Final Year Project Proposal: How ai is used in racing games?

Michael Hadley T7124630 GAV3027-N

Title:

How AI is used in racing games?

Contents

[Project description 1](#_Toc31375396)

[Rationale for project choice 1](#_Toc31375397)

[Areas for investigation 1](#_Toc31375398)

[Background research 1](#_Toc31375399)

[Literature review 2](#_Toc31375400)

[Methodology 2](#_Toc31375401)

[Research ethics 2](#_Toc31375402)

[Project plan 3](#_Toc31375403)

[Proposed project artefact/product 6](#_Toc31375404)

[References 6](#_Toc31375405)

# Project description

The project is to create a racing game using AI controlled opponents and to research how an AI controlled player can follow a track and learn how to race against a single player.

# Rationale for project choice

The idea behind the project is based off of my enjoyment of racing games and wanting create my own and by doing so using what I learn in AI for Games class I have running alongside the final year project. This way by the end of the year I should have made two games driven by AI one solo project and one group project.

# Areas for investigation

In this project I will look into how AI learns to adapt to its environment and how it reacts when it rammed by another car.

I will look at how the AI will overtake cars, safely efficiently and how they overtake depending on if they are on corners or straights.

In this project I will look into how Rubber Banding can be used to keep the AI close to the player during the race.

Look into autonomous cars which can be used as an example or inspiration of how the AI will drive around the track.

# Background research

In my background research I will look at how AI go around a race track using checkpoints as indicators for a specific direction of travel. How and when the AI should attempt to overtake an opposing player. I will look at how to keep the AI on the track and reduce crashes. How to make the AI competitive with the player so that the player isn’t just racing by themselves after the start of the race.

# Literature review

An Architecture Overview for AI in Racing Games article has helped me to get a further understanding of how AI is used to create single player racing games (Tomlinson and Melder, 2020).

The conference paper on Learning overtaking and blocking skills in simulated car racing has help get a further understanding of how to implement the correct techniques to allow cars to overtake one another. (Huang and Wang, 2015)

The development of a simulated car racing controller based on Monte-Carlo tree search article has helped me get a further understanding of the Monte Carlo tree search that I will be implementing in my project to create my racing controller. (Huang and Wang, 2016)

This video explains how to use way points to allow the AI car to manoeuvre around the map and to create a varied difficulty for the AI to compete against the player. I will be using this YouTube video to reference how to make a waypoint system. (Vegas, 2017)

This article Finite-State Machines: Theory and Implementation has helped me get an understanding of finite state machines and how I can implement them into my game. I will be using a finite state machine to control the logic of the AI to overtake opponents and driving quickly and safely around the race track. (Bevilacqua, 2013)

# Methodology

I will use Unity and C# to create the game that will be used to answer the question.

I will use Google Scholar, books and online tutorials to get a better understanding of my chosen subject.

I will use Git Hub to save and track my progression of my project using specific titles to every new commit pushed.

I will use waypoints set around the map for the AI to follow around the track. Adjust it to allow the AI to accelerate and break into corners at varying speeds depending on the corner and the difficulty the AI are set to by the player in the game. I will use Rubber Banding to keep the racers close to one another create a skill gap, this will be adjusted in the production of the game if the AI opponents become too difficult to beat making the unenjoyable.

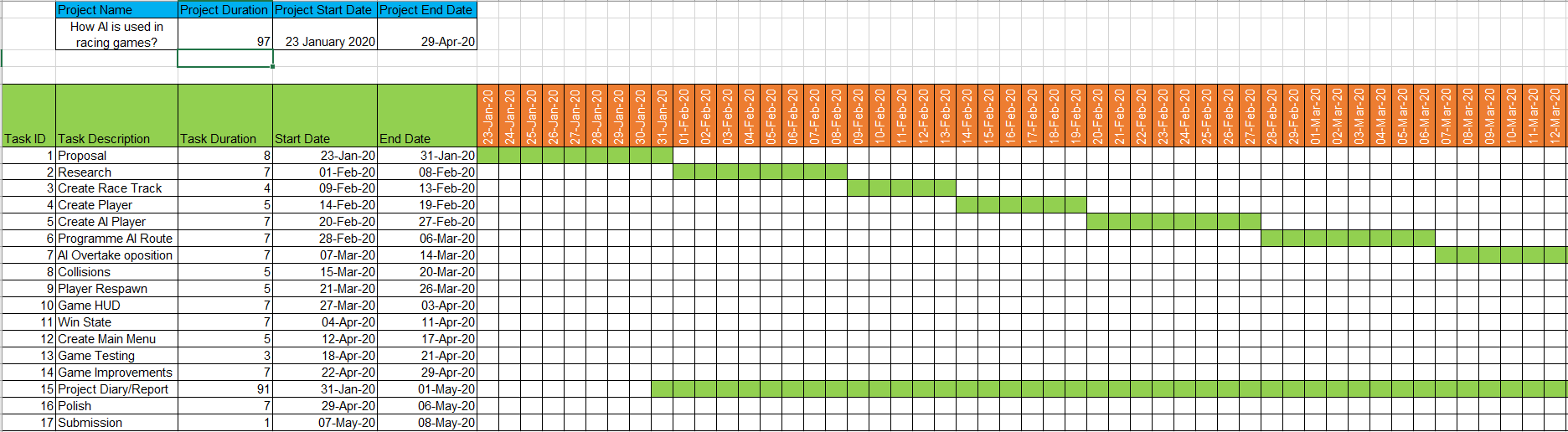
# Research ethics

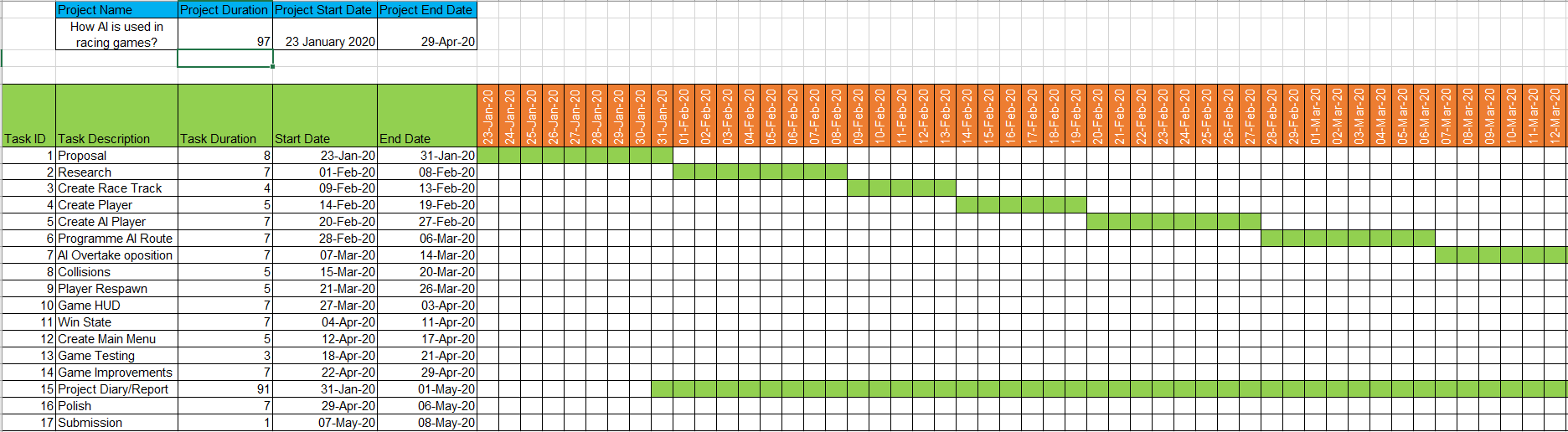
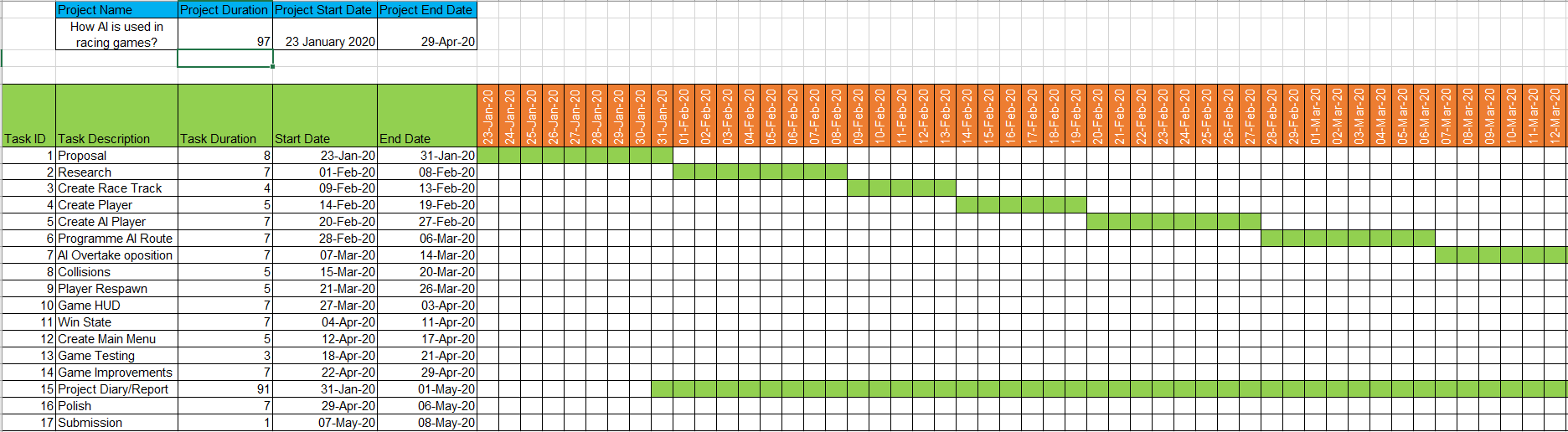
During play testing I will have a number of people to give me feedback of the game and I will keep the people anonymous by labelling the feedback sheet game tester 1, 2, 3 etc. With my game being a racing game only it would merit a PEGI 3 rating as it will be suitable for everyone to play as there will be no violence or bad language. This will limit any ethical issues.

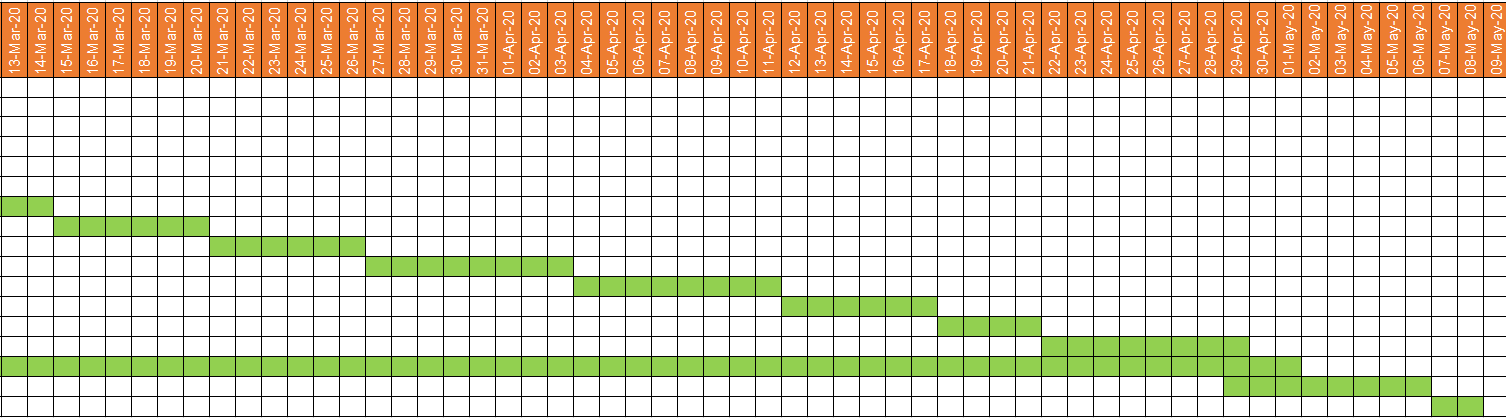
# Project plan

1. AI research to get an understanding of how AI is used in racing games (reference every source).
2. Create a race course.
3. Create the player character class with acceleration, breaking and cornering in mind.
4. Create the opposition class with acceleration breaking and cornering in mind.
5. Programme the route of course for the AI player to do one lap of the track.
6. Allow the AI to overtake one another.
7. Collisions between the player and the AI players.
8. Make a respawn feature when the player or the AI player goes off the track for more than five seconds the spawned back onto the track.
9. Create HUD with lap timer and race position.
10. Win state with lap times.
11. Create menu system that will allow the player to change the AI’s difficulty.
12. Game testing
13. Improvements based off of feedback
14. Project report
15. Project polish: remove any unnecessary files, code, any tests to create a new finished version of the project.
16. Submission

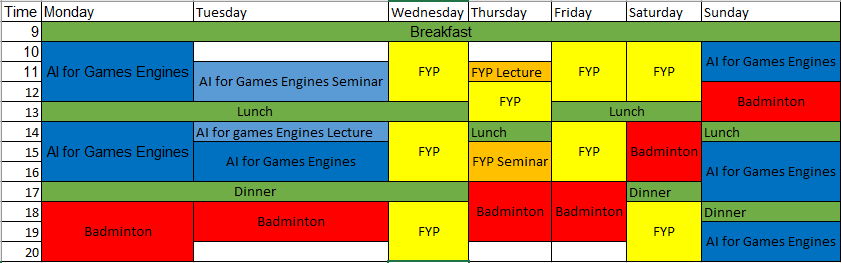
During the project I will test everything implemented and record errors and how they were fixed.

The Gantt chart below shows the timeline of the project plan:





Work schedule:



# Proposed project artefact/product

Using Unity to display AI within a racing game. The AI will know how to accelerate and break into corners, to be able to overtake opposing cars and will hit a top speed and come to a stop when they have crossed the finish line after the final lap. This will be a single player game with the ability to change the difficulty of the AI opponents.

# References

Tomlinson, S. and Melder, N. (2020). [online] Gameaipro.com. Available at: <http://www.gameaipro.com/GameAIPro/GameAIPro_Chapter38_An_Architecture_Overview_for_AI_in_Racing_Games.pdf> [Accessed 31 Jan. 2020].

<https://ieeexplore.ieee.org/abstract/document/7317916/references#references>

Huang, H. and Wang, T. (2015). Learning overtaking and blocking skills in simulated car racing. 2015 IEEE Conference on Computational Intelligence and Games (CIG).

<https://ieeexplore.ieee.org/document/7880111>

Huang, H. and Wang, T. (2016). The development of a simulated car racing controller based on Monte-Carlo tree search. 2016 Conference on Technologies and Applications of Artificial Intelligence (TAAI).

Vegas, J. (2017). *How To Make A Driving & Racing Game For FREE - Unity Tutorial #07 - AI & WAYPOINT SYSTEM*. [online] YouTube. Available at: <https://www.youtube.com/watch?v=TdNbxkF_f94> [Accessed 31 Jan. 2020].

Bevilacqua, F. (2013). Finite-State Machines: Theory and Implementation. [online] Game Development Envato Tuts+. Available at: <https://gamedevelopment.tutsplus.com/tutorials/finite-state-machines-theory-and-implementation--gamedev-11867> [Accessed 31 Jan. 2020].