Video Game Project

Software Project Management Plan

Project Manager:
Michael Swedo

Assistant Manager:
Jonathan Cabe

Revision 1.0
January 29, 2014

Contents

1. Overview

1.1 Project Summary

1.1.1 Purpose

This Software Project Management Plan outlines the management of the *Reignite* video game development. It contains requirements, development cycle plans, a timeline of progress, maintenance details for the application. Its intended audience is Dr. Concepcion.

1.1.2 Scope of the Project

The scope of this plan encompasses the development of *Reignite* during its first two major revisions (prototypes). It is assumed that development of this video game will continue after the timeframe outlined in this document. Anything not directly related to video game development as specified in the current SRS is to be considered outside of project scope.

1.1.3 Assumptions and Constraints

We make the following assumptions:

- The client has full ownership of the intellectual and physical property of video game.
- Members of the team understand and follow the approved SRS.
- Members of the team will work on the video game and underneath the direction of the management team.
- Members of the team will dedicate a reasonable time outside of class to work on the development of the video game.
- Members of the team will be proactive in obtaining the necessary skills in order to develop the video game.
- Members of the team will have access to necessary tools and facilities in order to develop the video game.

1.1.4 Project Deliverables

The first deliverable is an executable prototype of the game, focusing on the Adventure game section of the game. This prototype will include one or more scenes, a framework for the User Interface, and functionality for object interaction and movement.

The final deliverable is an executable prototype of the game, expanding upon the first prototype and including more content and the rhythm game section of the game in addition to the adventure game section.

1.1.5 Schedule and Budget Summary

Refer to Section 5.2.2 for the schedule summary. No budget is given for the project.

1.2 Evolution of the Plan

2. References

Software Project Management Plan IEEE 1058-1998 Student Advising Software Project Management Plan CSE 455, Inc. v1.0

3. Definitions, Acronyms, and Abbreviations

C# - A programming language used to program the logic within the application.

Client - For the extent of this project, this refers to Michael Swedo.

Github - A web-based hosting service for software development projects that use the Git revision control system.

MB - Shorthand for megabyte, a unit of measurement for digital information storage/transmission.

Note Chart - Visual representation of the pitch of music.

Puzzles - In general, tasks that players will have to take in order to progress in the story. These tasks will require a degree of thinking for success.

RAM - Random Access Memory. A form of computer data storage.

Splash Screen - Initial screen displayed when application is executed.

SPMP - Software Project Management Plan, this document.

SRS - Software Requirements Specification, including the Game Design Document..

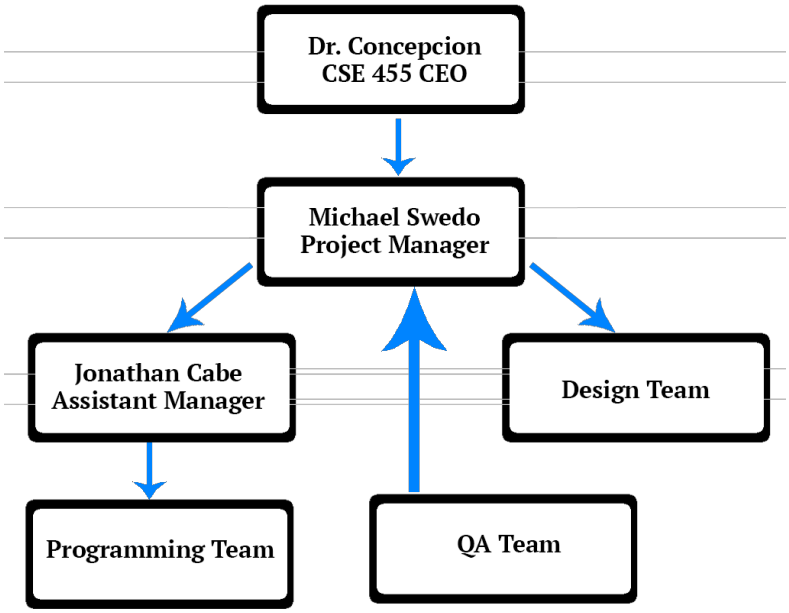
Target Platform - The hardware device that the program will operate on. Development takes this into account in order to ensure program compatibility.

Unity - A cross platform game engine with an integrated development environment.

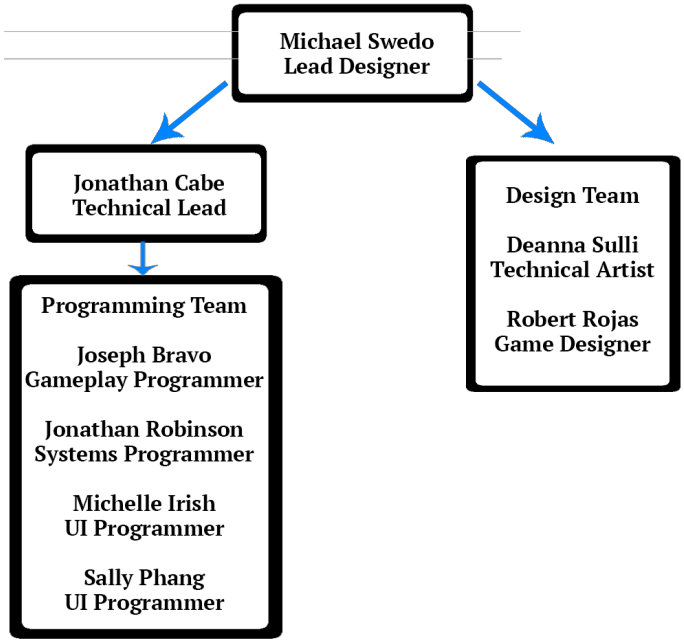
XML - A markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable

4. Project Organization

4.1 External Interfaces



4.2 Internal Interfaces



4.3 Roles and Responsibilities

Michael Swedo, as Lead Designer and Project Manager, is responsible for task assignment and follow up with the design team, as well as meeting control. As Lead Designer, he is responsible for game design, puzzle creation, sound and music creation and handling, and programming tasks related to input, music, and rhythm section gameplay.

Jonathan Cabe, as Technical Lead and Assistant Manager, is responsible for task assignment and follow up with the programming team, as well as meeting assistance. As Technical Lead, he is responsible for tool development for the dialog system and note chart generation, as well as tasks related to rhythm gameplay and the quest / game progression system.

Deanna Sulli, as Technical Artist, is responsible for the art direction, art asset creation, and implementation of all art assets. Because *Reignite's* initial design and story was her creation, she will also be heavily involved in the writing and story direction with Michael Swedo and Robert Rojas.

Robert Rojas, as Game Designer, will work closely with Michael Swedo and Deanna Sulli to design the game, puzzles, levels, and story ideas, and implement these in the game.

Joseph Bravo, as Gameplay Programmer, is responsible for in-game systems such as player interaction with objects and dialog, and the scoring section for the rhythm game.

Sally Phang, as UI Programmer, is responsible for the Main Screen and Rhythm section user interface's functionality, and will work closely with Deanna Sulli on the design of the UI.

Michelle Irish, as UI Programmer, is responsible for the functionality of the In-Game User Interface, including the inventory menu and dialog windows, and will work closely with Deanna Sulli on the design of the UI.

Jonathan Robinson, as Systems Programmer, is responsible for inventory management and the save/load feature, and will work with Jonathan Cabe on the quest / progress system.

5. Managerial Process Plans

5.1 Start-up Plan

The *Reignite* team's start-up plan was a branch of processes, many of which occurred simultaneously. These have already been completed or at the very least begun.

- Technologies research / tutorials
- Storyboarding / conceptualization
- Design overview

5.1.1 Staffing Plan

Our staffing for the *Reignite* team was done through a survey filled out by members of the Software Engineering course who were assigned to our team according to their interests and skills.

5.1.2 Resource Acquisition Plan

We do not foresee any required resources currently not in our team's possession. If any required components do arise, they will be requested from Dr. Concepcion as needed.

5.1.3 Project Staff Training Plan

All software engineers are expected to learn the required technologies independently, but workshops and tutorials will be provided by Jonathan Cabe and Michael Swedo in the Unity3D engine, C# programming, and any and all problems that may arise during development.

5.2 Work Plan

5.2.1 Work Activities

Conceptualization:

- Storyboarding
- Puzzle Planning and Design
- Writing
- Scene Design, Layout, and Relationships

Asset Generation:

- UI Art
- Sprite Sheet Building
- Character Art
- Character Animation
- Environment Art
- Object Art
- Sound Effects
- Rhythm Game Music
- Environment Background Music

Player Information Handling:

- Inventory Management
- Data Management (Save & Load)

Tool Creation:

- Music Chart Builder
- Dialogue Builder

User Interface Design and Programming:

- Rhythm Section UI
- Adventure Game UI
- Main Screen UI
- Dialog / Interaction Windows
- Functionality

Adventure Game Scene Design and Programming:

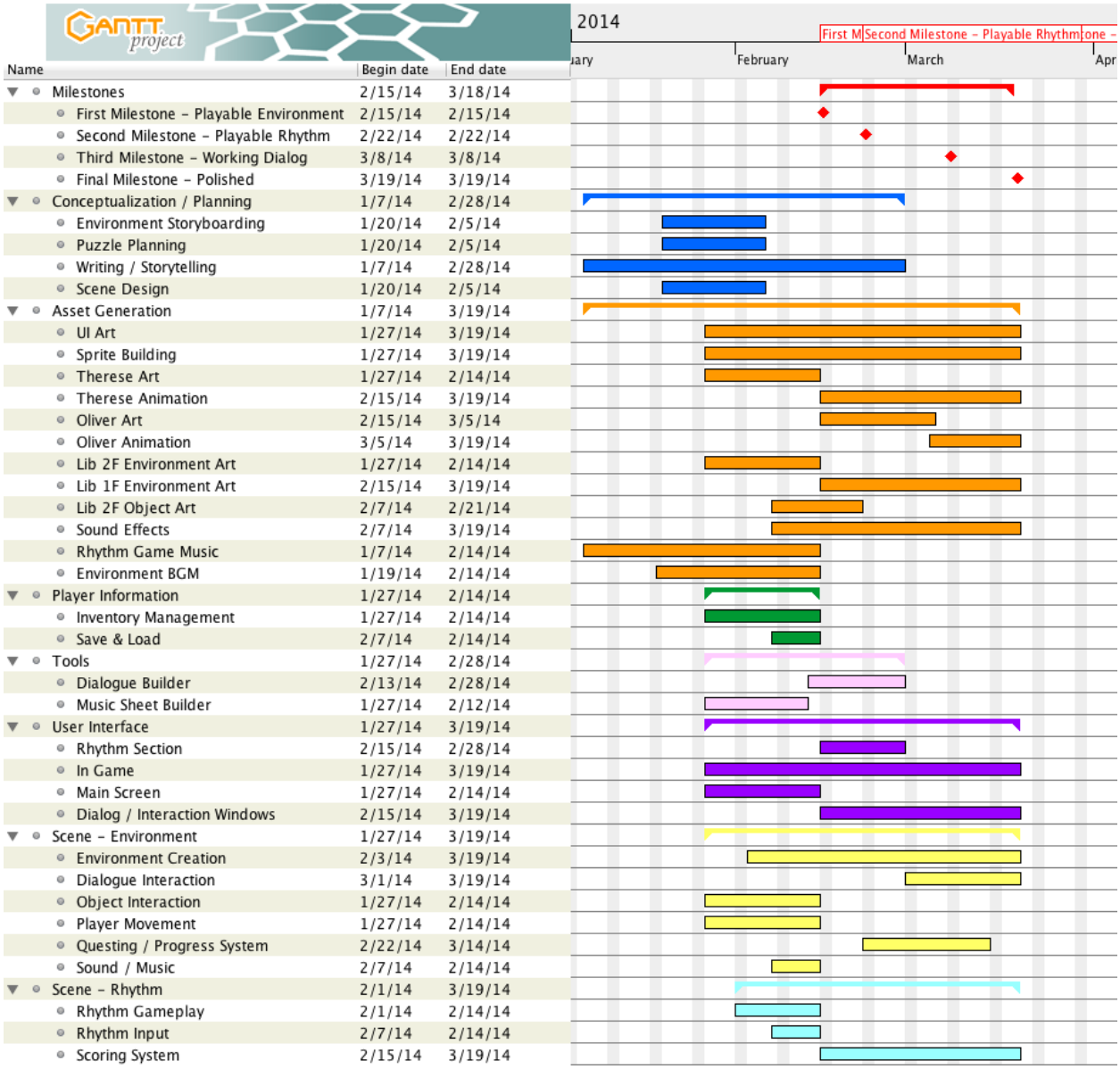
- Environment Creation
- Dialogue / Interaction
- Object Interaction

- Player Movement
- Questing / Progress System
- Sound / Music

Rhythm Game Scene Design and Programming:

- Gameplay Programming
- Input Handling
- Scoring System

5.2.2 Schedule Allocation



Name	Begin date	End date
Milestones	2/15/14	3/18/14
First Milestone – Playable Environment	2/15/14	2/15/14
Second Milestone – Playable Rhythm	2/22/14	2/22/14
Third Milestone – Working Dialog	3/8/14	3/8/14
Final Milestone – Polished	3/19/14	3/19/14
Conceptualization / Planning	1/7/14	2/28/14
Environment Storyboarding	1/20/14	2/5/14
Puzzle Planning	1/20/14	2/5/14
Writing / Storytelling	1/7/14	2/28/14
Scene Design	1/20/14	2/5/14
Asset Generation	1/7/14	3/19/14
UI Art	1/27/14	3/19/14
Sprite Building	1/27/14	3/19/14
Therese Art	1/27/14	2/14/14
Therese Animation	2/15/14	3/19/14
Oliver Art	2/15/14	3/5/14
Oliver Animation	3/5/14	3/19/14
Lib 2F Environment Art	1/27/14	2/14/14
Lib 1F Environment Art	2/15/14	3/19/14
Lib 2F Object Art	2/7/14	2/21/14
Sound Effects	2/7/14	3/19/14
Rhythm Game Music	1/7/14	2/14/14
Environment BGM	1/19/14	2/14/14
Player Information	1/27/14	2/14/14
Inventory Management	1/27/14	2/14/14
Save & Load	2/7/14	2/14/14
Tools	1/27/14	2/28/14
Dialogue Builder	2/13/14	2/28/14
Music Sheet Builder	1/27/14	2/12/14
User Interface	1/27/14	3/19/14
Rhythm Section	2/15/14	2/28/14
In Game	1/27/14	3/19/14
Main Screen	1/27/14	2/14/14
Dialog / Interaction Windows	2/15/14	3/19/14
Scene – Environment	1/27/14	3/19/14
Environment Creation	2/3/14	3/19/14
Dialogue Interaction	3/1/14	3/19/14
Object Interaction	1/27/14	2/14/14
Player Movement	1/27/14	2/14/14
Questing / Progress System	2/22/14	3/14/14
Sound / Music	2/7/14	2/14/14
Scene – Rhythm	2/1/14	3/19/14
Rhythm Gameplay	2/1/14	2/14/14
Rhythm Input	2/7/14	2/14/14
Scoring System	2/15/14	3/19/14

5.3 Control Plan

5.3.1 Requirements

Bi-weekly meetings will be conducted with the team.

5.3.2 Schedule

Outside of the bi-weekly course meetings, software engineers are responsible for their given tasks and must be responsible for their own time management. Milestones have been set and should be met with the appropriate work, and progress will be continuously evaluated.

5.3.3 Quality

Deanna Sulli will continuously evaluate the quality of the artistic design and direction of the project. Michael Swedo and Jonathan Cabe will review all other changes made to the program. The two major milestones of first prototype and final prototype will be reviewed by Dr. Concepcion.

5.3.4 Reporting

Dr. Concepcion will hold weekly board meetings where the status of the project is reported in detail.

5.3.5 Metrics Collection Plan

No metrics have been collected as of this draft.

5.4 Risk Management Plan

In case of human resource loss, other members of the project will pick up tasks assigned to them.

In case of equipment loss, students are held individually liable for the damages, as per CSUSB policy.

5.5 Closeout Plan

For this course, the closeout plan is the presentation of the final executable build to Dr. Concepcion for review. It will also include a review of the documentation by Dr. Concepcion and a plan for project continuation created by Michael Swedo.

6. Technical Process Plans

6.1 Process Model

An agile software development model will be utilized for the development of *Reignite.* Development will take place in multiple iterations, and have four milestones of features and content. A master, or stable, branch will be kept and updated with stable code on a daily basis. Builds between programmers will be pushed to the master branch when they have been checked for stability as detailed in Section 7.4.

Any work that was planned for a milestone but was not met will be carried over to the next milestone. Only the first milestone and the final milestone will be given to Dr. Concepcion for review and demonstration.

6.2 Methods, Tools, and Techniques

Methods:

- Agile Development (Rational Unified Process)

Tools:

- Unity3D
- Paint Tool SAI
- Adobe Suite (primarily Photoshop)
- Cubase Elements 7
- Self-created tools for Unity for Dialogue and Music Chart creation.

6.3 Infrastructure Plan

Continued maintenance and development of the game will be carried out by Michael Swedo, Deanna Sulli, and any and all other interested parties selected by Mr. Swedo and Ms. Sulli through an interview process. Further development for *Reignite* is planned and assumed.

Devices necessary for development will be provided by team members or by Dr. Concepcion.

6.4 Product Acceptance Plan

Dr. Concepcion and Michael Swedo will test the final product for acceptance. Primarily, functionality completeness will be the main criteria for the final product delivered through this phase of development.

7. Supporting Process Plans

7.1 Configuration Management

All project deliverables will be considered as configuration items. The named documents will be referred to by their abbreviations - SRS, SPMP - and a version number. Each code revision will be submitted to the Reignite GitHub branch of the programmer responsible for the code, and then reviewed as stated in Section 7.4 before it is pushed to the master branch.

7.2 Documentation

Documentation for programming will be provided with detailed comments in code. Design and story documentation will be maintained on the Reignite GitHub Wiki page.

7.3 Quality Assurance

Quality assurance will be performed both locally by the development team and a build will be submitted each week to the Quality Assurance team for additional testing.

7.4 Reviews and Audits

Jonathan Cabe or Michael Swedo will review and audit all code and work submitted to repository branches before it is posted to the master branch.

7.5 Problem Resolution

Problems will be resolved internally between members of the development team. If problems cannot be resolved, outside sources, such as Dr. Concepcion, will be consulted.

7.6 Process Improvement

As this is the beginning of the development, necessary adjustments will be made when necessary to improve performance.