

Development Plan

Group 8: Team Eight

Stanley Liu

Zhi Zhang

Toni Miharja

September 28 2017

Contents

1	Revision History	3
2	Team Meeting plan	4
2.1	When	4
2.2	Where	4
2.3	Frequency	4
2.4	Roles	4
2.5	Rules For Meeting Agendas	4
3	Team Communication Plan	5
4	Team Member Roles	5
5	Git Workflow Plan	5
5.1	Label Usage	6
5.2	Milestones	6
6	Proof of Concept Demonstration Plan	6
7	Technology	6
8	Coding Style	6
9	Project Schedule	7
10	Project Review	7

1 Revision History

Date	Developer	Change	Revision
September 29, 2017	Stanley Liu, Toni Miharja, Zhi Zhang	Development plan	Revision 0
December 6, 2017	Zhi Zhang	add table of contents, revision history table, project review.	Revision 1

Table 1: Revision History: Problem Statement

2 Team Meeting plan

2.1 When

- Regular meeting will be conducted on Mondays and Thursdays after lab as we will be attending labs and all members are free after the lab.
- Team members have agreed to keep the following timings free, reserved for the meeting:
 - Monday, 6:30 - 8:30 pm
 - Thursday, 4:30 - 6:30 pm
- The duration of the meeting is flexible, depending on the need.

2.2 Where

Labs or school libraries will be the main location for meetings.

2.3 Frequency

The team will meet at least twice ~~once~~ a week. The team will meet with higher frequency depending on need and tasks required to be completed.

2.4 Roles

- Team Leader: Toni
- Software Developers: Stanley, Zhi, Toni
- Software Tester: Toni
- Gitlab Manager: Stanley
- LaTeX Editor: Zhi

2.5 Rules For Meeting Agendas

- Meeting agenda will begin with topics that affect the entire team and continue with more specific details from each individual.
- The first topic will be to review the agenda to ensure that the team knows what to expect from the meeting.
- Meeting agenda will be based on input from team members.
- Agenda topics will be listed in questions.
- Each agenda item will have a realistic estimated time for discussion.
- One team member will be responsible to lead each topic.
- Meeting will end with a written statement on decisions made by the team.

3 Team Communication Plan

Our team will utilise a combination of communication channels to ensure effective communication:

- WeChat group chat serves as the main communication channel to organise meetings and talk about the project in general.
- Physical meeting on a weekly basis ensures that each member knows what the other members are doing and ensure that everyone is on the same page to ensure timely project completion.
- Git issues is also utilised to keep track of the main issues in the project development.

4 Team Member Roles

- Team leader: Toni
 - The team leader sets the direction of the team, chairs meetings and ensures that the project is on track to completion.
- Software developers: Stanley, Zhi Zhang, Toni
 - Software developer codes and develop the application.
- tester: Toni
 - Software tester checks that the requirements of the project are met.
- Git Manager: Stanley
 - Git Manager is in-charge of merging the different branches to master.
- LaTeX Editor: Zhi Zhang
 - Latex Editor is in-charge of ensuring that the documentations of the project are formatted well in LaTeX.

The members are aware that the roles can change and each member should be clear of which role he/she is taking.

5 Git Workflow Plan

We will adopt a feature-branch workflow plan. We will have a separate development (and the feature branches) and release branches.

5.1 Label Usage

Upon merging the different branches to master, we will tag the version of the product (for example ver 0.1, ver 1.2, etc).

5.2 Milestones

We have determined several key milestones in our project development and each milestone is broken down to several tasks. Currently we have planned for the following milestones (changes may be made along the way):

- Requirement document
- POC demonstration
- Version 1
- Version 2
- Testing
- Final Version

6 Proof of Concept Demonstration Plan

For our team's proof of concept demonstration, we would like to show the feasibility of the following:

- Creating a working tower class, the attacking unit that users can put on the field, that is capable of firing projectiles only within range.
- Creating a working enemy class, the unit that will walk through the designated path and have a hit point (HP).
- Ensuring a proper projectile collision detection between the projectile to the enemy.
- Creating a good layout design for the game.

7 Technology

The programming language we are using are Python 2.7 and pygame 1.9.3. For IDE we are using the Python default IDLE. Testing framework will be Python unittest. And we will use LaTeX for document generation.

8 Coding Style

The team understands the importance of having a coding standard. We will be following the Google Python Style Guide to ensure that the code that the team generate is consistent throughout.

9 Project Schedule

Please refer to Tower-Defender/ProjectSchedule/projectSchedule.gan and Tower-Defender/Doc/Rev1/DevelopmentPlan//ProjectSchedule.pdf

10 Project Review

The team communicated well developing the project and split work evenly between team members. Each team member completed his/her assigned work as expected, and learned the new language Pygame through the development. The design met all the functional and non-functional requirements of the original game, with addition of the team's own idea of game theme, game interface and optimization of game functionality such as adding three types of dragons that each has different effects on enemy.