SE 3XA3: Development Plan
Spaceshooter Remix

Team # 4, IRS Development
Ibrahim Malik maliki2
Ryan Schnekenburger schneker
Saad Khan khans126

Table 1: Revision History

Date	**Developer(s)**	**Change**
September 19, 2018	Ibrahim Malik	Creation of Document
November 24, 2018	Ibrahim Malik	Updates to document addressing TA's comments for Rev0

Our goal was to reimplement a modern version of a very classic game that an average computer user can enjoy on their local machines. This document is a development plan for our project that outlines the several key aspects of meeting our goals at the end of the semester. It was created after considerations from each member and is intended to be followed by the entire team.

1 Team Meeting Plan

The majority of the meetings can be held during labs on Tuesday night and Wednesday night. Each member lives on campus and is available from Monday-Friday. Weekend meetings will be done on Sunday night over online communications after handing out assignments to each individual for their portion of the project on the in-person team meeting on Wednesday. Towards the end of the semester, more meetings can start to occur preferably Thursday evenings after class at a location that is most convenient for everyone, most likely the Mills memorial library. At each meeting, the agenda will consist of overseeing previous developments, troubleshooting issues in those developments and then discussing how to proceed with the next phase of the project. After this, new tasks will be assigned to be completed before the next in-person meeting in Tuesday labs. Every member should complete all their assigned tasks and then present their challenges or feedback after during the Thursday meetings. Ibrahim has volunteered to chair the meetings initially to facilitate proper communication and flow. Ideally, meetings shall be at 45 minutes or so and this will be set aside by the members. Meetings will end wit ha written statement on decisions made and which members will be assigned which tasks.

2 Team Communication Plan

Each member has access to the git repo and has a responsibility of updating and pull information. Primary online conversations and meetings will be done over Skype and Facebook over the weekend. While in-person meetings are more for discussions of assigning tasks, actual coding and problems that arise on a day to day level will be discussed online.

3 Team Member Roles

There is no team leader because it is a relatively small group of 3 people and we all understand the responsibilities of this project. Ibrahim has volunteered to chair the meetings. Ryan has volunteered to be a scribe for the meetings to record the meeting's important discussion points. Ibrahim will focus the bulk on documentation has he has the majority of the compatible Latex software. Saad is in charge of running the git repo. The rest of the technology and coding will be shared equally by the members and each member has the newest version of Python installed on their computer.

4 Git Workflow Plan

For simplicity, we will avoid assigning branches until actual coding. Each member will ensure their local repo is updated and then each member will pull a branch to compile their code. Once they can ensure it will run without affecting other branches, they will merge this branch to the original repo. This will prevent errors in one person being uploaded to the entire repo and wiping that entire working copy out. Each person will be responsible for fixing any merge errors if they arise. There is an agreement in the team to always do a pull command before beginning any new task on their local machines to ensure that their local copies are up to date. Communication will be handled on Facebook and it provides an opportunity to alert other members of potential issues arising from the repo. For every milestone, once the assigned task is completed by the member, they will announce on Facebook and push their local copy. This way, each task completion will not interfere with corrupting the remote repo. We do not anticipate any problems in the Git Workflow because there each member has considerable experience working with GitLab and GitHub.

5 Proof of Concept Demonstration Plan

~~There might be difficult in testing certain movements and characters. Since this is a game, it can be difficult to create test cases and boundary cases for all the different types of possibilities. There will be testing for all the movements of the characters and the interactive objects in the game as well. Each character would be replayed over in every level to ensure all their movements follow the requirements and ensure against any errors. Abnormal scenarios will be checked to see if it causes any crashes in the game. There will be clean room testing to begin programming after developing proper specifications, functionality testing to identify bugs, compatibility testing as each member has a Windows, a Mac and Linux and a lot of ad hoc testing to randomly test certain aspects of the game. Finally playtesting to analyze the gameplay itself include the levels of difficulty etc. will be used by asking other users to play the game. There will also be load testing to evaluate whether the game will slow down under the running of several apps on the local machine.~~

We intend to produce a working version of the Spaceshooter game that can be played using basic controls during the PoC demonstration. The current program from the original source code has compilation issues, installability issues and crashes often during certain cases in the game. Our program will be a demonstration of a working game that accepts user inputs from the keyboard and uses Pygame to generate corresponding responses without those issues. It will not contain any added levels, features or powerups or any other enhancements to the gameplay. It is currently difficult to compile the project right now as it requires Python 2.7 and we shall reimplement it to allow a fully working copy in 3.6. Pygame is also required to be downloaded and it is not difficult to install but it has to be done through pip3 commands and homebrew. Although many Python 2.7 files work on Python 3, this particular file was unable to run on a windows machine with a new python version. There has to be a further investigation to evaluate this error and hopefully, as we rebuild the game on our machines, we will discover any potential issues to running these files on different issues.

6 Technology

The bulk of the programming will be done in Python 3.6 and 2.7 using Python's built-in IDLE software. The documents will be produced in Latex. Documentation for the MIS and MG will also be produced using Latex and PyUnit is intended to be used as a framework for testing.

7 Coding Style

The members have all agreed to adhere to the ~~Deprecated~~ Google Python Style and follow that guide to ensure a proper, universally readable document. We choose this style because it allowed for a project that can be consistent across all the modules and is easy to implement in our code.

8 Project Schedule

This folder can be found in the Git repo under:
`https://gitlab.cas.mcmaster.ca/khans126/IRS_Space_Shooter/tree/master/ProjectSchedule`

The PDF link is:
`https://gitlab.cas.mcmaster.ca/khans126/IRS_Space_Shooter/blob/master/ProjectSchedule/IRSGanttChart.pdf`

The gantt file :
`https://gitlab.cas.mcmaster.ca/khans126/IRS_Space_Shooter/blob/master/ProjectSchedule/IRSGanttChart.gan`

9 Project Review

This project was an exercise in the software engineering process requiring each member to elevate their programming abilities while also tackling the important issue of proper documentation and design. A lot of thought went into executing a re-invented version of Spaceshooter with an increased emphasis on improving the core features of the game. We have summarized 3 short reflections from each member of the group written below:

1. Ibrahim: We learned to design the software based on the principles of the engineering starting from devising the ideas, the requirements, the type of testing we can do and the proof of concepts we can create. The process facilitated the development of a better, working game which was more efficient and spanned several modules. New features were added as the weeks went on without damaging the core behavior of the game. All group members felt the project provided a better understanding of working in a team setting to split tasks and manage priorities to deliver goals that lead to an even bigger ultimate goal.

2. Ryan: The distribution of work was such that several tasks were assigned to each member who carried them resulting in the final implementation of the game itself. There are still several features that we feel we can implement in the future in the game to continue to improve its quality. Nevertheless, we all felt like this was a collective success in bringing team members together to putting forth a group project.

3. Saad: There were always challenges that we encountered in producing the best possible game within the time and workload of courses we had. Yet, this gradual development process allowed us to break down a large-scale task into a series of smaller, manageable activities that we could accomplish on our own. This resulted in the production of a game we can be proud of, but most importantly the knowledge of software development from an engineering perspective was invaluable.