Table 1: Revision History

Date	Developer(s)	Change
Jan 22, 2020	Kanakabha, Lucia, Raymond	creating problem statement
Jan 24, 2020	Lucia	adding notes for Workflow and PoC
Jan 28, 2020	Kanakabha, Lucia, Raymond	finalizing workflow, PoC and coding style
Jan 29, 2020	Kanakabha, Lucia, Raymond	finalizing PoC and adding GANTT Chart
Mar 28, 2020	Lucia	Revision 1 updates

SE 3XA3: Development Plan
Scrabble

Team #214, The Trifecta
Kanakabha Choudhri, choudhrk
Raymond Tu, tur1
Lucia Cristiano, cristial

January 31st 2020

This is the Development Plan for the Scrabble project created by Team Trifecta.

1 Team Meeting Plan

~~Team meetings will occur weekly during SFWRENG 3XA3 Lab Section 2 on Tuesdays~~. Team meeting will occur weekly on Mondays from 12:30-1:30pm in Thode Library on the McMaster campus. These meetings will be twenty to forty-five minutes in length depending on the needs of that week. Extra meetings will be scheduled as needed throughout the term. The meetings will be chaired by Lucia Cristiano. The chair will create the meeting agenda each week and ensure that all team members have input during the meeting. Each meeting Raymond Tu will act as scribe and take the necessary meeting minutes. The scribe will create a meeting minute document that will be filled in during each meeting.

The meetings will follow the format specified by the agenda. Firstly, the group will perform a check-in to see how the tasks that were assigned in the last week progressed or if they have been completed. The team will then reflect on past milestones and internal goals to talk about how they went and what lessons can be learned. Then the team will focus on immediate goals, coming up in the next one to two weeks. Finally, the group will look ahead to the future and brainstorm and assign individuals to work on future milestones and goals.

2 Team Communication Plan

The team will be using Git Issues to track and communicate errors in code, request edits on documentation and highlight any portions of code that need fixing. Essentially Git Issues will be used for anything pertaining specifically to code and documentation of the project.

As a second form of communication Facebook Messenger will be used for scheduling meetings, reminders, and general talking.

3 Team Member Roles

~~The roles of the team will be as follows. The team leader will be Kanakabha Choudhri. Team meetings will be chaired by Lucia Cristiano and during the meeting minutes will be taken by Raymond Tu, the scribe.~~

Table 2: Team Roles

Team Member	Role
Kanakabha Choudhri	Team Leader, Design Lead, Documentation Expert, Developer
Lucia Cristiano	Meeting Chair, UI Developer, Developer, Tester, Documentation Writer
Raymond Tu	Meeting Scribe, UI Developer Testing Lead, Developer, Documentation Writer

4 Git Workflow Plan

The workflow plan for project is loosely based on the article referenced in the lecture slides [1].

For the Scrabble project several branches will be used such as a master branch, development branch, and release branch. A master branch will be used for the main releases of the project such as Revision 0 and Revision 1. Only code items that are ready for release will be merged or pushed to master.

A development branch will be used to allow developers to work on code that is actively being prepared for future release. This branch merges back to master once ready for release.

A release branch will be branched off of the development branch once the project is close to a release. In this branch, last minute bug fixes and edits will be completed by developers. Once the project is ready for release, this branch is merged with master. The branch is also merged with development branch to ensure that all additions to the release are available for the next iteration.

5 Proof of Concept Demonstration Plan

5.1 Most significant risks

1. **Difficulties in Implementation**
   Since the Scrabble project is a game, team trifecta decided to use the Model-View-Controller(MVC) design architecture as this architecture is commonly used for games. Using MVC will ensure that the separate aspects of the game work cohesively with each other. The group members have limited experience using the MVC architecture, thus it may be a challenge to implement. Besides implementing this architecture, the project has a GUI component that will be created in Tkinter. No member of the group has used this software before, so there will be a learning curve associated with using this library.

2. **Difficulties in Testing**
   Testing will be difficult as we have to test it based on user behaviour. This requires a lot of play through and exploratory testing, to reveal issues and bugs. Exploratory testing can be a lengthy process and require and indidividual outside the project to test the software.

3. **Difficulties in Installing Libraries**
   The only library which may possibly give issues is Tkinter. Although Tkinter comes packaged with versions of Python 3, some users experience issues with having windows created using this library appear on the screen.

4. **Difficulties in Portability**
   Portability should not be a concern, any computer with Python 3 installed should be able to run the project. There is, however, potential for issues with Tkinter, as mentioned in the previous section.

5.2 Demonstration on How Risks Will be Overcome

1. **Implementation**
   To ensure proper implementation of the MVC architecture, the team members will conduct research, such as looking at examples and reading articles.
   To ensure that implementation goes smoothly the team members will go through tutorials found online to familiarize themselves with the Tkinter library. As the team members practice and understand how to use Tkinter, implementation will be easier.

2. **Testing**
   To ensure that there will be time for exploratory testing, the group members will test incrementally using unit tests as features are added to the project. Once the individual modules are tested, the team can focus efforts on exploratory testing to provide a good user experience. The group

will also ask colleagues to play the game as a form of exploratory testing.

3. **Installing Libraries**
   An additional software known as Windows Xming can be downloaded and a line added to the .bashrc file of the computer to remedy the issue of windows not being visible.

4. **Portability** To fix this issue, the steps mentioned in the Installing Libraries section can be followed.

5.3 Contents of the Proof of Concept Demonstration

During the proof of concept demonstration, a version of the interface without any connection to the back-end of the project will be shown. Demonstrating this GUI will prove that the team has gotten some familiarity with the Tkinter library, enough to make a viable interface. This addresses the risks associated with the implementation of the project. As well as ensuring that any risk regarding the installation and usage of the Tkinter library are handled. Creation of this interface will reduce the risk associated with testing as the functionality and usability of the interface must be tested before the demonstration. These tests result in less testing later in the life of the project.

Besides the showcasing the interface the team members will come prepared to answer any questions asked of them during the demonstration. This will be ensured through research as mentioned in Section 5.2, as well as all members of the project keeping up to date via meetings and the project GANTT Chart.

6 Technology

The project will be written in Python version 3 as the primary programming language. The IDE for coding will be based on the preference of the individual developer. Some team members prefer to use Notepad++ and others use VS-Code. For unit testing the project the team will use the PyTest as the testing framework. For the project documentation Doxygen comments will be used and automatically generated as a way of documenting the code for the project. Team Trifecta chose to use Python, PyTest and Doxygen as the main technology in our program as all the team members had experience with these platforms through coursework in the past years of university.

7 Coding Style

The coding style that will be used by team Trifecta for the Scrabble project is based off of the Pep 8 style guide for Python [2]. For indentation four spaces will be used rather than tabs. Each line can only have 79 characters maximum. All source files will be encoded in UTF-8 standard. Import statements for libraries will be on separate lines. For consistency, we will be using only single quotes for strings. White space will be avoided for consistency and readability. Trailing white space will also be avoided. Binary operators will be surrounded by a single space on each side for readability. Variable and class naming will follow Pep 8 conventions, with CamelCase for class names and variable names. Global variables will be named with the convention of prefixing with underscores. Constants will be named using all upper case letters.

In order to ensure consistency and that these rules are followed a linter called flake8 will be used to ensure that the code for the project is consistent and follows the rules of the Pep 8 style guide.

8 Project Schedule

[Click Here for Link to Gantt Chart]

9 Project Review

The Scrabble project when considered as a whole was done well and the code and documentation milestones associated with the project were completed in a timely and effective manner. The team members managed to overcome the hurdles and risks mentioned in this document regarding the usage of an unfamiliar library, Tkinter. As well as setting aside time for performing exploratory testing for the majority of the project. Overall, the project managed to stay on task for all the milestones as the team took the time to split up work long before the due date allowing each piece of documentation and module of code to be completed on time. A major challenge for the project was determining how to connect the front-end portion of the project with the back-end portion. None of the team members had experience with doing full stack development and there were few tutorials and guidelines to help with the process. Eventually, through researching some full stack methods and extensive testing the team was able to get connect the front-end and back-end portions of code to connect in a modular manner that did not compromise on performance. Another challenge was seen in deriving test cases for the project, as found in the test plan. Due to the random nature of Scrabble, (random tiles are given to each player), most of the testing was done manually. This made it harder to describe tests in systematic manner as manual testing often relies on implicit knowledge of the system that the tester is aware of, but hard to describe in a document.

In the future the team would modify the development plan to consider the risks regarding full stack development, as only front-end development was mentioned as a risk for the project. Another modification would be to assign specific roles such as "Testing Lead" and "UI Developer" so that once it came time for implementation, each individual already knew their area of focus. Rather than wasting time splitting up work, the team members could simply start on their next task. The team would also change the method in which the team meeting occurred. With the recent events, the team realized the benefits of using platforms such as Discord and MS Teams as an alternative way to hold team meetings, that did not rely on meeting in person. For the future the team would have a short online ”check-in” to update on tasks and a bi-weekly person meeting that discussed deliverables and the future of the project. Finally, the team also would have better time management when it came to coding the modules, as coding often happened in short spurts, rather than a continuous development over the life of the project. To remedy this the team would use a PERT chart to determine the dependencies between modules and thus have a plan to complete modules over a longer amount of time.

All in all, the team enjoyed working together and is ultimately proud of the final product of their Scrabble project. The team members learned a lot about project management and improved their abilities as developers.

References

[1] V. Driessen, “A successful git branching model,” Jan 2010. Accessed on 2020-01-28.

[2] G. Van Rossum, “Pep 8 – style guide for python code,” Aug 2013. Accessed on 2020-01-28.