**Sinister Transistor**

**Project Management Plan**

**COP 4331, Spring 2016**

Team Name: The Mega Bytes

Team Members:

* Greg Kelso
* Mark Boutwell
* Joel Gardyasz

Modification history:

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Who | Comment |
| v0.0 | 02/12/16 | Greg | Template |
| V0.8 | 02/12/16 | Joel | Changes to Deliverables, SW Life Cycle, and Risk Management |
| V0.9 | 02/12/16 | Mark | Added Project Overview, Reference Docs, Standards, Team Organization, and Tools |

Contents of this Document

Project Overview

Reference Documents

Applicable Standards

Project Team Organization

Deliverables

Software Life Cycle Process

Tools and Computing Environment

Configuration Management

Quality Assurance

Risk Management

Table of Work Packages, Time Estimates, and Assignments

Technical Progress Metrics

Plan for tracking, control, and reporting of progress

**Project Overview**

Sinister Transistor will be a 2D top-down action adventure game set in a fantasy universe. The game will be similar to the classic Legend of Zelda games but with modern gameplay and graphics. It will feature dungeon exploration, several enemies to fight, an item and inventory system, and player leveling.

**Reference Documents**

* Concept of Operations
  + <https://github.com/gregkelso/COP-4331/blob/master/CONOP.docx>

**Applicable Standards**

* Coding Standard :
  + We will be following the Microsoft .Net framework design guidelines outlined here:
  + https://msdn.microsoft.com/en-us/library/ms229042.aspx
* Document Standard
  + For Documents, we will be using 11pt Calibri with bold headings, 1” margins, and single spaced.
  + We will have a table of contents and a modification history showing all edits.
* Artifact Size Metric Standard
  + Time – milliseconds
  + Memory/Data – Megabytes

**Project Team Organization**

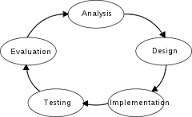
The team includes Greg Kelso, Mark Boutwell, and Gardyasz; with Greg Kelso being the project manager. To help us adapt and create new features and because we are such a small team, we will be using a loose organizational structure with a sizeable amount of autonomy between members. Communication will be done through several face-to-face meetings and online through a communication service called Slack. Program code will be shared via Github.

**Deliverables**

|  |  |
| --- | --- |
| **Artifact** | **Due Dates** |
| Meeting Minutes |  |
| Individual Logs |  |
| Team Reports |  |
| ConOps | Feb 5th |
| Project Plan | Feb 12th |
| SRS | Feb 19th |
| Project Management Report | Feb 19th |
| High-Level Design | Feb 26th |
| Detailed Design | Mar 4th |
| Test Plan | Mar 18th |
| User's Manual | April 22nd |
| Test Results | April 8th |
| Source, Executable, Build Instructions | April 22nd |
| Project Legacy | April 22nd |

**Software Life Cycle Process**

Our group will follow the Agile Software Development model. We chose this method because we felt that it would work best when creating a small 2D game. The Agile methodology will lead to iterative development and allow for flexibility and changes to the final product as time goes on.

This is a simple diagram of how our development process will operate. Its cyclical form will allow us to make changes to our design along the way as we learn more about what we wish to accomplish with this project.

**Tools and Computing Environment**

Operating System – Windows

Software – Unity, MonoDevelop, git

Program Language – C#

**Configuration Management**

<How will your group handle version control and change control? Who is responsible? What procedures will be followed?>

**Quality Assurance**

<What QA activities will your group do and when will each activity occur? ... Who is responsible for making sure this occurs? How will the results be reported?>

**Risk Management**

One potential risk for this project is the risk of either not including enough in the game or trying to implement too much. It is important to maintain a proper balance in this area because we want to feel accomplished with our project, but not overburdened. The risk will be managed with our agile development method by looking at what we’ve accomplished at regular intervals and determining what pace we are on and planning from there.

**Table of Work Packages, Time Estimates, and Assignments**

<Break down your project into a hierarchy of work packages. For each work package, estimate how much work time it will take to complete. For each work package, state who is responsible for its completion. It is expected that this information will be at a high-level at the beginning of the project.>

**Technical Progress Metrics**

<You must estimate and track your technical progress using appropriate metrics for each phase of your project. What is a useful metric for each phase of your project? For example, for requirements phase, the total number of requirements, the number of requirements changes, the number of TBDs, etc.>

<For OO analysis and design, you might want to count UML diagrams completed. For detailed design and code, you might want to count packages, classes, methods. You will also want to think about other technical metrics such as: memory usage, execution speed, size of various documents, complexity of code (using any of the complexity metrics). These can help in planning and in tracking your project work.>

<Choose your metrics carefully -- select metrics that will be easy to collect, easy to report, and easy to interpret. The goal is to give management insight into the progress and risks of your project.>

**Plan for tracking, control, and reporting of progress**

<Briefly describe what data to collect, when to collect it, how and when to interpret it, how and when to report it. Following is an example that you can base your team’s plan on.>

"At a minimum, each team member will post the following information weekly: individual time and activity log, individual status information, individual issues and problems, and individual defect log.

Each week, the project manager will: read and analyze the logs; examine the technical content of the work done to date; examine the technical progress metrics; consider the QA results; reassess the potential project risks; and take corrective action if necessary.

The project manager will issue a Project Management Report on the schedule as indicated in the deliverables section above. Updates will be posted to the Project Management Report every two weeks and will include the following information: 1 sentence description of overall status, 1 or 2 sentence of any planned changes to the project plan, graph of planned vs actual time, graph of planned vs actual for each technical progress metric."