PART III

**Project Plan**

****

**California State University Northridge**

**Computer Science Department**

**Scorch3d Earth**

By: Matthew Hoggan

Hangul Lim

David Schaffer

Davit Stepanyan

Author: Matthew Hoggan

Creation Date: 9/25/2010

Last Revised: 5/11/2011

Version: 1772

Contents

[Introduction 3](#_Toc290374518)

[*Purpose of Document* 3](#_Toc290374519)

[*Chronological Overview of Project Plan* 3](#_Toc290374520)

[Software/Hardware and Team Members Requirements 5](#_Toc290374521)

[*Software Requirements* 5](#_Toc290374522)

[*Hardware Requirements* 6](#_Toc290374523)

[*Team Member Organization* 6](#_Toc290374524)

[Project Schedule 7](#_Toc290374525)

[Effort Estimation 8](#_Toc290374526)

[Risk Analysis 10](#_Toc290374527)

[References 11](#_Toc290374528)

# Introduction

## *Purpose of Document*

As discussed in the System Requirements Specification Document:

“The purpose of this project is to remodel the original 2d Scorched Earth into a 3D version. This project will attempt to use the same general control system, random terrain generation, economic, physics, weapons, and environmental systems that were present in the original game. Thus, this project will attempt to reincarnate the original 2D game back to life in a new 3D body.”

Specifically, this document describes our plans to complete “Scorch3d Earth”. Presently the team is in the product start up phase, thus this document will describe how the “team” will break the development into increments and how resources will be distributed. Particularly, it will discuss how the work will be performed, to whom it will be allocated, and the overall development schedule.

This document will take into consideration the time constraints placed on the team by the University Calendar. Furthermore, it will consider other internal constraints such as the loss of a team member, inadequacies of teammates, and possible internal dissensions within the team. One last other important constraint it will consider are external ones, such as lack of budget, inadequate access to programming machines, as well as schedule conflicts.

Finally, this report will discuss possible assumptions made by the team, as well as possible risks that could deter the project from being completed.

## *Chronological Overview of Project Plan*

The following are the updates in this revision for our project plan. Up until phase IV of the list below our team was up-to-date. However, after phase IV there were some deviations from the plan. These will be addressed in the Gantt Charts and text below.

* Phase I: Finish Initial Documentation
* Phase II: Implement menu system
* Phase III: Unit test menu system components
* Phase IV: System test menu system
* Phase V: Implement “play game systems”
  + Terrain Sub-system
  + Tank Sub-system
  + Physics Sub-system
  + Projectile Sub-system
  + Background Sub-system
  + Inventory Sub-system
* Phase VI: Unit test each sub system Phase IV
* Phase VII: System test the “play game system”
* Phase VIII: Test and Debug entire system and Embed

Sounds

* Phase IX: Present final results

# Software/Hardware and Team Members Requirements

## *Software Requirements*

|  |  |  |
| --- | --- | --- |
| Requirement | Description | Reason |
| Subversion | A Content Management System (CMS) | Due to the complexity of the code and the normal structuring of a module when programming in OpenGL, the team will need group access to a consistent code base. |
| Microsoft Visual Studio 2008 | A Integrated Development Environment produced by Microsoft© | In order to maintain consistent project settings, and a uniform build environment, the team will be using Microsoft Visual Studio to build “Scorch3ed Earth” |
| OpenGL | A Set of Free Cross Platform Graphics Libraries | OpenGL will be at the heart of our program. Without OpenGL this game would not be possible. We chose OpenGL for our main tool because it is relatively comfortable for our team to use and provides an easy access of use at the computer lab 1618. |
| GLUT | Graphics Language Utility Toolkit (see glossary) | GLUT will be used to handle windowing, and user input via the keyboard or mouse. Furthermore, GLUT will provide our application with a timer function that will coordinate the game loop and the graphics pipeline. |
| SDL | Simple Direct Media Layer | Simple DirectMedia Layer is a cross-platform multimedia library designed to provide low level access to audio, keyboard, mouse, joystick, 3D hardware via OpenGL, and 2D video framebuffer. |
| AC3D | A 3D Design Program | AC3D will be used to generate models. Without AC3D we would have to render all models by hand (vertex by vertex), which would be extremely time consuming, if not impossible. |
| Doxygen | Doxygen is a documentation system for C++, C, Java, Objective-C, Python, IDL (Corba and Microsoft flavors), Fortran, VHDL, PHP, C#, and to some extent D. | It can help you in three ways:  It can generate an on-line documentation browser (in HTML) and/or an off-line reference manual (in ) from a set of documented source files. There is also support for generating output in RTF (MS-Word), PostScript, hyperlinked PDF, compressed HTML, and Unix man pages. You can configure doxygen to extract the code structure from undocumented source files. Doxygen is developed under Linux and Mac OS X, but is set-up to be highly portable. As a result, it runs on most other Unix flavors as well. Furthermore, executables for Windows are available. |
| Bugzilla | Bugzilla is server software designed to help you manage software development. | Bugzilla is a "Defect Tracking System" or "Bug-Tracking System". Defect Tracking Systems allow individual or groups of developers to keep track of outstanding bugs in their product effectively. Most commercial defect-tracking software vendors charge enormous licensing fees. Despite being "free", Bugzilla has many features its expensive counterparts lack. Consequently, Bugzilla has quickly become a favorite of thousands of organizations across the globe.` |

## *Hardware Requirements*

|  |  |
| --- | --- |
| System Component | Development Machine Requirement |
| CPU | 1.5 GHz or Faster |
| CPU Type | Intel Pentium 4 or AMD Athlon |
| Memory | 256 MB Internal RAM |
| OS | Any |
| Graphics Card | >= NVIDIA GeForce FX 5200  >= ATI Radeon 9600 |
| Graphics Card Memory | 128MB or greater |
| Hard Drive Space | 1 GB of free disk space |
| Mouse | Any |
| Keyboard | Any |

## *Team Member Organization*

|  |  |
| --- | --- |
| Team Member | Responsibility |
| Matthew Hoggan | Project Manager, Designer, Head of Infrastructure |
| Hangul Lim | Designer, Sound Engineer |
| David Schaffer | Lead programmer, assistant model designer |
| David Stepanyan | Assistant |

# Project Schedule

The updates to the project schedule will only reflect what remains in the last 5 weeks of school. It will not present things that have already been complete. If you would like to see the past schedules, please refer back to the previous version of the document.



Early in our development stage we adopted an agile approach to software development, in order to accommodate our school schedules, and Matthew’s school and work schedule. As a result much of the unit testing and system testing were done as a team. It was too difficult to keep track of exact dates at which these tests would occur.

# Effort Estimation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Information Domain Value | Count |  | Weighting Factor |  |  | Total |
|  |  | Simple | Average | Complex |  |  |
| External Inputs | 10 | 3 | 4 | 6 | = | 40 |
| External Outputs | 0 | 4 | 5 | 7 | = | 0 |
| External Inquiries | 0 | 3 | 4 | 6 | = | 0 |
| Internal Logical Files | 3 | 7 | 10 | 15 | = | 30 |
| External Interface Files | 0 | 5 | 7 | 10 | = | 0 |
| Unadjusted Function Points | -- | -- | -- | -- | --> | 70 |

For the count on External Inputs, instead of counting all the possible buttons the user could press in the original game, we simply counted the number of sub-menus which number about 10.

Internal Logical Files (ILFs) consit of the data associated with each player, their tank, and their inventory. All this data will be stored or pointed to by the tank object. Also there would have to be an ILF associated with terrain. Furthermore there will also be an ILF for the state of the physics engine. That make 3 ILFs.

To find the weighting factor for EIFs and ILFs there were three subgroups or Record Element Types (RETs) in the player ILF, two in the ILF associated with the physics engine, and only one for the terrain. This is a total of 6 RETs. For simplicities sake we counted only one Data Element Type (DET) per IFL. This in turn resulted in an average score for both EIFs and ILFs.

Using the Unadjusted Function Point tabulation from the table above, and the fact that our program will be programed in C++ and C the ratio found at http://www.spr.com/library/0Langtbl.htm gives us the average of 128 and 55 lines of code per each UFP. Therefore SLOC.

Now using the COCOMO II tool at http://sunset.usc.edu/research /COCOMOII/expert\_cocomo/expert\_cocomo2000.html the following tables were produced.

|  |  |
| --- | --- |
|  | Equivalent Size (SLOC) |
| New | 6105 |
| Reused | 114 |
| Modified | 0 |
| Total | 6219 |

Effort Adjustment Factor = 1.5

|  |  |
| --- | --- |
| **Effort** | **31.9 Person-months** |
| **Schedule** | **11.0 Months** |

Converting the person months over to hours we know there there are 153 person hours in a person month, therefore the person hours would be

required to complete the project. If we divide this by four, each team member would have to work a total of 1,220.175 hours. Now if we divide this by the total amount of weeks next semester minus one, this implies that we would have to work 81.345 hours per week over the next semester.

To conclude this section, the group has decided to face these numbers, and assume that somewhere in their subjectivity there are errors which will lead to reductions in the programming time.

This leads us to our second method of estimation, and that is based on expereience of three of our programmers in using OpenGL during a semester long course in game design and development. All three of us feel confident that we can accomplish the task in the given time period.

# Risk Analysis

As seen in the previous section the biggest risk for this project will be running out of time. However, this section will present a Matrix which will discuss all presently seens risks, and create a management plan for each.

Risks are at top and management plan objectives are labeled on left

|  |  |  |
| --- | --- | --- |
|  | Bad Time Estimates | Lack of Competance |
| Risk Number Priority | 1 | 2 |
| Risk Description | Due to the University Calender the team might not be able to submit the full project on time | If each team member is not compitant in their assigned task, the delays in completion of their given task will increase the probability of risk 1 |
| Risk Type | Estimation | People |
| Probability | 4 | 2 |
| Impact Factor | 3 | 3 |
| Impact Description | If our time estimates are not correct, and we can not acctually finish the project, then we have not overcome the risk and will receive a low grade on the project. | Lack of competence will hurt the group because it will keep us from staying on schedule, since other group members will have to shift from their responsibilities to fullfill someone elses |
| Avoidance Strategy | We have done our planning, and have all agreed to be persitent in our parts to finish this program | We have all agreed to work together upfront so we are all on the same page before we get too far along, and realize someone needs additional training. |
| Impact Minimization Strategy | To minimize the probability we have agreed to start on the porject over the winter break. | To absorb some of the imapct each team member has agreed to try and allocate extra time to help out if help is needed. |
| Monitoring Approach | We have established a rough schedule as seen in the “Project Schedule” section by discussing what needs to get done, and mapping it out using Gantt charts & we will monitor our progress against these charts weekly. | The only way to monitor this is for us to work together, and communicate. If anyone feels they cannot tell others they are behind then we will have serious problems. |
| Contingency Plan | If we fall off schedule quickly we will have to report this to the professor, and revise our planned project by cutting out features. | Report to the professor and redivide the problem amongst the competant members. Also try to include the less competent individual, by assigning any new task to him that arises ad hoc. |

# References

*C#, Visual Studio 2010, Silverlight, ASP.NET, WPF, WCF, WF, and Windows 7 Community*. Web. 27 Nov. 2010. <http://www.c-sharpcorner.com/UploadFile/imtiyazmulla/FunctionalPointAnalysisP211302005012541AM/FunctionalPointAnalysisP2.aspx>.