VeaRth Games

Software Design Description

Version 1.0

February 23, 2018

VR Circus

Brendan Baalke
Sangwoo Baek
Carson Reykdal
Isaac Shell

Table Of Contents

List of Figures

1. Introduction

1.1 Purpose

This document describes the design for the VR Carnival project. Specifically, this document will primarily provide an overview of the architectural structure, the data structure, and the user interface. The primary audience that this document is geared toward is future software developers working on this project.

1.2 Scope

This project is composed of three main parts, the main carnival game itself, a database for player information, and a prize server. However, the database and prize server plans are not detailed or complete. Therefore, the sections depicting their structures are subject to significant change, and will be given less focus in this document.

The current plan for the initial release is to create 3 games and a menu scene, namely, a shooting gallery, a catapult game, a paint toss game, and the Circus entrance. These games will be functionally sound and will be playtested for balance in terms of difficulty. Art assets are less of a priority, but if similar-quality textures can be found and applied across the game, they will be. Otherwise, the more artistic aspects of the project will be left for a future release for a team with superb artistic and 3d modeling talent. Overall, the main goals for the initial release are to create functionally complete games connected via the carnival entrance menu, and to create the game objects in such a way as to yield prefabs that future teams may continue to use and build off of.

1.3 Glossary

Player	The individual playing VR Circus
User	The individual playing VR Circus
Scene	The object containing all components of an area

VR	Virtual Reality
Prefab	A game object made from several smaller game objects

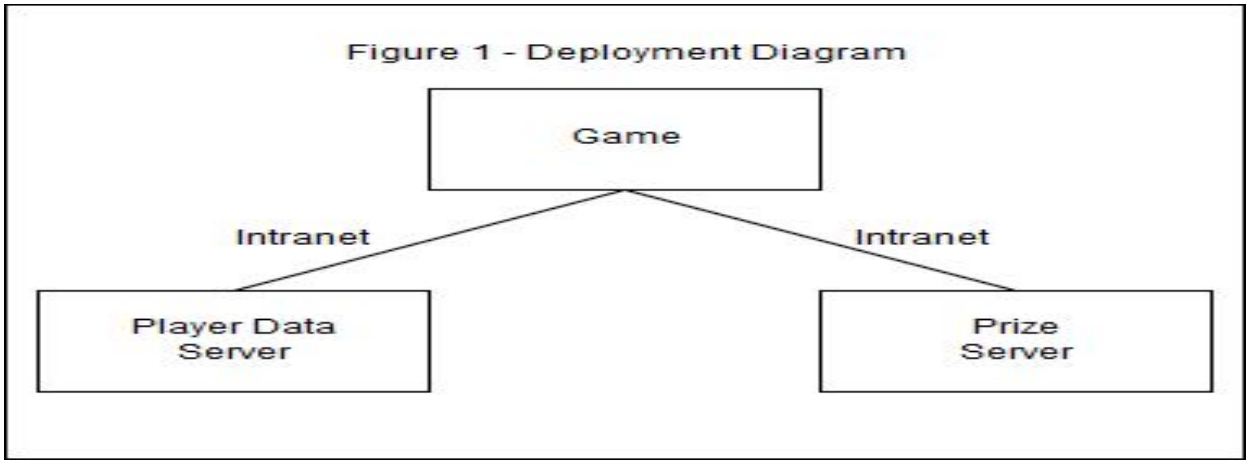
1.4 References

VR Circus UML Diagrams

1.5 Overview of Document

- Chapter 2 shows the Deployment diagram, which is the highest level structure that depicts the broad categories of the project that all parts of this project fall under.
- Chapter 3 contains the architectural design, which is a more detailed diagram that depicts more of the components behind the deployment diagram.
- Chapter 4 discusses a potential data structure design for the future database.
- Chapter 5 depicts the overall user interfaces for each of the current scenes being implemented in the game thus far.
- Chapter 6 is about how the help system will be implemented.
- Chapter 7 displays the sequence diagrams for the current scenes being implemented.
- Chapter 8 is a list of developer bios that are currently working on the project.

2. Deployment Diagram



For the deployment diagram above, the game box represents the overall carnival game made within Unity, the player data server box represents the database for storing user accounts, and the prize server is the database for storing various unlockables. To help get a clearer picture of the planned interactions between the boxes, the general flow that accompanies this diagram is:

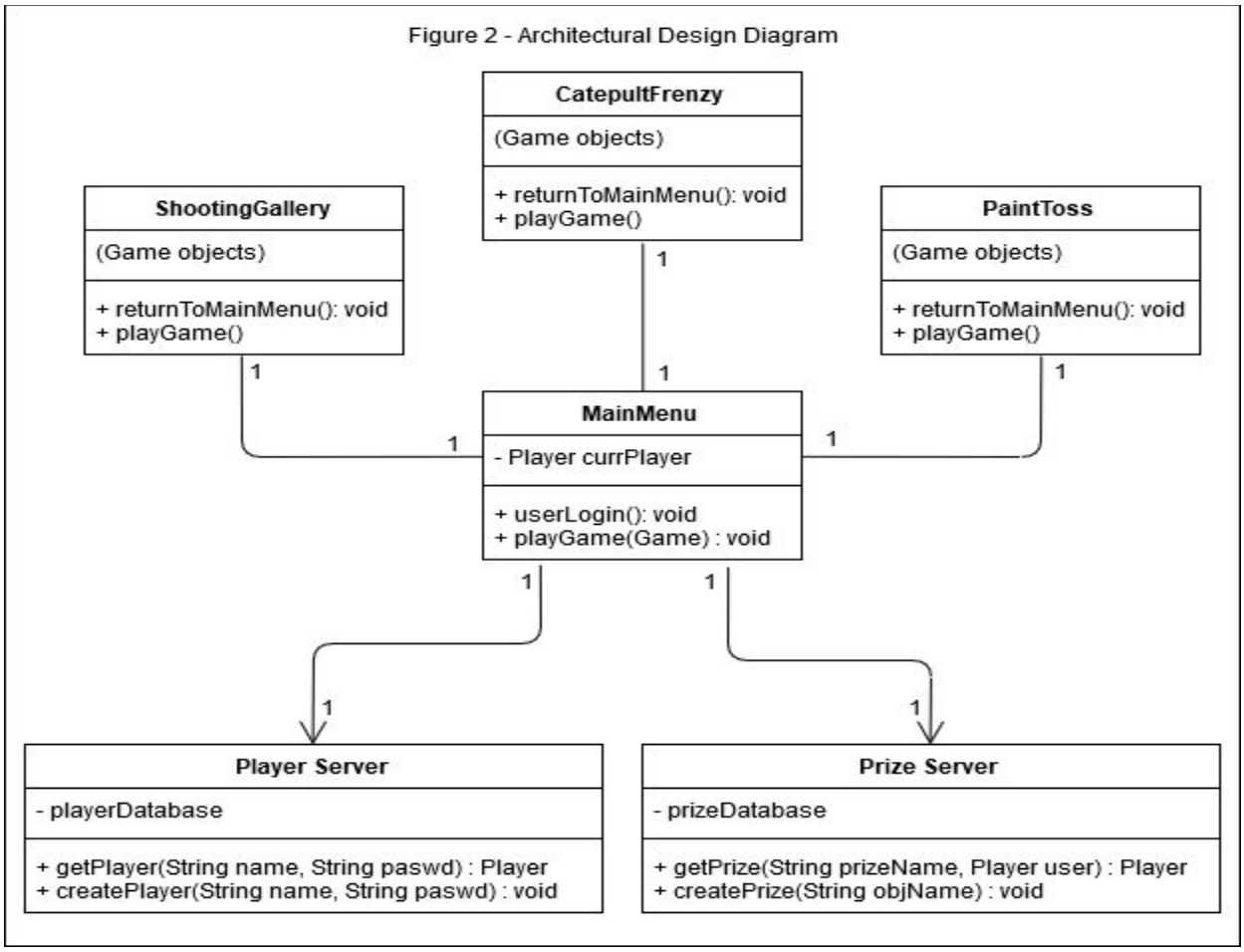
1. The user starts the game and is prompted to log in
2. The user selects an account, or creates a new one. The game responds by communicating with the player data server to load the correct player into the game.
    a. Retrieving unlocked prizes also occurs by communicating with the prize server.
3. The user plays the game with their unlocked prizes, and upon completion of specific goals, the game communicates with the prize server and player server to unlock a specified item.

Note: There are currently no servers in the VR Circus game. This is purely a basic plan for future development.

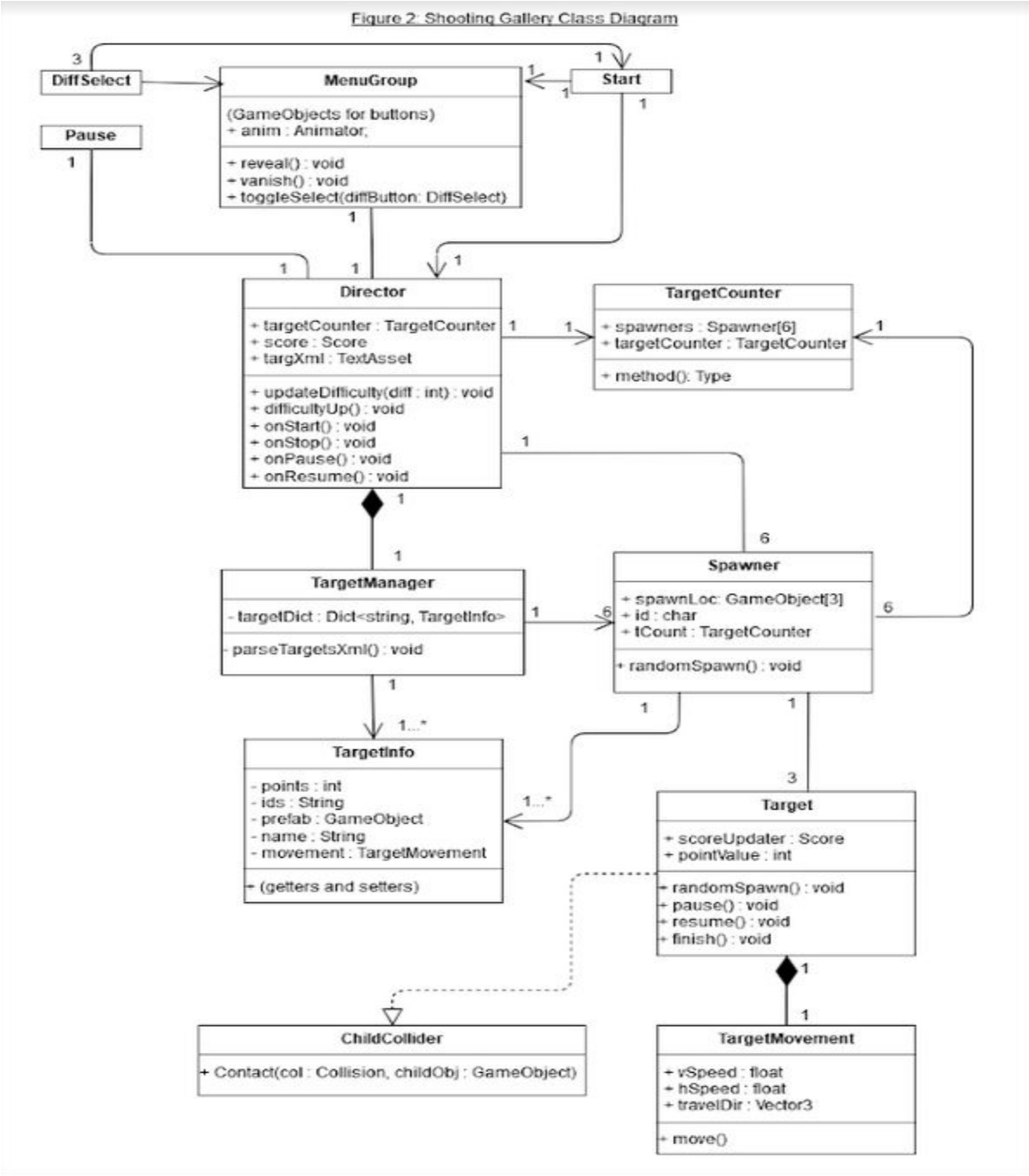
3. Architectural Design

Figure 2 below shows the basic components of the game and who they communicate with, including the servers mentioned above. Following this diagram are the specific class diagrams the games with more complex structures.

Figure 2 - Architectural Design Diagram



Note: playGame(Game) in the main menu is a game switching action, whereas playGame() in the other components are representations of starting each individual mini game.



Class Descriptions:

- MenuGroup: Responsible for handling menu animations and keeping the UI text on the menu correctly highlighted.
- Director: Responsible for starting, stopping, pausing, TargetCounter initialization, and

difficulty handling.

- TargetManager: Parses target information from the target-defining xml document specified by the director.
- TargetCounter: Keeps track of how many targets of each type remain. Updates the UI text that displays the targets remaining.
- TargetInfo: Defines a target, movement included.
- TargetMovement: Defines how a target should move
- ChildCollider: An interface that defines a function to allow a parent object to detect a collision that has struck another object.
- Target: The target itself. Contains collision-detection scripts
- Spawner: Spawns up to 3 targets at a time on its predetermined row. Keeps track of active targets, and never requests a sign type that has been exhausted. When out of targets, it communicates its death to the director.
- Score: Keeps track of the player's accumulated score.

4. Data Structure Design

Currently, player data, prize data, and unlocked prize data are the only scene-independent pieces of data that need to be stored. However, no database table structure has been discussed, since this will falls outside the scope of what will be finished before the next team takes over. A possible database structure could be to have a player-password table (PPT), a player-prize table (PPrT), and a prize list table (PLT), where the player name is the primary key of the PPT, the player name and prize name form the primary key of the PPrT, and prize name is the primary key of the PLT.

5. User Interface Design

The overall theme for the user interfaces is physical interaction; physical buttons or signs will be touched, shot, looked at, or hit to perform menu actions. In the following sections, each UI will be described for every scene in the game thus far.

5.1 Shooting Gallery UI

Upon loading into the shooting gallery, the player will be standing on the porch of a shooting-gallery building. In front of them, they will be presented with a large sign hanging just inside of a large open window. This big sign will have several smaller signs that, when shot with the egg gun provided to the player, start the game, exit back to the menu, change game modes, or allow the player to change guns. During the game, there will also be a small sign to the left of the window that can be shot to pause and resume the game in progress.

5.2 Paint Toss UI

Once the Paint Toss level loads, the player will see a few interactive menu buttons near the game board. The player will also see a game board where the target markers will appear. The player uses the VR controller of choice to click the select button and spawn a paintball in their hand. Variable skies and paved ground will cover the surrounding area of the game field. The game board that the player will interact with will be roughly 20 feet away from them. This board has two smaller secondary signs that correspond to a scoreboard and the selected challenge objective. While the player is participating in the current game they will be able to abort the game in progress via a sign toward the top right corner of the game board. All the signs/menu buttons with action options can be activated by throwing a paintball and hitting them.

5.3 Catapult Frenzy UI

When the game commences, the player will be in a large grassy field surrounded by a fence. To the left there will be 3 signs that have the words “Easy”, “Medium”, and “Hard” respectively. These will select the difficulty when the player points at them and presses the action button. The sign that is selected will appear change its text color so the player will always know which difficulty is selected. There will be another sign to the right which, when tapped, will return the player to the main menu. To begin, the player must hit the start sign on the right side to be transitioned over to the field of play. At this field, to the left of the player, there will be signs with pictures of different shaped blocks on them. Dragging the block from the sign will allow the player to place it in a 2 dimensional space directly in front of the player. After the player has finished building, the camera transitions to firing position by the slingshot-like structure. The player will hold the action button and drag back on the touchpad to pull the slingshot, then release to fire.

5.4 Main Menu UI (Carnival Entrance)

When the game loads up, the first scene the player will spawn into is a big grassy field surrounded by large mountains. The player will be facing the circus’s main door that slides down when it senses the player approaching, allowing entry into the game selection scene. Inside the game selection scene, there will be a desk in the middle of the room with a game selection panel. When the player points toward this panel, it will pop up and display the games. Once the game is selected with the action button, the player will be moved to the specific game scene.

6. Real-Time Design

Since this project is initially going to be a single-player game, there are not any concurrent user requirements at this time. However, if this project does move to include multiplayer, then there will be need for a multiplayer server to handle client interactions.

7. Help System Design

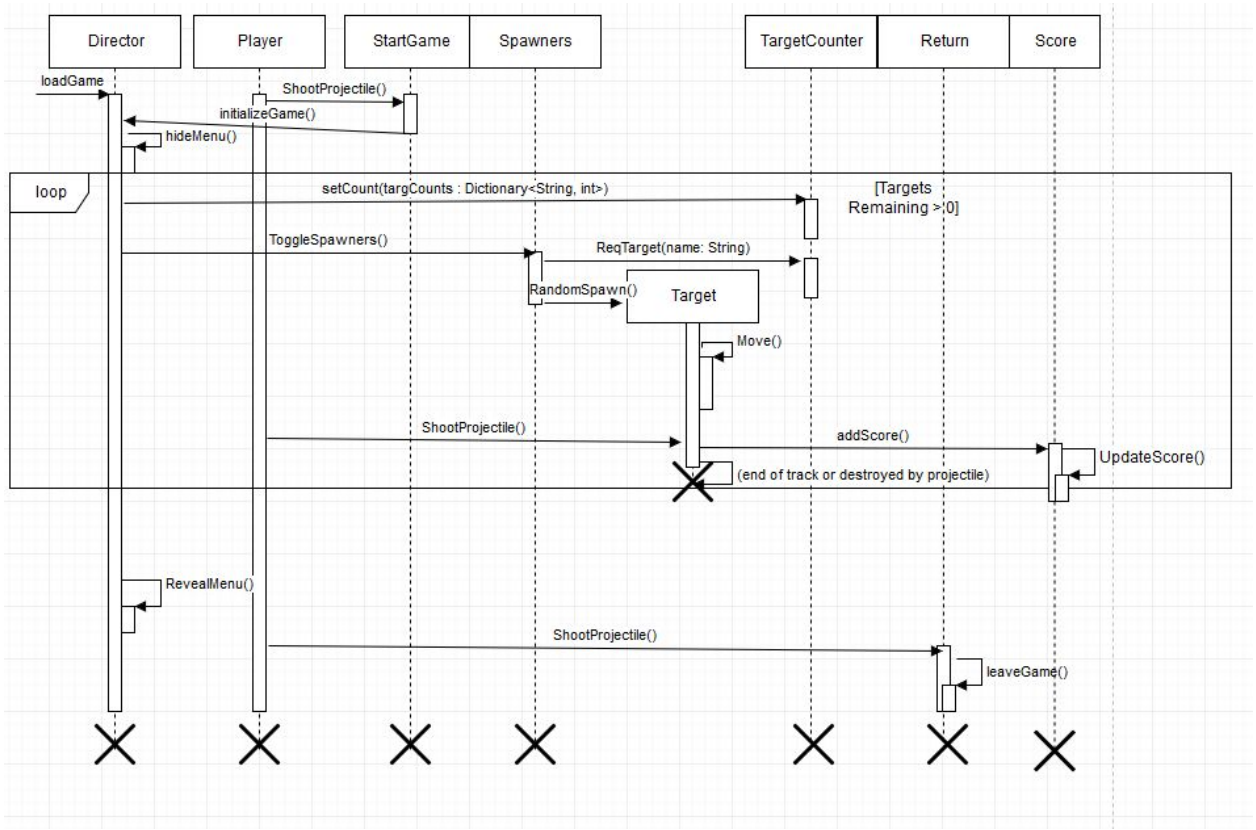
For the general user playing the game, most of the help system will come in the form of voiced instructions telling what the various options the player has in their current scene. For a system manager in charge of setting up and maintaining the systems, written instructions will be provided to explain how to set up any necessary servers, as well as any options to modify the databases.

8. Sequence Diagrams for Use Cases

Below are the sequence diagrams associated with each game. Each of these are based upon the use cases from the VR Circus UML Diagrams document.

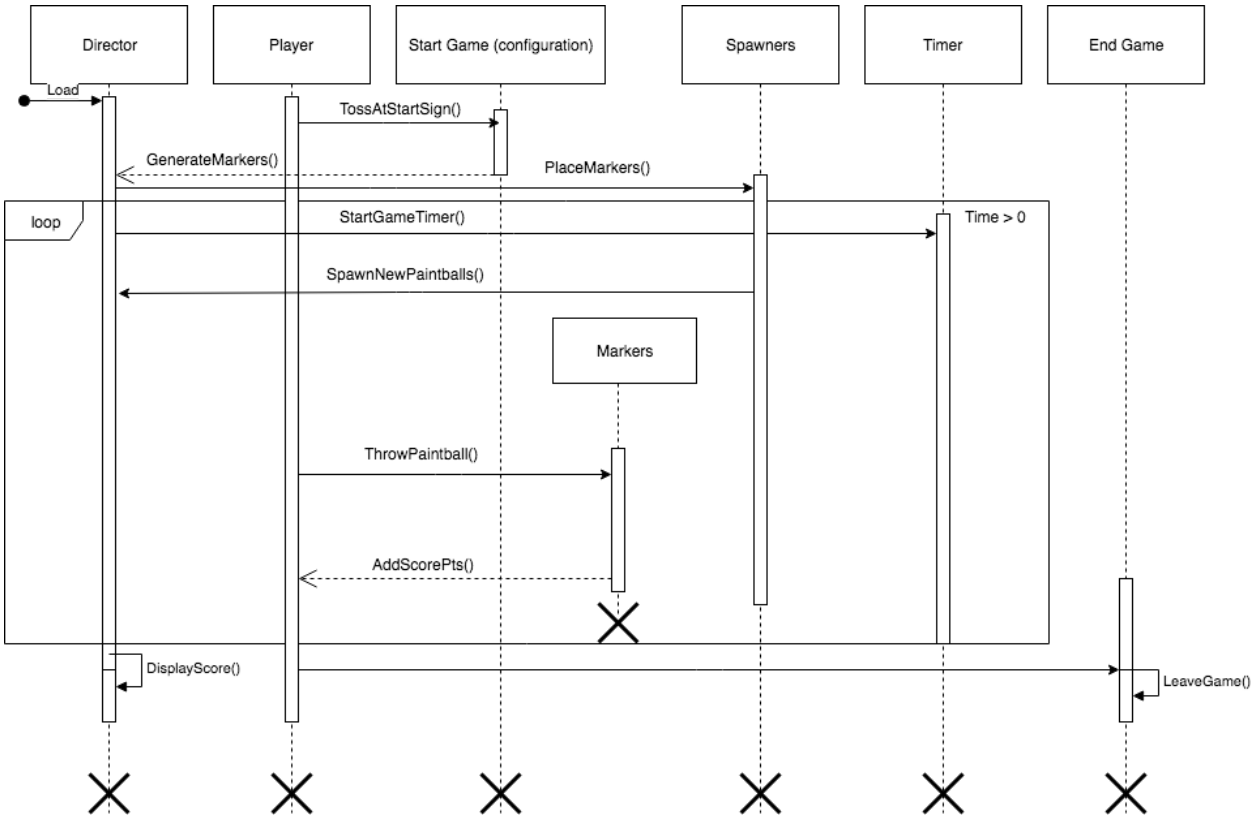
8.1 Shooting Gallery

Figure 3 - Shooting Gallery Sequence Diagram



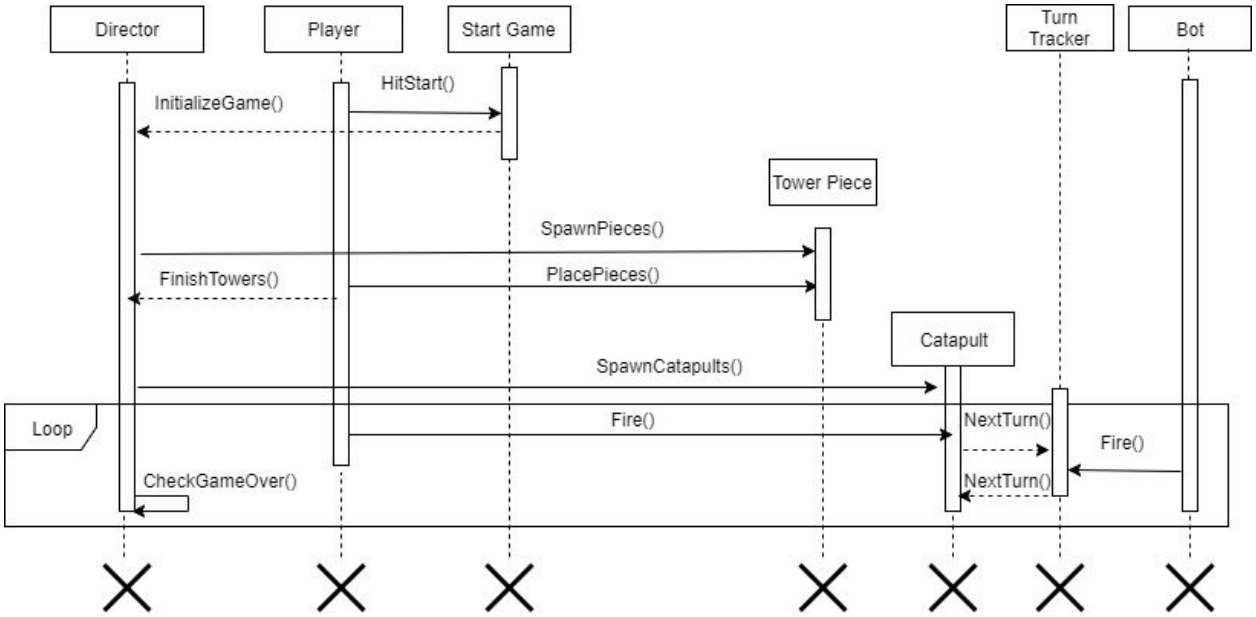
8.2 Paint Toss

Figure 4 - Paint Toss Sequence Diagram



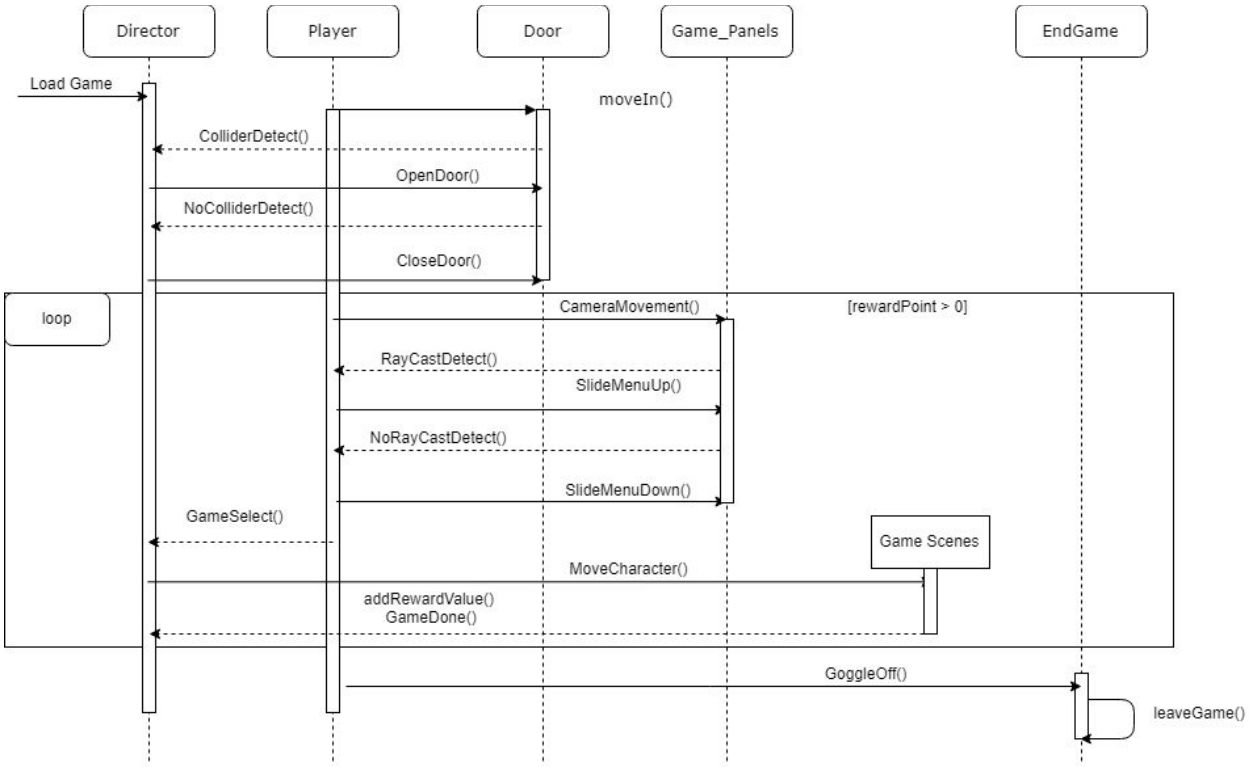
8.3 Catapult Frenzy

Figure 5 - Catapult Frenzy Sequence Diagram



8.4 Main Menu

Figure 3 - Main Menu/Entrance Sequence Diagram



9. Developer Bios

9.1 Brendan Baalke

Graduating from Western Washington University in either Spring or Summer with a BS in Computer Science. Technical interests include game design, user experience, and web development. A few of my non-technical interests include downhill skiing, graphic design, and playing video games. I plan on managing my own independent video game company within the next few years.

9.2 Sangwoo Baek

Graduating from Western Washington University in Spring of 2018 with a BS in Computer Science. Technical interests include game design, database, and mobile development. A few of my non-technical interests include hiking, researching, and playing video games.

9.3 Carson Reykdal

Graduating from Western Washington University in spring with a BS in Computer Science. Primarily interested in network backend development and artificial intelligence.

9.4 Isaac Shell

Graduating in Spring from Western Washington University with a BS in Computer Science. Technical Interests include game design and robotics, and his non-technical interests are drawing, leathercraft, and animation. Likes to help and teach others.