**Project Title: Time Attack Racer**

**Project Description:**

Arcade style split-screen 3D time attack racing game using xbox controller as input device. Tracks made using mod kits featuring a closed circuit with checkpoints to determine progress in the game.

Intent is to create a fully functional racing game with at least 1 car and 1 complete racetrack with checkpoints and a time tracking system.

**Project Must haves:**

* Must have a working car than can be used to drive around the track
* Must have 1 working track at least
* Must have a checkpoint system to track time and record laps/checkpoints

**Stretch Goals:**

* Split Screen MP
* Add a minimap for the level
* Adding a racing HUD (speedometer, time, checkpoint number)
* Adding multiple cars to choose from
* Adding multiple tracks to choose from
* Adding ghost car feature to race against the best time recorded
* Add UI for car select, track select and so on

**Super Stretch Goals if given the time:**

* Create a Level Editor to utilize the mod kit pieces and create custom tracks (rapid prototype using dear ImGUI)
* Create procedural tracks using mod kit pieces

**Some references to provide context:**

Driving Mechanics and checkpoints:

<https://www.youtube.com/watch?v=GPf9C2en-io&feature=emb_logo>

Track Design:

Note: This is a full game, it may take a while to load but ideally the rack track I intent to use would be very similar to the roads seen in this game (Ignore the buildings that is not what I am talking about, just the roads)

<https://smartdrivinggames.com/driving-games-city-car-driving-simulator/>

Camera Setup:

Same as what I did the last time with DFS 1, nothing fancy

A close up of text on a white background

Description automatically generated

What I have now:

A screenshot of a computer screen

Description automatically generated

Track System Layout:

A close up of text on a whiteboard

Description automatically generated

**Weekly deliverables:**

|  |  |  |  |
| --- | --- | --- | --- |
| Week | Date | Proposed Deliverables | Dues |
| 1 | Jan 21, 24 | Completed Project Plan  Setup Project on source control | Project Plan and Milestones |
| 2 | Jan 28, 31 | Have 1 completely designed track,  Start testing colliders for the track model in Engine |  |
| 3 | Feb 4, 7 | Finalize on track system (Mode kit pieces with individual colliders vs 1 track model and colliders model) |  |
| 4 | Feb 11, 14 | Tier 1 checkpoint system   * Setting up triggers along the track using data (xml) * Set specific checkpoints to be start and end points in the level |  |
| 5 | Feb 18, 21 | Tier 2 checkpoint system   * Correctly placed checkpoints for the track * Tested working intended behavior for the checkpoints * Keeping track of laps, time taken per lap, time between checkpoints and recording best runs |  |
| 6 | Feb 25, 28 | Racer HUD with a speedometer, lap counter, checkpoint counter, time taken, and best time recorded on track |  |
| 7 | Mar 3, 6 | Implement Race car audio  (Audio for acceleration, audio for gear switch and audio for braking) |  |
| 8 | Mar 10, 13 | Bug fixes and reassessment of project plan for mid term  Setup initial architecture and code for split screen MP support | Mid Term Presentation |
| 9 | Mar 17, 20 | GDC | GDC And Spring Break |
| 10 | Mar 24, 27 | Create MP split screen version |  |
| 11 | Mar 31, Apr 3 | Complete game flow  Make game UI (Main Menu, actual race, end of race menu) |  |
| 12 | Apr 7, 10 | Implement multiple cars  Build a car select screen |  |
| 13 | Apr 14, 17 | Implement more tracks (Model using mod kit + data)  Build a level select screen (Use framework from car select screen) |  |
| 14 | Apr 21, 24 | Bug Fixing and juice |  |
| 15 | Apr 28, May 1 | Bug fixing and juice | Final Presentations |

**Risks:**

The tack system needs some manual setup. I would need to develop the track and export as a model but have all the colliders export as separate models to account for concave hulls (i.e do this how unreal does it where I can generate simplex collision models for the track pieces and use those as n separate collision meshes that will be loaded into the scene)

This would need the use of UE4 (to load in my mod kit pieces and generate the collision mesh, then export them as fbx files) as well as some 3D software (Maya or Max will be used to then convert the fbx collision meshes into obj files that I can load up and use in the scene with PhysX)

Another risk I have to account for is PhysX, since it’s third party I am assuming there will inevitably be stupid things that will create roadblock.