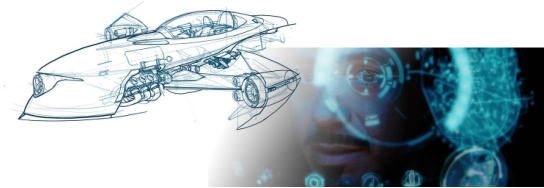
GDD Personal Project - Racing Game AI & Gameplay

Rodolfo Fava

160204 - Y4 Programmer



Change Log

|  |  |  |
| --- | --- | --- |
| Date | Version | Description |
| 13/02/2020 | 1.0 | Created the document |
|  |  |  |
|  |  |  |

# 

[**Objective**](#_cba9d8o0ipay) **4**

[**Approach**](#_hvh7iicx9ab7) **4**

[**Game Mechanics**](#_usqaozc81rxe) **4**

[Basic racing game mechanics](#_sbqwnn5a1vza) 4

[Add-ons](#_arfe9v5lkj17) 4

[**AI goals for Q3**](#_h79htxdb0ukl) **5**

[Context Steering with Racing Lanes](#_zcjiejj8g5ov) 5

[Case study of machine learning for racing games](#_1eiksvwes01a) 5

[Prototype of Machine Learning within my project](#_g62u5ieof0dv) 5

# 

# Objective

To create an engaging, elaborate anti-gravity racing game AI and a range of gameplay features. This will help me hone my skills and make me ready to work in a professional environment with a team. I will make use of my experience in Drivinity and my internship at Mechamania to create an innovative AI algorithm that will give the sense of an intelligent entity that is able to compete with the player. I want to experiment with machine learning as well, since it is a hot topic within the AI development at the moment, and should open many doors career-wise once I graduate.

# Approach

My goal for block C is to have a basic racing game done in *Unity 3D using C#*, with all the rules and mechanics locked, and functional on at least two racing tracks done with Samira’s (4th year BUAS artist) racing track tool. It should contain prototypes and a case study of machine learning for the NPCs.

# Game Mechanics

## Basic racing game mechanics

*(High Priority)*

* *Win-Lose conditions:* 
  + After a number of laps, the vehicle that is ahead of the others wins the game.
* *Multiplayer*
  + Up to 2 human players should be able to join simultaneously and play split-screen mode, as in the wipeout omega game (shown below).
  + 
* *Player vs NPCs*
  + The total amount of racers, players and NPCs, in any race shall be 4.
* *Best Laps*
  + As a complement to racing, each track will have a stored ranking of best laps (as in the time trial challenges). This should add an extra challenge to the game.

## Add-ons

(*Low Priority*) - Game Juice, player XP.

* *Power-ups (NPCs that shoot missiles, projecting the target's position)\**
* *Boost Pads*
* *Sound Effects*
* *Soundtrack*
* *HUD that displays enemies and areas of interests\* .* Also, vehicle stats like position and best laps would be interesting.
* **
* *\*: Ideas gotten from an intake assignment I did last summer for PlayMagic Ltd, a games studio in Malta. I hope to use this portfolio piece as my ticket to an Entry Programmer position at that company once I graduate*

# AI goals for Q3

## Context Steering with Racing Lanes

*(High Priority)*

(as shown by AI programmer Andrew Fray)

## Case study of machine learning for racing games

*(High Priority)*

Research the use of machine learning in racing games. What has worked in the past? What has not? What are the advantages of using it, what are the drawbacks? Refer to sources suggested by lecturers and professionals of the field:(AI programmers from AAA companies:

* Andrew Fray (AI programmer that worked on Codemasters’ F1.
* Arjen Beij (AI programmer from Guerilla games have had contact with).

## Prototype of Machine Learning within my project

*(Medium Priority)*

After the aforementioned research has been done, build a prototype of a machine learning AI that will “draw” the racing line automatically - Without the need for guess-work or develop some racing line drawing skill that I don’t have.

## Tools

* **Engine**: UE4. I chose it after talking to Bojan, and he said C++ is generally better for the game development industry and Unreal is also seen as more professional. I do, however, know that several studios out there (50%, last time I checked) use Unity3D as their main engine. Therefore, I intend to keep Unity side projects that I can work on during weekends in order to keep developing this skill.
* **Tracks:** After a meeting with Bojan, I decided to load track obj models from older games (e.g. Wipeout 1995) in order to optimize my development process, so I can focus only on the AI and Gameplay. I had also considered using Samira’s 3D track tool, but for now I will keep it simple. Could be an interesting test for the future.
* **Debugging Tools:** Make lots of debugging tools for live-coding.

## Reference Titles

Wipeout Omega (PS4 - 2017)



Redout (PC - 2016)



F-Zero X (Nintendo 64 - 1998)



What I like about this title is the arcadey feel to it. Quick maneuvers, bumping to the enemy, sending them flying away, recharging Energy levels to boost, all of that sounds like great fun. I want to bring this style of racing game into a modern anti-grav racing game, combined with modern graphics, realistic controls and physics.

## 