

Table 1: Revision History

Date	Developer(s)	Change
Jan 26, 2020	Alexander Samaha	Set-up document, initial changes
Jan 28, 2020	Daniel Noorduynd	POC and Roles Rough Draft
Jan 29, 2020	Daniel Noorduynd	POC and Roles Touched Up

SE 3XA3: Development Plan

Title of Project

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

Put your introductory blurb here.

1 Team Meeting Plan

By: David Need to include : When? Where? Frequency? Roles? Rules for agendas? Every Tuesday and Wednesday in ITB 236 from 2:30PM to 4:30PM. Informal meeting every Thursday at 6:00PM, if required.

David and Alex will log, and Dan will be chair.

Rules for agendas:

1. Items must be pertinent to next deliverable.
- 2.

Rubric: 5 marks.

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:

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

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

By: David

Pretty simple, how are our branches organized, is our repository centralized?
How do we label everything, how will milestones be used?

Rubric: 3 marks

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. A possible solution is creating an exe file for the users to run, which bypasses the need for users to download libraries. 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 gameplay 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 afore-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 MacBook. The demonstration will consist of a 2D plane where 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 gameplay 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

By: David We should follow a coding style, this can involve but its important that the code is consistent. Someone needs to research some python coding styles. Then we talk about it and how it can be helpful!

Rubric: 2 marks. Should be enough to just introduce the style and what advantages it brings.

8 Project Schedule

By: All Provide a pointer to your Gantt Chart.

Rubric: 5 marks. Needs to be detailed and accurate (realistic).

9 Project Review

By: All Looks like this is necessary for our revisions.

Rubric: 1 mark