**Feature Specification:**

|  |
| --- |
| Feature Information |
| Feature Name | Racing Game (TBD) |
| Area | Unity Game |
| Related Features |  |
| Requirement Specs |  |
| Document Location | https://github.com/AriGlockner/RacingGame |
| Spec Status | Initial Draft |

|  |
| --- |
| Contact Information |
| Developer | Ari Glockner ([s-glocknera@bsd405.org](mailto:s-glocknera@bsd405.org)) |
| Manager | Mr. Magendanz ([magandanzc@bsd405.org](mailto:magandanzc@bsd405.org)) |

|  |  |  |  |
| --- | --- | --- | --- |
| Revision Summary |  |  |  |
| Author | Date | Version | Comments |
| Ari Glockner | 2/12/2021 | Alpha 1 | Created Testing Scene |
| Ari Glockner | 3/9/2021 | Alpha 2 | Created free-play mode |
| Ari Glockner | 3/26/2021 | Alpha 3 | Created Beat the Clock mode |

Sections:

1. Functional Specification – Who is this for? What’s the project? What’s the Scope? Target Market…
2. Scenario Description – What’s the use of it
3. Feature Description – describe everything in Work Breakdown Structure
4. Boneyard – Put Cut Features Here

**Functional Specification:**

(Name of Game) is a racing game developed in Unity. The target audience for this game is for people who like racing games. This game includes a free-play mode, a beat the clock mode, a race against a ghost CPU (computer), and a race against a live CPU. The free-play mode will include a racetrack and a drivable car to race around by one’s self at their own leisure. The beat the clock mode will make the play complete 3 laps around the track in a timeframe determined by the length of the track. The ghost CPU mode will have a CPU race against the player following the track without interacting with the player. The race against a live CPU will have the player race against the CPU while both are trying to react to one another. If time allows, more racetracks will be added to the game.

**Scenario Description:**

Main Menu (might change later):

Graphical user interface, text, application

Description automatically generated

Pause Screen just pauses the game and changes the color of the screen and says Paused

Win Screen (might change design)

Graphical user interface, text, application, chat or text message

Description automatically generated

Lose Screen is same as Win screen except for it says You lose

Car:

A white car on a blue surface

Description automatically generated with medium confidence

Track 1:

A picture containing text

Description automatically generated

**Feature Description:**

Sections of the project:

1) Setting up an Environment for testers - Done

The first thing I’m going to want to do when I get started is research different vehicle assets and settings for the vehicles so I can get my evaluation team so they can help me determine the optimal experience of driving a car in an open Enviornment. While I am doing this, I am expecting to need to publish a lot of similar versions where I tweak various settings for them.

Initially, I will send out a link to simmer.io to my evaluators. I may need to change that into uploading a file to GitHub and having them download it if WebGL becomes a problem. If they have to download it from GitHub, the testers must be using a windows computer to be able to play the game.

For feedback, I will be using GitHub issues for errors, exceptions, and recommendations just like last semester except I plan on making my testers more involved in the process so it will be a better experience for everyone involved in the project.

2) Free-play mode - Done

While I’m giving my testers a chance to test, I will have time to research different assets for tracks, landscapes, etc. and find what I would like to use for my 1st racetrack. Once I finish free-play mode, I will not push the update out with the track until the testers are finished with step 1.

3) Main Menu and Pause Screen - Done

When I made Raven’s Adventure, I ran into issues with clicking buttons in the UI Layer so I will avoid this in this game by having all the selections being made by keys on the keyboard. For the Main Menu screen, I’m going to build a simple screen with options with number keys that will select the options. For adding a simple pause screen, I will have the player toggle if the game is paused by pressing the P key on the keyboard.

4) Game Selection - Done

When there’s only one option, this is no problem, but I will have to find a way to let the game know which mode it is in. I plan to do this by having a static String variable being named at the selection point determining which game objects are enabled to begin.

5) Beat the Clock Mode - Done

Start off with a countdown to enable the player’s movement, then have a clock in one of the upper corners to let the player know how much time is left. Use waypoints as laps to track how the player’s doing, and once finished go to a win screen or lose screen.

6) Ghost CPU – Done

Create a waypoint system (similar to what was used for the lap tracking, but more points on the track) and get the computer to follow the waypoints. Test the waypoints by having the camera follow the ghost. Use the layers to make sure the player cannot interact with the ghost.

7) AI CPU – Working on it now

Don’t know where to begin so when I get here, I’ll do research.

8) Add more tracks

Same process as making a track to begin with. Make sure they’re shorter tracks than the one I currently have.

There are a few risks to be aware of when I am making this project:

1) WebGL

As I created last semester’s project, Raven’s Adventure, I ran into a project that I could make a game as a Mac or Windows standalone build and everything worked perfectly, but when I built the game as a WebGL build, I ran into a problem where the UI layer stopped working. I plan on fixing this by inputting keys on the keyboard rather than dealing with clicking on the button with a mouse.

~~2) Switching between various modes~~

~~When I create the beat the clock mode, I am going to run into a problem with differentiating between what modes I will choose between. I plan to handle this by creating a static string variable when I make my mode selection to let the game scene know which mode to pick between so I can enable/disable certain game objects that will control the game mode.~~

3) Creating an AI to race against the player

I’ve realized that the same scripts that were used to create the ghost will also work for the AI to race against the player. The only difference that I will need to do will be to create a reset script in case the car gets itself into a situation it cannot get itself out of.

**Document Sensitivity:**

This specification is for internal project management purposes only.

**Boneyard:**

List of cut features:

None at the moment