APCSA Final Project | Design Document

Ms. Novillo

1. Intro

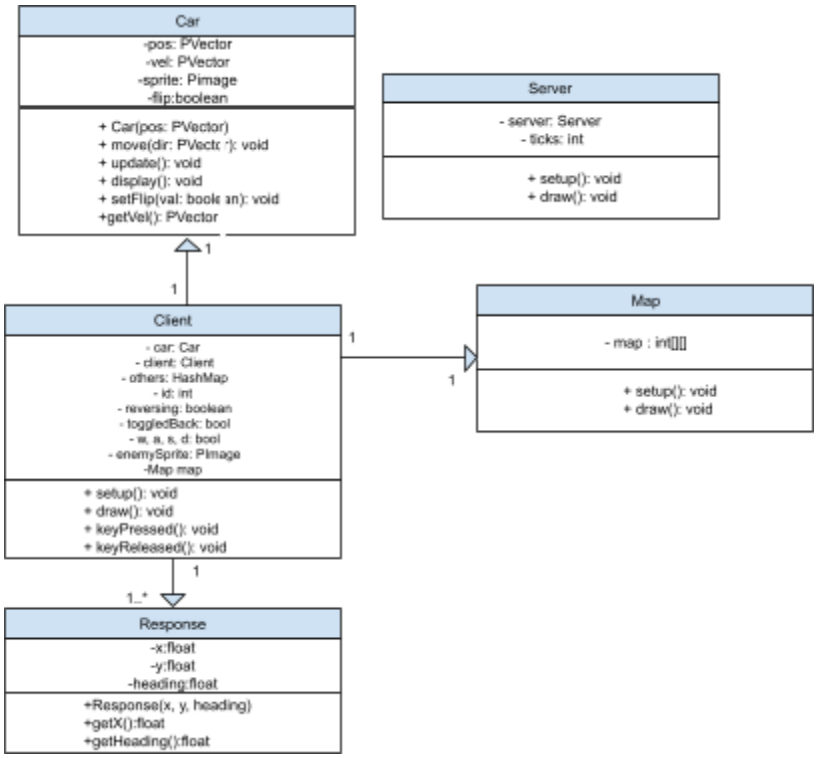
Period: 6th
Group Members: Owen, Jason, Vincent
Group Name: OJV
Project Title: 2D Multiplayer Racing Game

- Across different processing clients (and devices)

2. Description

This is a 2D multiplayer top-down racing game implemented in Processing using networked clients and a server. Each player controls a car that accelerates, decelerates, and turns based on directional key inputs (WASD). The game supports multiple players over a network and displays other players' cars on the screen. Each player's camera will follow their own car, providing a dynamic, centered view, and should load the rest of the screen as you move into the borders. The plan for the camera is to have an array representing the entire map, but the screen will only show the tiles around the player coordinates that will get updated with movement. The array representing the map will have integers that denote different forms of terrain (i.e. race track, grass, sand, barriers). Each surface will have a different impact on the car's speed, which will be done by implementing forces such as friction (which will also impact the drift ability of the car). We will be using the network library in processing to create a server for the multiplayer feature using client-server architecture. The library allows for multiple machines to run the game, as each client will be able to read data from and write data to the server and vice versa.

3. UML



4. How does it work?

To start the game, the server first needs to be initialized on one device so that other players/clients can connect. Each player can steer their car using the WASD keys (W to go forward, A and D to turn left/right, and D to brake and go backwards), and will have another key (space) to drift their car. Each car will be lined up at a start line, and there will be a countdown indicating when the game starts. The objective of the game is to be the first player to complete a set amount of laps around the track. At the end of the match, there'll be a scoreboard that displays each player's time and placement.

5. Functionalities / Issues

Player Car Movement:

- Controlled via WASD keys.
- Movement physics include acceleration, deceleration, friction, and reversing behavior.
- Smooth turning behavior based on velocity magnitude.

Multiplayer Support:

- Each client sends its car's state to the server.
- Server echoes data to other players for rendering remote cars.

Map System:

- Track image chosen and loaded onto screen
- System to detect road/grass using color pixels implemented (needs to be debugged)

Camera System (To be implemented):

- Each player's camera follows their own car.
- The background (track/map) and other players will be rendered relative to the camera.
- Screen center will align with the local player's position.

6. Preliminary & Future Log

Multiplayer support and syncing - Owen (Completed)
Player movement and drifting - Owen (Completed)
Game sounds, music, etc. - Owen (Doing)
Dynamic moving camera - Jason (Doing)
Map and terrain - Vincent (Doing)
Game UI, additional features, music, sounds, etc. - All (Task for next week)

- Basically polish it up