Sprint Plan 2

Product Name: VectorKart
Team: Calvin Owen, Yesenia Puga, Brian Chung, Gobe Fang, Gabriel Cardozo, Shauna Mahoney
Sprint Completion: Monday, 7/15/24

High-Level Goals:

Produce a minimally functioning singleplayer version of the game that links the graphics with the game logic.

User Stories/Tasks:

1. As a player I want to get live feedback on my gameplay so I can improve and learn from my mistakes. (As a teacher I want a game that intentionally makes my students think about vector addition and its applications.) (5 SP)
    a. Collision detection with edges of the map: Add game logic to detect when a player movement exits the map
        i. or crosses the finish line.
    b. Vector animation: Make vectors scale and move according to mouse hover position.
    c. Car movement and control: Make the Car class have an externally modifiable position. Link this position to player mouse clicks.
    d. Car lerp animation: Make the car model lerp smoothly from current position to a new position.
2. As a player I want to see statistics for my game so that I can share them and compete with my friends. (3 SP)
    a. Statistics tracking for the car: Create player statistics including velocity, acceleration, turns taken, and max velocity.
    b. Statistics dashboard: Create a dynamic HTML/CSS based dashboard that shows player statistics below the WebGL canvas.

Cleanup from Sprint 1

- Convert VisualVector to 3D model
- Integrate Map into repo
- Integrate Car into repo

Infrastructure

- Make Definition of Done and setup checklist
    - Jest
    - ESLint
    - User-facing acceptance
- Organize GameObject Hierarchy and modules

Team Roles:

- Scrum Master: Brian
- Product Owner: Gabe
- Developer Team: Everyone else

Initial Task Assignment:

- Calvin - GameObject Creation: Make base object (RenderObject and Vector3) for Car and Wall
- Gobe - Map Creation: Create 3 handmade maps with finish line and starting points
- Gabe - A function that takes a position and the Map and returns whether it is a collIsion
- Yesenia - The user clicks on the web page randomly, generating a random number for the car.obj object to have a random acceleration/velocity which will make the car move from the starting point to the end point(will not involve car movement animation)
- Shauna - Convert VisualVector to use visible 3D vectors
- Brian - Add statistics to the Car class including velocity, acceleration, turns taken, and max velocity. Display these statistics in an HTML/CSS dashboard below the canvas.

Task Backlog

- Car movement and control: Make the Car class have an externally modifiable position. Link this position to player mouse clicks.

- Car lerp animation: Make the car model lerp smoothly from current position to a new position.
- Car collision with finish line: Car can detect when it crosses the finish line (in the right direction)
- Win/Lose condition logic: Create logic that based on collision with finish line or crash will display a win/loss
- Game reset logic: Ensure game can be reinitialized to play again
- Textures: Add textures to various models
- Lighting: Add lighting to the WebGL canvas

Scrum Times:

- Tuesday 7/16: 11-11:45am (with TA)
- Wednesday 7/17: 12:30pm-
- Thursday 7/18: 11-11:45am (with TA)

Burnup Chart