SE 3XA3: Development Plan
Mini-Arcade
Team #104

Andrew Hum	William Lei	Arshan Khan	Jame Tran
huma3	leim5	khana172	tranj52
400138826	400125240	400145605	400144141

Table 1: Revision History

Date	**Developer(s)**	**Change**
1/29/2020	Andrew Hum	Sections 1 - 3
1/30/2020	William Lei	Sections 4, 7
1/31/2020	Arshan Khan	Section 5, 8, 9
1/31/2020	Jame Tran	Section 5, 6
4/2/2020	Andrew Hum	Revision 1
4/4/2020	Arshan Khan	Revision 1

This document provides information regarding the groups plan for the development of the product. This document will be referenced by the development team throughout the development process.

1 Team Meeting Plan

Typical meetings will occur during lab time on Mondays and Wednesdays from 9:30 am - 11:20 am in room 236 in the Information Technology Building on campus. These meetings will occur weekly with the exceptions of breaks. During these meetings, Arshan leads the group as the project manager, however, the group works as equals to assign appropriate work to each individual. If there is work left uncompleted after our allocated meeting slots, additional meetings will be scheduled amongst the team members. The individual in charge of the meeting agenda will switch each meeting and be in charge of recording minutes, written statement on decisions made and homework designations.

2 Team Communication Plan

The team's primary means of communication is through a group chat on Facebook. This allows easy communication to the entire group, as well as individual conversations if needed. For issues related to the software, we may use Git Issues to notify one another on current issues that need resolving. Each group member can contact one another through email if necessary.

3 Team Member Roles

Arshan Khan - Project Manager, Game Developer, Tester
Andrew Hum - Architect, Game Developer, Tester
William Lei - Git Expert, Launcher Developer
Jame Tran - Developer, Tester

4 Git Workflow Plan

The git workflow of the team will be centralized and branch-based. There will a total of 6 branches: master; development; and a personal branch for each team member.
The master branch is the central and main branch, it will always contain production-ready code. Team members cannot directly commit changes to the master branch, as it can only be updated by merging changes from the development branch. The main branch will be labeled by a version number along with a title, which will be updated after every merge.
The development branch is the main branch that will be used for the development of the project. It is initialized by branching off master and can be reset to master when needed.

Similar to master branch, team members cannot directly commit to this branch, as it can only be updated by merging from their personal branch.
Each team member will have a personal branch that only that person has the permission to commit to. This is the branch that the team members commit their code to during the development.
Each member can update the development branch anytime by merging their personal branch if the code in their personal branch is up to date with the changes made in the development branch (by other team members) and fully operational. When a milestone or a certain checkpoint is met, upon the agreement of all team members, the master will then be updated by merging from the development branch, and its label will be updated with a newer version number and a new title.

5 Proof of Concept Demonstration Plan

The most significant risk for our game refresh and launcher creation will be that recreating multiple games with refreshed visuals and different levels of difficulty will prove to be a challenge, especially in the given time frame of the project.

Testing the game may prove to be a challenge. We plan to have at least three games available to play, this means each game as well as the launcher needs to be sufficiently tested. While small unit tests can be quickly written using pytest, any system testing will have to be conducted manually by playing the game, or by automating parts of the game, which can be extremely time-consuming. Furthermore, any UI components or concerns, such as if the game runs pleasingly, will also have to be tested manually.

The required libraries are not too difficult to install, the most challenging aspect will be the actual programming of the games. That said, portability is expected to be very seamless as the project is intended to be downloaded as an executable file online. Once the executable file is downloaded, the game will install itself.

To overcome the above risks, we will spend time slowly building the games in advance. This is shown in the Gantt Chart in Section 8, where we spend nearly two weeks on the proof of concept to have one game and the launcher to be functional. Another risk to overcome is the amount of testing that needs to be done. We will spend time building test cases as we have learned in our other courses. Making good use of the many features of pytest such as mock patches, as well as following development techniques such as test-driven development will allow us to write quick, efficient unit tests. For system testing, the games being developed are not overly difficult, and can feasibly be beaten by the developers. We will be sure to allocate adequate time to manually play and test the games. The final risk we will face is packaging the game properly to install correctly on different machines. We will test this on multiple computers as well as different operating systems to ensure correct installation.

6 Technology

The project will be developed using Python 3. Python was chosen due to a large number of modules available for development, as well as its familiarity with the team developers. In this project, we will be using Pygame, which is a python wrapper for the SDL library, which controls the mouse, keyboard, sound, and video on the user's system. Pygame also adds functionality to the SDL library by providing collision detection, vector math, and MIDI support.

For IDEs, we will be using the Visual Studio Code text editor (VSCode). VSCode is a powerful, lightweight text editor with extensions that allow real-time linting, code style enforcement and debugging. Any additional tools can be run from the terminal.

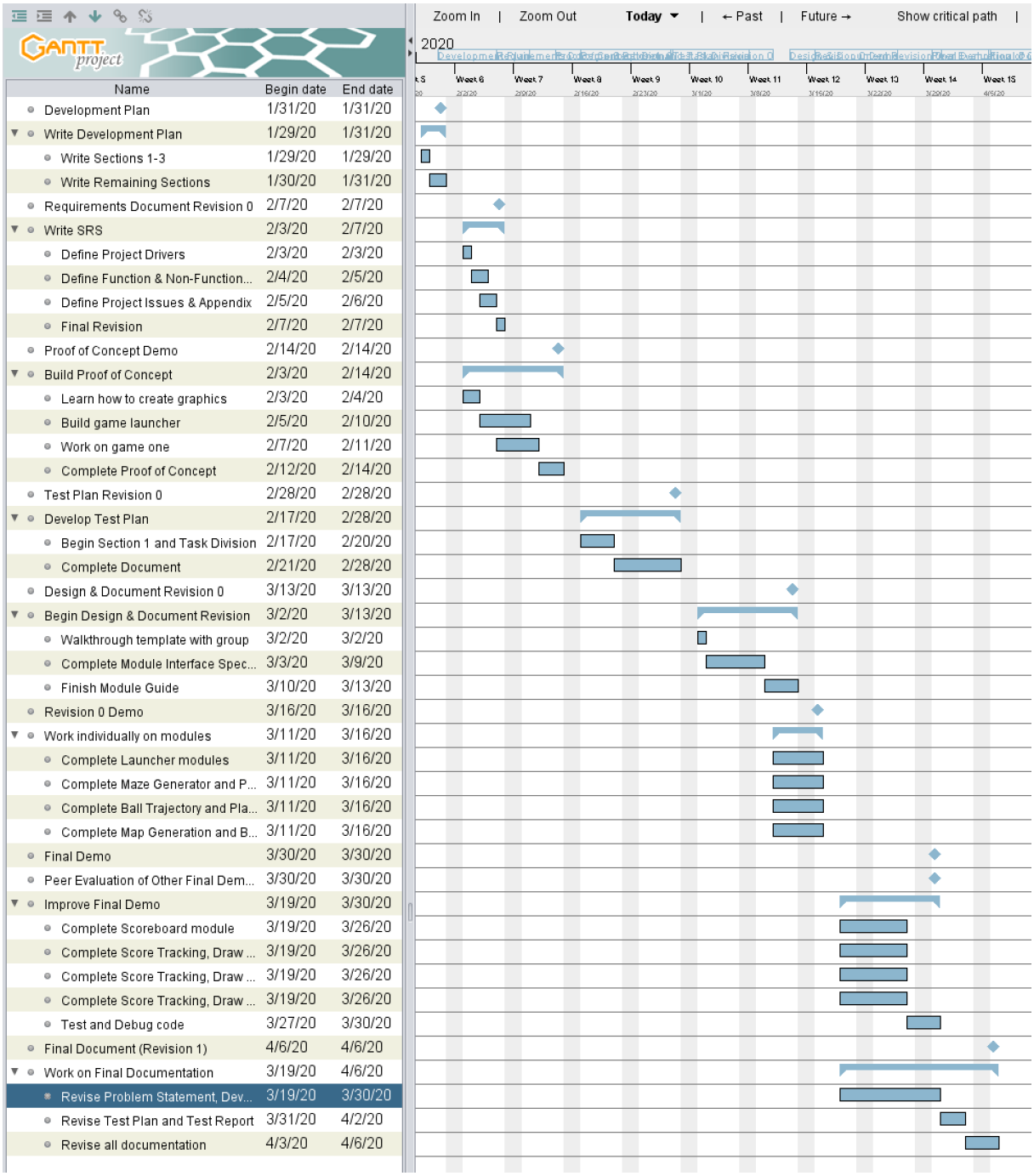
For the testing framework, we will be using pytest. pytest is a lightweight testing framework that provides several useful features, such as parameterized tests.

For document generation, we will be using Doxygen. Doxygen is a documentation generating tool that is run on specially annotated Python code. Doxygen can then create an online document browser using HTML, or an offline reference guide using LaTeX.

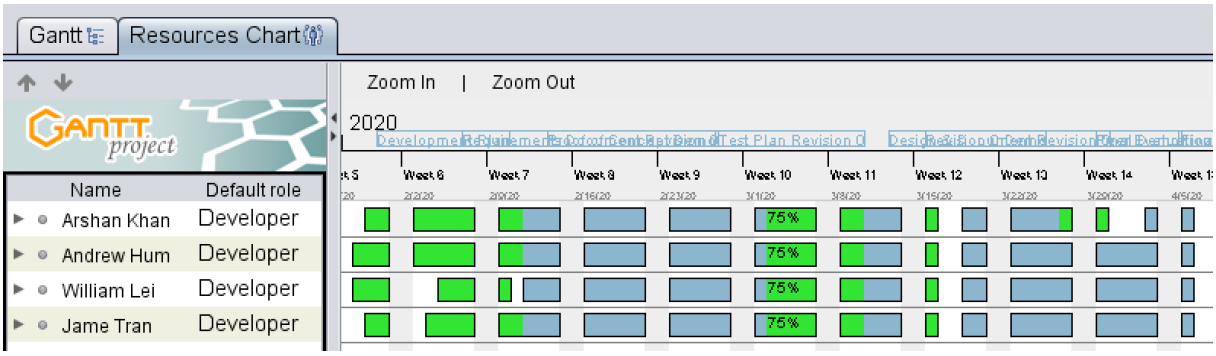
7 Coding Style

PEP 8 will be used as the coding style for this project. The detail style guide can be found on the official website of Python at http://www.python.org/dev/peps/pep-0008.
Doxygen will be used as the documentation and commenting style guide for this project. The detail style guide can be found on the official website of Doxygen at http://www.doxygen.nl/.

8 Project Schedule



Name	Begin date	End date
Development Plan	1/31/20	1/31/20
Write Development Plan	1/29/20	1/31/20
Write Sections 1-3	1/29/20	1/29/20
Write Remaining Sections	1/30/20	1/31/20
Requirements Document Revision 0	2/7/20	2/7/20
Write SRS	2/3/20	2/7/20
Define Project Drivers	2/3/20	2/3/20
Define Function & Non-Function...	2/4/20	2/5/20
Define Project Issues & Appendix	2/5/20	2/6/20
Final Revision	2/7/20	2/7/20
Proof of Concept Demo	2/14/20	2/14/20
Build Proof of Concept	2/3/20	2/14/20
Learn how to create graphics	2/3/20	2/4/20
Build game launcher	2/5/20	2/10/20
Work on game one	2/7/20	2/11/20
Complete Proof of Concept	2/12/20	2/14/20
Test Plan Revision 0	2/28/20	2/28/20
Develop Test Plan	2/17/20	2/28/20
Begin Section 1 and Task Division	2/17/20	2/20/20
Complete Document	2/21/20	2/28/20
Design & Document Revision 0	3/13/20	3/13/20
Begin Design & Document Revision	3/2/20	3/13/20
Walkthrough template with group	3/2/20	3/2/20
Complete Module Interface Spec...	3/3/20	3/9/20
Finish Module Guide	3/10/20	3/13/20
Revision 0 Demo	3/16/20	3/16/20
Work individually on modules	3/11/20	3/16/20
Complete Launcher modules	3/11/20	3/16/20
Complete Maze Generator and P...	3/11/20	3/16/20
Complete Ball Trajectory and Pla...	3/11/20	3/16/20
Complete Map Generation and B...	3/11/20	3/16/20
Final Demo	3/30/20	3/30/20
Peer Evaluation of Other Final Dem...	3/30/20	3/30/20
Improve Final Demo	3/19/20	3/30/20
Complete Scoreboard module	3/19/20	3/26/20
Complete Score Tracking, Draw ...	3/19/20	3/26/20
Complete Score Tracking, Draw ...	3/19/20	3/26/20
Complete Score Tracking, Draw ...	3/19/20	3/26/20
Test and Debug code	3/27/20	3/30/20
Final Document (Revision 1)	4/6/20	4/6/20
Work on Final Documentation	3/19/20	4/6/20
Revise Problem Statement, Dev...	3/19/20	3/30/20
Revise Test Plan and Test Report	3/31/20	4/2/20
Revise all documentation	4/3/20	4/6/20



9 Project Review

Will be completed in Revision 1.