

Table 1: Revision History

Date	Developer(s)	Change
November 1st	Evan Reaume	Edited Technology, Proof of Concept, Git WorkFlow and Communication
Date2	Name(s)	Description of changes
...

CS 4ZP6: Development Plan

Handits

Team Handits

Evan Reaume reaumee

Danny Stewart stewardl

Dustin Jurkaulionis jurkaudj

Ming Liu liumh@mcmaster.ca

Kong Zhijun kongz3@mcmaster.ca

1 Introduction

Our Goal for this project is to create and release a 2D co-op platformer game that uses creative game elements and fun themes to be genuinely enjoyable. the game is to be directed towards audiences of all ages, but will require a minimum of 2 players. Additionally this project will mostly focus on implementing good software development methodologies, and following scrum development for additional experience in agile development.

2 Team Meeting Plan

Team meetings will be held three days a week, at the following times: 2:30PM on Mondays, 3:30PM on Thursdays, and 2:30PM on Fridays. Using the scrum methodology we will be organizing our progress into Product Increments and Sprints. Each Product Increment will be 20-30 days in length, with the first meeting of the month being the Product Increment Planning Meeting in which we will identify our goals for the Product Increment, and identify the tasks required to get there. Additionally at the end of each Product Increment we will hold a Product Increment Review Meeting to individually identify what we accomplished, look at any problems we have encountered, and look at ways to improve our progress in the next Product Increment. At a lower level, each week will be a sprint, in which the Monday meetings will be sprint planning meetings where we distribute the tasks to be completed that week, and the Friday meetings are weekly Review meetings to identify what we've accomplished, what obstacles we faced to help prepare for the following sprint. Via Google Drive we will be creating Meeting Agendas and releasing the Meeting Minutes. In the case a member misses a meeting, another member of the group will fill them in, and they will be expected to review the meeting minutes. Meeting

Agendas should be online 1-2 days before the meeting, and minutes should be online ideally the day of the meeting itself.

3 Team Communication Plan

Our team will be using several social platforms for version control, project organization, and communication. For version control and sharing key information pertaining to our project we will be using Github; in addition to the source files themselves, we will be using GitHub to store any assets, and any design documents. GitHub issues will be used to delegate tasks, and to store design documents. For project organization we will be primarily using Trello, but tasks and git issues will likely be distributed in parallel to individual team members; GitHub Issues will be used for detailed descriptions of tasks, while Trello will be used for it's deadlines. In Trello we will also be storing the Product Backlog which will contain the set of all tasks to be accomplished, and the Sprint log, which will be filled with elements from the Product Backlog in each Sprint Planning Meeting. Finally, aside from in meetings, the bulk of our communication will be done in Facebook Messenger. Facebook Messenger is readily available in our daily lives, and as such is quick to notify us of important updates; we chose Messenger over Slack for these reasons.

4 Team Member Roles

Our team will be fluid, but roles will help designate leaders and avoid democratic stalemates and to help with this distribution of tasks. Scrum development has the three positions of Product Owner, Scrum-Master, and the Scrum Team itself. As we are a team of 5, we will all be committed to development, organization, etc.

4.1 Product Owner \Lead Designer

Evan was selected as the product owner, which will assign him the final say in major design decisions. In Scrum Methodology, the product owner should assure the product meets the desires of potential stake-holders, which in this case would be the capstone supervisor and the capstone course instructor. As a small team however, we will all have contact with these parties, be involved in the verification and validation of our product, and will always communicate design decisions and idea conflicts as much as possible.

4.2 Scrum Master

Kong, Ming and Evan were selected to share the position of Scrum Master. The Scrum Master is tasked with assuring scrum practices are followed, and maintaining direction during scrum meetings. As we will all be equally present in development, research and design of this project, it is better to have the

position shared between members; having this position split will also help team cohesion in meetings.

4.3 Game Designer

Kong and Ming were selected to split the role of game designer. The role of Game Designer holds the final say in design decisions involving game mechanics, and game play, and should include the distribution of tasks related to that field of development.

4.4 Level Designer

Dustin and Danny were selected to split the role of level designer. The role of Level designer holds final say in decisions involving level design, and should lead in the distribution of tasks related to level design and creation.

4.5 UI Designer

Danny was selected as the UI Designer. The UI Designer holds final say in decisions involving User Interaction. the UI designer should lead in the distribution of tasks related to UI, and will likely be more involved with the tasks himself.

4.6 Art Director

Danny and Evan will split the role of Art Director, pertaining to tasks related to sprites, game assets, visual design, etc.

4.7 Audio Director

Dustin will hold the role of Audio Director, pertaining to tasks related to music composition, sound effects, etc.

5 Git Workflow Plan

In each product increment we will have a develop branch for that product increment. During the product increment we will create a separate branch for every feature we develop. After the feature is finished and reviewed, it can then be merged with the develop branch. After a sufficient amount of features has been added, a quality assurance branch would be created, and all tests would branch from the QA branch. If bugs are found, then the develop branch would be updated. After sufficient testing, we would then create a release branch. from here, once the product owner approves, the release branch will be merged with the master branch. This development cycle would have a period of one product increment.

6 Proof of Concept Demonstration Plan

A Proof of Concept (PoC) version will be developed to demonstrate the feasibility of the project on a minimal scale. The PoC will serve as the first iteration of a playable game which will demonstrate the core mechanics and user interface(UI) of the game. The PoC will explore at least one of each type of gameplay mode and shell menu with a flow board to model their relationships. The following features will be implemented in the PoC:

- One fully designed
- Complete player control, including developed interaction model.
- Enemy AI
- Player interaction with elements of the environment including enemies, items on the map or other players
- Two-Player Local Co-op

7 Technology

We will be using Unity as a game engine, and we will be using GitHub as a version control tool. GitHub issues will be used for in-depth descriptions of tasks, but Trello will be used to divide tasks and to store scrum artifacts like the product log and sprint log. We will also be using Google Drive to store any additional files we required involved with things like brainstorming and meeting minutes. Facebook Messenger will be used to communicate frequently about the project. Additional software will be used for asset creation like sprites and music. Also we will need to look into additional software for multiplayer networking in the case Unity is unable to support it on it's own.

8 Coding Style

We will hold a consistent style across our project, from code style, to project structure. The code made will be embedded C scripts to be used within Unity, and so, we will be following coding conventions from Microsoft's C Programming guide. As we will be developing in Unity, a large portion of the project will be organizing and maintaining scripts, assets, and scenes stored within the Unity project, and so we will attempt to structure our project rigidly. Although we may decide to change the structure, we will structure our files by type (sprite, animation, script, etc.) relevant to each individual level/scene. We will sort our scripts into the following categories: BaseInputController, BaseGameController, BasePlayerManager, BaseUserManager, and SceneManager. This is based on the Textbook *Game Programming Cookbook for Unity 3D*.

9 Project Schedule

Active steps in our project can be seen from our Trello board and our GitHub.
Please click [here](#) for the Gantt chart in PNG format

10 Project Review