**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 |
| V1 | 02/12/16 | Greg | Added Configuration Management, Quality Assurance, Risk Management, Work Packages, Technical Progress Metrics, and Tracking/Reporting of Progress |

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>
* Project Management Plan
  + <https://github.com/gregkelso/COP-4331/blob/master/ProjectManagementPlan.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**

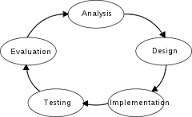
The team includes Greg Kelso, Mark Boutwell, and Joel Gardyasz; with Greg Kelso being the project manager. To help us adapt and create new features and because we are such a small team, we will be using a loose organizational structure with a sizeable amount of autonomy between members. Communication 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**

Greg is hosting a public software repository on his github page at <https://github.com/gregkelso/COP-4331>. We are storing all documentation and project files in this repository. Everyone is responsible for the changes that are made, and all merge conflicts are discussed before being handled.

**Quality Assurance**

At the end of each sprint, we will review the state of the assigned tasks and rate their completion status according to the NPLF rating scale found in ISO/IEC 15504:

* Not achieved (0 – 15%)
* Partially achieved (>15% - 50%)
* Largely achieved (>50% - 85%)
* Fully achieved (>85% - 100%)

**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**

Due to the short duration of the course, we will have one work package for each person, and the project manager will take on the integration of all packages.

* Work package integration: Greg, 7 days
  + Initial creation of software structure, game objects, entity controllers, and other necessary methods of ensuring an object-oriented software design: Mark, 21 days
  + Level design, integration of art assets with game objects: Joel, 21 days
  + Creation of artificial intelligence for enemies and NPCs: Greg, 14 days

**Technical Progress Metrics**

1. Algorithm and resource analysis. Deliverable: first decision made regarding all art resources, algorithms, and level design so the team can start work on the project.
2. Environment development: Creation of physical game world, in addition to all necessary software structures being created (such as entity controllers, sensor classes, etc).
3. Algorithm development: Includes pathfinding, artificial intelligence, sensors, and others.
4. Integration and testing: All elements are combined and tested to ensure compliance in accordance to requirements and quality standards.

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

Each week the team will meet and discuss their progress over the previous sprint. Each member will track time spent, in addition to the percentage of completion they achieved for their assignment. For assignments with over 85% completion, the individual will move onto the next assignment; under 85% requires more work on the assignment and a reevaluation of the requirements and the planned technical progress.

Individuals will also keep a weekly issue tracking log, and at each meeting the logs will be discussed, and added to a communal issue log. This will help us keep an idea of what problems might frequently arise, and have a database of solutions to go with the problems.