S. Tanenbaum

Since its earliest formation, the US Military has had an interest in standards. It has standardized everything from rations and rifle loads, to buildings and training, and much more. The concept of being able to reduce almost anything to fundamental components that can be quickly and efficiently interchanged has been woven into the very fabric of US military thinking. This approach had definite advantages in industrial-age warfare, where the attrition of brute force-on-force clashes demanded rapidly replacing people and parts. But in today's volatile and fast-paced environment, maintaining a proper balance between innovation and standardization has become difficult.

The DoD defines standardization as:

> "[T]he process by which [it] achieves the closest practicable cooperation among the Services and Defense agencies for the most efficient use of research, development, and production resources, and agrees to adopt on the broadest possible basis the use of: a. common or compatible operational, administrative, and logistic procedures; b. common or compatible technical procedures and criteria; c. common, compatible, or interchangeable supplies, components, weapons, or equipment; and d. common or compatible tactical doctrine with corresponding organizational compatibility."[58]

(Contrary to popular usage, the government maintains a distinction between "standards" and "specifications,"[59] and this document does not use the terms interchangeably.)

---

[58] US Department of Defense. "DoD Dictionary of Military Terms." Washington, DC: Joint Doctrine Division, Aug 2006, *www.dtic.mil/doctrine/jel/doddict/*.

[59] According to the Government Accountability Office, military specifications "describe the physical and/or operational characteristics of a product," while military standards "detail the processes and materials to be used to make the product." *http://government.ihs.com/standards/military-specifications/mil-specs-learn-more.htm.*

DSP and the "Perry Memorandum"

DoD's belief in standards is expressed in the *Defense Standardization Program, Policies and Procedures (DSP)*, which states, "Standardization is *beneficial* [emphasis added] in achieving interoperability, ensuring products meet certain requirements, commonality, reliability, total cost of ownership, compatibility with logistics systems, and similar defense-related objectives."[60] However, there is evidence that standardization in and of itself is not always beneficial. In the early 1990s, the government recognized that the almost 30,000 DoD standards (some estimates range as high as 45,500) imposed unnecessary restrictions, increased costs, and hindered using the latest technology. So strong were the arguments against military standards that in June 1994, Secretary of Defense William Perry issued what has now become known as "the Perry Memorandum," a directive to severely reduce the use of military specifications and to rely wherever feasible on voluntary commercial standards, whether official, *de facto*, or *de jure*.[61]

Nevertheless, vendors working with DoD should be sensitive to the concerns and sensitivities expressed in the DSP statement. The government still desires quality, efficiency, reliability, fitness of use, and a low total cost of ownership, and standards still play an important role in pursuing those goals.

The Defense Modeling and Simulation Office (DMSO)

Interestingly, 1994 also saw the DoD issue Directive 5000.59 to establish policy, assign responsibilities, and prescribe procedures for managing modeling and simulation. As part of that directive, the Defense Modeling and Simulation Office (DMSO) was created to build a community of interest, oversee planning, provide direction, and coordinate amongst the Services.[62] As a result, whether through default or design, coordinating standards with cross-service application within M&S came to reside within DMSO.

DMSO soon took ownership of emerging M&S standards, such as the Synthetic Environment Data Representation and Interchange Specification (SEDRIS) and the

---

[60] US Department of Defense. "Defense Standardization Program, Policies and Procedures (DSP)." Washington, DC: Office of the Under Secretary of Defense (Acquisition, Technology and Logistics), Mar 2000. *www.dsp.dla.mil/policy/docs/4120-24m.pdf.*

[61] Perry, William J. SECDEF Memo "Specifications & Standards—A New Way of Doing Business." DTD 29 Jun 94, *https://acc.dau.mil/CommunityBrowser.aspx?id=32397*.

[62] US Department of Defense, 1994.

High Level Architecture (HLA). It is not this report's task to debate the pros and cons of those standards or their implementation. However, it is appropriate to note that each took over eight years to develop (SEDRIS and all its elements are still not completely though the standards process), and each has been met with varying opinions of success. In addition, despite management pressure, both of these standards are often disregarded when economic necessity dictates.

Thus, while the Perry Memorandum encourages using and adopting "voluntary standards," the reality is that the department still has a predisposition to enforce standards through top-down management directives, rather than a bottom-up approach whereby projects voluntarily adopt the standards that seem appropriate. And while it has made progress since 1994, DoD still has a long way to go before it is a "voluntary" market-based organization with regard to standards.

In 2006, DoD's management approach to M&S changed and, as of this writing, DMSO's charter is being revisited and realigned. Originally DMSO, as directed through the SD-1 "Defense Standardization Program," was the Lead Standardization Activity (LSA) for the Modeling and Simulation Standardization Area (MSSM).[63] But now it is difficult to understand who within DoD will bear the Standards responsibility, whether this responsibility will remain within the scope of DMSO's replacement, or whether it will be dispersed amongst interested parties.

Simulation Interoperability Standards Organization (SISO)

Another organization that influences the DoD M&S standards arena is the Simulation Interoperability Standards Organization (SISO), which originated in 1989 with a small two-day conference called "Interactive Networked Simulation for Training." The conference attracted approximately 60 people who were concerned that the few existing networked simulation efforts were operating in isolation. The group believed that the technology would advance more rapidly if there were some means to exchange information between companies and groups. They also believed that as the technology began to stabilize, standards would become necessary, and that these standards could be established by consensus agreement of the community.

---

[63] A list of standardization documents for which the DSPO has primary responsibility can be found at *www.dsp.dla.mil/documents/sds.htm*.

The conferences soon evolved into the Distributed Interactive Simulation (DIS) Workshops, and became focused on creating standards around SIMNET, the first real-time distributed combat simulator. In 1996, DIS became SISO and is now an international organization dedicated to promoting M&S interoperability.[64] The organization is recognized as a Standards Development Organization by NATO and as a Standards Sponsor by IEEE.

The Advanced Distributed Learning (ADL) Initiative and SCORM®

Another standardization effort that might affect game developers working with DoD is the Advanced Distributed Learning (ADL) initiative, which was jointly established in 1997 by the DoD, the White House Office of Science and Technology, the Department of Labor, and others as "a collaborative effort to harness the power of information technologies to modernize structured learning," employing "a structured, adaptive, collaborative effort between the public and private sectors to develop the standards, tools and learning content for the learning environment of the future." [65] While ADL has little to say about the standardization of game production tools and technologies, their SCORM® specification will have a great impact on designing and implementing games required to be SCORM-compliant.

SCORM is the "Sharable Content Object Reference Model," a collection of standards and specifications designed to "provide a comprehensive suite of e-learning capabilities that enable interoperability, accessibility and reusability of Web-based learning content."[66] The specification's Content Aggregation Module (CAM) describes the components of a learning experience and how to package those components to exchange between learning systems. Its "simple sequencing" component is a standardized way for a learning system to know in what order to deliver content to learners. According to ADL, "SCORM is the first step on the path to defining a true learning architecture." [67]

---

[64] SISO's website is *www.sisostds.org*.

[65] Advanced Distributed Learning, "About ADL," OUSD P&R, 2005, *www.adlnet.gov/aboutadl/index.cfm*.

[66] Advanced Distributed Learning. "SCORM 2004." Third edition, 2004, *www.adlnet.gov/scorm/index.cfm*.

[67] Advanced Distributed Learning, "Introduction to SCORM and the ADL Initiative," presentation, *www.adlnet.gov/aboutadl/index.cfm*.

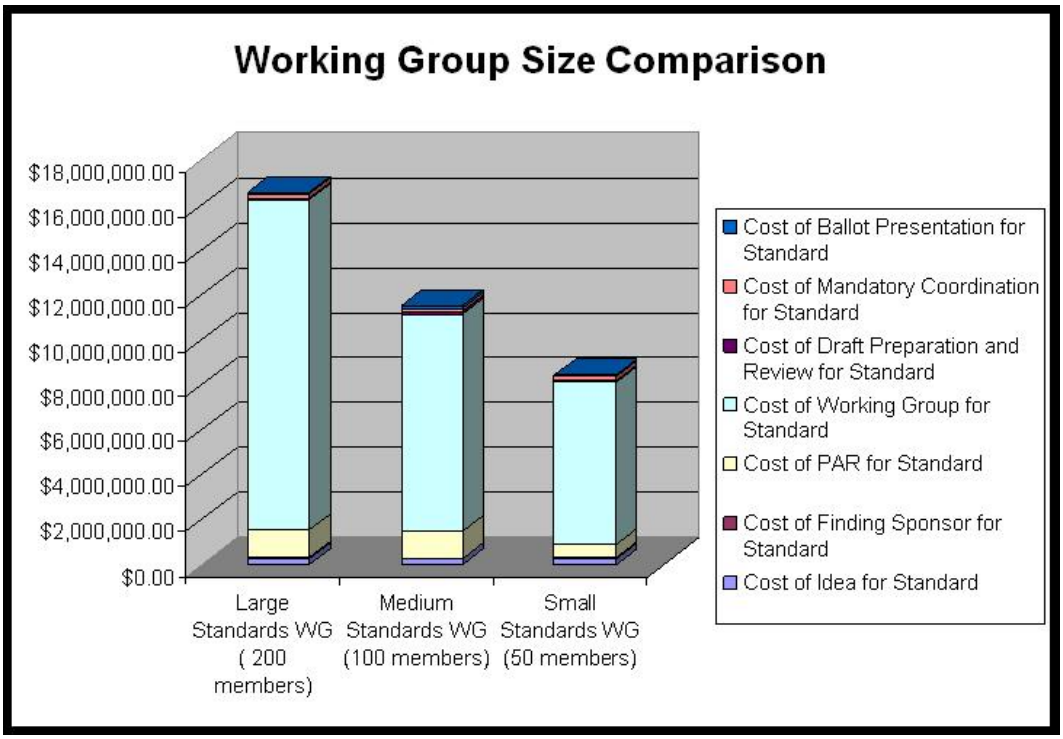
Other Organizations

In addition to the entities mentioned above, several other individuals and organizations are interested in standards-setting in both the M&S and gaming environments. Their interest usually stems from their involvement in designing or developing specific programs that require cooperating with commercial game companies. Institutions with this program-centric interest include the Army's Institute of Creative Technology (ICT), the Army's Research Development and Engineering Command (RDECOM), The Naval Post Graduate School, and the Marine Corps' Program Manager of Training Systems (PMTRASYS).

Speed of Technology vs. Speed of Standards

Despite the many standards that have benefited and improved the efficiency of both the commercial and government sectors, there remains a troubling discrepancy between the speed of technical change and speed of standards approval. In a 2005 review of DoD standards, of the 16 standards examined, the average development time-to-standard was 9.2 years with an average cost of standardization activities (not including first application) of $44.3 million. This is not exclusively a DoD problem—a review of the generalized IEEE standards process revealed an average time-to-standard of approximately four years, with costs ranging from $8–16 million.[68] Significantly, the cost of developing a standard grew in direct proportion to the size of the working group (See Figure 6). This is relevant to DoD because on any given topic, the constituent base is so large that it drives up the size of the working group equivalent, thereby increasing the cost and time-to-standard.

[68] Institute for Defense Analyses. *DMSO Historical Standards Characterization and Costs Status*. Alexandria, VA: IDA, Oct 2005.



Source: Institute for Defense Analyses

Figure 6. Relationship of Working Group Size to Standards Development Cost

Given this DoD timeframe for developing standards, and given the speed of technical change and innovation in the PC marketplace, it may not be practical for DoD to pursue standards development with regard to PC games. By the time a standard can be developed, the hardware and operating systems have moved on.

The same is not true, however, for console games, where hardware platforms such as the Playstation and Xbox have a targeted replacement cycle of seven-to-ten years. Because the platform is likely to be stable for longer than the standards development process, it may be possible to implement and benefit from a standards program.

When NOT to Standardize

The idea of exempting PC game projects from some standardization efforts is not as foreign to DoD as it may seem. The situation is anticipated and accounted for in the DSP which, in section C3.5, specifies "when not to standardize." The policy states "If

the answer is yes to any of these questions, then standardization should not be a primary consideration." [69]

- Is the technology unstable?
- Is it preferable not to freeze design in order to take advantage of technical advances?
- Will standardization unacceptably inhibit design flexibility and innovation?
- Is the primary goal to satisfy the customer preferences?

Certainly in the case of game projects, the answer to a number of these questions will be yes.

This does not mean that all standards should be forsaken. Many of the standards groups mentioned in this report have stated that the community discussions of design and engineering concepts have greatly enhanced applications development. The implication is that DoD should remain active and participate in various volunteer standards bodies and working groups, even though budget constraints make that difficult. In 1994, when Secretary Perry issued his memorandum, there were approximately 2,200 DoD personnel participating in voluntary standards committees. "By 2000 that number had dropped to just over 400 and we are not sure what that number is today."[70] This is of particular concern because many of the technologies and standards groups in the commercial gaming market did not exist at that time, and therefore participation in this larger number of efforts requires even more resources, not fewer.

DoD should actively participate in gaming standards discussions, but it must learn to accept the role of a minority player and to cultivate ways for standards to develop more organically within its constituency. As mentioned above, in the absence of a strong business rationale, no matter how well-intentioned a participant may be, standards can fall by the wayside. Both the commercial and government sectors will surely benefit by being patient and persistent.

---

[69] Defense Standardization Program, "DoD 4120.24-M: DSP Policies & Procedures C3.5," Mar 2000, *www.dsp.dla.mil/documents/4120.24-M/chapter3.htm#C3.5.*

[70] Saunders, Gregory E., Director, DSP Office, DLA, testimony before US House of Representatives Committee of Science, Subcommittee on Technology, Washington, DC, Mar 15, 2000, *www.house.gov/science/saunders_031500.htm.*

VIII. DoD Specific Requirements

Game companies wishing to work with the government—and particularly with the Defense Department—should be aware of the many unique requirements that accompany working on federal projects.

Acquisition and Accounting Regulations

While working with the government can be challenging for many reasons, possibly the greatest is dealing with the large number of bureaucratic requirements. If combing through a 2,000-page regulatory document seems daunting, don't worry—the FARS volumes 1 and 2 are only 1,996 pages! The Federal Acquisition Regulation (FAR)[71] and the Defense Federal Acquisition Regulations Supplement (DFAR)[72] spell out all the rules and regulations that must be followed when doing business with the government. Familiarity with these documents will help avoid pitfalls and costly legal issues down the road.

Intellectual Property Rights

Traditionally, the government has demanded broad and far-reaching rights to intellectual property (IP) developed under the projects it funds. There has been little, if any, thought given to the impact of these demands on the financial bottom line of the companies with which it does business. This, along with formidable accounting and security requirements, has been a large deterrent to doing business with the government.

Recently, however, attitudes have begun to change. Since 1980, industry-funded R&D has significantly exceeded federally-funded efforts (Figure 7), and the government is no longer the leading force in research and development. Thus, while the government still desires to see the R&D it funds become widely disseminated to maximize its impact, it now recognizes the legitimate interest of private companies to retain co-developed IP rights that can help them establish and maintain advantages over their competitors.

---

[71] Federal Acquisition Regulation (FAR), *www.acquisition.gov/far/*.

[72] Defense Federal Acquisition Regulations Supplement (DFAR), *www.acq.osd.mil/dpap/*.

The definition of IP is broad and relates not only to code but also to trade secrets.[73] As individuals in the government have become increasingly sensitive to the complexity and range of IP rights, they have begun to educate others in DoD about ways to "seek flexible and creative solutions to IP issues, focusing on acquiring only those deliverables and license rights necessary to accomplish the acquisition strategy." [74]

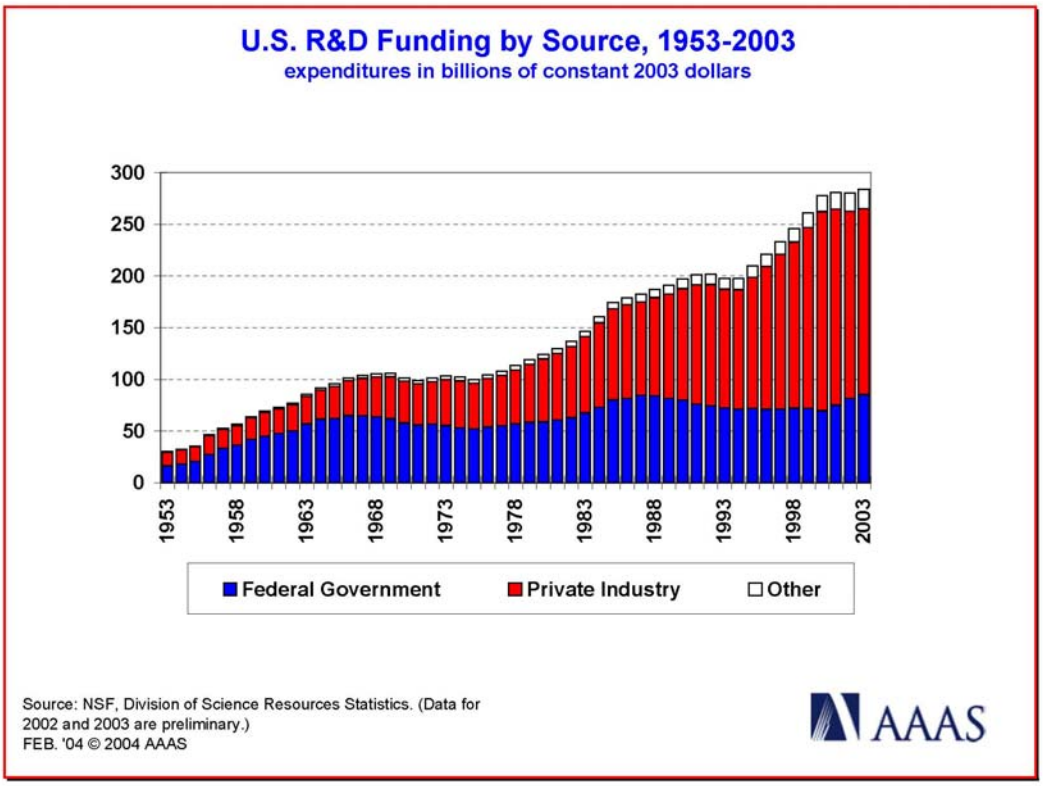


Figure 7. Research and Development Funding by Source, 1953–2003

The IP of interest to game developers is not just the code and technical data that comprise their programs, but also patents, trademarks, copyrights, and trade secrets. The

---

[73] The Economic Espionage Act of 1996 (18 USC 1831-39) defines trade secrets as "all forms and types of financial, business, scientific, technical, economic or engineering information, including patterns, plans, compilations, program devices, formulas, designs, prototypes, methods, techniques, processes, procedures, programs, or codes, whether tangible or intangible, and whether or how stored, compiled, or memorialized physically, electronically, graphically, photographically." *http://rf-web.tamu.edu/security/secguide/S2unclas/Propriet.htm.*

[74] US Department of Defense. *Intellectual Property: Navigating Through Commercial Waters*. Washington, DC: Office of the Under Secretary of Defense for Acquisition, Technology and Logistics, Oct 15, 2001. *www.acq.osd.mil/dpap/Docs/intelprop.pdf*

ownership rights to all of these must be clearly defined in the contract phase, as they determine who will control the use of the product for sequels, expansions, and other commercial activities. One need only consider the success of *Full Spectrum Warrior* and *America's Army* to understand the importance of controlling trademarks and copyrights in an industry that is heavily hit-driven and franchise-oriented. While these two products have been handled very differently, each is representative of a cooperative trademark/copyright approach between government and industry.

With regard to code itself, the government distinguishes between acquiring commercial "off-the-shelf" software (COTS), and non-commercial software. With the former, the government holds itself in the same light as the general public and expects only to receive copies of the software and a license to use it. For non-commercial software (i.e., software developed at least partially at government expense), the government will almost always require broader licensing agreements and will sometimes require access to underlying technical data. For example, if a game company were to develop a game engine to support a DoD-funded game project, the government might expect to acquire rights in that engine. If the parties agree to keep the rights separate, the contract must reflect that, and it becomes very important to segregate the engine work from the game application work, with separate budgets and accounting for each.

Security

Clearances

Government-related work is often classified. Even in game-related projects, the data may be secret, and its release might harm national security. Regulations governing classified information held by contractors can be found in National Industrial Security Program (NISP) Executive Order 12829, as amended.[75]

The most frequently-used levels of classification are "secret" and "top secret," and they can apply to both facilities and to individuals. Some special access programs (SAP) require a Sensitive Compartmented Information (SCI) clearance to work with them. Most work, however, is classified only at the secret level.

---

[75] National Industrial Security Program, *www.archives.gov/isoo/oversight-groups/nisp/*. For defense-specific regulations, see also The National Industrial Security Program at Defense Security Service (DSS), *www.archives.gov/isoo/oversight-groups/nisp/dss.html*.

Being cleared to work in the classified arena can create significant opportunities for an individual, but obtaining a clearance is time-consuming and involves a lot of paperwork. Qualifying for a "secret" clearance is relatively straight-forward, but it requires permitting the government to investigate personal habits and history. Individuals interested in obtaining a clearance would do well to consult the NISP website and to attend government-sponsored sessions that explain the application requirements and procedures.

It is significantly more difficult and expensive to get facility and system classifications. These are still relevant to individuals, however, because for a person to do classified work, either his company must already be certified by the DoD to hold his clearance, or he must find a company that is certified to hold one for him.

Companies with both cleared and non-cleared personnel can maximize their opportunities to work on classified projects by carefully partitioning their workforce according to the security level of the various tasks. For example, in government simulations and game-based applications, it is likely that the data will be classified, while the algorithms and underlying software will not. In this case, most of the team—those who lack security credentials—may be able to work on the unclassified software, while the cleared workforce are the only ones to see the classified data.

Malicious software and aggregated information

Another aspect of security is the danger that malicious code or information might be inserted into DoD software. Incidences such as the infamous *GTA: Vice City* "Hot Coffee" mod, which allowed an in-game sex scene to be unlocked via an Internet hack, has helped shine a spotlight on the problem. This worry is only made worse as more and more companies move development offshore.[76] While DoD is probably less concerned about pornography than about security, the "Hot Coffee" incident is a graphic demonstration of how malicious information can be hidden within a software product without the funding organization's knowledge.

Hidden messages can take many forms. One of this report's authors worked on a game that was found to contain an anti-Semitic statement that appeared in a single frame, and was discovered only after thousands of hours of play-testing. Secret

[76] *GTA Vice City* was developed by a Scottish subsidiary of Take Two Interactive, and the modder who unlocked the hidden scene was Dutch.

information can be encoded into graphics, much as Nazi agents did during the Second World War when they relayed sensitive military information using Morse Code dots and dashes incorporated into fashion drawings.[77]

Inappropriate information is not the only concern. Malicious code can make a program behave improperly, either overtly or in subtle ways that may be difficult to detect. Trojan Horses can infect host computers with viruses that corrupt data and cause malfunctions. Worms can install keystroke loggers that transmit secret information such as passwords.

Companies in the game industry typically have very few formalized procedures in place to detect whether their employees have inserted malicious code or inappropriate material into a game. Master disks are usually verified to be virus-free by running them through commercial security checking packages. As for inappropriate images, most managers simply rely on trust.

Each government entity has its own programs and policies for sanitizing code. The Defense Security Service has an Industrial Security Information Assurance Branch which has a Certification and Accreditation Process and also provides security assistance and training.[78] In addition, the National Security Agency's Information Assurance Directorate offers a suite of tools that project managers can use to engineer secure systems.[79]

Verification, Validation and Accreditation (VV&A)

VV&A is another set of government-specific requirements that game contractors must become familiar with. Verification is demonstrating that you built the right thing, validation is stating that you built the thing right, and accreditation is should it be used, is it sufficient for the sponsor's task at hand.

These concepts are deeply ingrained into government thinking about models and simulations. One could not imagine that the models our government uses to understand nuclear arms aging and decay should not be VV&A'd after all, since live testing is now banned by treaty, all decisions regarding the aging nuclear weapon stock pile is based on

---

[77] Griffiths, Peter. "Britain cracked WW2 spies' secret 'dress code'." London, UK: Reuters, Sept 3, 2006, *http://news.yahoo.com/s/nm/britain_spies_dc.*

[78] More information is at the DSS website, *www.dss.mil/infoas/index.htm.*

[79] US National Security Agency, Information Assurance Directorate. *www.nsa.gov/ia/government/index.cfm.*

models. It seems reasonable that those models should bear rigorous review. In fact, in the DoD world where serious games really do have serious consequences, the idea of VV&A is natural and appropriate. From the incredibly complex models of how our nuclear stockpiles are aging to the comparatively simple simulation of an attack on a jeep convoy, the government subjects most M&S programs to rigorous review through VV&A.

Although the process may seem foreign to game developers, it is quite similar to the steps a game goes through as it progresses through design specification, functional specification, design review, and play testing.

Submitting game projects to VV&A scrutiny is controversial because the process adds cost and time to projects intended to be cheap and fast. One of the reasons DoD is attracted to games is the belief that by leveraging technology investments the commercial sector has already made, the government can get high-quality products quickly and inexpensively. Little work has been done to adapt VV&A to today's fast-paced environment, however, so adding the cost and delay of this process seems contrary to many projects' basic goals. One solution to this problem might be to integrate the VV&A team into the development process at an early stage, just as agile-development entertainment teams strive to include all their "customers" from the start of a project.[80]

A more subtle problem with subjecting games to VV&A is that models and sims traditionally have excluded human behavior as much as possible, whereas the very heart of games is their "Human in the Loop" activities. It is difficult to segregate the human from a game's evaluation or, as one researcher put it, "How do you VV&A the play of the human?" This problem exists not just for game-based products, but for any M&S program where human interaction is integral.[81]

The "A" in VV&A may be the most important of the three elements because the ultimate goal of a model/sim/game is to satisfy its sponsor's needs. Verification and valida-

---

[80] For a more complete discussion of the cost and time issues associated with VV&A, see Kilikauskas, Michelle L. and David H. Hall. "Cost-Effective VV&A: Get Credit For What You're Already Doing" Belcamp, MD: SURVICE Engineering Co., 2006. *www.survice.com/SIPapers/CostEffectiveVV&A.pdf*

[81] For more information see Pace, Dale K. "Modeling and Simulation Verification and Validation Challenges." *Johns Hopkins APL Technical Digest*, vol. 25, no. 2, 2004. *www.jhuapl.edu/techdigest/td2502/Pace.pdf*

tion simply help inform the accrediting agent. "All models are inaccurate, some are useful" as one saying goes, but also "Unaccredited models can produce great insights." [82]

In the end, it is up to the sponsor to decide whether a game is sufficient for the sponsor's purpose. Unfortunately, this has sometimes meant that games (as have models and sims) have been approved based solely on their "way cool factor," and it was only discovered later that that they were inappropriate for their stated purpose.

The best way to ensure that a program receives a fair appraisal is for the developer to be as transparent as possible in explaining and demonstrating the product, and to constantly monitor the sponsor's satisfaction through frequent internal program reviews (IPRs) and other forms of contact. To game developers familiar with agile development, this will be nothing new.

Some of the friction between the VV&A and game design communities is simply due to lack of exposure to each other. As more projects are developed, the groups will undoubtedly develop a common language and establish better ways of working together.[83]

The "Ilities"

Since July 22, 1991, and the creation of the Defense Modeling and Simulation Office (DMSO), DoD has been formally struggling with the "Ilities," those non-functional attributes of reusability, scalability, composibility, reliability, affordability, usability maintainability, manageability, adaptability, survivability, and while not strictly an Ility, quality of service.

The Ilities requirements are set forth in DoD Directive 5000.59 and its resulting instructions and replacements.[84] The debate over their value is not unique to DoD—software designers and their managers have long wrestled with the conflicting desires to make a particular program as small and efficient as possible on the one hand, and to have it be reusable and interoperable on the other.

---

[82] Coyle, Philip E. Modeling and Simulation Conference, presentation, May 1998, *www.dote.osd.mil/presentations/coyle051398/sld038.htm.*

[83] For more information *www.informs-cs.org/wsc00papers/107.PDF*, *www.dtic.mil/whs/directives/corres/pdf/i500061_051303/i500061p.pdf*, *http://vva.dmso.mil/Glossary/Glossary-pr.pdf*

[84] US Department of Defense, 1994.

The problem is exacerbated within DoD because of the participation of multiple stakeholders, many of whom have different priorities, not all of which are apparent or discussed at the outset. This lack of clearly defined business drivers has encouraged DoD's M&S community to pursue generalized optimal solutions, resulting in large, expensive attempts to build such all-inclusive systems as JSIMs and JWARs.

Interestingly, since most DoD programs are compartmentalized, individual program managers have narrowly focused business drivers which often conflict with the Department's overall needs. For example, a particular program may have no business rationale for making its model composible or maintainable, and this puts its business needs in direct opposition to those of the larger community. Optimization at the program level is not optimization at the community level.

This report cannot present an in-depth discussion of the Ilities or offer a resolution to this debate. Suffice to say that game developers should be aware of the Ilities and of the pressures within DoD to accommodate them. The Ilities may not be required in early-stage development or demonstrations, but they are a formal part of past and present DoD M&S master plans, and therefore it would be prudent for developers and program sponsors to decide at an early stage of negotiations how they will be observed.

Data

Games typically lack analytical tools that allow significant data from gameplay sessions to be examined after-the-fact. This functionality would be tremendously valuable to DoD and may one day be formalized as a requirement. Interestingly when one asks game developers "Can you capture any data?" they answer, "We can capture every key stroke if you like." While this is true, it does not get at the heart of the problem. Many games can record the data from a gameplay session and play it back for viewing enabling after action reviews (AAR), but it does not allow analysts to examine and mine the data for additional information, or to aggregate data from multiple gameplay sessions.

To illustrate the problem, suppose a soldier had an in-game training objective to destroy a target with his rifle. Even if examining the raw data stream revealed the target was destroyed, it probably would not be easy to determine whether it was destroyed by a rifle bullet or a hand grenade, by the soldier or one of his squad mates, or even whether it blew up as a result of splash damage from an unrelated action. Currently, the only way an instructor or analyst could determine what happened is to visually examine a real-

time replay, a process which is expensive, time-consuming, and prone to error, and which does not help answer such questions as "what percentage of soldiers who played this game achieved this particular objective," or "how close must the average soldier get to the target before he is able to destroy it?"

One game-based training program that has at least partially achieved this goal is the *Adaptive Thinking and Leadership* application that was jointly developed by Sandia National Laboratories and Virtual Heroes, building on top of the *Unreal* engine and the *America's Army* platform. The application is being used at the JFK Special Warfare Center and School at Ft. Bragg to train Special Forces soldiers in critical skills.[85] The AAR portion of this product includes not only a playback feature for instructor review and analysis of individual sessions, but also the capability to gather data of interest to subject matter experts so they can analyze statistical tracking data across multiple gameplay sessions.[86]

The ability to extract and analyze data from games in a easy and user-friendly way would significantly enhance the credibility and value of these products. Analysts need tools with intuitive graphical user interfaces that will allow them to easily "slice and dice" data, instead of laboriously examining raw data from individual sessions. These tools should be included and budgeted for in the initial product design. We believe DoD will soon recognize the glaring omission of this much-needed functionality, and will one day require it in the projects that it sponsors.

Conclusion

Government specific requirements are generally more autocratic, structured, and bureaucratic than the ones game developers are accustomed to. However, none of the problems discussed in this section are insurmountable, especially if industry producers and government program managers work together to identify and plan for them during contract negotiations, and maintain good communication throughout the course of the project.

---

[85] Wilson, Bret, press release, "Sandia National Laboratories," US Army, America's Army Information Site, Jul 26, 2005, *http://info.americasarmy.com/press.php?id=1*.

[86] VirtualHeroes_answers_AAR_Questions. Unpublished paper from Virtual Heroes, Inc., 2006.

3
WHEN WORLDS COLLIDE

I. Introduction

What happens when the culture and business practices of the commercial game world collide with highly regulated procedures of the government, and especially those of the military? The result is a challenging—but not impossible—landscape that both sides must traverse to find common ground. This section explores current "serious games" efforts, details the research that supports their effectiveness, and outlines road-blocks to the increased use of games by the Department of Defense. It focuses on the business and technical challenges of public/private collaborations and concludes with a glimpse into the future and specific recommendations about how the government and the game industry can work together more effectively.

II. Current Efforts, Research & Roadblocks

The first two sections of this report were written as primers for the Commercial and Government worlds of games and modeling and simulation. In this section, we discuss those points where friction exists between the two worlds, beginning with a snapshot of current defense-related gaming MS&G efforts, and then covering the business and technical challenges of commercial and government working together. By way of recommendations to help smooth the way, we present a roadmap to lead the way to future collaboration and innovation.

Here we describe the landscape of current game-based projects, summarize research on the effectiveness of gaming technologies, present viewpoints and ideas from experts in the field, and list significant obstacles barring the way to beginning similar efforts. Given the rapid pace of development in the field, it will undoubtedly overwhelm this "screenshot." Our attempt here is simply to capture and present what exists now in an effort to create a place for discussion and a means to act.

Current Efforts

Most current DoD-sponsored game-based projects began as research efforts or as speculative projects whose effectiveness was intuited, but not proven. This motivation is changing rapidly as studies are published documenting the effectiveness of gaming technologies and as a new industry appears around the specialty of serious games.

Areas of Interest

Many people think of serious games only in the context of training. While this is the most obvious use, serious games have much broader potential: they are also used in education (as distinct from training), information sharing, health, experimentation, re-

cruitment, and operations. "All the branches [of DoD] now employ a variety of games, in a wide variety of formats, for almost everything the military does." [87]

Education

Education is often confused with training, but the distinction is that training focuses on specific skills, while education addresses general capabilities.

Education-oriented projects include:

- *Adaptive Thinking & Leadership*, Sandia, JFK SWCS, and OEMA[88]
- *SCUDHunt*, ThoughtLink, ARI
- *Critical Leadership Analysis System*, Army Research Center, Institute for Creative Technologies

"The requirement for adaptive thinking—being able to make good decisions on the fly—is very important to Special Forces." —Dr. Elaine Raybourn, Sandia National Laboratories[89]

"Education…is concerned with the development of the mind, of the intellect, while training deals with learning specific skills." —Dr. Samuel Blumenfeld[90]

"Training emphasizes standard procedures and expected performance. Education emphasizes critical thinking without a 'correct' answer or process." —LtCol Gary Morgan, Squadron Office College[91]

"The military and business have noted the potential for simulation and gaming technology to develop higher order thinking skills; in particular, they see potential

---

[87] Maguire, Flack, Michael van Lent, Marc Prensky and Ron W. Tarr. "Defense Combat Sim Olympics — Methodologies Employing the 'Cyber Gaming Culture.'" 2002. *www.marcprensky.com/writing/*

[88] Information on this project can be found at *http://info.americasarmy.com/*.

[89] Burroughs, Chris. "Sandia-developed game helps Special Forces learn adaptive thinking, problem solving." *Sandia Lab News*. Sandia National Laboratories, May 13, 2005. *http://sandia.gov/LabNews/*

[90] Blumenfeld, Samuel L. "Education vs. Training." World Net Daily website, Feb 7, 2000. *www.worldnetdaily.com/news/article.asp?ARTICLE_ID=16292*

[91] Morgan, Gary C. "Developing Air and Space Leaders, Shaping the Air Force Future." Presentation. Maxwell AFB, AL: US Air Force, 2005, *www.jointadlcolab.org/newsandevents/ifests/2005/*.

in such areas as problem solving, metacognition, and decision making." —C. J. Bonk and V. P. Dennen[92]

"Certain skills gained and practiced by gamers in massive multiplayer online gaming environments closely parallel those required by a military transforming itself to operating under the concept of network centric warfare. The technologies and practice methodologies employed in multiplayer games also hold great potential to provide appropriate network centric warfare training environments." —C. J. Bonk and V. P. Dennen[93]

"Serious games are a way to explore the possible effects of human decisions, to try to get a handle on the uncertainties—even more, to try to learn 'what we didn't know we didn't know'." —Dr. Peter Perla, Center for Naval Analyses[94]

Information Sharing

Soldiers use user-modifiable games to capture their experiences in the field and quickly share their new knowledge with comrades and those further up the chain of command.

Projects that focus on information-sharing include:

- *Ramadi Convoy Exercise*, KUMA, Army Combined Arms Support Command
- *DARWARS Ambush!*, BBN, DARPA
- *Land Warrior*, Raytheon. Although not designed as a game itself, the *Land Warrior* project uses game-like technologies that permit soldiers to create experiences and allow others to witness activities in which the soldiers participate.

These logs of personal experiences are called "glogs." In the current environment, soldiers from all ranks and units have begun using text-based blogs to create an incredible base of day-to-day knowledge that provides real-world accounts of adaptation, evolution, and innovation. Glogs take the concept a step further by moving these experiences into a virtual world where any other soldier can step into the glogger's shoes and attempt to understand, solve, experiment with or improve the real-world events;

---

92 Bonk, Curtis J. and Victoria P. Dennen. "Massive Multiplayer Online Gaming: A Research Framework for Military Training and Education." Advanced Distributed Learning DoD, Technical Report-1, Mar 2005. *www.adlnet.gov/downloads/files/186.cfm*

93 Bonk, Curtis J. and Victoria P. Dennen, 2005.

94 Tochen, Dan. "Serious Games in the Services: Army vs. Navy." (Perla quoted), *GameSpot*, Oct 31, 2005. *www.gamespot.com/news/6136900.html*

where instructors and mentors can wander, building lessons on events with immediacy and relevancy to the warfighter.

> "A CyborgLog (often abbreviated to 'glog) is a first-person recording of an activity, in which the person doing the recording is a participant in the activity. Examples of cyborglogs include recordings made by assistive technologies such as a visual memory prosthetic, or a seeing aid that links to remote computational or human elements. Although cyborglogs have a 30-year history dating back to wearable computers in the 1970s, modern technologies like cameraphones make it much easier to create cyborglogs in everyday life" —Wikipedia[95]

> "I wanted…to capture the experience of veterans inside a computer game that any soldier can play." —Dr. Ralph Chatham, DARPA[96]

> "Personalized, effective, engaging, deployable, affordable on-demand training is not only possible, but is being developed and deployed now." —Dr. Ralph Chatham[97]

Health

Games are being developed that deal with all aspects of a soldier's health, from general healthcare, to combat medical training, to post-action mental health. This specific area of interest in the serious games space has generated so much interest that it has spawned its own community and set of conferences.[98]

Representative projects include:

- *Combat Medic,* Legacy Interactive
- *Virtual Iraq,* new therapies to treat Post-Traumatic Stress Disorder, Virtually Better
- *Interactive Trauma Trainer*, battlefield surgeon training, Blitz Games, UK
- *Top Gun*, laproscopic surgery training, Dr. James Rosser
- *Pulse*, virtual health delivery training system, ONR, Breakaway

---

[95] Wikipedia, entry for CyborgLog, updated June 2006, *http://en.wikipedia.org/wiki/CyborgLog.*

[96] Bray, Hiawatha. "War Games 2.0." *The Boston Globe*, (Chatham quoted), Dec 8, 2004. *www.boston.com/business/globe/articles/2004/12/08/war_games_20/*

[97] BBN Technologies, press release "DARWARS Training Superiority Program Participants Demonstrate Systems at I/ITSEC." (Chatham quoted), Dec 6, 2004. *www.bbn.com/News_and_Events/Press_Releases/2004/*

[98] Information about this community and the "Games For Health" conference can be found at *www.gamesforhealth.org.*

"We are assessing the landscape, considering a strategy for the future, and actively managing development of several video game efforts." —Harvey McGee, MM&S & Advanced Medical Technologies, TATRC[99]

"Game technology is transforming military medicine." —CDR Russell Shilling, Office of Naval Research[100]

Experimentation/Acquisition

Wargames have always been beneficial testbeds for new strategies and tactics. Bringing wargames to the computer leverages commercial companies' time and manpower investments and allows the government to focus on its unique needs. As far as DOD is concerned, it reduces the time and manpower needed to test a scenario, and increases the number of scenarios (and iterations of each scenario) that can be tested. And in a networked environment may accommodate a more diverse and richer pool of ideas.

Projects geared towards experimentation include:

- Mosbe™, "lite" simulation toolkit, Breakaway Games

"Games and experiments evaluating prospective new technologies, organizations and concepts are needed to explore features of the emerging fitness landscape involving virtual capabilities and "big Bets" that can not otherwise be evaluated." —John Hanley, Office of Force Transformation[101]

Recruitment

In the absence of a Draft, efficient measures are needed to recruit an all-volunteer military force. Games have proven to be an enormously cost-effective way to communicate the values of the military to prospective candidates.

Recruitment games include:

- *America's Army*
- N*aval Training Exercise: Strike and Retrieve*

---

[99] McGee, Harvey, Telemedicine and Advanced Technology Research Center, email to authors, Oct 11, 2005.

[100] Shilling, Russell, presentation "Game Technology is Transforming Military Medicine: Is the Inverse Also Possible?" Games for Health Conference. May 2006. *www.gamesforhealth.org/archives/000135.html*

[101] Hanley, John. "Rapid Spiral Transformation." *Transformation Trends*. Washington, DC: Office of Force Transformation, DoD, Feb 3, 2003. *www.oft.osd.mil/library/library_files trends_172_Transformation %20Trends--3%20February%20Issue.pdf*

"Survey results find *America's Army* to be the Army's most effective sponsorship effort for reaching young Americans." —Chris Chambers, *America's Army*[102]

"19 percent of this year's freshman class (at the military academy at West Point) said they had played *America's Army: Operations*." —Col Casey Wardynski, *America's Army*[103]

"We've seen results from a tracking study that says people's image of the Navy has shifted a little and we like to draw a correlation between that change and (NTE: Strike and Retrieve)." —Joe Gaulzetti, Campbell-Ewald[104]

Training

Training represents the most common use of games in the military. Whether they are adaptations of commercial products or built-to-order, games are being used as effective training tools in all branches of the armed forces.

Training games include:

- *Convoy Trainer,* Army game project, Zombie Studios
- *Bottom Gun*, Naval Air Warfare Center Training Systems Division
- *DARWARS Tactical Language Trainer,* USC Center for Advanced Research in Technology for Education, DARPA
- *Forward Observer*, BMH Associates, ONR
- *Mission Rehearsal Exercise*, ITC
- *Quick Strike*, Mak Technologies, USAF

"For training games, the primary goal is to introduce and train a particular domain, or to practice a skill that needs further refinement." —Dr. James Belanich, US Army Research Institute for the Behavioral and Social Sciences[105]

"Games are getting remarkably sophisticated. The simulations and graphics are incredible; they feature a lot of artificial intelligence; and you can attack them from

---

[102] Callaham, John. "America's Army Interview." (Interview with Chris Chambers.) Firing Squad website, Jun 21, 2006, *www.firingsquad.com/games/americas_army_interview/*.

[103] Gwinn, Eric. "Army Targets Youth With Video Game." (Col Casey Wardynski quoted.) *Chicago Tribune*. Nov 7, 2003. *www.notinourname.net/resources_links/video-game-7nov03.htm*

[104] Beerman, S. et al. "From Campus to Combat, Military Spurs Controversy." (Gaulzetti quoted.) Medill, 2006. *http://observer.medill.northwestern.edu/301-wi06-sec04/04recruitment_challenges/01main_story/*

[105] Tochen, Dan. "Serious Games in the Services: Army vs. Navy." (Belanich quoted.) *GameSpot*, Oct 31, 2005. *www.gamespot.com/news/6136900.html*

many different angles. In short, they do all of the things that the learning scientists told us worked well." —Kay Howell, Federation of American Scientists

Operations

Games are increasingly being used in Rapid Mission Rehearsals, which "will exploit technical innovation and integration to provide any U.S. soldier with the ability to open a laptop computer and rehearse a specific mission in the relevant geo-specific terrain." [106]

Such games include:

- *DARWARS Ambush!*, Total Immersion Software, DARPA
- *RealWorld A-10 Desktop Trainer*, Total Immersion Software, DARPA

"The DTT project is an exciting undertaking where we are able to create a synergy between the technical abilities and skills from the entertainment gaming business and the creative and innovative training approach that the Air National Guard has always provided to the U.S. Air Force... This A-10C DTT project is one of the first applications that we will be integrating into our RealWorld rapid mission rehearsal project." —Carl Norman, A-10C DTT[107]

Representative Sample of Companies in the Field

- Anark
- Anteon
- BBN
- BHM
- Big Fun
- Blitz
- Bohemia
- Breakaway
- DigitalMill
- Dynamic Animations Systems
- Forterra
- Games2Train
- Immersive Education Ltd.
- Kuma
- Legacy
- Mak
- Mind Miners LLC
- Persuasive
- Symmetron
- ThoughtLink
- Total Immersion
- Ultramundum
- Valador
- Virtual Heroes
- Will Interactive
- Wireframe Interactive
- Zombie

---

[106] Defense Sciences Office, Rapid Mission Rehearsal description, Daniel Kaufman, program manager. *www.darpa.mil/DSO/thrust/biosci/rmr.htm*

[107] ETS News, press release "Low Cost Training Simulation will Provide ANG Pilots with a Portable and Flexible Solution." May 15, 2006. *www.ets-news.com/third.php?id=735*

The definition of IP is broad and relates not only to code but also to trade secrets.[73] As individuals in the government have become increasingly sensitive to the complexity and range of IP rights, they have begun to educate others in DoD about ways to "seek flexible and creative solutions to IP issues, focusing on acquiring only those deliverables and license rights necessary to accomplish the acquisition strategy." [74]

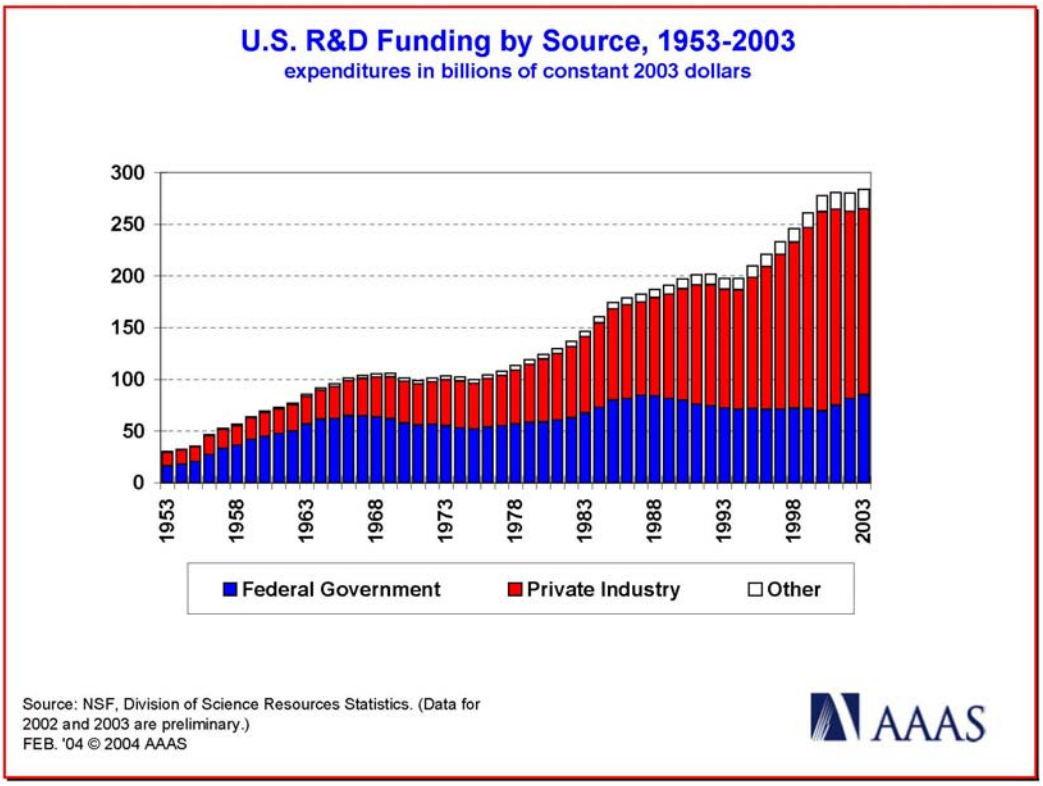


Figure 7. Research and Development Funding by Source, 1953–2003

The IP of interest to game developers is not just the code and technical data that comprise their programs, but also patents, trademarks, copyrights, and trade secrets. The

---

[73] The Economic Espionage Act of 1996 (18 USC 1831-39) defines trade secrets as "all forms and types of financial, business, scientific, technical, economic or engineering information, including patterns, plans, compilations, program devices, formulas, designs, prototypes, methods, techniques, processes, procedures, programs, or codes, whether tangible or intangible, and whether or how stored, compiled, or memorialized physically, electronically, graphically, photographically." *http://rf-web.tamu.edu/security/secguide/S2unclas/Propriet.htm.*

[74] US Department of Defense. *Intellectual Property: Navigating Through Commercial Waters*. Washington, DC: Office of the Under Secretary of Defense for Acquisition, Technology and Logistics, Oct 15, 2001. *www.acq.osd.mil/dpap/Docs/intelprop.pdf*

many different angles. In short, they do all of the things that the learning scientists told us worked well." —Kay Howell, Federation of American Scientists

Operations

Games are increasingly being used in Rapid Mission Rehearsals, which "will exploit technical innovation and integration to provide any U.S. soldier with the ability to open a laptop computer and rehearse a specific mission in the relevant geo-specific terrain." [106]

Such games include:

- *DARWARS Ambush!*, Total Immersion Software, DARPA
- *RealWorld A-10 Desktop Trainer*, Total Immersion Software, DARPA

"The DTT project is an exciting undertaking where we are able to create a synergy between the technical abilities and skills from the entertainment gaming business and the creative and innovative training approach that the Air National Guard has always provided to the U.S. Air Force... This A-10C DTT project is one of the first applications that we will be integrating into our RealWorld rapid mission rehearsal project." —Carl Norman, A-10C DTT[107]

Representative Sample of Companies in the Field

- Anark
- Anteon
- BBN
- BHM
- Big Fun
- Blitz
- Bohemia
- Breakaway
- DigitalMill
- Dynamic Animations Systems
- Forterra
- Games2Train
- Immersive Education Ltd.
- Kuma
- Legacy
- Mak
- Mind Miners LLC
- Persuasive
- Symmetron
- ThoughtLink
- Total Immersion
- Ultramundum
- Valador
- Virtual Heroes
- Will Interactive
- Wireframe Interactive
- Zombie

---

[106] Defense Sciences Office, Rapid Mission Rehearsal description, Daniel Kaufman, program manager. *www.darpa.mil/DSO/thrust/biosci/rmr.htm*

[107] ETS News, press release "Low Cost Training Simulation will Provide ANG Pilots with a Portable and Flexible Solution." May 15, 2006. *www.ets-news.com/third.php?id=735*

Table 6. Current DoD-Related Projects

Branch	Project Titles
Air Force	Air Force Delta Storm・Avant Guard・Falcon 4.0・JVID and Finflash・Project X・Quick Strike—Time Sensitive Targeting Trainer・Starcraft
Army	America's Army—Operations・Army Research Lab Trainer for X-Box・Asymmetrical Warfare Virtual Training Technology・Battle Command 2010 (BC2010)・Battlefield 1942・Civil Support Team Trainer・Critical Leadership Analysis System (CLAS)・DARWARS Ambush! Lessons Learning Game・DARWARS Tactical Language Trainer・Decisive Action・Delta Force V / Land Warrior・Digital Warrior・Full Spectrum Command・Full Spectrum Leader・Full Spectrum Warrior・Gator Six, Battery Command Virtual Experience・Guard Force・M1 Tank Platoon・Mission Rehearsal Exercise (MRE)・Operation Flashpoint・Saving Sergeant Pabletti・Spearhead II・Steel Beasts・TAC-OPS
Joint Forces	Anti-Terrorism Force Protection・DARWARS—The DARPA Training Superiority Program・Entropy-Based Warfare・Joint Force Employment・Peloponnesian War・Stability Operations: Winning the Peace・Warlords
Marine Corps	Close Combat Marines・Infantry Tool Kit・Marine Air-Ground Task Force—MAGTF XXI・Marine Doom (no longer used)・Medal of Honor・Soldier of Fortune・Virtual Battlefield Simulation 1 (VBS1)
Navy	Battle Stations 21・Bottom Gun・Electro-Adventure・Forward Observer・Harpoon2・Jane's Fleet Command・Leadership Training - Center for Naval Leadership・Microsoft Flight Simulator・Navy Anti-terror Simulation Game・SOCOM: US Navy Seals・Sub Command
Other	Other Military Games・Comanche 4・Command & Conquer—Generals・Counter-Strike・Cyberwar XXI・Ghost Recon・Raven Shield (Rainbow Six 3)・Return to Castle Wolfenstein・Rogue Spear・Team Apache

Source: www.dodgamecommunity.com

Table 7. Other Games Relevant to the Military

Game Title	What it's used for	Who uses it	Who makes it
Aces High	flight training	Air Force Academy	Hi Tech
America's Army	spin-offs created for specific scenarios	Army	US Army
Embedded trainer	Bradley, Javelin, CROWS		
TOW	anti-tank missile trainer		
CyberSeige	computer security	Navy	Rivermind
Incident Commander	disaster management	Homeland Security	Breakaway
Interactive Trauma Trainer	battlefield surgeon trainer	Human Factors Integration Defense Technology Center	Blitz Games
91W-TC3	care under fire	ECS/RDECOM/AMEDD Center & School	US Army
Pulse!!	virtual healthcare training	ONR	Texas A&M
SCUDhunt	enhance shared situation awareness	DARPA	ThoughtLink
Swarmada	unmanned aerial vehicle trainer	Navy	Big Fun
Tactical Iraqi	spinoff from Tactical Language Trainer	DARPA	Adelo
24 Blue	flight deck operations simulator	Navy	Breakaway
Virtual Iraq	PTSD treatment	ONR	Virtually Better

"Persuasion" Games

Games are increasingly being used as tools of persuasion. Organizations as diverse as the United Nations (*FoodForce*), Hezbollah (*Special Force*), The Canadian Department of Foreign Affairs (*Pax Warrior*), and MTV (*Darfur is dying*), are funding games intended to win hearts and minds and to drive people to action.

"Persuasion" games include:[108]

- *Darfur is dying*, MTV
- *Special Force*, Hezbollah
- *Food Force*, United Nations
- *September 12* and *Madrid*, Newsgaming.com
- *Peace Maker*, Multicultural student group that became ImpactGames

[108] For other "persuasion" games, see *www.gamesforchange.org/conference/2006/expo.asp*, which lists the games demonstrated at the 2006 "Games for Change" conference.

- *A Force More Powerful*, International Center on Nonviolent Conflict
- *Take Back Illinois*, US Republican Party
- Untitled game about gerrymandering, USC Annenberg School for Communications
- *Under Ash*, AkfarMedia, Syria
- *Left Behind*, Left Behind Games
- *Pax Warrior*, Canadian firm 23 YYZee, partially funded by Canadian Department of Foreign Affairs' Peacebuilding Fund
- *Four Years in Haiti*, Microsoft
- *Real Lives*, Educational Simulations
- *Generation Rx*, Kentucky River Community Care

"The first part of activism is getting something under your skin, and having a personal identification with it, and the immediacy of playing a [video] game can often do that." —Stephen Friedman, mtvU[109]

"Games give you the opportunity to live a culture and I think that is dramatically more powerful and persuasive than a million leaflets or 60,000 Peace Corps volunteers." —Edward Castronova, Indiana University[110]

Research About the Effectiveness of Game Technologies

For years, the adoption of serious games was hampered by a lack of empirical evidence that they are effective tools. That is rapidly changing as the field has gained in popularity, and hundreds—if not thousands—of gaming technology research projects are now underway.

"Given emerging research on how video games and associated pedagogies work in designed settings (Shaffer 2005), it seems the important question is not whether educators *can* use games to support learning, but *how* we can use games most effectively as educational tools. The explosion of research initiatives, conferences, books, and software focused on educational games suggests that computer and

---

[109] Biyd, Clark. "Darfur Activism Meets Video Gaming." (Friedman quoted.) *BBC News,* Jul 6, 2006. *http://news.bbc.co.uk/2/hi/technology/5153694.stm?ls*

[110] Musgrove, Mike. "Video Game World Gives Peace a Chance." (Castronova quoted.) *The Washington Post*, Oct 16, 2005. *www.washingtonpost.com/wp-dyn/content/article/2005/10/15/AR2005101500218.html?referrer=email*

Table 7. Other Games Relevant to the Military

Game Title	What it's used for	Who uses it	Who makes it
Aces High	flight training	Air Force Academy	Hi Tech
America's Army	spin-offs created for specific scenarios	Army	US Army
Embedded trainer	Bradley, Javelin, CROWS		
TOW	anti-tank missile trainer		
CyberSeige	computer security	Navy	Rivermind
Incident Commander	disaster management	Homeland Security	Breakaway
Interactive Trauma Trainer	battlefield surgeon trainer	Human Factors Integration Defense Technology Center	Blitz Games
91W-TC3	care under fire	ECS/RDECOM/AMEDD Center & School	US Army
Pulse!!	virtual healthcare training	ONR	Texas A&M
SCUDhunt	enhance shared situation awareness	DARPA	ThoughtLink
Swarmada	unmanned aerial vehicle trainer	Navy	Big Fun
Tactical Iraqi	spinoff from Tactical Language Trainer	DARPA	Adelo
24 Blue	flight deck operations simulator	Navy	Breakaway
Virtual Iraq	PTSD treatment	ONR	Virtually Better

"Persuasion" Games

Games are increasingly being used as tools of persuasion. Organizations as diverse as the United Nations (*FoodForce*), Hezbollah (*Special Force*), The Canadian Department of Foreign Affairs (*Pax Warrior*), and MTV (*Darfur is dying*), are funding games intended to win hearts and minds and to drive people to action.

"Persuasion" games include:[108]

- *Darfur is dying*, MTV
- *Special Force*, Hezbollah
- *Food Force*, United Nations
- *September 12* and *Madrid*, Newsgaming.com
- *Peace Maker*, Multicultural student group that became ImpactGames

[108] For other "persuasion" games, see *www.gamesforchange.org/conference/2006/expo.asp*, which lists the games demonstrated at the 2006 "Games for Change" conference.

- *A Force More Powerful*, International Center on Nonviolent Conflict
- *Take Back Illinois*, US Republican Party
- Untitled game about gerrymandering, USC Annenberg School for Communications
- *Under Ash*, AkfarMedia, Syria
- *Left Behind*, Left Behind Games
- *Pax Warrior*, Canadian firm 23 YYZee, partially funded by Canadian Department of Foreign Affairs' Peacebuilding Fund
- *Four Years in Haiti*, Microsoft
- *Real Lives*, Educational Simulations
- *Generation Rx*, Kentucky River Community Care

"The first part of activism is getting something under your skin, and having a personal identification with it, and the immediacy of playing a [video] game can often do that." —Stephen Friedman, mtvU[109]

"Games give you the opportunity to live a culture and I think that is dramatically more powerful and persuasive than a million leaflets or 60,000 Peace Corps volunteers." —Edward Castronova, Indiana University[110]

Research About the Effectiveness of Game Technologies

For years, the adoption of serious games was hampered by a lack of empirical evidence that they are effective tools. That is rapidly changing as the field has gained in popularity, and hundreds—if not thousands—of gaming technology research projects are now underway.

"Given emerging research on how video games and associated pedagogies work in designed settings (Shaffer 2005), it seems the important question is not whether educators *can* use games to support learning, but *how* we can use games most effectively as educational tools. The explosion of research initiatives, conferences, books, and software focused on educational games suggests that computer and

---

[109] Biyd, Clark. "Darfur Activism Meets Video Gaming." (Friedman quoted.) *BBC News,* Jul 6, 2006. *http://news.bbc.co.uk/2/hi/technology/5153694.stm?ls*

[110] Musgrove, Mike. "Video Game World Gives Peace a Chance." (Castronova quoted.) *The Washington Post*, Oct 16, 2005. *www.washingtonpost.com/wp-dyn/content/article/2005/10/15/AR2005101500218.html?referrer=email*

video games will have some part in education, just as all media before them have been used for learning." —Kurt Squire, University of Wisconsin-Madison[111]

"Games...can teach higher order thinking skills such as strategic thinking, interpretive analysis, problem solving, plan formulation and execution, and adaptation to rapid change." —Henry Kelly, Federation of American Scientists[112]

"A good instructional game…would pick its domain of authentic professionalism well, intelligently select the skills and knowledge to be distributed, build in a related value system as integral to gameplay, and clearly relate any explicit instructions to specific contexts and situations." —James Paul Gee, University of Wisconsin-Madison[113]

"Additional explanation of why games are effective tools comes from Dr. Robert Ahlers and Rosemary Garris of the Navy's NAWCTSD Submarine Lab. (Ahlers and Garris, in Press) They found that games create a self-perpetuating virtuous cycle in users, as players initiate and control game play, practice skills, solve problems, persist to the end and strive to win (which often translates as 'learn'), a process which then leads to re-initiation." —Maguire et al[114]

"Both Whitehall and McDonald (1993) and Ricci, Salas, and Cannon-Bowers (1996) found that instruction incorporating game features led to improved learning. In addition, Ricci, et. al (1996) proposed that instruction that incorporated game features enhanced student motivation, which led to greater attention to training content and greater retention." —R. Garris and R. Ahlers, Naval Air Warfare Center Training Systems Division[115]

"This study demonstrates that well-designed action-adventure video games can significantly improve learning, skill development, and behavior change. Video games can be highly appealing and motivating learning environments in which

---

[111] Squire, Kurt. "Changing the Game: What Happens When Video Games Enter the Classroom?" *Innovate*, Aug/Sept 2005, vol. 1, no. 6. *www.innovateonline.info/index.php?view=article&id=82*

[112] Kelly, Henry. Invitation to Summit on Educational Games. Federation of American Scientists, Oct 2005. *www.fas.org/gamesummit/reading/invite.pdf*

[113] Gee, James Paul. "What Would a State of the Art Instructional Video Game Look Like?" *Innovate*, Aug/Sept 2005, vol. 1, no. 6. *www.innovateonline.info/index.php?view=article&id=80*

[114] Maguire, Flack, Michael van Lent, Marc Prensky and Ron W. Tarr. "Defense Combat Sim Olympics—Methodologies Employing the 'Cyber Gaming Culture,'" 2002. *www.marcprensky.com/writing/*

[115] Garris, R. and R. Ahlers. "A Game-Based Training Model Development, Application, and Evaluation." I/ITSEC, 2001.

players have unlimited opportunities to rehearse skills, receive immediate feedback on performance, obtain and make use of information, experience social support when interacting with other players, and develop the confidence and ability to carry out new skills in their daily lives." —Brown, Lieberman, Gemeny, Fan, Wilson, and Pasta[116]

Playing action games may improve visual perception skills useful in preparing soldiers for combat. "Players can process visual information more quickly and can track 30 percent more objects than nonplayers. Several game players even achieved perfect scores on tests barely doable for nongamers." —C.S. Green and D. Bavelier, Dept. of Brain and Cognitive Sciences, University of Rochester[117]

"In one of the largest and most rigorous experimental studies of gaming effectiveness, Dr. Baranowski and colleagues showed that game-based activities could stimulate school-aged children to increase their fruit and vegetable intake by one serving a day, an increase of about 30%." —From Challenges and Opportunities in Game-Based Learning conference[118]

"Success in a MMOG requires developing new literacies, understanding intricate and intersecting rule sets, thinking creatively within constraints, collaborating with other participants towards shared goals, and perhaps most importantly, taking on new identities as players (via their avatars) inhabit game spaces (Gee 2003). Such properties offer significant potential for educational contexts, as indicated by the emergence of MMOGs specifically designed to enable student interactions and centered on instructional topics." —M. Young, P. G. Schrader, D. Zheng[119]

---

[116] Brown, S. J. et al. "Educational video game for juvenile diabetes: Results of a controlled trial." *Medical Informatics*, 1997, vol. 22, no. 1, pp. 77–89. *www.comm.ucsb.edu/faculty/lieberman/*

[117] Green, C. Shawn and Daphne Bavelier. "Action Video Game Modifies Visual Selective Attention." *Nature*, vol. 423, May 29, 2003. *www.nature.com/nature/journal/v423/n6939/pdf/nature01647.pdf*

[118] Baranowski, Tom. Speaker on the topic of Learning and Behavioral Outcomes for Games: Research Studies at the conference "Challenges and Opportunities in Game-Based Learning." The National Academies, Nov 2, 2005. *www.nae.edu/nae/caseecomnew.nsf/weblinks/NFOY-6HXRHX?OpenDocument*

[119] Young, M., P. G. Schrader, and D. Zheng. "MMOGs as Learning Environments: An Ecological Journey into Quest Atlantis and The Sims Online." *Innovate,* Apr/May 2006, vol. 2, no. 4. *www.innovateonline.info/index.php?view=article&id=66*

"If you are to make good edutainment games you must turn to commercial game companies." —Simon Egenfeldt-Nielsen, Game-Research[120]

In addition to the above, a survey of the research conducted by the Northwest Educational Technology Consortium referenced the following findings:[121]

- Simulation environments and modeling have unique capabilities for enhancing learning. (Gordin and Pea, 1995)
- Gaming teaches competition strategies, cooperation and teamwork, and conflict resolution. (Neubecker, 2003)
- The effectiveness of gaming relies on the degree to which the games simulate real life. (Hood, 1997)
- When students are able to represent and explore new information in science classrooms using modeling tools, they are able to explore and deepen their understanding, as well as share it with others. This helps them understand the phenomena they are investigating. (V. Michalchik, A. Rosenquist, R. Kozma, P. Kreikemeier, P. Schank, and B. Coppola, *in press*)
- Games are dynamic, intrinsically motivating, and involve high levels of involvement. They provide immediate feedback to participants, and mistakes do not result in actually losing assets. (Hood, 1997)
- Games have been found to serve a range of functions in education including tutoring, exploring and practicing skills, and attitude change. (Dempsey et al., 1994)
- Simulations can provide students engaging experiences towards learning crisis-management, communication and problem solving, data management, and collaboration. (Gredler, 1994)
- The effective use of games differs depending on the educational areas where the games are employed. The best results were found to be in the areas of mathematics, physics, and language arts (as opposed to social studies, biology and logic). The beneficial effects of gaming are most likely to be found when specific content is targeted and objectives precisely defined. (Randel et al, 1992)

However, it is important to point out that games are not a panacea, or a "one-size-fits-all" tool that can be used across the board in education and training.

"If we continue to preach only that games can be effective, we run the risk of creating the impression that *all* games are good for *all* learners and for *all* learning

---

[120] Egenfeldt-Nielsen, Simon. "Thoughts on Learning in Games and Designing Educational Computer Games." Game-Research.com, Jul 2003. *www.game-research.com/art_educational_games.asp*

[121] Northwest Regional Educational Laboratory. "Focus on Effectiveness: Research-Based Strategies." Northwest Educational Training Consortium, 2005. *www.netc.org/focus/strategies/simu.php*

outcomes, which is categorically *not* the case. What is needed now is (1) research explaining *why* Digital Game Based Learning is engaging and effective, and (2) practical guidance for *how* (when, with whom, and under what conditions) games can be integrated into the learning process to maximize their learning potential" —Richard van Eck, University of North Dakota[122]

"If a game fails to teach, it may not be because it is a game, but because it is a *poorly designed* game." —Ben Sawyer, DigitalMill[123]

Roadblocks & Recommendations

Several roadblocks stand in the way of widespread government adoption of gaming technologies.

Despite the growing body of research that documents the effectiveness of games as educational and training devices, there has not been an official statement from an authority within DoD that games should have a place alongside more traditional methods. The very word "game" has frivolous connotations, and career officers may perceive advocating games as risky to their advancement, especially when this light-hearted word is juxtaposed to the life-and-death consequences often associated with the development of effective programs. DoD should fund its own research into effectiveness and issue definitive guidelines for the use of gaming technologies.

Game companies typically don't plan a "proof of effectiveness" phase into their development schedules. In the commercial world, effectiveness is measured by sales, and companies routinely plan for sanity checks during the development process to ensure that the game will meet its commercial goals. With a serious game, where effectiveness is defined as reaching the goal the game was funded to meet, there is frequently no formal evaluation phase during production that tests whether the game will be effective.

The cross-section of talent and expertise needed to create a successful serious game makes predicting success difficult. Games need to be fun and challenging, and serious games need to be effective. This requires a team of experienced game designers,

---

[122] van Eck, Richard. "Digital Game-Based Learning: It's Not Just the Digital Natives Who Are Restless." *EDUCAUSE Review*, Mar/Apr 2006, vol. 41, no. 2, pp. 16–30. *www.educause.edu/apps/er/erm06/erm0620.asp?bhcp=1*

[123] Sawyer, Ben, private communication with the authors, 2006.

subject matter experts, training/educational personnel, and qualified technical and artistic talent. The intersection of these capabilities does not often occur by chance. In order for serious games to be successful, workers in all these fields must be motivated to come together to work on these projects.

Government standards requirements also remain a roadblock to rapidly and effectively implementing serious games. The history and efficacy of standards are dealt with in the DoD Standards and M&S section of this report, but it is appropriate to note here that whenever the government imposes interoperability requirements on a game, the development cycle for the project will be longer, more expensive, and less reactive to changing world conditions.

III. Business Challenges

The fundamental business challenge facing game developers and the government is the misalignment of work environments and supporting infrastructures by which each can meet the other's needs. As described in the Commercial and Government sections, these environments are extremely different. This lack of a common "way of doing business" affects almost every phase of a business relationship, including how organizations find each other, contracts bidding, long-term funding commitments, project development, payment cycles, complying with regulations, accountability, and acceptance criteria.

Finding Each Other

How does a government organization find and qualify a game company capable of developing a particular project? How does a game company find government organizations with projects it wants developed?

Government officials who want to work with game companies frequently don't know how to contact them or how to gauge the quality of their work. There is no independent rating system for software developers. There are game listing websites such as www.dodgamecommunity.com, but they are not well-maintained, and they have fallen short of providing the same type of marketplace and community that their commercial counterparts such as Gamespy.com and IGN.com have achieved, due in large part to lacking a sustainable business model. A marketplace needs a way to bring a buyer and seller together, establish value, and enable the exchange of goods and services. In the DoD game arena, there is no ebay or Gamespy-type central marketplace for services; hence, government agencies must rely on a patchwork of sources to find developers.

As the serious game industry grows, we can expect some of those conditions to change. While it is unlikely there will ever be a ratings system for developers, it is quite likely that forums will be created where companies interested in working on serious games can announce their interest and capabilities.

Therefore, government program managers need to manage game projects and game companies differently. Developers require different kinds of oversight than the government uses to manage DoD-focused contractors. Moreover, the management skill set to run game projects isn't generally available in DoD. (See the Project Development in the DoD section.)

In some cases, past experience has made the government and its contractors wary of each other, which has led to an almost adversarial relationship between them and is in part responsible for the burdensome government acquisition process. Unfortunately, this tension is not conducive to the kind of collaboration and cooperation that game developers rely on to develop good products.

Anecdotally, during our teleconference interview sessions, both government and developers alike stated that the mid-level officers "get it," and that game designers also "get it," but that the administrators (on each side) who stand between them create problems because each side recognizes that a project's requirements are likely to change while it is being built, and neither side wants to be unduly at risk for absorbing the inevitable cost increases that will follow. While the concept of risk aversion and over-rides is not unique to the government, the commercial sector has not developed contracting mechanisms to accommodate these issues.

The standard government cost-plus contracting process encourages feature-creep, because contractors rarely have an incentive to say "no" to a request. If the client asks for a change, the contractor generally profits by making the change, even if it is detrimental to the project, so the contractor rarely speaks out against making changes. Game company administrators, on the other hand, vigorously fight feature creep because it extends development cycles and eats into profits. This cultural difference will have to be recognized and accounted for at the time the contract is being negotiated, simply because the organizations are likely to enter the relationship with vastly different frames of reference for dealing with requests for changes. Establishing how change management will be implemented and paid for is a vital step in negotiating the contract.

Intellectual Property

Some game companies are nervous about working with the government because they fear their source code—their "family jewels"—may become known to their competitors in the commercial world. While this is an extremely remote possibility, worries persist.

The government would do well to publicize that it goes to great lengths to honor its confidentiality agreements, and recognizes that it has a huge liability if it does not.

On the government side, some project managers believe it is essential for the government to acquire all IP rights to a project in order to avoid being held ransom by the IP owner. The reality is that with a little foresight and planning, the IP desires of both partners can almost always be satisfied.

The best approach to IP issues is for both sides to have a firm understanding of what is important to them, and to recognize that a project's components are not a single indivisible property, but a collection of definable assets, each of which can be separately negotiated for.

Recommendations

> There is nothing more difficult to take in hand, more perilous to conduct, or more uncertain in its success, than to take the lead in the introduction of a new order of things. Because the innovator has for enemies all those who have done well under the old conditions, and lukewarm defenders in those who may do well under the new.
>
> —Niccolo Machiavelli

Some of the business challenges we have outlined could be addressed by creating a government entity (or more preferably a government-chartered non-profit separate entity) to act as an ombudsman, clearinghouse, and facilitator between government and industry; but the solution that addresses *all* these issues is to take a step further and empower that entity to create a true marketplace for serious games.

The limited version of this entity is conventional in concept and is a band-aid of sorts whose mission is simply to improve the "interface" between DoD and the game industry. This approach has found success in the likes of small business bureaus, trade organizations, arbitrators, and expediters.

This organization could help:

- Game companies learn about government contracting vehicles.
- Developers learn of game-related projects within government.
- Government project managers learn which contracting vehicles may be appropriate to apply to game-related projects.
- Government project managers find developers with experience in specific areas of expertise.

- Government project managers understand what genres of games may be appropriate to achieve their individual goals.
- Government Project Managers learn how to effectively structure working relationships with game companies engaged in agile development.
- Each side understand the legal implications of working with the other, including ways of protecting intellectual property.

But, as this portion of the report dramatically highlights, there is no organized marketplace in which buyers and sellers with problems and solutions can find each other, establish value, and conduct a transaction. And it is this lack of a marketplace which is critical.

During a Senate Armed Services Committee hearing on January 11, 2001, Defense Secretary Rumsfeld said, "…simply tinkering with the present acquisition system will not provide the innovation and speed necessary to satisfy future military needs." His subsequent statements frame the problem not as one associated with resource capital constraints, but with the fundamental problem of establishing a marketplace in which the real issue is speed-to-market with the right product for the correct customer.

To understand the value of creating such a marketplace, one need only think of the accelerated dynamics that marketplace creators such as ebay and Gamespy have enabled by expanding upon the basic clearinghouse concept with their user community ratings, easy distribution, and facilitated financial transaction mechanisms.

> The ultimate good desired is better reached by free trade in ideas [and] the best test of truth is the power of the thought to get itself accepted in the competition of the market.
>
> —Oliver Wendell Holmes, Jr.

An active and viable marketplace will translate to more suppliers, better products, and a better match between those who develop projects and those who need them. These additional capabilities of the proposed organization might include:

- Clearinghouse of information
- Sample downloads
- User and supplier reviews
- Peer reviews
- Screen shots
- Updates
- Communities of interest
- Business forums
- Legal forums
- Project announcements area
- Transaction payments

Some have suggested that such an organization might take on additional tasks such as funding activities to help smooth the problems of year-on-year availability of government money, and helping to set standards that address evaluation capabilities and tools for data analysis.

This is the key message and most far-reaching recommendation of this report: If you want to change things, then change them.

IV. Technology

This section of the report deals with the technology challenges that arise when the government and the game industry collaborate to create serious game products for the military.

Rapidly Advancing Commercial Hardware

One of the reasons DoD is interested in gaming technologies is the desire to deliver material to modern soldiers in a format they are already familiar and comfortable with: games. But a large part of the appeal of these soldiers' favorite commercial products depends on using hardware more advanced than is widely available in the military.

As we have mentioned previously in this report, the state-of-the-art in console and personal computer technology is being driven more by consumer demands than by military requirements. The commercial sector is pushing the technology along at a dizzying pace. This past year alone has seen drastic increases in processing speed and capability, graphics capability, physics processing, artificial intelligence, and networking—all driven by demand within the commercial interactive media marketplace.

A desktop point of reference: The authors recently acquired a new desktop gaming computer. It contains the latest CORE 2 Duo Extreme 2.93 GHz CPU, 4GB DDR2 low latency memory, *two* NVidia GeForce graphics processors, a *separate* Ageia Physics Processor (PPU), and a *two* 250GB harddrive. The speed with which this machine renders graphics and processes detailed physics is phenomenal. Game developers will soon be taking full advantage of these capabilities to create breathtaking scenes and visual effects that will become the standard against which soldiers will judge other computer-based programs.

A console point of reference: The Cell processor now being used in the Sony PS3 far exceeds the capabilities of even the Cray X1E supercomputer that was introduced as recently as March of 2005 (See Table 8). The Graphics Processing Unit of the PS3 yields a stunning 1.8 TFLOPS floating point performance. In addition, Ageia has joined

with Sony to provide real-time physics for PS3 applications. The combined power of these units will soon challenge developers to re-define the boundaries of their art.

Table 8. Single-precision Dens Matrix Multiplication (GEMM) for various processors

Processor	Gigaflops/sec
Itanium 2	3.0
AMD 64	7.8
Cray X1E Supercomputer	29.5
Cell*	204.7

*Note: Because Cell hardware wasn't available for testing, those generating the data in this table used a combination of performance projections and benchmarks on a cycle-accurate simulation of Cell that IBM has released. Real-world results should be comparable to those here, if not better.

Source: Lawrence Berkeley National Laboratory, 2005.

Both of the above-mentioned machines possess far greater power than is needed to drive the general-use applications that most DoD computers are purchased for, such as word-processing, spreadsheet, and database programs. But while DoD will not want all its computers to be capable of playing state-of-the-art games, if it is to pursue a goal of making game-based training and other programs more widely available to soldiers, it should establish a minimum specification for new machines that make them media capable. It is beyond the scope of this report to identify those specifications, but once they are established, serious game developers will be able to develop towards that specification with confidence that they are delivering the best possible experience for the installed base.

Console vs. PC Gaming

DoD has a PC-centric (rather than game-console oriented) technical environment. Yet the soldiers coming into the military are likely to have played more games on consoles than on the PC.[124] If the department is to provide programs on the platforms that soldiers are most familiar with, it will have to consider purchasing game console units and sponsoring projects developed for those platforms.

[124] In 2005, console games outsold computer games by more than six-to-one: 2005 console game sales, $6.06 billion; 2005 computer games sales, $953 million. Entertainment Software Association, *Essential Facts about the Computer and Video Game Industry*, 2006. *www.theesa.com/archives/files/Essential%20Facts%202006.pdf*

Interestingly, the capabilities of next generation console chips may make such purchases more attractive to DoD. Both the Xbox 360 and the PS3 feature network capabilities. In addition, the architecture of the Cell enables it to "roam over a network, allowing the processor to perform a type of distributed or grid computing" [125] Because the cost of the Cell chip has been driven down by consumer demand, it is sufficiently inexpensive that one could easily imagine it becoming an almost ubiquitous presence across many military platforms (Humvees, Bradleys, LAVs, UAVs, etc.). This would create a truly smart information-processing *fabric* (as opposed to a simpler node-based architecture), where the computational power of a particular device could support its assigned task, be partially re-tasked to handle service interruptions, or be totally re-tasked during idle time to perform computationally-heavy calculations for another application. Think of it as a distributed super computer spread across a platoon, company or battalion, at a cost significantly lower than most of the current architected solutions. Distributed applications such as this are already in place for the PC, and have recently been announced for the PS3, most notably Stanford University's Cure@PS3 project, which seeks to use the down-time of consumers' gaming consoles to create "simulations to further study of protein folding and related diseases, including Alzheimer's Disease, Huntington's Disease, and certain forms of cancer." [126]

Leveraging Game Industry Technical Methods

Just as hardware improvements have prompted re-evaluation of long-held M&S assumptions, game industry advances in technical methodologies may also be leveraged to benefit government programs.

For example, a study presented at I/ITSEC in 2005 examined the game industry's practice of moving Line of Sight calculations and route-planning algorithms onto the GPU, and compared it to the JSAF and OneSAF programs' practice of leaving such calculations on the CPU. The study found that adopting the game industry's method resulted in a significant 10x–20x improvement in performance.[127] One suspects that even greater enhancements could result from moving other calculations onto a PPU. Separate

---

[125] Becker, David. "PlayStation 3 Cell chip aims high." CNET News.com, Feb 4, 2005. *http://news.com.com/PlayStation+3+Cell+chip+aims+high/2100-1041_3-5563803.html*

[126] Stanford University, "Folding@Home FAQ," Oct 22, 2006, *http://folding.stanford.edu/FAQ-PS3.html.*

[127] Verdesca, Marlo, et al. "*Using Graphics Processor Units to Accelerate OneSAF: A Case Study in Technology Transition.*" I/ITSEC 2005, *www.iitsec.org/documents/R_2121_000.pdf.*

GPUs and PPUs did not exist at the time ModSAF (the genetic ancestor of JSAF and OneSAF) was created. Advances such as these and the game industry's uses of them suggest it may be appropriate to re-examine some of the basic M&S assumptions that were in place when the concepts of Semi-autonomous Forces (SAF), HLA, DIS, TENA, etc., were conceived.

Resolving Different Approaches to Optimization

Commercial PC game developers are economically driven to make their games transmit as little data as possible, because they pay huge connectivity charges to keep their servers online. Therefore they optimize for minimum bandwidth usage, and transfer as much of the load as possible onto consumers' machines (which they don't pay for). They minimize their costs by driving consumers to load up their machines with high-end features.

Commercial console developers optimize differently: because they sell their hardware at a loss and make their profits on software sales, they optimize for minimum memory usage, as memory is the single most expensive component of their hardware. They minimize their costs by making the hardware as inexpensive as possible, and driving consumers to pay a lot for the software.

The government, on the other hand, has neither of these constraints. It has access to large bandwidth at low cost, and it is motivated to keep the price of its hardware purchases to a minimum by foregoing expensive alternatives like high-end CPUs, GPUs, PPUs, etc. Therefore projects optimized for government use shouldn't be optimized for bandwidth or for minimum memory use, but to run on the "lowest common denominator" machine possible.

Resolving this is not as easy as simply recognizing the problem. The economic drivers of the respective businesses are reflected deeply within their system architectures and underlie everything those companies know about developing games. In the end, project managers may have to settle for only partial optimization for the government environment, because the expense of rebuilding everything "from scratch" would surely wipe out the economic benefits that full optimization might bring.

MMOG Technologies and Network-Centric Warfare

> Certain skills gained and practiced by gamers in massive multiplayer online gaming environments closely parallel those required by a military transforming itself to operating under the concept of network centric warfare. The technologies and practice methodologies employed in multiplayer games also hold great potential to provide appropriate network centric warfare training environments.[128]
>
> —DOD Technical Report

There are few, if any, places other than an MMOG where one can be truly immersed in network-centric behavior. In fact, MMOGs may currently be the *only* place where the principles of Network Centric Warfare can be tested and explored. MMOG architected games may come to form the first National Training Center in the network-centric world.

Independent of the many other reasons to study MMOGs (such as for their social and psychological interactions, their effectiveness in education and training, and their relevance in a military context), there are strong technical reasons for DoD to research this area through partnerships with commercial developers. Many commercial service considerations—network attacks, information assurance, bandwidth optimization, server optimization, account accreditation, content creation, and production issues—are amazingly similar to the issues in a network centric environment and can therefore serve as a fertile field for informing DoD's understanding of persistent networks for warfare.

To fully appreciate these technologies and the efficiencies the commercial market has attained, DoD would be well-served to dedicate some resources to actively monitoring and interacting with the leading edge of the commercial MMOG industry so as to have a fully informed view of the evolving state of the art. DoD should also support some application research through academic institutions or federal research laboratories to explore DoD-specific topics of interest in an environment free of the parochial interests of specific services and of game developers.

---

[128] Bonk, Curtis J. and Vanessa P. Dennen. Mar 2005. *www.adlnet.gov/downloads/189.cfm*

Engines

Most engine discussions begin by focusing on the initial cost of acquisition. We believe that is a mistake. The most important driver of engine selection should be the technical and operational requirements of the individual project.

If fast implementation is paramount, then the best "engine" may well be the free software distributed with many commercial products that allow them to be "modded" quickly by individuals with only moderate technical skills. So, for example, if the desired product is a "quick-and-dirty" representation of knowledge acquired in-theatre for rapid dissemination to other troops, a game mod might be a very good choice.

Other sets of fast, inexpensive, and easy-to-use technologies are also appearing on the scene. Linden Labs offers content creation tools for its "Second Life" virtual environment that allow fast prototyping of content and "simple 3D mock-ups with audio-video and scripting features [that] can be created and presented online within minutes."[129] In August 2006, Microsoft announced the release of its XNA Game Studio Express, "which will democratize game development by delivering the necessary tools to hobbyists, students, indie developers and studios alike to help them bring their creative game ideas to life while nurturing game development talent, collaboration and sharing that will benefit the entire industry."[130]

For products of moderate complexity, an open source engine that provides commonly-needed features may be the best choice.

> A review of most of these simulations reveals that almost all of them have essentially the same features; probably 90% of the functionality each provides is basically the same, with minor differences. The differences between all of them essentially boils down to the most advanced features each provides. However, these advanced features are not needed for the vast majority of simulations.[131]

This choice, however, should not be made because there is little or no initial acquisition cost for the engine. It should be made, instead, upon determining that the engine features satisfy the program requirements.

---

[129] Second Life, "*Second Life Features and Technology,*" *http://secondlife.com/developers/features.php.*

[130] Microsoft, Inc., "Microsoft Invites the World to Create its own Xbox 360 Console Games for the First Time." Press Release, Seattle, WA: Microsoft, Aug 13, 2006. *www.microsoft.com/presspass/press/2006/aug06/08-13XNAGameStudioPR.mspx*

[131] Microsoft, Inc., Aug 13, 2006.

The initial acquisition cost of an engine is relatively small compared to the overall development costs of most products, and while a particular engine may be inexpensive to *acquire*, it may be expensive to *use*.

Hidden costs of open source engines may include poor documentation, inadequate support, and licensing restrictions that may affect the use and security of the program.

Documentation for open source programs is usually out-of-date, almost always incomplete, and sometimes simply wrong. One study of open source projects concludes, "It was apparent...that online documentation was usually dated, and subsequently inconsistent with current functional capabilities or system commands."[132]

Poor support for open source products is also not uncommon. Although the above study asserts that there are usually informal processes whereby correct information can be found, one might question whether these processes, while adequate for amateur or volunteer development efforts, are sufficiently timely and reliable to use in professional projects, which cannot afford to lose time to faulty information or waiting for the correct information to appear. The same study notes that the problem of informal documentation exists partly because "the developers are generally end-users of the systems they develop, whereas in traditional software requirements engineering efforts, developers and users are distinct, and developers tend not to routinely use the systems they develop. Perhaps this is why open software systems can suffice with reliance on software informalisms, while traditional software engineering efforts must struggle to convert informal requirements into more formal ones."[133] This developer-as-user model will certainly not be the case for most government M&S projects.

Open source projects also carry licensing restrictions that may create hidden costs. The Office of Management and Budget warns that the total cost of ownership must include "lifecycle maintenance costs, the costs associated with risk issues, including security and privacy of data, and the costs of ensuring security of the IT system itself" and that the licensing restrictions of open source projects "may affect the use, the security,

---

[132] Scacchi, Walt, "Understanding the Requirements for Developing Open Source Software Systems" *IEE Proceedings—Software*, Feb 2002, 149(1), p. 24–39. *www1.ics.uci.edu/~wscacchi/Papers/New/Understanding-OS-Requirements.pdf*

[133] Scacchi, Walt, Feb 2002.

and the total cost of ownership of the software and must be considered when an agency is planning a software acquisition."[134]

A "special case" open source project is the Delta3D engine that has been developed by the MOVES institute. Unlike many projects whose origin, scope, and size of community are unknown, the Delta3D project is well-funded and was built with DoD requirements in mind. Perry McDowell, Executive Director of the project, says it was developed "to fit a need noted by many in both the modeling/simulation and gaming industries: a robust, easy-to-use engine which would prevent vendor lock-in."[135] The engine also has the benefit of making the game development process more broadly available to individuals and small groups, the benefits of which we have commented on elsewhere in this report.

Large, complex projects with custom features and advanced requirements should probably be built using commercially-licensed engines that are full-featured and well-documented, and that come with targeted support from the vendor. As game-based projects become more common, DoD should consider entering centrally-negotiated enterprise-wide license agreements with several vendors. This will present project managers with a broad array of engine choices enabling them to make decisions based not on license fees, but on capabilities and use.

Standards

As we noted in The Commercial World section, the game industry has historically resisted organized standards efforts and embraced instead the *de facto* standards that emerged as a result of competition in the marketplace. Additionally, the speed of technological advances, as compared to the slowness of the standards-setting process, makes the organized development of standards even more problematic (although in the case of consoles in general and the Cell processor in particular, the longer lifespan of stable platforms may make standards development more feasible).

The government, although it has attempted to reduce overusing standards, still has strong and legitimate needs for attention to standards in serious games, particularly with

---

[134] Evans, Karen S. and Robert A. Burton, "Software Acquisition," Memorandum M-04-16, Office of Management and Budget, Executive Office of the President, Washington, DC, Jul 1, 2004. *www.whitehouse.gov/omb/memoranda/fy04/m04-16.html*

[135] McDowell, Perry et al. "Delta3D: A Revolution in the Methods of Building Training Simulations," MOVES Institute, Naval Postgraduate School, Monterey, CA, 2005. *www.siaa.asn.au/get/2411856047.pdf*

regard to data output, analytical tool sets, and VV&A. Standardized data output will allow aggregation of information across products. Good analytical tool sets will enable researchers to more easily understand and use that information. Better VV&A will help determine whether a particular game succeeded or failed based on its appropriateness to the task, and on how well it was implemented in the first place.

Game developers who wish to partner with the government must recognize these special needs. More time and money must be budgeted into individual projects for building in relevant data extraction, analytical tools, and VV&A procedures. Additionally, both industry and government should participate in organized efforts to develop appropriate and useful standards in these areas.

Unfortunately, DoD's participation in this community of interest has been waning and it seems possible, given the recent realignments of DMSO and other M&S activities, that this involvement will continue to decline and perhaps become fractured across a multitude of entities. We hope instead, and strongly suggest, that DoD's renewed customer-focused M&S orientation will prompt it to reinvigorate its participation in these industry and standard-setting organizations so as to be fully aware and appreciative of the commercial sector's direction and to remain an influential part of that community, rather than a bystander.

Legacy Systems

One of the potential barriers to DoD's adopting new technology is its commitment to legacy systems. It is always difficult to know when to abandon maintaining and upgrading an existing system in favor of embracing a new one. Within the government, this decision is influenced by the lack of depreciation considerations in capital investment, the high cost (both in time and money) of developing new assets, and the political weight behind existing technologies and their supporting industries.

The government is currently faced with just such a decision in the case of JSAF, and to some extent OneSAF, JSIMs, and JWARs. These systems still contribute significantly to the overall M&S effort. However, as DoD evaluates its future investment decisions, it must decide whether the original design specifications of these systems are too constraining to economically afford significantly expanding their scope to meet new requirements, or whether these expanded needs should be met by new systems, programs, or processes.

Among the criteria that must be considered when deciding are the capabilities and availability of commercial alternatives—another compelling reason why DoD should continue its connection to, interest in, and support of the serious games industry.

Recommendations

1. DoD should establish a minimum specification for new desktop computers that make them media capable.

2. DoD should sponsor projects for game consoles such as the Nintendo, Xbox and Playstation, which are the gaming platforms that soldiers are most familiar with.

3. Future MS&G efforts should aggressively seek to leverage game industry techniques and processes.

4. Government project managers should be aware of the technology optimization strategies that are driven by their industry partners' business plans, and try to optimize instead for the government environment, so long as doing so is not economically disruptive to the project.

5. The demands of Net-centric warfare suggest that DoD should stay current with commercial MMOG technology and support academic research in this field.

6. DoD should consider entering centrally-negotiated enterprise-wide license agreements with several vendors. This will present project managers with a broad array of engine choices that will enable them to make decisions based not on license fees, but on capabilities and use.

7. DoD should stay active in industry standard-setting organizations and remain an influential part of that community.

8. When new MS&G requirements emerge, DoD should consider whether they are best met by expanding the capabilities of legacy systems, or by developing new game-based programs.

V. The Future

Certain trends that will influence the military's use of games in the future are already discernible.

Games Will Play a Large Role in the Military

As many as 100 serious game projects are underway. Anecdotal accounts indicate they are valuable tools in many situations. Academic research is beginning to support those accounts and to better define circumstances in which games might be most effective.

Game Companies Will Become More Interested in Serious Games

The When Worlds Collide: Current Efforts section of this report lists 26 companies who are already involved in military-oriented serious games. This number will almost certainly grow. Ben Sawyer, founder of the Serious Games Conference, said in May 2006, "I believe that every company in the games space will have a serious games related business position in the next ten years."[136] Although he was referring to the entire field of serious games (including non-military applications), the ongoing consolidation within the game industry suggests that game companies will increasingly be looking to diversify into serious games as a continuing source of revenue.

This interest, however, will probably not lead to steady-state expansion: some of the companies who enter the field will succeed and some will fail. It is likely that the challenges of working with the government will lead to waves of expansion and contraction as companies enter and subsequently leave the serious games space. This has the benefit of giving contracting officers more companies to chose from for their pro-

---

[136] Dobson, Jason, "Serious Games Initiative/Games For Health At E3 2006: A Look Back," SeriousGamesSource.com, May 26, 2006, *http://seriousgamessource.com/features/feature_052606_games_for_health.php*.

jects, but the corresponding disadvantage of being more difficult to identify which companies will make good partners for those projects.

DoD Will Need More Agile M&S

Past large-scale, long-term efforts such as OneSAF, HLA, JSAF, Millennium Challenge, JSIMS, JWARS, etc., are viewed in some quarters as not only costly, but ineffective in meeting today's challenges.

As a 2002 RAND study concluded, "The "old" [M&S] business model is characterized as being both fiscally wasteful and a hindrance to innovation because it created a system of inefficient long-term commitments to what are effectively contractor-proprietary simulation systems."[137]

In an era when significant threats come from small, agile, and widely-dispersed enemies, systems are needed that can be rapidly designed and implemented. In this environment, speed and adaptability are paramount. Yet, unless it significantly changes its internal culture, DoD will continue to wrestle with the dichotomies between large, monolithic systems, "systems of systems," and lightweight, agile, and even disposable models and games.

Commercial Games will Continue to Innovate, but...

The consumer gaming industry will continue to spawn technological advancements which the government can take advantage of. However, there will always be governmental M&S needs that the industry will not fulfill. When the government looks ahead to anticipate its needs, it is clear that some will be met at little or no R&D cost by piggybacking on industry efforts. This is arguably true, for example, in areas such as graphics cards, physics processors, and user interface design. This trend is also evident in the use of gaming technologies to create cyberlogs (or "glogs") that transmit in real time the personal experiences of soldiers in the field.

However, there will also be gaps between what the industry will produce as a result of market forces, and what the military will need to fulfill its special mission. DoD must consider where these gaps may develop, and decide what investments it should

---

[137] Paul, Christopher, et al. *Implementing and Evaluating an Innovative Approach to Simulation Training Acquisitions.* Rand National Defense Research Institute, 2006. *www.rand.org/pubs/monographs/2006/RAND_MG442.pdf*

make in order to fill those gaps either through its own research and development, or through partnering with industry leaders.

For example, in the field of chip design for radiation environments, the Department of Energy has acquired a royalty-free license from Intel to develop "rad hard" Pentium processors for US space and defense purposes. According to a 1998 SpaceDaily.com article, "In recent years, the rapid pace of design innovation for commercial integrated circuit (IC) applications, such as personal computers, has outdistanced the budgetary ability of military and space designers to design comparable performance ICs for radiation environments."[138]

Government won't Continue to Drive Innovation and Standards

In the past, the government was the major customer for technological advances, and as such, was the primary driver for creating many hi-tech standards and specifications. While this remains true in many cases, advances in the field of non-mainframe computing are now driven primarily by consumer demand, and the government does not wield the standard-setting power it once did.

Furthermore, technology innovation tends to be accelerated by technology *use,* and users in the private sector far outnumber those within government. Because ideas for new technology tend to originate with the users of current technology,[139] it is reasonable to expect more innovation to come from outside the government than within.

With many millions of non-government users now driving markets and innovation, new standards will tend to emerge from the consumer and serious game marketplace. The government will undoubtedly still participate in standards-setting activities, but it will do so in collaboration with non-governmental bodies as one of many voices at the table.

---

[138] "Sandia To Develop Intel Rad-Hard Chips," press release, SpaceDaily.com, Dec 9, 1998, *www.spacedaily.com/news/radiation-98c.html.*

[139] "Technological learning is the essential step that paces adoption and diffusion. 'Learning-by-doing' contributes to reductions in production costs, and adopters of new technology contribute to ongoing innovation through 'learning by using.' Widespread adoption, in turn, accelerates the incremental improvements from learning by users and producers, further fueling adoption and diffusion." *Lessons Learned from U.S. Technology and Innovation Policies.* PEW Center on Global Climate Change, Arlington, VA. *www.pewclimate.org/policy_center/policy_reports_and_analysis/technology/lessons.cfm*

Offshoring Implications

Offshoring will have capability and security implications for military projects. Government agencies who want to ensure that applications are developed in the United States may face increasing difficulties in years to come because more and more companies are moving vital game development processes overseas to save money. This "brain drain" will reduce the amount of US-based game-making capability, with a correspondingly higher security risk that comes with code written overseas. In addition, this loss of resident intellectual knowledge and innovation will have both national security and commercial implications.

Globalization & Open Sourcing

The globalization of the industry and open sourcing will also have security implications. Even if a government agency partners with a commercial US-based company, there is a high probability that the company's software will contain code written outside the United States. Even "home-grown" engines often contain sub-engines developed elsewhere. To take just one example, the popular Unreal3 engine which will drive the next *America's Army* game and which has been licensed by over 70 other developers, comes with a physics engine developed by Havok, a company whose development takes place in Dublin, Stockholm, Calcutta, Munich, and Tokyo.

Other popular engines also include open source code whose country (or countries) of origin may be impossible to trace.

As the rate of globalization increases, more and more projects will include code that is either open source or has been written overseas. Establishing that this code is secure will become increasingly difficult.

Games as Ambassadors

Countries and organizations will continue to use games as a way of explaining their cultural values to the world. As we documented earlier in this section, games are increasingly being used as tools of persuasion. Organizations whose aims are as diverse as the United Nations (*FoodForce*), Hezbollah (*Special Force*), The Canadian Department of Foreign Affairs (*Pax Warrior*), and MTV (*Darfur is dying*), are funding games intended to win hearts and minds and to drive people to act.

In some cases, these games are simply meant to make a political point and bolster nationalism, as in the recent state-funded Iranian mod to *CounterStrike*[140] The Iranian addition allows players to sink a US oil tanker in the Strait of Hormuz. "We show in this game…how easily we can spoil their [the US] party by shutting down their oil artery," said Ahmadreza Nouri, the game's designer.[141]

This trend shows no signs of abating. More than twice as many people attended the June 2006 *Games For Change* Conference as the year before.[142] Conference speakers pointed out that activist games are being developed by groups spanning the ideological spectrum, from white supremacists and hate groups, to developers interested in world peace and global justice. At the conference, the president of the Serious Games Initiative called for the creation of a Corporation for Public Gaming.

Democratization of Development

Lower barriers to entry will make it easier for individuals and poorly-funded organizations to create agenda-oriented games. The increasing availability of free technology and the democratization of game-development tools will bring an increase in games expressing anti-American views and advancing agendas potentially harmful to the United States. It is important that DoD thoroughly understand these technologies to evaluate how they may be exploited both for and against American interests.

Keeping Pace with Foreign Governments

Foreign governments will continue to fund their game-creating industries as an important part maintaining their overall technical infrastructure. As was noted in the "Commercial: Future" section of this report, China will be spending $1.8 billion over the next five years, Singapore will invest $614 million over the next 10 years, and other countries are investing significantly in their domestic game industries.

---

[140] "Iranian Video Game Offers Chance to Blow Up U.S. Tanker," (Reuters), as reported in *The New York Times*, Sep 30, 2006.

[141] "Iran Video Game: Sink U.S. Oil Tankers," NewsMax.com, Oct 1, 2006, *http://newsmax.com/archives/ic/2006/10/1/160833.shtml.*

[142] Jana, Reena, "Getting Activist Video Games to Market," *Business Week Online*, Jun 30, 2006, *www.businessweek.com/innovate/content/jun2006/id20060630_662842.htm.*

Entities such as the European Union will also create communities of interest to further their games industries. For example, the "Game Tools Project" is an EU-sponsored effort under its "6th Framework Programme that brings together leading European computer graphic experts from universities in Austria, France, Hungary and Spain with European industrial partners from the fields of computer game development and virtual reality to create next generation realtime 3D libraries for geometry, visibility, and global illumination."[143]

Testing Virtually

Massively Multiplayer Games and Net-based "sandbox" environments will become increasingly important test-beds. With the military's embrace of Net-Centric Warfare comes the need for a space in which to test its applications. When a team is virtual—when they've never met each other, haven't trained together, yet have a common goal—the problems of building team intuition, creating trust, and establishing other factors that make teams work, become much more difficult. In commercial MMOs, these challenges are successfully met every day when groups of gamers meet and go on "raids." Players who have never met in real life form teams of specialists who plan and carry out intricately orchestrated sequences whose success depends on trust and cooperation. MMOs are an ideal training ground for the military of the future.

MMOs will also continue to harness the creative power of thousands or millions of inventive gamers to test government solutions to difficult problems. An example of this principle can be seen in the RSA's DES efforts to create unbreakable encryption methods, which they then open up to the world, only to find that the "million minds" can crack them with disheartening speed.

MMOs can also be used as test-beds to study:

- Factors that influence an individual to join an affinity group.
- How such groups communicate.
- How groups collaborate to further their goals.
- Potential flaws in security systems. (Another example of this is Microsoft making advance copies of *Vista*, their next operating system, available to individuals

[143] Game Tools, "Game Tools Project Overview," GameTools. com, *www.gametools.org/html/overview.html.*

> at the notorious Black Hat and DefCon "Hackers" conferences to see if the "bad guys" could discover exploitable weaknesses in the software.[144]

Economics professor Edward Castronova wrote in a 2006 article in the *Yale Economic Review:*

> Synthetic worlds are a special kind of host for human society: being synthetic, they can be designed for a purpose. Worlds can be built that teach anyone in the world about Renaissance England or about practical democracy. Perhaps even more promising, they can be built to explore the evolution of social patterns, effectively becoming social science Petri dishes. These controlled environments for studying the evolution of macro-level forces of government, law, economics, sociality, learning, and culture present an unprecedented opportunity to open a new frontier in the understanding of human affairs.[145]

Evolution of Collaborative Virtual Spaces

Collaborative, three-dimensional spaces will also become useful. Like MMOGs, non-game online spaces like Croquet and Second Life (SL) offer environments for dispersed teams to work together, as well as opportunities to tap the creativity of thousands of users.[146]

These virtual destinations provide both an outlet for in-world creativity, and for real-world entrepreneurial opportunities. For example in Second Life, virtual-world clothing designers create their own fashions (which they can sell for real-world money), and established clothing retailers such as American Apparel and Adidas sell items in Second Life that mimic apparel they sell in their brick-and-mortar stores.[147]

Virtual worlds may also become predictors of real-world behaviors and fashions. "The banking giant Wells Fargo built its own branded island inside SL, designed to train young people to be financially responsible. Wal-Mart, American Express and Intel are looking at using SL for their corporate training. And why not? With its natural interac-

---

[144] Evers, Joris, "Microsoft gets good reception at Black Hat," CNETNews.com, Aug 3, 2006, *http://news.com.com/Microsoft+gets+good+reception+at+Black+Hat/2100-7355_3-6102068.html?tag=nl.*

[145] Castronova, Edward, "The Data Game: How Economists Can Learn From Online Video Games," *Yale Economic Review*, Summer 2006. *www.yaleeconomicreview.com/issues/summer2006/datagame*

[146] Croquet and Second Life's home pages, respectively, are *www.opencroquet.org/* and *http://secondlife.com/*.

[147] LaVallee, Andrew, "Now, Virtual Fashion," *The Wall Street Journal*, Sep 22, 2006, p. B1. *http://online.wsj.com/public/article/SB115888412923570768-HtFYrBweWpF25yJkL0CdXvkFRkY_20070922.html*

tivity and open platform for creation, Second Life, or something like it, may very well be the next generation of the Web."[148]

An Innate Videogame Military Generation

Games will continue to profoundly influence the decision-making of tomorrow's soldiers. Soldiers entering tomorrow's army will have grown up playing videogames. Their approach to group interaction and problem-solving will be different from the generations that preceded them. As Mark Prensky points out in his presentation "Innovations in eLearning,"[149] playing complex games will have taught them to:

- Cooperate, collaborate, work in teams—i.e., to work effectively with others
- Make effective decisions under stress
- Take prudent risks in pursuit of objectives
- Make ethical and moral decisions
- Employ scientific deduction
- Quickly master and apply new skills and information
- Think laterally and strategically
- Persist and solve difficult problems
- Understand and deal with foreign environments and cultures
- Manage business and people

In addition, the principles that shape these soldiers' decision-making will be transformed. In the book *Got Game*, authors Beck and Wade point out that gamers have logged thousands of hours during which they rapidly analyze new situations, interact with people they don't know, and learn to solve problems quickly and independently. The authors also summarize the problem-solving rules that drive the gamer generation:

- If you get there first, you win.
- There's a limited set of tools, and it is certain that *some* combination will work. If you choose the right combination, the game will reward you.
- Trial and error is the best strategy and the fastest way to learn.
- Elders and their received wisdom can't help; they don't understand even the basics of this new world.

---

[148] Newitz, Annalee, "Your Second Life is Ready," Popsci.com, Sep 2006, *www.popsci.com/popsci/technology/7ba1af8f3812d010vgnvcm1000004eecbccdrcrd.html.*

[149] Prensky, Marc, "Innovations in eLearning," presentation, Jun 7, 2006, *http://view.dau.mil/dauvideo/view/viewEvent.jhtml;jsessionid=ECCZNLKB0OUYRAF4VLHSFEQ?contentid=2198&eventid=1045.*

- You will confront surprises and difficulties that you are not prepared for. But the sum of those risks and dangers, by definition, cannot make the quest foolish.
- Once you collect the right "objects" (business plan, prototype, customers, maybe even profits), you'll get an infusion of gold to tide you over.
- While there may be momentary setbacks, overall the trend will be up. [150]

Innovative training ideas will come from "digital natives," rather than from the older generation of training personnel and M&S managers. The most likely sources of new plans and new tools are gaming "natives" with backgrounds in education (both institutional and organizational), sociology, anthropology, and entertainment.

Implementing Usability Research

The military will benefit from the game industry's usability research. The game industry has figured out how to deliver very complex information to users in ways they can quickly understand. The best game interfaces are highly efficient and allow the player to process multiple streams of information simultaneously.

The usability research behind game controllers will also be adapted by the military to make the operations of their systems easier and more intuitive, especially to the "digital natives" of tomorrow's armed forces.

Like the soldiers who adapted the Playstation controller to drive Abrams tanks and Bradley fighting vehicles, the military may find it effective to adapt gaming technology for more serious purposes. Another example of this is the robot truck "Dragon Runner" whose controller is modeled after the PlayStation2, "because that's what these 18-, 19-year-old Marines have been playing with pretty much all of their lives," according to project manager Maj. Greg Heines.[151]

New Opportunities

The convergence of the Internet, mobile communications, and other advances in connectivity will create new gaming opportunities for both the private sector and for DoD. In the past, most games were developed for a single platform or device. Now,

---

[150] Beck, John C. and Mitchell Wade. *Got Game: How the Gamer Generation Is Reshaping Business Forever*. Boston, MA: Harvard Business School Press, 2004.

[151] Gomes, Pedro, "Army Video Games and Gadgets," InfoSatellite.com, Jun 24, 2003, *www.infosatellite.com/news/2002/06/p240602army.html*.

however, games can be accessed from almost anywhere and played on almost any device. As an example, the GPS capabilities built into all cell phones starting in 2005[152] have enabled a whole new genre of games, called location-based entertainment, which represents the first step toward augmented reality. These games link actual locations and real events to made-up stories and contests in the gameworld. One such 3G mobile technology game gives away prizes to customers who find the most "treasures" at major Hong Kong subway stations before a specified date.[153]

It is not difficult to imagine a game set in a real city in which a team of players linked only by a network connection must improvise routes through the city, interact with the population and environment, and collaborate with each other to develop new plans and strategies as they adapt to whatever unexpected challenges present themselves.

When a game can be accessed from anywhere and the users are not tied to their PC or console, new kinds of interactions emerge and new kinds of gameplay develop that are extremely useful in examining the conditions of a network-centric life.

Collaborating Across Borders

The international flavor of the gaming community will increase contact and collaboration across borders. This will have both positive and negative effects. On the negative side, it may enable hostile elements to find each other more easily, and the unmonitored communications within gameworlds may present a security risk. On the positive side, increased contact across cultures may help humanize opponents instead of demonizing them, undermining popular support for an institutionally-generated conflict.

The emergence of automatic language translators ("auto translators") will accelerate these trends, allowing people who speak different languages to communicate and exchange ideas in-game.

This technology is here today. The authors know of one 14-year-old who plays *Final Fantasy XI* almost exclusively on a Chinese server. He cannot speak Mandarin—the game's auto-translator allows him to use English to speak to his Chinese friends.

---

[152] Adomatis, Doug, "Using the GPS for People Tracking," TravelbyGPS.com, Aug 2006, *www.travelbygps.com/articles/tracking.php*.

[153] Hong Kong Wireless Development Centre, "E 3G Cyberport Treasure Hunt," press release, 2006, *www.hkwdc.org/3g_application.html*.

Large, supra-national communities will be sufficiently large to wield political and economic influence. The economies within some gameworlds are approaching the size of developed countries. The game *World of Warcraft* had 6.6 million subscribers worldwide as of June 2006.[154] As long ago as 2001, economics professor Edward Castronova studied the *EverQuest* world of Norrath, where "the exchange rate between Norrath's currency and the U.S. dollar is determined in a highly liquid (if illegal) currency market, and its value exceeds that of the Japanese yen and the Italian lira…Norrath's GNP per capita easily exceeds that of dozens of countries, including India and China."[155]

More recently Castronova wrote, "The resulting synthetic worlds have evolved from mere gamespaces into fully functional microsocieties that parallel our own yet exist only in cyberspace, outside the bounds of any nation's sovereignty."[156]

How long will it be before the cyber-citizens of these synthetic societies seek political representation and begin to wield economic power in the real world?

Recommendations

Given the important role that games will play in the future of the military, what stance should the government take in order to gain the maximum benefit from gaming technology? The authors of this report believe there are places where government involvement would be beneficial, and others where a *laissez-faire* policy would produce the best results.

As we recommended at the end of the "Business" section of this chapter, DoD needs to establish a viable and sustainable serious games marketplace.

A dynamic "ecosystem" is a major contributor to the success of any technical community. Good supply of talent, good demand from customers, and an efficient system to help them find each other, all combine to create a healthy environment from which all participants benefit. Low barriers to entry create a fertile breeding ground for new artists, new ideas, and new technologies.

---

[154] MMOGChart.com, *www.mmogchart.com/*.

[155] Whiting, Jason, "Online Game Economies Get Real," *Wired*, Nov 6, 2002, *www.wired.com/news/games/0,2101,55982,00.html.*

[156] Castronova, Edward, "The Data Game: How Economists Can Learn From Online Video Games," *Yale Economic Review*, Summer 2006. *www.yaleeconomicreview.com/issues/summer2006/datagame*

The emergence in the commercial game world of IGN, Gamespy, and similar news, distribution, and community sites has encouraged the development of just such an ecosystem where people talk about, compare, recommend, try out, and buy products. The result is an environment in which good creative talent and superior products rise quickly to the top, while inferior talent and products rapidly disappear.

To facilitate this, at a minimum DoD should set up a short term task force or working group comprised of people from the commercial and government marketplace to encourage information-sharing among the grass-roots efforts that have sprung up in various offices. This group can help vet game companies, share acquisition and management tips, and generally improve the quality and efficiency of existing and future DoD game-related efforts.

But to "do the job right," DoD should create a government-chartered non-profit entity that is empowered to create a true marketplace for serious games.

DoD should formally acknowledge the value of games for specific uses and issue guidelines that enable contracting officers to arrange for their acquisition and development without having to "creatively circumvent" current acquisition policies.

DoD should try to avoid the consolidation of suppliers that has occurred in other areas of acquisition. Instead of four or five massive suppliers, the department should try to work with a wide range of small companies to take advantage of the agility, diversity, and creativity the game industry has to offer.

DoD should not try to mandate industry standards, but instead should collaborate with the industry as standards emerge.

DoD should sponsor a "Grand Challenge" that encourages the industry to advance the medium of games as an art form.

More specifically, we propose a Grand Challenge that targets and rewards the achievement of "subtlety" within a game.

Currently, all the subtle information that people process to make decisions is incapable of being reproduced in our gaming environments, and consequently is lost. When one compares the crude representations of an in-game "agitated" crowd to the subtle indicators of a real-world gathering in Baghdad on the verge of erupting, one quickly realizes how far the industry has yet to go, and how vital it is for that gap to be closed.

Until we can use subtlety to create deep emotions within players, we will be limited to emulating only the grossest human behaviors. But if we can achieve this goal of subtlety, we will open the floodgates of the industry, enable it to evolve to a higher level of artistry, and provide our soldiers with a truly immersive learning environment.

DoD should develop or acquire code-checking programs that can efficiently analyze code for security risks. The NSA may already be involved in this area,[157] as well as the Department of Homeland Security's Science and Technology Directorate.[158]

DoD should evaluate the effectiveness of "persuasive" games used by hostile countries and organizations, and determine whether it would be appropriate to develop games that explain American values to others around the world.

DoD should recognize the fundamental shift in the analytical and strategic problem-solving skills and techniques of the next generation of soldiers, and adapt its training and motivational methodologies accordingly.

In summary, DoD should create a viable marketplace, leverage commercial investments, and extend the art and science of immersive environments through close collaboration with the games industry.

---

[157] Onley, Dawn S., "NSA to take the lead on DoD info assurance" InfoSec News, Dec 15, 2004. *http://lists.jammed.com/ISN/2004/12/0053.html*

[158] Evers, Joris, "Homeland Security helps secure open-source code." CNETNews.com, Jan 10, 2006, *http://news.com.com/Homeland+Security+helps+secure+open-source+code/2100-1002_3-6025579.html.*

Appendix A: Acronyms & Abbreviations

AAR	after action review
AASG	Army Advanced Studies Group
ABM	Air Battle Model
ACES	Air Command Exercise System
ACT	Accelerated Combat Timeline
ADL	Advanced Distributed Learning
AFTRA	American Federation of Television and Radio Artists
AI	artificial intelligence
AIISC	Artificial Intelligence Interface Standards Committee
AoA	Analysis of Alternatives
AP	assistant or associate producer
API	application programming interface
ARPANET	Advanced Research Projects Agency Network
BAA	broad area announcement
CAL3D	Character Animation Library
CAM	Content Aggregation Module
CCTT	Close Combat Tactical Trainer
CDD	Capabilities Development Document
COCOM	Combatant Command
COGS	cost of goods
COLLADA	COLLAborative Design Activity
COTS	commercial off-the-shelf
CPD	Capability Production Document
CPU	central processing unit
CRPG	computer role-playing game, also RPG
DARPA	Defense Advanced Research Projects Agency
DFAR	Defense Federal Acquisition Regulations
DGA	Director's Guild of America
DIS	Distributed Interactive Simulation
DMO	Distributed Mission Operations

DMSO	Defense Modeling and Simulation Office
DOD	Department of Defense
DODD	Department of Defense Directive
DSP	Defense Standardization Program, Policies and Procedures
DTRA	Defense Threat Reduction Agency
EADSim	Extended Air Defense Simulation
ESA	Entertainment Software Association
ESRB	Entertainment Software Rating Board
EULA	end user license agreement
FAR	Federal Acquisition Regulation
FCS	Future Combat Systems
FFPAS	FireFinder Position Analysis System
FMV	full motion video
FPS	first person shooter
FRPD	full rate production and deployment
FX	effects
GUI	graphical user interface
GNE	game networking engine
HLA	high level architecture
HMMWV	high mobility multi-purpose wheeled vehicles
IDA	Institute for Defense Analyses
ICD	Initial Capabilities Document
ICT	Institute of Creative Technology
IEEE	Institute of Electrical and Electronic Engineers
I/ITSEC	Interservice/Industry Training, Simulation, and Education Conference
I/O	input/output
IGDA	International Game Developer's Association
IP	intellectual property
IOT&E	Initial Operational Test and Evaluation
IPR	internal program review
JAWP	Joint Advanced Warfighting Program
JCIDS	Joint Capabilities Integration Development System
JFCOM	Joint Forces Command
JSAF	Joint Semi-Automated Forces
JSF	Joint Strike Fighter
JSIMS	Joint Simulation System

JWARS	Joint Warfare Simulation
LAN	local area network
LD	level design
LRIP	low rate initial production
M&S	modeling and simulation
MDA	Milestone Decision Authority
MDF	market development fund
MIDI	musical instrument digital interface
MMOG	massively multiplayer online game
MMORPG	massively multiplayer online role-playing game
MOD	modification
MPEG	moving picture experts group
MS&G	modeling, simulation and gaming
MSRR	Modeling and Simulation Resource Repository
MSSM	Modeling and Simulation Standardization Area
MUD	multi-user dungeon
MURI	Multi-disciplinary research program of the University Research Initiative
NES	Nintendo Entertainment System
NDA	non-disclosure agreement
NIH	not invented here
NISP	National Industrial Security Program
NPC	non-player character
NSA	National Security Agency
ODE	Open Dynamics Engine
OEM	original equipment manufacturer
OMA	Open Mobile Alliance
OSD	Office of the Secretary of Defense
OSG	Open Scene Graph
OUSD	Office of the Under Secretary of Defense
PPBE	Planning, Programming, Budget Execution
PMTRASYS	Program Manager of Training Systems
PSW	persistent state world
PvE	player versus entertainment
PvP	player versus player
QA	quality assurance
RDECOM	Research Development and Engineering Command

RDT&E	research, development, testing and evaluation
RFP	request for proposal
ROI	return on investment
RPG	role-playing game
RTS	real-time strategy
SAG	Screen Actors Guild
SAP	special access program
SBA	Simulation-Based Acquisitions
SBIR	Small Business Innovative Research program
SCI	sensitive compartmentalized information
SCORM	Sharable Content Object Reference Model
SEDRIS	Synthetic Environment Date Representation and Interchange Specification
SIM	simulation
SISO	Simulation Interoperability Standards Organization
SLAMEM	Simulation of the Locations and Attack of Mobile Enemy Missiles
STTR	Small Business Technology Transfer program
SVGA	super video graphics array
TBD	to be done
TDS	Technology Development Strategy
TRADOC	Training and Doctrine Command
VGA	video graphics array
VOIP	voice over Internet protocol
VV&A	Validation, Verification and Accreditation
WAN	wide area network
WYSIWYG	what you see is what you get

Appendix B: References

Adomatis, Doug, "Using the GPS for People Tracking," TravelbyGPS.com, Aug 2006, *www.travelbygps.com/articles/tracking.php*.

Advanced Distributed Learning, "About ADL," OUSD P&R, 2005, *www.adlnet.gov/aboutadl/index.cfm*.

———, "Introduction to SCORM and the ADL Initiative," presentation, *www.adlnet.gov/aboutadl/index.cfm*.

———. "SCORM 2004." Third edition, 2004, *www.adlnet.gov/scorm/index.cfm*.

Alberts, David S. and Richard E Hayes. *Code of Best Practice, Campaigns of Experimentation*. Washington, DC: CCRP Publications, Mar 2005, p. 15. *www.dodccrp.org/files/Alberts_Campaigns.pdf*

Baranowski, Tom. Speaker on the topic of Learning and Behavioral Outcomes for Games: Research Studies at the conference "Challenges and Opportunities in Game-Based Learning." The National Academies, Nov 2, 2005. *www.nae.edu/nae/caseecomnew.nsf/weblinks/NFOY-6HXRHX?OpenDocument*

BBN Technologies, press release "DARWARS Training Superiority Program Participants Demonstrate Systems at I/ITSEC." (Chatham quoted), Dec 6, 2004. *www.bbn.com/News_and_Events/Press_Releases/2004/*

Beck, John C. and Mitchell Wade. *Got Game: How the Gamer Generation Is Reshaping Business Forever*. Boston, MA: Harvard Business School Press, 2004.

Beck, Kent, et al. "Manifesto for Agile Development." 2001. agilemanifesto.org

Becker, David. "PlayStation 3 Cell chip aims high." CNET News.com, Feb 4, 2005. *http://news.com.com/PlayStation+3+Cell+chip+aims+high/2100-1041_3-5563803.html*

Beerman, S. et al. "From Campus to Combat, Military Spurs Controversy." (Gaulzetti quoted.) Medill, 2006. *http://observer.medill.northwestern.edu/301-wi06-sec04/04recruitment_challenges/01main_story/*

Biyd, Clark. "Darfur Activism Meets Video Gaming." (Friedman quoted.) *BBC News,* Jul 6, 2006. *http://news.bbc.co.uk/2/hi/technology/5153694.stm?ls*

Blumenfeld, Samuel L. "Education vs. Training." World Net Daily website, Feb 7, 2000. *www.worldnetdaily.com/news/article.asp?ARTICLE_ID=16292*

Bonk, Curtis J. and Vanessa P. Dennen. *Massive Multiplayer Online Gaming: A research Framework for Military Training and Education.* Technical Report 2005-1. Washington, DC: DOD, Office of the Under Secretary of Defense for Personnel and Readiness, Advanced Distributed Learning Initiative, Mar 2005. *www.adlnet.gov/downloads/189.cfm*

Bray, Hiawatha. "War Games 2.0." *The Boston Globe*, (Chatham quoted), Dec 8, 2004. *www.boston.com/business/globe/articles/2004/12/08/war_games_20/*

Brewer, Garry. *The War Game.* Cambridge, MA: Harvard University Press, 1979, pp. 130–131.

Bridge Ratings LLC, press release, Feb 20, 2006, www.bridgeratings.com/.

Brown, S. J. et al. "Educational video game for juvenile diabetes: Results of a controlled trial." *Medical Informatics*, 1997, vol. 22, no. 1, pp. 77–89. *www.comm.ucsb.edu/faculty/lieberman/*

Burroughs, Chris. "Sandia-developed game helps Special Forces learn adaptive thinking, problem solving." *Sandia Lab News*. Sandia National Laboratories, May 13, 2005. *http://sandia.gov/LabNews/*

Caffery, Matthew. "Chapter 2: Wargaming and the World Wars." Unpublished work, matthew.caffrew@wpafb.af.mil.

Callaham, John. "America's Army Interview." (Interview with Chris Chambers.) Firing Squad website, Jun 21, 2006, *www.firingsquad.com/games/americas_army_interview/*.

Castronova, Edward, "The Data Game: How Economists Can Learn From Online Video Games," *Yale Economic Review*, Summer 2006. *www.yaleeconomicreview.com/issues/summer2006/datagame*

Center for Advanced Research in Technology for Education. "The DARPA Tactical Language Training Project." Defense Advanced Research Projects Agency, *www.isi.edu/isd/carte/proj_tactlang/description.html.*

COLLADA, About COLLADA, 2006, *collada.org/public_forum/welcome.php.*

Computer Industry Almanac, press release, Nov 14, 2005. *www.c-i-a.com/pr1105.htm*

Coyle, Philip E. Modeling and Simulation Conference, presentation, May 1998, *www.dote.osd.mil/presentations/coyle051398/sld038.htm.*

Defense Acquisition University, Defense Acquisition Policy Center, CJCSI 3170.01 E and B, May 11, 2005, *http://akss.dau.mil/dapc/index.html.*

Defense Federal Acquisition Regulations Supplement (DFAR), *www.acq.osd.mil/dpap/*.

Defense Sciences Office, Rapid Mission Rehearsal description, Daniel Kaufman, program manager. *www.darpa.mil/DSO/thrust/biosci/rmr.htm*

Defense Standardization Program, "DoD 4120.24-M: DSP Policies & Procedures C3.5," Mar 2000, www.dsp.dla.mil/documents/4120.24-M/chapter3.htm#C3.5.

Dobson, Jason, "Serious Games Initiative/Games For Health At E3 2006: A Look Back," SeriousGamesSource.com, May 26, 2006, *http://seriousgamessource.com/features/feature_052606_games_for_health.php.*

Ebert, Roger. "Movie Answer Man." *Chicago Sun Times*, Nov 27, 2005. *http://rogerebert.suntimes.com/apps/ pbcs.dll/article?AID=/20051127/answerman/511270304/1023*

Egenfeldt-Nielsen, Simon. "Thoughts on Learning in Games and Designing Educational Computer Games." Game-Research.com, Jul 2003. *www.game-research.com/art_educational_games.asp*

Electronic Arts, press release, Jun 17, 2005, *www.info.ea.com/news/pr/pr645.pdf.*

ETS News, press release "Low Cost Training Simulation will Provide ANG Pilots with a Portable and Flexible Solution." May 15, 2006. *www.ets-news.com/third.php?id=735*

Evans, Karen S. and Robert A. Burton, "Software Acquisition," Memorandum M-04-16, Office of Management and Budget, Executive Office of the President, Washington, DC, Jul 1, 2004. *www.whitehouse.gov/omb/memoranda/fy04/m04-16.html*

Evers, Joris, "Homeland Security helps secure open-source code." CNETNews.com, Jan 10, 2006, *http://news.com.com/Homeland+Security+helps+secure+open-source+code/2100-1002_3-6025579.html.*

Evers, Joris, "Microsoft gets good reception at Black Hat," CNETNews.com, Aug 3, 2006, *http://news.com.com/Microsoft+gets+good+reception+at+Black+Hat/2100-7355_3-6102068.html?tag=nl.*

Faris, Stephan. "Meet the Next Disney." *Fortune Magazine*, Nov 28, 2005. *http://money.cnn.com/magazines/fortune/fortune_archive/2005/11/28/8361976/index.htm*

Federal Acquisition Regulation (FAR) 16.101(b), General Services Administration FAR Secretariat, *www.arnet.gov/far/current/html/Subpart%2016_1.html#wp1085495.*

——— 6.102(d)(2), General Services Administration FAR Secretariat, *www.arnet.gov/far/current/html/Subpart%206_1.html#wp1087648.*

Federation of American Scientists. "Future Combat Systems (FCS)." Washington, DC: Federation of American Scientists, May 17, 2000. *www.fas.org/man/dod-101/sys/land/fcs.htm*

Fuilford, Deb. "Distributed Interactive Simulation: Its Past Present and Future." Proceedings of the 1996 Winter Simulation Conference, J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain, eds., p. 179.

Game Tools, "Game Tools Project Overview," GameTools. com, *www.gametools.org/html/overview.html.*

Garris, R. and R. Ahlers. "A Game-Based Training Model Development, Application, and Evaluation." I/ITSEC, 2001.

Gee, James Paul. "What Would a State of the Art Instructional Video Game Look Like?" *Innovate*, Aug/Sept 2005, vol. 1, no. 6. *www.innovateonline.info/index.php?view=article&id=80*

Gomes, Pedro, "Army Video Games and Gadgets," InfoSatellite.com, Jun 24, 2003, *www.infosatellite.com/news/2002/06/p240602army.html.*

Green, C. Shawn and Daphne Bavelier. "Action Video Game Modifies Visual Selective Attention." *Nature*, vol. 423, May 29, 2003. *www.nature.com/nature/journal/v423/n6939/pdf/nature01647.pdf*

Griffiths, Peter. "Britain cracked WW2 spies' secret 'dress code'." London, UK: Reuters, Sept 3, 2006, http://news.yahoo.com/s/nm/britain_spies_dc.

Gwinn, Eric. "Army Targets Youth With Video Game." (Col Casey Wardynski quoted.) *Chicago Tribune*. Nov 7, 2003. *www.notinourname.net/resources_links/video-game-7nov03.htm*

Hammack, Lonnie and E. L. Perry. "The Accelerated Combat Timeline." Conference paper, I/ITSEC 2001.

Hanley, John. "Rapid Spiral Transformation." *Transformation Trends*. Washington, DC: Office of Force Transformation, DoD, Feb 3, 2003. *www.oft.osd.mil/library/library_files trends_172_Transformation %20Trends--3%20February%20Issue.pdf*

Hong Kong Wireless Development Centre, "¡E 3G Cyberport Treasure Hunt," press release, 2006, www.hkwdc.org/3g_application.html.

Institute for Defense Analyses. *DMSO Historical Standards Characterization and Costs Status*. Alexandria, VA: IDA, Oct 2005.

Irlam, Gordon and Ross Williams. "Software Patents: An Industry at Risk." Cambridge, MA: League for Programming Freedom, 1994. lpf.ai.mit.edu/Patents/industry-at-risk.html

Jana, Reena, "Getting Activist Video Games to Market," *Business Week Online*, Jun 30, 2006, *www.businessweek.com/innovate/content/jun2006/id20060630_662842.htm.*

Joint Chiefs of Staff. "Catalog of Wargaming and Military Simulation Models." 12th ed. Washington, DC: The Joint Staff, DoD, 1992, p. ii.

Juniper Research. "Research and Markets: Research Has Lead To Estimates That The Mobile Games Market In 2004 Was Just Above $3bn, With A Sharp Rise Forecast For The Next Five Years." As reported in *Business Wire*, Research and Markets, Feb 15, 2005.

Katz, Warren. "Industry Perspectives on M&S Reuse." Simulation Interoperability Standards Organization, MAK Technologies, Inc. Jan 26, 2000. www.amso.army.mil/smart/conf/2000/day-1/katz.ppt

Kelly, Henry. Invitation to Summit on Educational Games. Federation of American Scientists, Oct 2005. *www.fas.org/gamesummit/reading/invite.pdf*

LaVallee, Andrew, "Now, Virtual Fashion," *The Wall Street Journal*, Sep 22, 2006, p. B1. *http://online.wsj.com/public/article/SB115888412923570768-HtFYrBweWpF25yJkL0CdXvkFRkY_20070922.html*

Maguire, Flack, Michael van Lent, Marc Prensky and Ron W. Tarr. "Defense Combat Sim Olympics —Methodologies Employing the 'Cyber Gaming Culture.'" 2002. *www.marcprensky.com/writing/*

McConnell, Steve. *Rapid Development Taming Wild Software Schedules*. Redmond, WA: Microsoft Press, 1996.

McDowell, Perry et al. "Delta3D: A Revolution in the Methods of Building Training Simulations," MOVES Institute, Naval Postgraduate School, Monterey, CA, 2005. *www.siaa.asn.au/get/2411856047.pdf*

McDowell, Perry, (executive director, MOVES Institute, Naval Post Graduate School), private conversation with authors, Sept 19, 2005.

McGee, Harvey, Telemedicine and Advanced Technology Research Center, email to authors, Oct 11, 2005.

Melissinos, Chris. DataCenter News, Article 12480, vol. 73, no. 3, Mar 15, 2000.

Microsoft, Inc., "Microsoft Invites the World to Create its own Xbox 360 Console Games for the First Time." Press Release, Seattle, WA: Microsoft, Aug 13, 2006. *www.microsoft.com/presspass/press/2006/aug06/08-13XNAGameStudioPR.mspx*

———. XNA FAQ, *www.microsoft.com/xna/faq.aspx*.

Mirza, Yahya, H. and Henry da Costa. "Introducing the New Managed Direct3D Graphics API in the .NET Framework." *MSDN Magazine*, vol. 18, no. 7, 2003.

Morgan, Gary C. "Developing Air and Space Leaders, Shaping the Air Force Future." Presentation. Maxwell AFB, AL: US Air Force, 2005, *www.jointadlcolab.org/newsandevents/ifests/2005/*.

Musgrove, Mike. "Video Game World Gives Peace a Chance." (Castronova quoted.) *The Washington Post*, Oct 16, 2005. *www.washingtonpost.com/wp-dyn/content/article/2005/10/15/AR2005101500218.html?referrer=email*

National Industrial Security Program, *www.archives.gov/isoo/oversight-groups/nisp/*. For defense-specific regulations, see also The National Industrial Security Program at Defense Security Service (DSS), *www.archives.gov/isoo/oversight-groups/nisp/dss.html*.

National Research Council. *Modeling and Simulation in Manufacturing and Defense Acquisition: Pathways to Success*. Washington, DC: The National Academies Press, 2002, p. 12.

Newitz, Annalee, "Your Second Life is Ready," Popsci.com, Sep 2006, *www.popsci.com/popsci/technology/7ba1af8f3812d010vgnvcm1000004eecbccdrcrd.html.*

NewsMax.com, "Iran Video Game: Sink U.S. Oil Tankers," NewsMax.com, Oct 1, 2006, *http://newsmax.com/archives/ic/2006/10/1/160833.shtml.*

Northwest Regional Educational Laboratory. "Focus on Effectiveness: Research-Based Strategies." Northwest Educational Training Consortium, 2005. *www.netc.org/focus/strategies/simu.php*

Onley, Dawn S., "NSA to take the lead on DoD info assurance" InfoSec News, Dec 15, 2004. *http://lists.jammed.com/ISN/2004/12/0053.html*

Open Mobile Alliance, About Open Mobile Alliance, 2006, www.openmobilealliance.org.

Parks Associates, press release, Jun 29, 2006, www.parksassociates.com/press/press_releases/2006.

Paul, Christopher, et al. *Implementing and Evaluating an Innovative Approach to Simulation Training Acquisitions.* Rand National Defense Research Institute, 2006. *www.rand.org/pubs/monographs/2006/RAND_MG442.pdf*

People's Daily Online. "China to invest US$1.8 billion to develop online games." Jul 31, 2005. http://english.peopledaily.com.cn/200507/31/eng20050731_199405.html

Perry, William J. SECDEF Memo "Specifications & Standards—A New Way of Doing Business." DTD 29 Jun 94, https://acc.dau.mil/CommunityBrowser.aspx?id=32397.

Prasithsangaree, P. et al. "UTSAF: A Simulation Bridge between OneSAF and the Unreal Game Engine." Pittsburgh, PA: University of Pittsburgh, copyright IEEE, 2003.

Prensky, Marc, "Innovations in eLearning," presentation, Jun 7, 2006, *http://view.dau.mil/dauvideo/view/viewEvent.jhtml;jsessionid=ECCZNLKB0OUYRAF4VLHSFEQ?contentid=2198&eventid=1045.*

PricewaterhouseCoopers. "Global Entertainment and Media Outlook 2005-2009." New York, NY: Pricewaterhouse Coopers, 2005.

Reuters, "Iranian Video Game Offers Chance to Blow Up U.S. Tanker," as reported in *The New York Times*, Sep 30, 2006.

Saunders, Gregory E., Director, DSP Office, DLA, testimony before US House of Representatives Committee of Science, Subcommittee on Technology, Washington, DC, Mar 15, 2000, www.house.gov/science/saunders_031500.htm.

Sawyer, Ben, private communication with the authors, 2006.

Scacchi, Walt, "Understanding the Requirements for Developing Open Source Software Systems" *IEE Proceedings—Software*, Feb 2002, 149(1), p. 24–39. *www1.ics.uci.edu/~wscacchi/Papers/New/Understanding-OS-Requirements.pdf*

Schiesel, Seth. "An Online Game, Made in America, Seizes the Globe." *NY Times,* Sept 5, 2006.

Shilling, Russell, presentation "Game Technology is Transforming Military Medicine: Is the Inverse Also Possible?" Games for Health Conference. May 2006. *www.gamesforhealth.org/archives/000135.html*

Singapore Economic Development Board, press release, Dec 5, 2005, www.sedb.com/edb/sg/en_uk/ index/news_room/news/2005/edb_will_earmark_s.html.

SpaceDaily.com, "Sandia To Develop Intel Rad-Hard Chips," press release, Dec 9, 1998, *www.spacedaily.com/news/radiation-98c.html.*

Squire, Kurt. "Changing the Game: What Happens When Video Games Enter the Classroom?" *Innovate*, Aug/Sept 2005, vol. 1, no. 6. *www.innovateonline.info/index.php?view=article&id=82*

Stanford University, "Folding@Home FAQ," Oct 22, 2006, *http://folding.stanford.edu/FAQ-PS3.html.*

Steinman, Jeffrey S. and Douglas R. Hardy. "Evolution of the Standard Simulation Architecture." 2004 CCRT Symposium, US Department of Defense Command and Control Research Program. www.dodccrp.org/events/2004/CCRTS_San_Diego/CD/papers/067.pdf

Telephia, "Mobile Game Report Q1 2006," www.telephia.com.

Tochen, Dan. "Serious Games in the Services: Army vs. Navy." (Belanich quoted.) *GameSpot*, Oct 31, 2005. *www.gamespot.com/news/6136900.html*

———. "Serious Games in the Services: Army vs. Navy." (Perla quoted), *GameSpot*, Oct 31, 2005. *www.gamespot.com/news/6136900.html*

Toyon Research Corp., "Introducing the SLAMEM/GVS User Group," 2006. www.toyon.com/slamem_gvs.asp.

Trott, Kevin. "Command, Control, Communications, Computer, Intelligence, Surveillance and Reconnaissance (C4ISR) Modeling and Simulation Using Joint Semi-Automated Forces (JSAF)." Rome, NY: Northrop Grumman Inc., 2003.

US Army. "Close Combat Tactical Trainer," www.fas.org/man/dod-101/sys/land/wsh/70.pdf.

US Department of Defense. "Defense Standardization Program, Policies and Procedures (DSP)." Washington, DC: Office of the Under Secretary of Defense (Acquisition, Technology and Logistics), Mar 2000. www.dsp.dla.mil/policy/docs/4120-24m.pdf.

———. *DoD Dictionary of Military Terms*. Washington, DC: Joint Doctrine Division, Aug 2006, www.dtic.mil/doctrine/jel/doddict/.

———. DoD Directive 5000.59 "DoD Modeling and Simulation (M&S) Management," Jan 4, 1994. www.dtic.mil/whs/directives/corres/html/500059.htm

———. DoD Instruction 5000.1 "Operation of the Defense Acquisition System" Paragraph 4.2. May 12, 2003.

———. DoD Instruction 5000.2 "Operation of the Defense Acquisition System" Section 3. May 12, 2003.

———. *Intellectual Property: Navigating Through Commercial Waters*. Washington, DC: Office of the Under Secretary of Defense for Acquisition, Technology and Logistics, Oct 15, 2001. www.acq.osd.mil/dpap/Docs/intelprop.pdf

US Small Business Administration, "How the Government Buys." *www.sba.gov/businessop/basics/buys.html.*

van Eck, Richard. "Digital Game-Based Learning: It's Not Just the Digital Natives Who Are Restless." *EduCause Review*, Mar/Apr 2006, vol. 41, no. 2, pp. 16–30. *www.educause.edu/apps/er/erm06/erm0620.asp?bhcp=1*

van Genuchten, Michiel. "Why is Software Late? An Empirical Study of Reasons for Delay in Software Development." Los Alamitos, CA: IEEE Computer Society, 1991, pp. 143–152.

Verdesca, Marlo, et al. "*Using Graphics Processor Units to Accelerate OneSAF: A Case Study in Technology Transition.*" I/ITSEC 2005, *www.iitsec.org/documents/R_2121_000.pdf.*

VirtualHeroes_answers_AAR_Questions. Unpublished paper from Virtual Heroes, Inc., 2006.

Whiting, Jason, "Online Game Economies Get Real," *Wired*, Nov 6, 2002, *www.wired.com/news/games/0,2101,55982,00.html.*

Wikipedia, entry for CyborgLog, updated June 2006, *http://en.wikipedia.org/wiki/CyborgLog.*

Wilson, Bret, press release, "Sandia National Laboratories," US Army, America's Army Information Site, Jul 26, 2005, http://info.americasarmy.com/press.php?id=1.

Young, M., P. G. Schrader, and D. Zheng. "MMOGs as Learning Environments: An Ecological Journey into Quest Atlantis and The Sims Online." *Innovate,* Apr/May 2006, vol. 2, no. 4. *www.innovateonline.info/index.php?view=article&id=66*

Zittel, Randy C. "The Reality of Simulation-Based Acquisition." *Acquisition Review Quarterly*, Spring-Summer 2001. www.findarticles.com/p/articles/mi_m0JZX/is_2_8/ai_81763216/

Zweben, Stuart. "2004-2005 Taulbee Survey: Ph.D. Production at an All-Time High with More New Graduates Going Abroad..." *Computing Research News*, vol. 18, no. 3, 2006.

Appendix C: Bibliography

For further reading.

America's Army, *http://info.americasarmy.com/*.

Carter, Ashton B. and John P. White, eds. *Keeping the Edge: Managing Defense for the Future*. Cambridge, MA: Preventive Defense Project, Sept 2000.

Chew, Jennifer and Cindy Sullivan, "Verification, Validation, and Accreditation in the Life Cycle of Models and Simulations," Proceedings of the 2000 Winter Simulation Conference, J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, eds., 1998. *www.informs-cs.org/wsc00papers/107.PDF*

COLLADA, *collada.org/public_forum/welcome.php*.

Croquet, www.opencroquet.org/.

Defense Modeling and Simulation Office, "VV&A Recommended Practices Guide Glossary," Oct 15, 2001. http://*vva.dmso.mil/Glossary/Glossary-pr.pdf*

Defense Security Service, *www.dss.mil/infoas/index.htm*.

Defense Standardization Program Office, *www.dsp.dla.mil/documents/sds.htm*.

DoD Instruction 5000.61, *www.dtic.mil/whs/directives/corres/pdf/i500061_051303/i500061p.pdf*

Entertainment Software Association, Essential Facts about the Computer and Video Game Industry, 2006. *www.theesa.com/archives/files/Essential%20Facts%202006.pdf*

Federal Acquisition Regulation (FAR), *www.acquisition.gov/far/*.

Games for Change, *www.gamesforchange.org/conference/2006/expo.asp*.

Games for Health, *www.gamesforhealth.org*.

Heppell, Stephen, et al, "Unlimited learning - Computer and video games in the learning landscape," ELSPA (Entertainment and Leisure Software Publishers Association), London, Sept 2006. *www.Elspa.com*

JSAF, *www.cs.odu.edu/~tang_j/dtic/pdf/ADA417477.pdf*.

Kilikauskas, Michelle L. and David H. Hall. "Cost-Effective VV&A: Get Credit For What You're Already Doing" Belcamp, MD: SURVICE Engineering Co., 2006. *www.survice.com/SIPapers/CostEffectiveVV&A.pdf*

Lippitz, Michael J., Sean O'Keefe, and John P. White with John Brown. "Advancing the Revolution in Business Affairs," Chapter 7 in *Keeping the Edge: Managing Defense for the Future*, Cambridge, MA: Preventive Defense Project, Sept 2000, pp. 163–202.

Microsoft, *www.microsoft.com/xna/faq.aspx*.

MMOGChart.com, *www.mmogchart.com/*.

ModSAF, *www.aiai.ed.ac.uk/~arpi/SUO/MODULES/modsaf.html*.

OneSAF, *www.onesaf.org/public1saf.html; Delta3D www.delta3d.org/*.

Open Mobile Alliance, *www.openmobilealliance.org*.

Pace, Dale K. "Modeling and Simulation Verification and Validation Challenges." Johns Hopkins APL Technical Digest, vol. 25, no. 2, 2004. *www.jhuapl.edu/techdigest/td2502/Pace.pdf*

PEW Center on Global Climate Change, "Lessons Learned from U.S. Technology and Innovation Policies." Arlington, VA. *www.pewclimate.org/policy_center/policy_reports_and_analysis/technology/lessons.cfm*

Prensky, Marc, "Digital Natives, Digital Immigrants," in *On the Horizon,* NCB University Press, vol. 9 no. 5, Oct 2001.

Second Life, "Second Life Features and Technology," *http://secondlife.com/developers/features.php*.

Silverman, Barry G., et al., "Athena's Prism—A Diplomatic Strategy Role Playing Simulation for Generating Ideas and Exploring Alternatives," University of Pennsylvania under a task order from the Institute for Defense Analyses, 2004.

SISO, *www.sisostds.org.*

The Economic Espionage Act of 1996 (18 USC 1831-39), *http://rf-web.tamu.edu/security/secguide/S2unclas/Propriet.htm*.

US National Security Agency, Information Assurance Directorate, *www.nsa.gov/ia/government/index.cfm.*

Williams, Samuel, et al, "The Potential of the Cell Processor for Scientific Computing," Berkeley, CA: Association for Computing Machinery, 2005.

Appendix D: Glossary

Acquisitions—An M&S genre within DoD that may include the processes of developing concepts for new systems, assessing effectiveness in the field, designing and manufacturing, and training in use.

After Action Review—A facilitated assessment via discussion conducted after a project or major event that allows participants and leaders to learn what happened and why.

Alpha, Alpha Testing—Alpha is an early stage of product development. Alpha testing is generally geared towards resolving gameplay issues.

Analysis—An M&S genre within DoD that covers the tactical and operational levels, but also the strategic level for certain kinds of tasks, including intelligence work. Analysis M&S focuses on helping users understand functionality of systems, reasons for particular outcomes, and other data pertinent to DoD missions.

Beta, Beta Testing—Beta is a late stage of product development, when the game is nearly complete. Beta testing generally focuses on finding and fixing bugs.

BAA—*Broad Agency Announcement*. General statement about a technology requirement that science and technology developers within the government use to convey their needs to industry, and published to solicit bids.

Build—(Noun) The current version of the game. (Verb) To assemble all subcomponents of the game into a working version.

CODEC—*Coder-Decoder*. Compression format typically used on audio and video files. Files are compressed with a certain codec when they are saved and then decompressed by the codec when played back. Common codecs for video files include MPEG and AVI, and WAV and AIFF for audio files.

COGS—*Cost of Goods*. The cost to create all the physical objects that go into the game box, including the box itself, the CD or DVD disc, the manual, the jewel case, and so on.

Continuing Resolution—Measure passed by Congress to fund the government at current levels when approval of the budget is delayed.

CRPG—*Computer Role-Playing Game*. See *RPG*.

Cut Scene—A pre-rendered scene, usually shown between rounds of gameplay, that is designed to move the plot forward.

Defense Acquisition Management Framework—A DoD process structure that details the interaction between capabilities development, acquisition management, and the planning, programming, budgeting, and execution process.

Developer—(1) A company with whom a publisher contracts to create the software for a game; (2) An individual programmer, also known as a *coder*.

DirectMusic—A music delivery system developed by Microsoft for the PC.

End Cap—The display space at the end of the aisle in a retail store.

Experimentation—An M&S genre within DoD that includes the development, exploration, and assessment of new Joint Concepts, organizational structures, and emerging technologies through a process of discovery, innovation, adaptation and integration.

Federal Acquisition Regulation—*FAR*. The procedures that government contracting officials follow to acquire supplies and services, and hence that contractors must accommodate when supplying those to goods and services.

FMV—*Full Motion Video*. Filmed segments that are inserted into a game.

High Concept—The one- or two-sentence response to the question, What is your game about?

HUD—*Heads-up Display*. A portion of the screen that supplies crucial game-related information to the player.

Human-in-the-Loop—Term used to identify the presence of an active human component in a simulation or model; human-in-the-loop affects a simulation's outcome unpredictably and hence inhibits its reproducibility.

Ilities—An expression used to refer to the grouping of those non-functional attributes of models and sims, typically ending in the letters "ility." There are at least as many as 50 expressions, such as composibility, reusability, adaptability, etc. They have varying importance and priority based on the intended use of the model or sim.

IP—*Intellectual Property* (1) All the ideas, code, art, and other material your company develops. (2) Shorthand for a franchise or brand you license to or from another company.

Localization—The process of creating foreign-language versions of a game. The term covers a broad range of activities, including translating text, writing subtitles, dubbing voices, altering content that is deemed unsuitable for some markets, and creating new content altogether.

Long Tail—A statistical term used to describe the economic business model of online retailers, whose lack of physical inventory allows them to "stock" more items than bricks-and-mortar stores. This phenomenon, when graphed, shows a "long tail" of low-selling products that, taken together, add up to more sales than the smaller number of "best-selling" items stocked by traditional retailers.

MIDI—*Musical Instrument Digital Interface*. A standard that allows a composer to store and play music from data files rather than from recordings.

MMO, MMOG, MMP, MMORPG—Any acronym beginning with MM will be "Massively Multiplayer." The "O" will stand for "Online." The "G" will stand for Game. The "P" will be some variant of the word "Play" or "Player"

MOD—Short for *modification*. A version of a popular game that has been changed or added to by the amateur gaming community.

Model—A representation of an object or event in the real world. A model contains the model creators understanding, abstraction and assumptions of the phenomena and, as a result are necessarily incomplete. Models can allow complex systems and

behaviors to be understood within the scope of the model, but may give incorrect descriptions and predictions for situations outside the realm of their intended use. A model may be used as the basis for simulation.

Motion Capture—A studio process whereby an actor's movements are digitally captured and transferred to a model in an animation program.

MPEG— *Moving Picture Experts Group*. A video compression scheme that comes in two flavors, MPEG-1 and the higher-resolution MPEG-2.

MP3—Short for *MPEG-3* (*Moving Picture Experts Group Layer-3 Audio*). A scheme to compress audio for quick transmission and easy playback.

NDA—*Non-disclosure Agreement*. A document whereby someone agrees not to reveal a company's trade secrets or other confidential information.

Net—(1) The Internet, also known as the *Web*. (2) A local area network (LAN) within an office that connects the workers. A central depository for team and company information.

Non-compete—An agreement prohibiting an employee from working for another game company while working at his current company, and sometimes for a specific period of time thereafter.

NPC—*Non-player Character*. Any character appearing in a game that is not controlled by the gamer.

OEM—*Original Equipment Manufacturer*. Usually, a computer maker or a peripheral manufacturer who is interested in bundling your game with his hardware.

Offshoring—Relocating business processes (i.e., production, manufacturing, or services) from one country to another, typically done to lower total costs and possibly to smooth production cycles. For example, someone in Croatia can be debugging code while workers in the US sleep and visa versa.

Ogg Vorbis—A fully open, non-proprietary, patent-and-royalty-free, general-purpose compressed audio format for mid- to high-quality (8kHz-48.0kHz, 16+ bit, polyphonic) audio and music at fixed and variable bitrates from 16–128 kbps/channel.

One-year money—A duration of government funding that specifies how long a contractor has to spend the money. Almost all other funding is one-year except RDT&E money which can be expensed over a longer three-year period.

PPBE—*Planning, Programming, Budget Execution*. The process DoD uses to assemble its spending requests and set it's long-term investment goals. The PPBS is the system under which it is implemented.

Planning—M&S genre used to determine the size and composition of a military force and to learn how to plan the missions of military forces.

Port—A game version created for a different hardware platform than the original. Also called a *conversion*. (Verb) To create such a conversion: "They ported the game from the playstation to the PC."

Price Protection—The lowering of a game's wholesale price. Usually, this comes in the form of a credit to the retailer for units he has on the shelves but hasn't sold

through. The markdown is taken as a game's rate of sales slows, to encourage the retailer to keep the game in stock rather than return it to the publisher.

QWERTY keyboard—Term derived from the first six letters in the top alphabet row of a keyboard. It is also called the "universal" keyboard. Originally appearing in 1878 under the patent held by C.L. Sholes, it is still the keyboard arrangement that appears with most computers.

RedBook Audio—A fancy name for the digital standard developed by Phillips and Sony to record regular CDs that go in your stereo. So called because the original specification was in a book with a red binder.

Research—A broad and varied M&S genre, typically application based, that may include tools used to consider new problems and gather more information about old problems.

RFP—*Request for Proposal*. Government document that identifies and specifies a needed product or service and invites industry contractors to explain how they would fulfill that need and at what price.

ROI—*Return on Investment*. An estimate of how much money the game will make, usually expressed as a percentage of income to expense. This number is derived from the numbers on the P&L statement.

RPG—*Role-Playing Game*. A genre in which the player directs a group of heroes on a series of quests, usually in a story-based environment.

RTS—*Real-Time Strategy*. A genre of games played in real time (as opposed to turn-based) in which the emphasis is on managing a limited set of resources to achieve a goal.

Security Clearance—As it relates to the DoD, an administrative determination by a competent authority that an individual is eligible, from a security stand-point, to access various levels of classified information.

Sell-through—The number of units that are actually sold at retail.

Simulation—The technique of representing the real world by a computer program or virtual environment by imitating not only the results of the thing being simulated, but also the internal processes involved.

Specifications—As relating to the DoD, guidelines used to describe the physical and/or operational characteristics of a product, differentiated from *Standards*.

Spiral development—A development model originally conceptualized in the mid-1980s as a way to reduce risk on large software projects. It includes recurring feedback and revision cycles during development and has evolved slightly differently based on whether applied to commercial game or government development.

Standards—As relating to the DoD, detail the processes, materials and configurations to be used to make a product, differentiated from *Specifications*.

Storyboard—A sequence of pencil sketches that rough out what a scene will look like; to create the sequence.

Studio—(1) An independent development house (or *developer*) that develops game software. (2) A division of a large company that acts as a semi-autonomous unit to de-

velop games. (3) A soundproof room for recording actors' voices, also known as a *voice studio*. (4) An interior location for filming.

Testing—An M&S genre that includes those models and sims used to try out and evaluate anything from a vehicle to a fighter jet to countermeasures of attacks using weapons of mass destruction. Usually associated with some form of analysis.

Training—M&S genre that includes any number of levels from the tactical to operational and strategic that focuses on instilling a certain level of competency regarding a particular skill set.

VGA—*Video Graphics Array*. Analog graphics standard introduced with the IBM PS/2 series. Backwards compatible with EGA at the BIOS level, but provides higher resolutions. Supports a maximum resolution of 640 x 480 pixels in 16 colors out of a palette of 262,144 colors.

Wargame—A simulation game where participants seek to achieve a specified military objective given pre-established resources and constraints; for example, a simulation in which participants make battlefield decisions and a computer determines the results of those decisions.

Waterfall—Development model historically used by the government where all end requirements are specified at the beginning and rigid boundaries are between each development stage.

WYSIWYG—*What You See Is What You Get*. Any interface that allows you to see what material will look like on the computer screen while you are creating it.

Appendix E: MMOG Engine Information

Tables E-1 and E-2 follow.

Table E-1. MMOG Engines

Engine Name	Short Description	Company Name	Location	Contact	URL	Cost
Full MMOG Engines						
Forterra	There.com engine for government uses	Forterra Systems Inc.	San Mateo, California	650-433-4000 business@forterrainc.com	www.forterrainc.com	email to normal business contact
There.com	---	Makena Technologies Inc.	San Mateo, California	650-433-4040 opportunities@thereinc.com	www.there.com	N/A, see Forterra
BigWorld	Relatively complete solution; licensed to DoD through Total Immersion.	BigWorld Pty Ltd	Sydney, Australia	+61-2-8570-5400 bw_licensing@bigworldtech.com	www.bigworldtech.com	---
Emergent (formerly Butterfly.net)	Full tech and service provider	Emergent Game Technologies	Glendale, California	818-291-6304	www.emergentgametech.com	---
Kaneva	Full tech and service provider	Klaus Entertainment Inc.	Atlanta, Georgia	770-551-2760	www.klausentertainment.com	free, pay for hosting, split the profits
MM-Suite	Owned by Square	Community Engine Inc.	Tokyo, Japan	---	www.celab.net	---
Horizons	---	Tulga Games LLC	Tempe, Arizona	davidb@tulgagames.com	www.tulgagames.com	
NEL	Much is open source	Nevrax	France	---	www.nevrax.org/tiki-index.php	€ 500k negotiable, no royalty
Potential MMOG Engines						
Torque	Limited MMOG capabilities	Garage Games	Eugene, Oregon	541-345-3040	www.garagegames.com	---
Second Life	---	Linden Lab	San Francisco, California	415-243-9000 /contact@lindenlab.com	secondlife.com	does not license, only sells land in current game
Q Technology	Not an MMOG, but a game engine	Qube Software	London, UK	info@qubesoft.com	www.qubesoft.com	ranges from £16k for PC to £150k for source
RealmForge	For small-scale client/server; really an API	---	---	---	source-forge.net/projects/realmforge	---
Leviathon	Server engine	---	---	---	www.climaxgroup.com/technology	---

RealmCrafter	Amateur MMO builder	---	---	---	www.realmcrafter.com	---
Parts of MMOG Engines						
Terraplay	Wireless MMOG Engine	Terraplay AB	Solna, Sweden	+46-8-764-9100	www.terraplay.com	---
Net Z and Eterna	All but Eterna is a peer-to-peer solu-tion; no MMOGs	Quazal	Montreal, Canada	+1-514-395-4646	www.quazal.com	---
SIMinterNET	All military solutions	MÄK Technologies	Cambridge, Massachusetts	617-876-8085	www.mak.com	---
Twisted	Written in Python	Twisted Matrix Labs	---	---	www.twistedmatrix.com	---
Stream Engine	No MMOGs yet, no code	---	---	---	source-forge.net/projects/streamengine	---
Open Skies	---	Cybernet Systems	Ann Arbor, Michigan	---	www.openskies.net	---
TeraZona	---	Zona Inc.	Santa Clara, California	408-844-9646	www.zona.net/products/datasheet .html	email to normal business contact

Table E-2. MMOG Features

	Forterra	BigWorld	Kaneva	Horizons	NEL	X-SRC
Platform	PC/Linux	PC, XBOX	PC/?	PC, Windows servers	PC/Linux	PC
Licensing	varying	multi-component	$50k	varying	closed license with fee	---
Programming language	C++	C++	C++	C++	C++	C++
Scripting language support	yes, proprietary	yes	yes	yes	some	yes
Sample application	yes	yes	yes	yes	no	yes
Shipped product	yes	no	no	yes, Horizons	yes	yes
Auto Build	yes	no	yes	no	yes	
Testing harness	yes	no	no	no	no	no
Metrics collection framework	yes	server performance	yes	no	no	---
System-wide update deployment	yes	yes	yes	no	yes	
Customer service tools	yes, open source	yes	no	no	some	some
Backup system	yes	flexible	yes	yes	yes	yes
Server framework	yes, seamless	yes, seamless	yes, zoned	yes	yes, seamless	yes, zoned
Network layer API	yes	yes	yes	yes	yes	yes
CPU allocation	yes	flexible	no	yes	depends	yes
Persistence framework	yes	yes	yes, but	yes	yes	yes
Chat	yes	yes	yes	yes	yes	yes
Server monitoring	yes	yes	yes	limited	yes	yes
Patching system	yes	yes	yes	yes	yes	yes
Incremental patching	yes	yes	yes	yes	yes	yes
Registration/login server	yes	yes	yes	yes	yes	yes
Subscription billing	yes	no	yes	no	no	yes
Asian billing	no	no	no	no	no	no
Per item billing	some	no	no	no	no	yes
Logging system	yes	yes	yes	yes	yes	yes
AI framework	no	yes	yes	yes	no	yes
Server collision/verification	yes	yes	yes	yes	yes	---

Security services	yes	some	yes	yes	no	no
Bandwidth control/throttling	yes	yes	yes	yes	yes	
In-game systems framework	yes	yes	yes	no	no	yes
NPC creation/programming	yes	yes	yes	no	no	yes
Object management system	yes	yes	yes	no	no	---
Quest/content tools	yes	no	yes	no	no	yes
UI tools	no	some	yes	no	no	yes
Worldbuilding/art pipeline tools	yes, internal tool	yes	yes	yes	yes	some
3D exporter	yes	yes	yes	yes	yes	yes
Client framework	yes	yes	yes	yes	yes	yes
Terrain	yes	yes	yes	yes	yes	yes
Renderer	yes	yes	yes	yes	yes	yes
Collision/movement system	yes	yes	yes	yes	yes	yes
Path finding	no	yes	yes	some	some	yes
Other physics (ragdoll, IK)	yes	no	no	no	no	some
Animation system	yes	yes	yes	yes	yes	yes
Particle system/editor	maybe	yes	yes	yes	yes	no
Character customization	yes	---	no	no	no	yes
LOD system	yes	yes	yes	yes	yes	yes
Internationalization (unicode support)	yes	yes	yes	yes	yes	yes
Visibility culling system	some	yes	yes	yes	yes	yes

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE (DD-MM-YY)	2. REPORT TYPE	3. DATES COVERED (From – To)
July 2006	Study (Draft Final)	

4. TITLE AND SUBTITLE	
MS&G, When Worlds Collide: A Primer for Potential	5a. CONTRACT NO. DASW01-04-C-0003 and W74V8H-05-C-0042
	5b. GRANT NO.
	5c. PROGRAM ELEMENT NO(S).
6. AUTHOR(S) Brian Williams, task leader; Bob Bates, Tara McGovern, Terry Heuring, Gordon Walton	5d. PROJECT NO.
	5e. TASK NO. AK-8-246201
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Joint Advanced Warfighting Program – Institute for Defense Analyses, 4850 Mark Center Drive, Alexandria, VA 22311-1882	8. PERFORMING ORGANIZATION REPORT NO. IDA Document D-3335
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Defense Modeling and Simulation Office, 1901 N. Beauregard Street, Suite 500, Alexandria, Virginia, 22311-1705, USA	10. SPONSOR'S / MONITOR'S ACRONYM(S) DMSO
	11. SPONSOR'S / MONITOR'S REPORT NO(S).

12. DISTRIBUTION / AVAILABILITY STATEMENT

13. SUPPLEMENTARY NOTES

Unclassified

14. ABSTRACT

Commercial game companies operate in an extremely competitive environment, creating products on the bleeding edge of technology. Companies survive by evolving streamlined processes that allow them to efficiently develop these complex products and to deliver them to market quickly. What happens when the culture and business practices of the commercial game world collide with highly regulated procedures of the government, especially those of the military? The result is a challenging—but not impossible—landscape that both sides must traverse to find common ground. These two seemingly similar communities of interest (DoD M&S and Commercial Gaming, (MS&G)) have evolved differently due to the customers' needs and the market dynamics that surround them, resulting in philosophical and operational differences that must be understood for the communities to work effectively together.

This document attempts to provide a basic understanding of the history, technology, product development processes, and business practices of the game industry and DoD/M&S. Furthermore, it lays out those areas of conflict and fusion while proposing some concepts and recommendations to help these two groups build sustainable, successful working relationships in the pursuit of serious games.

15. SUBJECT TERMS

Serious Games, DoD and Gaming, MS&G, Modeling, Simulation and Games, Serious games in DoD, Gme history, M&S History, When worlds collide, Persuasion games, collaborative virtual spaces, Government sponsored game engines, MMOG game engines, massive multiplayer on-line games, serious games marketplace,

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NO. OF PAGES	19a. NAME OF RESPONSIBLE PERSON
			UU	269	COL Jerry A. Glasow
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER (Include Area Code)
U	U	U			703-824-3456