

# Escape Fate (Horror Game)

Software Project Management Plan



Fall 2022  
Version 2.0

Nicholas Montalvo (PM)  
Daniel Alcala (APM)  
Daniel Calderon  
Logan Ashbaugh  
Valerie Mora  
Peter Leiva

## Table Of Contents

<b>1. <u>Overview</u></b>	
<u>1.1 Project Summary</u>	4
<b>1.1.1 Purpose, Scope, and Objectives</b>	4
<b>1.1.2 Assumptions and Constraints</b>	4
<b>1.1.3 Project Deliverables</b>	4
<b>1.1.4 Schedule and Budget Summary</b>	5
<u>1.2 Evolution of the Plan</u>	5
<b>2. <u>References</u></b>	5-6
<b>3. <u>Definitions</u></b>	5-6
<b>4. <u>Project Organization</u></b>	7
<u>4.1 External Interfaces</u>	7
<u>4.2 Internal Structure</u>	8
<u>4.3 Roles and Responsibilities</u>	8
<b>5. <u>Managerial Process Plans</u></b>	8
<u>5.1 Start-up Plan</u>	8
<b>5.1.1 Estimation Plan</b>	8-9
<b>5.1.2 Staffing Plan</b>	9
<b>5.1.3 Resource Acquisition Plan</b>	9
<b>5.1.4 Project Staff Training Plan</b>	9
<u>5.2 Work Plan</u>	9
<b>5.2.1 Work Activities</b>	9
<b>5.2.2 Schedule Allocation</b>	10
<b>5.2.3 Resource Allocation</b>	10
<b>5.2.4 Budget Allocation</b>	11
<u>5.3 Control Plan</u>	11
<b>5.3.1 Requirements Control Plan</b>	11
<b>5.3.2 Schedule Control Plan</b>	11
<b>5.3.3 Budget Control Plan</b>	11
<b>5.3.4 Quality Control Plan</b>	11
<b>5.3.5 Reporting Plan</b>	11
<b>5.3.6 Metrics Collection Plan</b>	12
<u>5.4 Risk Management Plan</u>	12
<u>5.5. Closeout Plan</u>	12
<u>5.6 Gantt Chart</u>	13
<b>6. <u>Technical Process Plans</u></b>	13
<u>6.1 Process Model</u>	13
<u>6.2 Methods, Tools, and Techniques</u>	14
<u>6.3 Infrastructure Plan</u>	14

<u>6.4 Product Acceptance Plan</u>	14
<b>7. <u>Supporting Process Plans</u></b>	14
<u>7.1 Configuration Management Plan</u>	14
<u>7.2 Verification and Validation Plan</u>	14
<u>7.3 Documentation Plan</u>	14
<u>7.4 Quality Assurance Plan</u>	14
<u>7.5 Reviews and Audits</u>	14
<u>7.6 Problem Resolution Plan</u>	15
<u>7.7 Subcontractor Management Plan</u>	15
<u>7.8 Process Improvement Plan</u>	15

<b>Name</b>	<b>Date</b>	<b>Reason For Changes</b>	<b>Version</b>
Project Fate: Horror Game	9/2/2022	Creation of SPMP	1
Project Fate: Horror Game	10/2/2022	<ul style="list-style-type: none"> <li>● Table of contents reformatted</li> <li>● Gantt chart added to section 5.6</li> </ul>	2

# **1. Overview**

## **1.1 Project Summary**

This Software Project Management Plan will describe the details of the steps our team took in order to design and code for Project Fate the Horror Game. It includes the development cycle, organization, specific roles, projected timelines and testing protocols.

### **1.1.1 Purpose, Scope & Objectives**

The Software project management plan (SPMP) will present all the details regarding the development plan and the development cycle designing Project Fate (The Horror Game). This document will explain the assigned roles that each team member has regarding the development process of this video game. It will also show deadlines to tasks that our team members were assigned to finish this game within the final deadline. We will also be closely following the specifications included in our current version of the SRS that was developed. The first prototype of our game will include a fully functional UI design for our Main Menu as well as our Pause Menu. This UI will also have design details befitting to our Horror games theme. Lastly the prototype will have a designed environment that the player will be able to walk through and interact with no major detail added. The reason we are designing this type of game is to attract a genre of players that enjoy a fun single player experience with some horror aspects added in. We hope that our game will be fun and scary to those who dare to play.

### **1.1.2 Assumptions and Constraints**

The list of all assumptions and constraints:

1. Team members are expected to attend all meetings.
2. Team members will do their best to meet all deadlines set.
3. Team members will follow all requirements set by the SRS unless specified otherwise.
4. Team members will work on the game design outside of class and whenever available to do so.
5. The game will only be playable on windows version 10 and above.

### **1.1.3 Project Deliverables**

1. Clickable desktop shortcut that is playable on windows devices capable of running the game.
2. SRS, SPMP, Source Code, Maintenance Manual

### **1.1.4 Schedule & Budget Summaries**

There is no budget planned for this game. Everything will be designed using free engines and platforms. According to the class schedule the first prototype will be finished by week 7 and the final prototype for the game will be finished by the final day of the class.

## 1.2 Evolution of the Plan

The management team met with our client Jennifer Jin on the second week of classes. We had been assigned the idea of developing a video game. After further discussion with the client we had decided on the genre of the game to be a first person Horror game. After having discussed the genre and the storyline of the game we were ready to head into talks about the first prototype design. The management team put together a well designed SRS that explains how the layout design of the UI will look like. We have a very detailed schedule that was put together in order to keep us on track with tasks to have our final prototype of the game ready to play by the final deadline.

## 2. References

1. IEEE Software Engineering Standard Committee, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998.
2. Epic Developer Community. 2022. *Epic Developer Community Learning | Tutorials, Courses, Demos & More – Epic Developer Community*. [online] Available at: <<https://dev.epicgames.com/community/learning>>.
3. Awesome Tuts, 2021. *Monster Chase Game Tutorial Unreal Engine C++*. [video] Available at: <<https://youtu.be/DwjehZh5YQ0>>.
4. Isocpp.org. n.d. *Standard C++*. [online] Available at: <<https://isocpp.org/>>.
5. Visual Studio- Microsoft. “Visual Studio Code - Code Editing. Redefined.” *RSS*, Microsoft, 3 Nov. 2021, <https://code.visualstudio.com/>.

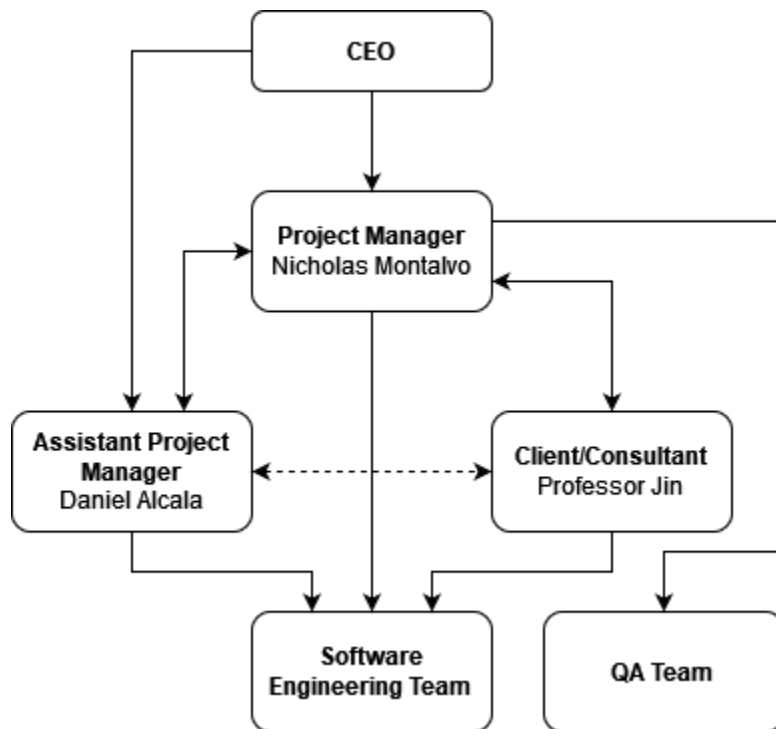
## 3. Definitions

- **C++** - is an object-oriented programming (OOP) language.
- **Computer** - A digital piece of hardware that can carry out mathematical calculations, functions, and operations.
- **Game Screen** - The page where the app is played, accessible from Main Menu.
- **Git** - An open source software that helps with version control, tracking files during development.

- **Github** - A internet hosting software that integrates Git to allow collaborative development and programming across multiple devices.
- **Launch Page** - The first page of the app that the user can interact with.
- **Main Menu** - The launch page for the app to access Settings, Game Screen, and more.
- **Mixamo** - A website that develops and sells services for 3D applications, including animations.
- **Pause Screen** - A function in a game that stops the game in the current state and allows the user to change Settings, quit, and potentially more.
- **Software** - A set of computer programs, documents, and data that the user can interact with easily.
- **SRS** - This Document, a Software Requirement Specification.
- **UI** - User Interface, the different elements the user can see and interact with.
- **UML** - Unified Modeling Language (UML), a standardized modeling language enabling developers to specify, visualize, construct, and document artifacts of a software system.
- **Unreal Engine** - Video game engine used for developing games on multiple platforms, especially PC.
- **Use Case Diagram** - A representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.
- **User** - Someone who uses the application or software specified in this Software Requirement Specification.
- **Video Game** - An electronic game that displays generated visual feedback when a user creates input through a device, such as a controller or joystick.
- **Visual Studio** - An integrated development environment developed by Microsoft that is used to develop programs in different languages.
- **Website** - A collection of web pages and similar content that can be accessed by a user under a domain.

## 4. Project Organization

### 4.1 External Interfaces



**CEO** - Overviews performance and development of the project and provides feedback and guidance to the managers of the project.

**Nicholas Montalvo (PM)** - Creates communication, develops ideas, and allocates roles to all members of the project team. Involved in making decisions to help further the development along, as well as helping with documentation and programming.

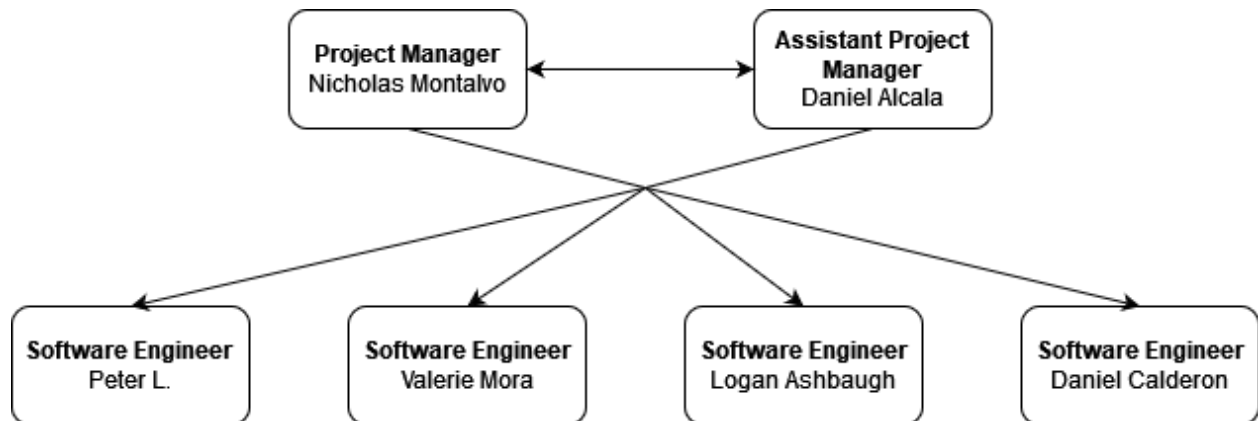
**Daniel Alcala (APM)** - Communicates with the PM to help support and foster ideas for the project team. Documents any changes, tasks, or dates completed by the team. Can help the PM with any duties that they might have to help make sure the project goes smoothly and according to the specifications. They may also help with documentation and programming.

**Professor Jin (Client/Consultant)** - They help with figuring out any rough decisions or issues the team or project managers may have. Can help gear a direction for the team if needed, and give insight to the managers on directions to run their project towards. They do not have any direct involvement with the project.

**Software Engineering Team** - The team that codes and develops the software for the project application.

**QA Team** - The team that tests and produces feedback for the project team to go over based on prototypes.

## 4.2 Internal Structure



A diagram showing the connection between all the members of the project, including the project manager, assistant project manager, and all software engineers.

## 4.3 Roles and Responsibilities

- Nicholas Montalvo (PM) - Communicates with the team to direct, produce, and conduct the project in the direction of completion. Assists the team with the mechanics, story, and theme of the video game.
- Daniel Alcalá (APM) - Collaborates with the project manager to assist in any decision making, as well as keep track of any meetings or logs the project produces. May help in any programming section if needed.
- Peter Levia (SE) - Sound Design for the game, including SFX, any music if implemented, and ambience noise.
- Valerie Mora (SE) - Environmental Design, which includes the surroundings of the level as well as the outside of the level.
- Logan Ashbaugh (SE) - UI design, which includes the main menu, pause screen, options, and any other components.
- Daniel Calderon (SE) - Character and Enemy Design, includes animations, 3D model, and the playable character. They are also helping with the main game code to make sure the game runs correctly.

## 5. Managerial Process Plans

### 5.1 Start-Up Plan

#### **5.1.1 Estimation Plan**

1. Discuss with the client all the specifications for the video game.
2. Research all the technologies required to accomplish those specifications.
3. Discuss with the team how much could actually be done within the given time.
4. Create guidelines for prototypes 1 and 2.



5. Make estimated deadlines for and task completion deadlines for both prototypes so that the project is done on time.

### **5.1.2 Staffing Plan**

We were able to form a full team for this project by discussing project ideas with each other during class time. We have had no new members join our team so we have been steady with 6 team members which has proven to be more than enough help.

### **5.1.3 Resource Acquisition Plan**

All the resources required for this project are available free for download. Such as the Unreal engine where we will be developing most of our game design. We will also be using Visual studio Code to develop our AI for the Enemy in the game. Github is going to be our resource control where we share our edited files of the game to the rest of the group. As of right now we do not have any budget plans for resources.

### **5.1.4 Project Staff Training Plan**

During the first three weeks of the class each team member is required to watch and learn what they can from the Unreal Engine tutorial videos. After having watched the video tutorials each team member is encouraged to learn any additional technologies that might be necessary in the future.

## **5.2 Work Plan**

### **5.2.1 Work Activities**

Work Units:

- **UI/UX Development Prototype 1:** The first prototype will have a fully functional UI such as the Main Menu screen and the Pause Menu screen. This will be developed by Logan and Nicholas.
- **Enemy AI:** The enemy AI will be a base model that will have all the functions necessary for our horror game. This will be developed by Daniel Calderon.
- **Sound Design:** Our sound design is necessary to help create the environment for the horror game. This will be included in our final prototype of the game and will be developed by Peter.
- **Final Prototype:** The Final Prototype of the game will include completed versions of the work previously stated. As well as completed environment design and a working playable character that the user can control.
- **Documentation:** All work will be documented in our Github so that the team can collaborate and work on the files together.

## 5.2.2 Schedule Allocation

9/27/22	9/29/22	Prototype #1 Coding (UI/Main Mechanics)
10/4/22	10/6/22	Prototype #1 Coding (Character/Sounds)
10/11/22	10/13/22	Prototype #1 Coding (Environment/Enemy AI)
10/18/22	10/20/22	Prototype #1 Coding (Bug Fixes)
10/25/22	10/27/22	Prototype #1 Presentation
11/1/22	11/3/22	Prototype #2 Coding (SFX Touch-Up)
11/7/22	11/9/22	Prototype #2 Coding (Score Counter)
11/15/22	11/17/22	Prototype #2 Coding (Environment Touch-Up)
11/22/22	11/24/22	Prototype #2 Coding (Bug Fixes)
11/29/22	12/1/22	Prototype #2 Presentation (Final)

## 5.2.3 Resource Allocation

Each team member will have access to the same amount of resources. Except for our team member Valerie who is unable to download Unreal Engine 5 on her laptop and has no access to a desktop due to work requirements. Other than that each member will have access to resources listed below:

- Visual Studio Code
- Unreal Engine 4.2.7
- GitHub
- Discord
- Asana
- Google Docs
- Google Sheets

## **5.2.4 Budget Allocation**

No budget has been allocated for this project.

## **5.3 Control Plan**

### **5.3.1 Requirements Control Plan**

All team members are required to attend every meeting to the best of their ability. Each team member is also to follow the guidelines and meet every deadline as stated in the SRS. IF there are any unexpected technical issues or change requests made by the client they are to be discussed first by the management team.

### **5.3.2 Schedule Control Plan**

The team is required to attend every class session to the best of their ability as well as any required outside meetings to stay on schedule. Our managers will be consistently checking in on team members to make sure that all tasks are being done correctly. As well as being completed on time. The project manager and the assistant manager will be in close contact with one another to keep up to date on all tasks and discuss any new deadlines that might come up.

### **5.3.3 Budget Control Plan**

There is no allocated budget for Project Fate (The Horror Game). However the Fate studios team will do everything in their power to use all the free resources necessary to complete this project without a budget.

### **5.3.4 Quality Control Plan**

The project manager as well as the assistant manager will be performing quality control checks on the game every other week. This will give the team time to fix any issues with the games software and continue to work on any other tasks assigned to them. The project manager will be working closely with the client to make sure any design choices made by the client are met as well. Before the 1st prototype is due the project manager and assistant manager will be performing a test run of the game such as UI main menu and pause menu testing.

### **5.3.5 Reporting Plan**

The project manager will be in direct contact with the client to keep them up to date on our progress of the game. The project manager will keep a record of attendance of all team members as well as write this SPMP to keep the client up to date on the plan for creating this game.

### **5.3.6 Metrics Collection Plan**

The managing team will make sure that every developer is completing their tasks for the week on time and help out wherever necessary. The managing team will also be testing the game software developed as it is completed.

## **5.4 Risk Management Plan**

### Development

- The team will be meeting regularly to check if there are any issues in the development process.
- There is a scheduled deadline for all tasks to keep the project running on time.
- Each team member will be kept up to date regularly, they will also keep the managing team informed of any issues that arise from development.

### Project Failure

- If the game can not be put together as described in the SRS. The team will get together and do its best to make the game a still fun and viable experience when playing.

### Server Failure

- There is no server in this stage of game development.

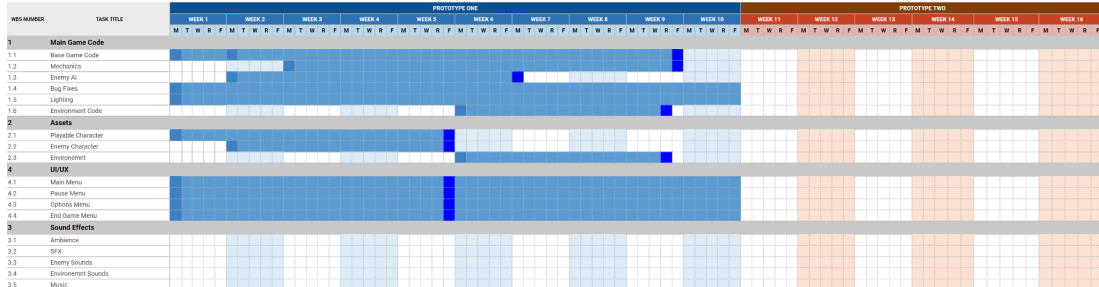
## **5.5 Close-out Plan**

The team will submit all variables to the GitHub repository by their scheduled deadlines. The team will also participate in the final presentation on the scheduled final date.

## 5.6 Gantt Chart

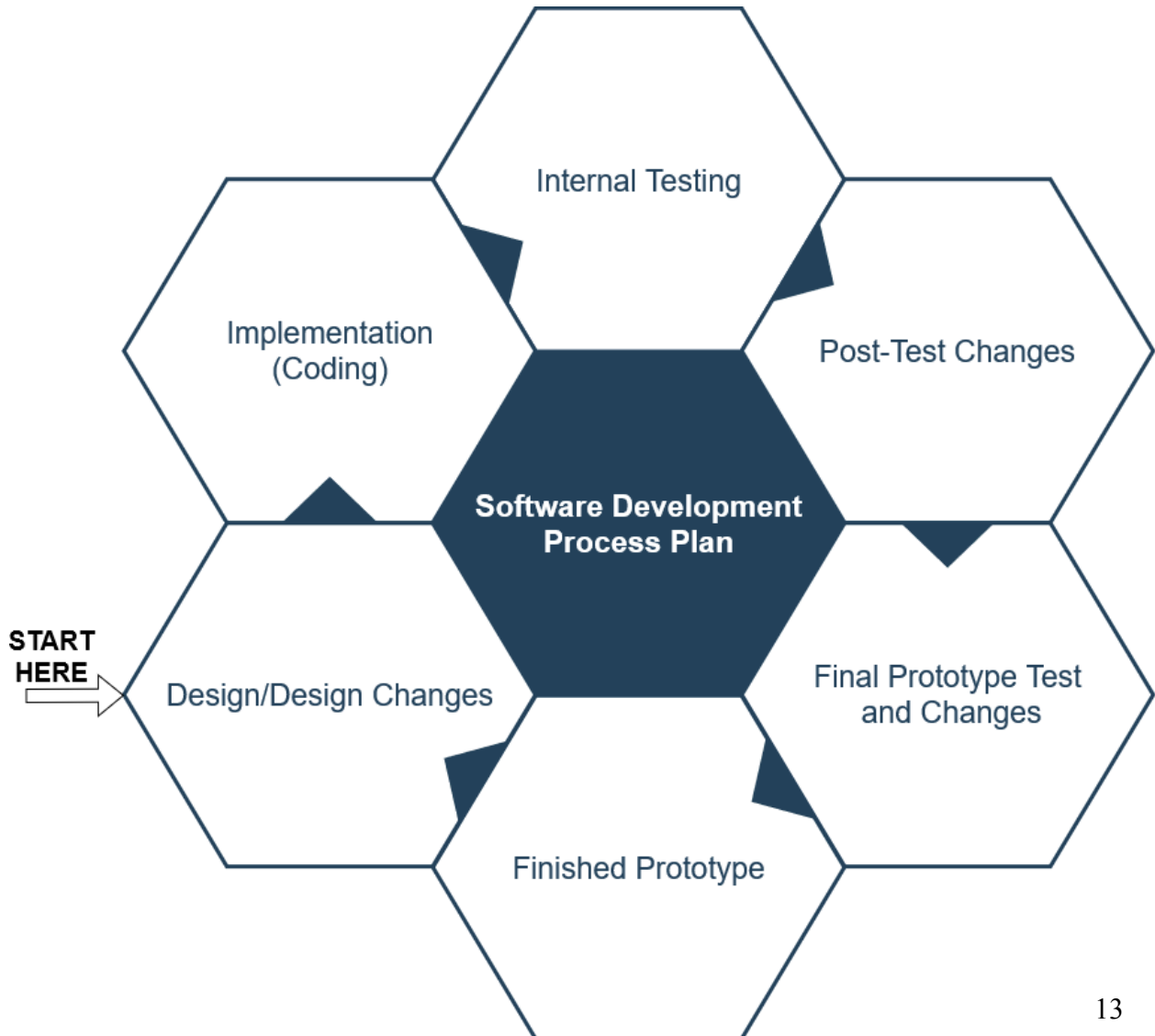
### Fate Studios

PROJECT TITLE: Project Fate  
 COMPANY NAME: Fate Studios  
 PROJECT MEMBERS: Nicholas Morfalis, Daniel Abulsi, Daniel Callahan, Logan Ashbaugh, Valeric Moin, Peter Lohr  
 START DATE: 8/23/22



## 6. Technical Process Plan

### 6.1 Process Model



The chart above demonstrates our Software Development Process Plan, which follows Agile Software Development standards, in how we wish to accomplish each of the tasks for each step of the development process.

## **6.2 Methods, Tools, and Techniques**

- Method: Incremental Development Model
- Tools: Github, Discord, Asana, Google Drive
- Techniques: Regular meetings with the client.

## **6.3 Infrastructure Plan**

There is no server setup planned for this game application because it is going to be a singleplayer game. Therefore there is no need for an infrastructure plan.

## **6.4 Product Acceptance Plan**

After development of the game the Q&A team will test the game application and make sure that all prototypes are working and playable. During this time the client will be kept in the loop throughout the development process.

# **7. Supporting Process Plans**

## **7.1 Configuration Management Plan**

We are using GitHub for the configuration. It will contain all changes made to the game and will let us make a smoother configuration of the game.

## **7.2 Verification and Validation Plan**

The verification and validation of this game will begin after the first prototype is completed into a desktop version of the shortcut executable. From then on we will be applying this process through the shortcut.

## **7.3 Documentation Plan**

The management team will work on the SRS and SPMP. The development team will keep records of the design and architecture of the game's progress.

## **7.4 Quality Assurance Plan**

The Q&A team will assure the quality of the game application.

## **7.5 Reviews and Audits**

While in the process of development the development team will perform consistent testing of the game to assure that everything is working properly. If any bugs or glitches are discovered in the code or design they are to be reported and analyzed for immediate fix or future fixes.

## 7.6 Problem Resolution Plan

Every member of the development team will report any issues with the games programming to the managing team as soon as possible. The managing team will collaborate with the developers on how to solve the issues that arise. The entire team will collaborate on how to solve these issues so that the project runs as smoothly as possible and is completed by the deadline date.

## 7.7 Subcontractor Management Plan

This project has no subcontractors.

## 7.8 Process Improvement Plan

In order for our team to improve the process and development of the software it is crucial to have proper documentation of the design and source code of the game. It is also crucial that we have strong building blocks for our code so that later on in the process there will not be much to fix if we wish to improve the quality of the game or change something. We also are planning ahead of time for any problems that might arise throughout the development process. This means planning ahead to have our team test the game before we release the first prototype to the Q & A team as well as the final prototype. By creating the SRS and SPMP we are also assuring ourselves that we have a well established plan and design to put forward into developing.