

Table 1: Revision History

Date	Developer(s)	Change
Jan 26, 2020	Alexander Samaha	Set-up document, initial changes
Jan 28, 2020	Daniel Noorduyn	POC and Roles Rough Draft
Jan 28, 2020	Alexander Samaha	Technology and Communication Rough Draft and Final
Jan 29, 2020	Daniel Noorduyn	POC and Roles Touched Up
Jan 29, 2020	David Jandric	TMP, GWP, CS Rough Draft and Final
Jan 29, 2020	Alexander Samaha	Project Schedule completed and linked
Jan 31, 2020	Daniel Noorduyn	Finshing Touches
April 6, 2020	Daniel Noorduyn	Rev 1 Touch-ups

SE 3XA3: Development Plan super-refactored-mario-python

Team 203, Abstract Connoisseurs
Daniel Noorduynd, noorduyd
David Jandric, jandricd
Alexander Samaha, samahaa

This document lays out the plan for developing, testing and reviewing our project.

1 Team Meeting Plan

Meetings will take place every Tuesday and Wednesday in ~~ITB 236 from 2:30PM to 4:30PM~~ **Thode Library from 4:30PM to 5:30 PM**. Informal meetings will be held on Thursdays at 6:00PM, if required. Alex will log, Dan will be chair, David will act as technology expert. Rules for agendas:

1. Items must be pertinent to next deliverable.
2. All team members should have some input into the direction of the meetings.
3. Items in agenda should be posed as questions.
4. Give times for each item.
5. Agenda should be shared with each member a day before the meetings.

2 Team Communication Plan

The development group have all exchanged Facebook Messenger credentials, phone numbers and academic email handles. This allows the team three channels of communication outside of the predetermined weekly meetings.

The team has set up a Facebook Messenger group chat for discussing work related activities among all group members. Messenger provides an important avenue to discuss minor issues related to the project, in contrast, the weekly meetings will be used to discuss major issues and activities in detail. This could involve current bugs or issues with the program, blockers where a team member is unsure how to proceed, as well as current updates on development of

features. Casual work discussion not fitting the previous criteria would also be appropriate on Facebook Messenger. Update conference calls will also happen on this platform through the conference call functionality. This should occur before a development sprint and throughout the development cycle to update team members on features to be completed and manage a prioritization schedule.

More time sensitive issues involving individual team members will be pursued through telephone communication. Phone calls or text messages would be appropriate to communicate issues such as critical blockers that need to be fixed.

The team members have access to GitLab through our academic emails. This directly connects the team to the source code-base to communicate merge requests and issues. Real-time or synchronous messages will be avoided with this channel since it is not likely to elicit a fast response relative to phone and Facebook Messenger communication. Furthermore, our academic emails open a direct channel between our team and supervising TA. This will be used to pass information both ways regarding different stages of the project and feedback on progress.

3 Team Member Roles

A project with this level of coordination required amidst developers and designers results in the need for a chair (or team leader). The chair's responsibility is making sure each team meeting runs smoothly within the allotted time frame and makes sure each team member is communicated with each other during the designing and building phases. The team has democratically voted in Daniel Noorduynd to fulfill this position. During the team meetings, a scribe will be required to jot minutes down. These minutes will be referred to by individual team members to recall all team decisions during individual work. The team has democratically voted in Alexander Samaha. During the design phase, a technology lead is required to determine if certain project aspects are feasible. The team has voted that David Jandric fulfill this position.

Each member of the team has unique expertise that will benefit the overall project. The breakdown is in the following table:

Old table:

Team Member	Expertise
David Jandrie	Documentation
Alexander Samaha	Git
Daniel Noorduynd	LaTeX
David Jandrie	Technology
Alexander Samaha	Design

New table:

Team Member	Expertise
David Jandric	Documentation Technology
Alexander Samaha	Git Design
Daniel Noorduyn	LaTeX

Table 2: Team Member Expertise

These roles and expertise fields are subject to change as different challenge arise during the project. Each team member will be filling in multiple roles from the start, and therefore understand how fluid the assigned roles are.

4 Git Workflow Plan

Our repository is centralized around the master branch, and new branches are created for different features and modules. ~~This type of plan is called a Feature Branch workflow plan. These branches are~~ Each branch is destroyed once merged with master. Labels will be used in order to track bugs, stylistic changes, documentation requirements, issues that may require discussion, and suggestions. We will be using milestones in order to track the due dates of the deliverables.

5 Proof of Concept Demonstration Plan

The ambitious goal of creating a Super Mario Bros game that is accessible on every platform results in some possible hurdles that will need to be overcome if the project is seen as feasible. We the team want to create a game that blends elements from the original Super Mario Bros with features from the newer versions. However, considering the popularity of the game, both past and present, users will have strong opinions as to what features the game should possess and how they should be implemented. Therefore, a minimum viable product (MVP) will be created that users can interact with and critique. This feedback will help us continue creation of the game in a way that results in a wanted final product.

There are also some strictly technological issues that will have to be addressed. Our game will be created using PyGame, a Python library. This library is difficult to install and run properly on an Apple Computer. Therefore, even though Python is compatible with most operating systems, portability with respect to Apple computers will be an issue due to the incompatible library. The implementation of our game is quite straightforward and feasible, but difficulties may arise when many different methods must work together in real time to create a seamless game-play for the user.

There will be difficulties in testing our product as it relies heavily on human input. Therefore, multiple individuals are needed to play the game and report

any bugs. It is important to note that not all testing will be done by external players, as different methods and algorithms can be tested individually. This logical testing will be performed using PyTest. However, for testing overall interaction of the program attributes, we will have to look outside ourselves.

To present proof that the previously mentioned risks can be overcome and the game can be developed, we will have a demonstration. To underline that the library issue was overcome, this demonstration will be run on a Mac-book. The demonstration will consist of a 2D plane where in a character can be controlled to move left and right as well as jump up and fall down. If this can be achieved the overall project should be able to be completed.

6 Technology

Programming Language:

The programming language to be used for the project is Python. We will be taking advantage of the PyGame library which is beneficial for rapid prototyping and smaller scope games, it is also the most widely recognized game development library through Python. The game we are producing does not require intensive resources and as such, Python utilizing the PyGame library would be adequate to satisfy the games requirements. The original project is already utilizing this language and library which allows the team to efficiently continue from what was left off. This solution allows for an easy set-up for users of the project, requiring the Python interpreter to be installed along with a version of PyGame through the pip installer.

IDE:

There will be no mandatory IDE to develop the project, however, the preferred development environment will be Microsoft VSCode. Although not a dedicated Python development platform, VSCode provides many advantages that make development more efficient and convenient. VSCode houses tools that perform linter checks and code completion for Python, helping the development team release higher quality and consistent code. VSCode also has terminal support which speeds up the process of testing the code since it can be run in-app. The chosen IDE also supports Git integration, which encourages best practices such as branching and frequent code committing to the source code repository.

Testing Framework:

Testing during the course of the project will be achieved using pyTest and manual testing. pyTest is a popular Python testing framework that allows us to perform dynamic white box testing and ensuring appropriate code coverage for all parts of the program. The team already has experience using this framework which makes it easier to implement for this solution. Manual testing will be

done to ensure that in-game assets and game-play are behaving as expected when run.

Documentation:

The project will be documented in full using LaTeX and made available for all users with access to see on GitLab. LaTeX allows the development team to create efficient and consistent documents that can be easily compiled into PDF format. All development team members have extensive use of LaTeX in previous projects. For code documentation, proper development practices related to code commenting will be followed. In-line comments will be used to describe the function and use of the sequence of code. Further, Doxygen will be used to encompass the functionality of the entire source code base. This will allow us to track the logic of each class, method and field across the whole project and act as a reference for future development.

7 Coding Style

Since our project will be coded in Python, we will be following the [PEP8](#) style standard for Python, which comes from the creators of Python. A coding style like this can be helpful by keeping the structure and look of our code consistent, which helps with maintaining the code base. Using a consistent coding style can also help other developers read and understand the code.

8 Project Schedule

A link to our project schedule can be found in the project repository linked below:

Link to Gantt chart and Resource chart

This Gantt chart is updated throughout the project

9 Project Review

Not Applicable.