**Dung Eon : A Dungeon Crawler**

**Project Management Plan**

**COP 4331, Spring, 2016**

Team Name: Bobby Sandwich

Team Members:

* Jacob Crandall
* Robert Bland
* Jacob Jiskoot

Modification history:

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Who | Comment |
| v1.0 | 2/12/2016 | All members | Put initial items for everything in the table of contents |
| ... |  |  |  |

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**

Our project is a video game meant for mobile platforms. The game will be a two dimensional dungeon crawler with a top down perspective. A user will control a hero battling through randomly created mazes with a dungeon theme. The goal of the game is not to reach a certain point or beat a certain level, but rather to reward the user with finding new items and conquering new or harder enemies. A side goal of this project is to help us learn skills for designing cross-platform mobile applications.

**Reference Documents**

* Concept of Operations
* Project Management Plan
* https://github.com/jacobmcrandall/POOPCrawler
* <any other relevant documents>

**Applicable Standards**

* **Coding Standard**We will be using the coding standard outline at :  
  <http://www.dofactory.com/reference/csharp-coding-standards>  
  And enforcing it with style cop :   
  <http://stylecop.codeplex.com/>
* **Document** **Standard**  
  Font : Calibri 11  
  Margins : 1” all around  
  Each document (unless it is a diagram) will have a table of contents and cover page on the first page. After that the items will be listed in the order presented on the table of contents.  
  Each cover page will have a list of changes and the author’s names, as well as the project name and class information.  
  Grammar checking will be done using Microsoft word’s automatic spell and grammar checking.  
  Tables will be done using Microsoft’s standard table creator. Tables will have one blank row to indicate that they may still be modified.
* **Artifact Size Metric Standard**We are classifying the “size” of our project by the number of objects that we have. We will break these objects into how many child objects we have, such as items as the parent object and then armors / weapons, etc. as the child objects.

**Project Team Organization**

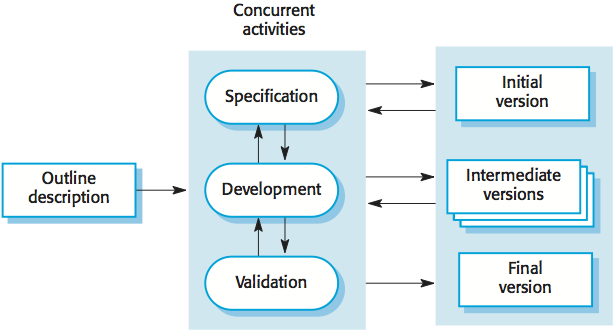
* Robert Bland, Jacob Crandall and Jacob Jiskoot are all members of our development team.
* We will be splitting up the project into classes. Each member will be responsible for a category of classes. For example the levels or the items. We have yet to divide these responsibilities.  
  Major documents such as this or the concept of operations will be done as a group. However, more specific documentation will be handled by whoever is in charge of that respective category of classes.
* We have a chat group for issues that may arise during the week. We shall have weekly in person meetings where we will get caught up with other’s work.

**Deliverables**

|  |  |
| --- | --- |
| **Artifact** | **Due Dates** |
| Meeting Minutes | Weekly |
| Individual Logs | Weekly |
| Team Reports | Weekly |
| ConOps | 2/05/2016 |
| Project Plan | 2/12/2016 |
| SRS | 2/19/2016 |
| High-Level Design | 2/26/2016 |
| Low-Level Design | 3/04/2016 |
| Detailed Design | 3/11/2016 |
| Test Plan | 3/19/2016 |
| User's Manual | 4/22/2016 |
| Test Results | 4/08/2016 |
| Source, Executable, Build Instructions | 4/22/2016 |
| … |  |

**Software Life Cycle Process**

We will be using the Incremental Development Model. We will be following along the lines of the waterfall model to get an initial prototype to work with. After this point we will fully enter into using the Incremental Development Model by adding on extra features until we reach a final product. We are using it because we want to develop an initial prototype and continue pushing updates and new content to the game. This is also important because the prototypes will allow us to test new content easily.



**Tools and Computing Environment**

Our development team will be using the windows operating system.  
We are designing the project to run on android initially and potentially other mobile platforms.  
We will be programming in C# using the Visual Studio IDE with Xamarin’s tools for development along with the CocoSharp game engine library.

**Configuration Management**

We will be using GitHub to handle version control. Jacob Crandall will be in charge of making sure the GitHub is maintained properly.

**Quality Assurance**

There will be incremental testing as we develop to make sure that each new feature works properly and integrates with the overall system. Each new feature added to our current program will be reviewed and tested by the other two members. Results will be reported by opening an issue in GitHub.

**Risk Management**

* One risk is that we do not finish our project on time. We can manage this risk by having prototypes so that we can end development early on a certain feature if needed.
* Having an inefficient application that bogs down the user’s phone. We can solve this by testing the code in our prototypes and trying to detect a slow system.

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

Initial System – All members are responsible for : EST 2.5 Weeks  
 - Including initial UI base (Starting / Main Menu)  
Level System – Jacob Jiskoot – Mid March  
Characters – Robert Bland – End of March  
Item System –Jacob Crandall – Mid March

**Technical Progress Metrics**

* **Technical Progress  
  Requirements phase**

Total number of planned requirements - 10 requirements

**Design phase**

Total number of planned UML diagrams - 6 diagrams

**Implementation**

Total number of planned classes - 17 classes

**Testing Phase**

Total number of planned classes to be test - 17 classes

**Increment Phase**

Total of number of prototypes - 5 prototypes

* **OO analysis and design**Total number of UML diagrams completed - 0 diagrams

Total number of classes completed - 0 classes

Total number of classes tested - 0 classes

Total Memory Usage - 0 B

Size of Concept of Operations - 5 pages

Size of Project Management Plan - 6 pages

Total number of prototypes - 0 prototypes

Total number of packages - 0 packages

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

In each weekly meeting each individual will summarize new code and changes that they have made.  These individuals will also be able to bring up complex questions and resolve issues that were not able to be solved in online meetings.  While the code will be summarized by the individual the related documentation will be presented by the individual and review by the other members.  If there are issues between features interacting the group members it affects will be able to plan a course of action to try to resolve this issue.  Should this issue continue to be an issue the team will try to address it together until a solution is found.  Once all questions and issues are resolved each individual shall outline the next steps they plan to take for the week.

All of these items discussed in the meeting will be summarized in a rolling weekly meeting log.  So each week there will be three summaries, one from each individual.  
These will be one to two sentences for: changes made the previous week, questions and answers about certain parts of the program, complex issues (if applicable) and future plans.