

**A STUDY ON CREATING GAMES AND VIRTUAL WORLDS FROM A  
SOFTWARE ENGINEERING PERSPECTIVE**

**MARK TEE KIT TSUN**

A Master's Project submitted in partial fulfilment of the requirements for  
the degree of Master of Software Engineering

Centre for Graduate Studies  
Open University Malaysia

2011

# **A STUDY ON CREATING GAMES AND VIRTUAL WORLDS FROM A SOFTWARE ENGINEERING PERSPECTIVE**

**MARK TEE KIT TSUN**

**JAN 2011**

## **ABSTRACT**

The industry of developing games and virtual environments have come a long way, but it is not without its share of problems. In many ways, virtual worlds entail similar development processes as games, as they both require expertise from creative and technical facets. While facing similar difficulties as other commercial projects, creating games presents its own unique challenges due to its multi-disciplinary nature. Because of this, beginner and indie developers face complications in attempting to apply Software Engineering methods (which are efficiently tailored to guide general software projects) to reduce the chance of project failure. This study aimed to examine the construction of games and integrate it into the Software Process. First, the background of game development and Software Engineering were explored, deriving the introductory guides for both. Then, a guidelines draft is developed to offer advice on grafting aspects of the Software Process onto existing game or virtual world projects. They are aimed at independent developers as they represent the future workforce of the game development industry. To test the draft, an experimental project is carried out using the guide as a supplementary resource. The project resulted in a 3D action-RPG game platform that can be used as a starting point for advanced feature placement and content development. It demonstrated that the guidelines were educative but lacks depth and correctness. Revisions on these guidelines will transform it into a stepping stone resource that introduces novice game developers to integrating Software Engineering processes and practices into their projects.

### **Keywords:**

Game Development, Software Engineering

## TABLE OF CONTENTS

<b>TITLE PAGE</b>	<b>i</b>
<b>DECLARATION</b>	<b>ii</b>
<b>ABSTRACT</b>	<b>iii</b>
<b>ABSTRAK</b>	<b>iv</b>
<b>ACKNOWLEDGEMENTS</b>	<b>v</b>
<b>TABLE OF CONTENTS</b>	<b>vi</b>
<b>LIST OF TABLES</b>	<b>ix</b>
<b>LIST OF FIGURES</b>	<b>x</b>
<b>CHAPTER 1 INTRODUCTION</b>	<b>1</b>
1.1 Background of Study	1
1.2 Problem Statement	2
1.3 Research Question	3
1.4 Research Objective	3
1.5 Scope of Study	3
1.6 Significance of Study	4
<b>CHAPTER 2 LITERATURE REVIEW</b>	<b>6</b>
2.1 History of Computer Game Development	6
2.2 History of Software Engineering	10
2.3 Overview of Parallels between Software Engineering And Game Development	12
2.4 Theoretical Framework	15
<b>CHAPTER 3 RESEARCH METHODOLOGY</b>	<b>22</b>
3.1 Introduction	22
3.2 Phase 1 – Data Gathering	23
3.3 Phase 2 – Analysis	24

3.4	Phase 3 – Development Case Study	25
3.5	Conclusion	31
<b>CHAPTER 4 A GUIDE TO GAME DEVELOPMENT</b>		<b>33</b>
4.1	Introduction	33
4.2	Overview of Game Development	36
4.3	Development Methodology	38
4.4	Game Design	45
4.5	Game Architecture	48
4.6	Game Engine Fundamentals	52
4.7	Content Creation	54
4.8	Conclusion	56
<b>CHAPTER 5 INTRODUCTION TO SOFTWARE ENGINEERING</b>		<b>58</b>
5.1	Introduction	58
5.2	Process Activities	60
5.2.1	Software Specifications	60
5.2.2	Software Development	63
5.2.3	Software Validation	66
5.2.4	Software Maintenance	68
5.3	Development Methodologies	69
5.4	Standards for Software Quality and Process Improvement	74
5.5	Conclusion	75
<b>CHAPTER 6 PROPOSED SE - GAME DEVELOPMENT ADAPTATION</b>		
<b>GUIDE</b>		<b>77</b>
6.1	Introduction	77
6.2	Guide Overview	78
6.3	Software Engineering for Game Development Guidelines	79
6.3.1	Phase 1: Initial Planning	79
6.3.2	Phase 2: Requirements	81
6.3.3	Phase 3: Implementation	86

6.3.4	Phase 4: Validation	91
6.3.5	Phase 5: Maintenance and Evolution	93
6.4	Conclusion	95
<b>CHAPTER 7 PROJECT REPORT, GUIDELINES ASSESSMENT AND</b>		
<b>CONCLUSION</b>		<b>96</b>
7.1	Development Project Plan	96
7.2	Technical Development Process	100
7.3	Software Engineering for Games Guidelines Application	
	Review	104
7.4	Conclusion	107
<b>REFERENCES</b>		<b>108</b>
<b>APPENDICES</b>		<b>110</b>
APPENDIX A	Historical Timeline of Computer Games and Platforms	111
APPENDIX B	Software Engineering Timeline	112
APPENDIX C	Project Proposal	113
APPENDIX D	Project Feasibility Report	116
APPENDIX E	Software Requirements Specifications	118
APPENDIX F	Design Models and Diagrams	127
APPENDIX G	Initial Project Schedule	133
APPENDIX H	Project and Software Manual	134

## LIST OF TABLES

Table 4.1	Example components separated by categories	33
Table 4.2	Some example items declared under Initial Design and Gradual Refinement	46
Table 4.3	Difference between DirectX and OpenGL	53
Table 6.1	Some example tasks enlisted under Creative, Technical and Management	83
Table 7.1	Summarized Developer Log	100
APPENDIX D		
Table 1.1	Feasibility Study Discussion Results	116
APPENDIX E		
Table 1.1	Definitions, acronyms and abbreviations used through the SRS report	118
APPENDIX G		
Table	Initial Project Schedule	133

## LIST OF FIGURES

Figure 2.1	The software process	16
Figure 2.2	Motion game system architecture	18
Figure 2.3	General Game-Engine Based Framework	19
Figure 3.1	Hybrid Waterfall-Prototyping model	28
Figure 3.2	The Prototyping Software Life Cycle Model	30
Figure 3.3	The Prototyping Process in the Hybrid Model	31
Figure 4.1	General Games Development Processes	38
Figure 4.2	The Structured “Water-Fall” Model	41
Figure 4.3	The Spiral Approach	41
Figure 4.4	The Prototyping Approach	42
Figure 4.5	The Scrum Agile Approach	44
Figure 6.1	Game Development Process	78
APPENDIX A		
Figure	Historical Timeline of Computer Games and Platforms	111
APPENDIX B		
Figure	Software Engineering Timeline	112
APPENDIX E		
Figure 2.1	Use Case Diagram showing the classic interface activities found in general action RPG games	121
Figure 2.2	Use Case Diagram showing the classic gameplay activities found in general action RPG games	121
Figure 3.2.1	State Diagram modelling the general states that the game should revolve in, based on reviewing the general structure of commercial game software	124

Figure 3.2.2	Class Diagram describing the conceptual gameplay components as visualized in an object oriented approach	125
--------------	--	-----

## APPENDIX F

Figure 6.1.1	State Diagram showing the Level-0 of the game states	127
Figure 6.1.2	Class Diagram derived from the initial conceptual counterpart	128
Figure 6.2.1	Activity Diagram depicting the back-end of the Main Menu	128
Figure 6.2.2	This set of Activity Diagrams visualize the technical mechanics behind each in-game action that can be performed by the Player Character	129
Figure 6.2.3	The second set of Activity Diagrams for In-Game basic actions cover the Character Panel and Barter Interface	129
Figure 6.3.1	The Sequence Diagram for Main Menu selections	130
Figure 6.3.2	The Sequence Diagram depicting the initial design for Dialog and Barter In-Game actions	131
Figure 6.3.3	The Sequence Diagram for Starting a New Game	131
Figure 6.3.4	The Sequence Diagram for Continuing from a Saved Game	132

## **CHAPTER 1 INTRODUCTION**

### **1.1 Background of Study**

This project studies the lifecycle of games and virtual worlds from inception until construction, highlighting the similarities and contrasts between these and conventional software applications. The difference between virtual worlds and games should be made clear first, as a virtual world application consists of the environmental elements that simulate a particular environment (Jakala & Pekkola, 2007). A game is in fact a virtual world that is populated with multimedia content (usually centred on entertainment since the objective of every game). The degree of such content depends on both the developer and the audience. However, for the sake of this discussion and study, the creation process for both will be referred to as games development. While the creation of computer games is technically situated within the realm of general software development, there may be complications in applying software engineering practices to games. This is because the process of assembling one requires a multi-disciplinary approach that melds not just aspects of software development, but also works of literature, entertainment and in some cases, social studies (Diefenbach, 2008). Schedules, budget allocations, team composition and design vary from project to project, and this poses an added factor that increases the chance of software failure (Rabin, 2005). Traditionally, software engineering practices were honed and matured to the point where the industry has benefitted from it greatly, resulting in more timely delivered and higher quality software (Kotonya, 1998). However, the same cannot be said for the games development industry as the variation and dynamic nature of the

process demands guidelines that cannot be contained by the comprehensive software engineering standards. Regardless, this statement does not deem that existing development guidelines and standards are redundant for the creation of games, but more flexible explanations should be made available for game developers in order to understand and adapt the said standards towards producing more passable products in terms of quality and correctness.

## **1.2 Problem Statement**

The cause of such software failure in games development and virtual world projects can be mostly attributed to incorrect, inadequate or non-existent application of software engineering influences (Sommerville, 2007). The target sample in question refers to 'indie' or individual developers instead of established development studios because a more stable organization will most likely have access to training and specialists who either homebrew their techniques or advocate effective application of software engineering (Maurina, 2006). An individual developer will be hard pressed to find guidelines for effective development other than online resources and general texts. What are generally missing are directories that point to correct references as well as introductory guidelines on best practices. Therefore, the main problem statement is that 'games development project failures stem from incorrect, inadequate application of software engineering practices'. This main problem consists of the following sub-problems:

- General software engineering standards and guidelines are formalized to apply to all general purpose application development, but are not usually flexible enough to accommodate the varying requirements of games, simulators and virtual worlds.

- Individual developers do not have a wide choice of easily accessible information on game development-specific guidelines and software engineering standards.
- Failures still occur even with adequate software engineering resources because there is inadequate number of formal demonstrations on how software engineering practices is adapted to creating games. (Leon, 2004)

### **1.3 Research Question**

In response to the problem statement, the research question for this study is ‘how can software engineering practices be adapted to ‘indie’ games development?’

### **1.4 Research Objective**

The objectives of this study in order to answer the research question, is divided into 3 parts:

- To document the general games development lifecycle in order to set the background on which software engineering practices can be applied
- To study and discern methods to interpret software engineering resources and standards in order to fit portions of the games development lifecycle
- Demonstrate how planning and implementation of adapted software engineering activities can be carried out for a game development project by documenting the process undergone for the construction of a virtual world environment.

### **1.5 Scope of Study**

This study will centre on the process of constructing virtual worlds and games in general as will be undertaken by individual developers instead of software houses. Therefore, the resources outlined therein will be predominantly introductory in depth but adequate for anyone to kick-start a project. The document as well as the

demonstration application will follow the COTS game engine style as this is practiced in general even amongst leading developers. Software engineering standards and techniques that will be covered in the study shall encompass the software development lifecycle, software quality assurance, as well as leading quality assurance standards including CMMI and ISO9001. To satisfy the second objective of this project, a discussion on the adaption of said materials will be done within the scope of general simulation development (excluding entrances to development of graphical, audio and literal content).

The study will not extensively cover every possible method of developing simulators, virtual worlds and games, as the emphasis is on applying software engineering techniques for the purpose of reducing the chance of project failure. Neither will the development of game engines, multimedia content and game design be discussed, although an explanation of each portion in general will be included, as in fulfilment of the first object. Finally, the software development process that will be outlined and applied for the demonstration project will allow variations between several classic methodologies which include the waterfall model, prototyping and Agile development. More recent and popular methodologies such as SCRUM, extreme programming and SAP will not be discussed as each deserves a dedicated branch of the study. The demonstration system will be used a case study for the purpose of illustrating how the software engineering process may be applied in real time. Thus, the end product will be a prototype instead of a full release of an agent simulator environment.

## **1.6 Significance of Study**

Games development has already been accepted as a significant branch of software development, to the point that it has earned dedicated degree programs and specialist

courses throughout the world (Diefenbach, 2008; Goulding & Ditrolio, 2007; Mikami et al., 2009; Ritzhaupt, 2009). Spanning more than just entertainment, it has catered to niche markets that require simulators, virtual reality and projection applications, each demanding more advances as hardware graphics, audio and human interface technology continue to stack up (Jakala & Pekkola, 2007). Even then, resources and guidelines are still rather limited to come by, save for enthusiast websites and largely basic introductory texts that do not break enough grounds to reduce project failures. Software studios have managed to survive only by sheer chance of developing best practices and having a sizeable amount of luck (Keith, 2010). This study will contribute to the rising number of resources that aims to remedy this problem and hopefully assist the games development to achieve higher recognition of significance.

## REFERENCES

Achterbosch, L., Pierce, R. & Simmons, G. (2008). *Massively Multiplayer Online Role-playing Games: The past, present and future*. ACM Comput. Entertain., 5, 4, Article 9, March 2008.

Adams, J. (2004). *Programming Role Playing Games with DirectX* (2<sup>nd</sup> Ed.). Pearson-Course Technology.

Barry, I. (2005). *Game Design. Introduction to Game Development* (1<sup>st</sup> Ed.). Boston, USA: Charles River Media.

Blow, J. (2004). *Game Development: Harder Than You Think*. ACM Press Queue, February 2004.

Diefenbach, P. (2008). *Creating a Multi-Discipline Gaming Curriculum: Avoiding Mistakes, Missteps and Growing Pains*. SIGGRAPH Asia 2008, Singapore.

Flynt, J. (2004). *Software Engineering for Game Developers*. Pearson-Course Technology.

Goulding, T. & Ditrolio, R. (2007). *Complex Game Development by Freshman Computer Science Majors*. Inroads - SIGCSE Bulletin. Vol. 39, No. 4, Publication date: December 2007.

Jakala, M. & Pekkola, S. (2007). *From Technology Engineering to Social Engineering: 15 Years of Research on Virtual Worlds*. The DATABASE for Advances in Information Systems, Vol. 38, No. 4, Publication date: November 2007.

Keith, C. (2010) *Agile Game Development with Scrum*. Boston: Pearson Education Inc.

Kim, G. J., Kang, K. C. & Lee, J. (1998). *Software Engineering of Virtual Worlds*. VRST 1998, November 2-5, Taipei, Taiwan: ACM Inc.

Kotonya, G. & Sommerville, I. (1998). *Requirements Engineering: Processes and Techniques*, West Sussex, England: Wiley.

Leon, A (2004). *Software Configuration Management Handbook* (2<sup>nd</sup> ed.). Norwood: Artech House Inc.

- Llopis, N. (2005). *Game Architecture*. Introduction to Game Development (1<sup>st</sup> Ed.). Boston, USA: Charles River Media.
- Maurina, E. F. (2006). *The Game Programmer's Guide to Torque: Under the Hood of the Torque Game Engine*. Wellesley: AK Peters Ltd.
- Menchaca, R., Balladares, L., Quintero, R. & Carreto, C. (2005). *Software Engineering, HCI Techniques and Java Technologies Joined to Develop Web-Based 3D-Collaborative Virtual Worlds*. CLIHC 2005, October 23-26, Cuernavaca, Mexico: ACM Inc.
- Mikami, K., Watanabe, T., Yamaji, K., Ozawa, K., Ito, A., Kawashima, M., Takeuchi, R., Kondo, K. & Kaneko, M. (2009). *Construction Trial of a Practical Education Curriculum of Game Development by Industry/University Collaboration*. SIGGRAPH Asia 2009, Yokohama, Japan, December 16-19.
- Petrillo, F., Pimenta, M., Trindade, F. & Dietrich, C. (2009). *What Went Wrong? A Survey of Problems in Game Development*. ACM Comput. Entertain., 7, 1, Article 13, February 2009.
- Rabin, S. (2005). *Introduction to Game Development* (1<sup>st</sup> ed). Boston, USA: Charles River Media.
- Ritzhaupt, A. D. (2009). *Creating a Game Development Course with Limited Resources: An Evaluation Study*. ACM Trans. Comput. Educ. 9, 1, Article 3 (March 2009).
- Sheldon, L. (2004). *Character Development and Storytelling for Games* (1<sup>st</sup> ed). Course Technology PTR.
- Stacey, P. & Nandhakumar, J. (2008). *Opening Up to Agile Games Development*. Communications of the ACM. Vol. 51, No.12, Publication date: December 2008.
- Sommerville, I. (2007). *Software Engineering* (8<sup>th</sup> ed.). Harlow, Essex: Addison-Wesley.
- Taylor, M. J. & Baskett, M. (2009). *The Science and Art of Computer Games Development for Undergraduate Students*. ACM Computers in Entertainment. Vol. 7, No. 2, Article 24, Publication date: June 2009, New York: ACM Inc.
- White, W., Koch, C., Gehrke, J. & Demers, A. (2009). *Better Scripts, Better Games*. Communications of the ACM, Vol. 52, No.3, Publication date: March 2009.