Shiva Project Plan

Project 1 - Phase 3

Prepared By: Kai Mizuno and Jared Schneider

Instructor: Professor Mark Baldwin

February 1, 2020

# Section I: Introduction

Team Foxtrot has been put together a plan for the development of a new turn-based strategy/tower defense game called Shiva. This game will put players in a post-apocalyptic world overrun by aliens, where the player takes control of a band of survivors that must explore the world, gain resources, and defend their town from endless waves of aliens.

This document details the organization, task plan and project management behind the development of Shiva. It will go into detail on who Team Foxtrot is, what tasks need to be completed by what day, the resources required to complete each task, the risks behind the development of this project, and the quality assurance and collaboration techniques being practiced.

# Section II: Organization

Team Foxtrot currently consists of two members: Kai Mizuno, and Jared Schneider. These two individuals will be working together and properly delegating tasks to ensure a high-quality product is created. Kai will be responsible for project management and back-end code development. Jared will be responsible for front-end code development. The team has broken down the work required to complete this project into tasks that have been delegated based on these established roles. Kai will be handling tasks that have to do with the base mechanics of the game such as AI and resource management, while Jared works on developing the interface and graphics that allow the player to interact with these mechanics. Kai will also be responsible for maintaining the project schedule and ensuring the proper tasks get delegated at the proper times.

The team will be using text and email to communicate the status of tasks and establish meeting times. During meetings, the team will brief each other on the work they have done this week, go over new designs and/or code that has been created to receive feedback, evaluate if the team is proceeding on schedule, and reaccess task delegation as needed.

# Section III: Task Overview

This section will detail how Team Foxtrot has broken down the work required to complete this project. A timeline has been created to ensure the team is aware of what needs to be done in order to finish the project in the allotted time. Also, milestones have been established for deliverables.

## 

The development of Shiva requires many different components. Tracking these components is done using a Gantt chart (Figure 1). This method provides a visual aid to the team to reference throughout the project to ensure the project is completed on time. The chart breaks down the work into tasks and sub-tasks, which each gets scheduled and established dependencies.

From the time of writing this paper to February 10th, 2020, is established as a period for finishing the planning and writing of the game design document. This encapsulates multiple design tasks including: 

* Finish the designs and concept art for the game’s map, interface, player units, weapons, vehicles, and alien units.
* Finish establishing the mechanics behind player units, weapons, vehicles, and alien units.
* Finish designing the resource and research economies.
* Finish establishing the background story of the game.

Team Foxtrot has already discussed many of the details behind these remaining design considerations. The team needs to finish concept art and work together to put all the discussions in a cohesive game design document. This is estimated to take another 8 hours of work. This step is vital to maintain the course of this project because all of the following steps will be based on the designs established in the game design document.

The next milestone is on February 24th. By this time, Team Foxtrot plans to deliver a game with basic map generation, weapons, and vehicles. Jared will be tackling developing the tiles used on the game’s map and the procedural generation that sets up these tiles at the start of each game. This will require pixel art for the tiles and objects inhabiting them (objects include terrains such as forest or lakes and enemy spawn pods) and the code which will attach these tiles together to create the game board. Kai will work on the battle interactions between friendly soldiers, civilians, defenses, and alien units. This does not require an established game board to create so it is an ideal task to work on while Jared works on tile generation. Kai would be working on creating the base objects for the friendly soldiers, defenses, and weapons laid out by the game design document, and at least 1 enemy unit. Soldiers and defenses would each have unique attack functions and characteristics that will later be called at the appropriate times during the development of a standard game turn. The base enemy type will be required and implemented to help test the attack functions of the friendly units. Placeholder sprites will be used until Jared later implements the front-end code behind these objects that allow for smooth game flow and animation. These deliverables are expected to take both team members around 10 hours to finish.

For the second milestone on March 9th, 2020, Team Foxtrot plans to have a polished map with tile generation, movement functions and sprites for units and defenses, and the implementation of the resource system detailed in the design document. Jared will be finishing the creation of the map including new features such as tiles being ‘visible’ or not to the player and how units/defenses will appear on the map. The objects created by Kai for the previous deliverable are required to finish the pathing and movement functionality. Jared will also be working on the base pathing for units and sprite animations to indicate movement. Kai will be working on the resource system and economy. This will require the map objects created by Jared for the first deliverable because hometowns generated on the map will have the functionality to delegate civilians to certain resource management tasks. Civilians will also be able to move to other tile objects on the map and discover new resources. The resource system will be able to keep track of food and research. The food system will be implemented and units will die without the needed resources. Research progression will not be implemented until the player’s turn is flushed out but will require the underlying resource management developed here. These deliverables are expected to take both team members around 6 hours to finish.

Alpha delivery is due March 30th, at which point Team Foxtrot plans to have a working game. To deliver this, the team will be working on the UI and logic behind player turns. These will be developed by taking all the objects the team developed individually and putting them together in a cohesive form. Jared will develop the UI, which will show important information such as player health, what units need to be given orders, and the current state of the player’s resources. Kai will be starting to put together the pieces of a player’s turn. The player should be able to click on a unit, move them to an allowable spot, collect resources, place defenses, and end their turn. Spawn points created during the map’s generation should spawn a wave of aliens between the player’s rounds which attempt to attack the player’s buildings and units. New spawn points will then be created based on the current round (higher rounds have different amounts of spawn points). This cycle should be repeated until the player’s town is destroyed. The state of the player round developed here is dependant on the objects created for the previous deliverables. These deliverables are expected to take both team members at around 10 hours to finish.

For the Beta and Final deliveries on April 10th and April 20th respectively, Team Foxtrot plans to refer to a compiled list of issues/unfinished parts of the game to polish. This may include the creation of more defenses or aliens, or implementation of better sprites and/or UI. The team will also compare the current state of the game with the game design document submitted at the start of the project and discuss what parts still seem to be missing or could still be expanded upon.

# Section IV: Project Management

The project will be evenly managed by both members. The team will be working on different systems at the same time, with frequent weekly check-ins on progress. Continuous comparisons to the Team Gantt Chart will ensure the team is meeting deadlines and getting systems complete so that the project can be completed with as little complications as possible.

## 4.1: Risk Analysis

Most of the risks associated with this project involve time management. If the team does not find the time to complete a given task, then all of the following tasks get slowed down up to the dependencies between tasks. If a team member falls behind on a given task, then the team will work together to come up with a new schedule to help get the project back on track. If one team member has gotten ahead on their work, then they will help split the other team member’s work to ensure that work gets finished on time. If it is not realistic to expect the other team member to help catch up on work, then tasks will need to be prioritized. Team members will work together to determine which tasks need to be done ASAP, vs what can be put off more. For example, during the development of objects and AI, the friendly and alien units are core to the gameplay of Shiva so they should be prioritized, while entities such as vehicles are additions that can likely be put off or omitted altogether without changing the core design of Shiva.

If the team gets too focused on the development of the systems and mechanics of the game, then there is a chance there will not be time to flesh out developing sprites and animations. If this becomes the case, then the team will fall back on using free sprites from the internet as placeholders. If the team finds that they get ahead on the work on the mechanics, then they will take time to go back and finish the sprites and artwork.

As the team learns more about the capabilities of GameMaker, they may come up with new ideas of features to add to Shiva. If a team member starts to add these features without consulting the rest of the team, then there is a chance it will interfere with the development or project vision of the other team member. To avoid this, the team will be continuously checking and updating the changelogs to ensure everyone is aware of any change. If someone wants to implement a major change to the game’s design, then they need to talk with the other team member during quality assurance meetings.

## 4.2: Quality Assurance

In order to ensure quality, Team Foxtrot plans on using a spreadsheet to keep track of found and fixed bugs, which will help save time when attempting to patch the game in the Alpha and Beta stages of development. To keep track of what has been updated, team members will be actively using GitHub commits with detailed changelogs. When a team member finishes one part of the design requirement, they will indicate such in the Gantt chart that informs the team on what needs to be discussed at the next meeting. There will be a separate indicator in the Gantt chart to specify when the team has talked about the design task and both believe it is complete. During team meetings, team members will brief each other on their progress since the last meeting and go over each of the commits and changelogs. If another member of the team notices any mistakes or improvements, then the team will work together to make a plan to carry out the changes. These changes will be added to the Gantt chart, or if the task seems to be sufficiently completed then it will be marked as completed.

# Conclusion

Team Foxtrot is very excited to begin working on Shiva. The team has put together a thorough plan and design for this turn-based strategy tower defense game. Team members Kai Mizuno and Jared Schneider have split the work according to back-end and front-end development respectively. Using a Gantt chart has aided in the team in establishing a timeline, identifying the time commitment for each task, and identifying dependencies between tasks. The team has established rules to aid in reducing risks and ensuring the project maintains high quality. Moving forward the team will be continuing to reference the plan outlined in this document and using it to implement the design laid out in Shiva’s game design document.