



AI Truck Parking Game

Project Engineering

Year 4

Vaclav Vincent Hornak

G00366783

Bachelor of Engineering (Honours) in Software and
Electronic Engineering

Galway-Mayo Institute of Technology

2021/2022



Figure 0-1 Project Renders

Declaration

This project is presented in partial fulfilment of the requirements for the degree of Bachelor of Engineering (Honours) in Software and Electronic Engineering at Galway-Mayo Institute of Technology.

This project is my own work, except where otherwise accredited. Where the work of others has been used or incorporated during this project, this is acknowledged and referenced.

Vincent Hornak

Acknowledgements

I would like to thank my supervisor, Paul Lennon for continues support and feedback through our project classes in Semester 1 and continuing help fulfil the teamwork aspect of the project with his class every Friday. I found this a big help and a confidence boost for getting adapted to a real work environment.

I would like to thank the lecturers, Niall O Keeffe, Michelle Lynch, Brian O Shea for the support throughout the classes during the week. Michelle put a lot of emphasis on the deliverables and the structure in which they should be laid out which made it very helpful. Niall gave me great motivation and showed interest in my project each week making sure I was going along the right path and keeping up to date with the progress.

Lastly, I would like to thank my classmates who helped with weekly feedback throughout our 15-minute stand-up meetings. During these meetings the feedback and engagement was essential for the progression of the project.

Table of Contents

1	Summary	7
2	Poster	8
3	Introduction	9
4	Project Tools	10
4.1	Diagrams.net	10
4.2	Paint.net	10
4.3	Zmodeler 3	10
4.4	Microsoft Visual Studio 2019	10
4.5	Unity	11
4.6	ML Agents.....	11
4.7	GitHub	11
4.8	Jira	11
5	Project Architecture.....	12
6	Project Plan	13
6.1	Initial Timeline.....	13
6.2	Revised Timeline	14
6.3	Jira Project Management	15
7	Project Description and Code	16
7.1	Models.....	16
7.1.1	Truck.....	16
7.1.2	Trailer	17
7.1.3	Map	17
7.2	RigidBody.....	18

7.3	Box Collider	18
7.4	Wheel Collider	19
7.5	Movement Controller Script	20
7.6	Parking Spot Script	22
7.7	Sky Texture	22
8	Ethics	23
9	Conclusion	24
10	References	25

1 Summary

The goal of my game was to create an interactive player-controlled parking game where you drive a truck with a semi-trailer attached to it and park it into a designated parking spot using basic keyboard controls. I also had an interest in AI which brought the idea of machine learning into my mind. I did some research about AI built into games and came across AI being taught to play hide and seek. [1] The idea of AI learning from itself really fascinated me and was something I was really interested in.

When doing my research, I found that reinforcement learning was the best type of machine learning for the project I had in mind. I wanted to see how the AI would match up against a player in terms of speed and accuracy to park the truck in the designated spot. Reinforcement learning is basically a reward and punishment system for teaching the AI. This data is sent to the Neuralink which is basically the brains of the AI.

Getting the main game working was a fun and interactive process that really brought my attention to how much I love game development and will look to work further in this field. Machine learning was something I underestimated and is something I know I would really enjoy with being able to put time into it.

I really enjoyed the research into ML Agents, the package that Unity provides for teaching agents. Although I struggled to implement this into my project with the time I had, I will continue to work on it in my spare time.

The software I will be using for my project will be C# for my unity scripts that are used to run my game. Python is a crucial part of the ML-Agents package which controls the learning aspect of the AI. I used Zmodeler 3 as my tool for building models that will be used as the assets in my game. I decided to use these tools as I've worked with C# while on my work placement with Avaya and have used Zmodeler 3 for my hobby where I create 3D models and python is something new to me that I have an interest in further developing my learning.

2 Poster



Figure 2-1 Poster

3 Introduction

During the time of picking my final year project, I looked for relevant problems in the world that aligned with my interests and how I could implement something that would help the issue at hand once fully developed.

For my final year project, I decided to look into developing a game which I have had interests in for years and within that game, try to teach AI to park a truck with a trailer. The idea game from my hobby of truck simulation and the recent shortage of HGV drivers around Europe which effected the economy in a negative way. I think this project can be developed and used to teach future drivers how to park with a large truck which is a skill people struggle with due to the sheer size of the vehicle.

4 Project Tools

4.1 Diagrams.net

Diagrams.net is a free and open-source cross-platform graph drawing software developed in HTML5 and JavaScript. [2] I've worked with this program in several modules within my course in GMIT and felt it would be the best tool to develop my Architecture Diagram and Project Timeline.

4.2 Paint.net

Paint.net is a freeware raster graphics editor program for Microsoft Windows, developed on the .NET Framework. [3] Paint.net is a program I've used in my hobbies of 3D model creation for conversion and creation of materials. With my experience in this tool, I felt confident in using it for my Project Poster.

4.3 Zmodeler 3

ZModeler (or Zanoza Modeler) is a 3D modeling application developed by Oleg Melashenko. It is aimed at modelers who model vehicles and other objects for computer games such as Euro Truck Simulator. [4] This product is a highly detailed modelling software with a paid subscription. I've created a hobby out of creating detailed 3D models for video games which is where my inspiration to create a model and create a game originally started from.

4.4 Microsoft Visual Studio 2019

Microsoft Visual Studio is an integrated development environment from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. [5] Visual Studio 2019 is a tool I am very familiar with from work placement and several C coding modules from college. I felt this tool is the one I am most comfortable with out of all the tools I've worked with.

4.5 Unity

Unity is a cross-platform game engine developed by Unity Technologies, the engine has since been gradually extended to support a variety of desktop, mobile, console and virtual reality platforms. It is particularly popular for iOS and Android mobile game development. [6] Unity was a software I had an interest in, way before even considering using it for my final year project. I looked into game creation before and found Unity to be the most beginner friendly with great documentation available online which was a big part in me deciding to go with it.

4.6 ML Agents

The Unity Machine Learning Agents Toolkit (ML-Agents) is an open-source project that enables games and simulations to serve as environments for training intelligent agents. [7] ML-Agents is a package designed by unity to allow the training of agents and simulation of agents. This allows for the creation of AI in games or the simulation of events to see how fast computers can learn.

4.7 GitHub

GitHub, Inc. is a provider of Internet hosting for software development and version control using Git. It offers the distributed version control and source code management functionality of Git, plus its own features. [8] GitHub was a program I wish we spent more time working with as a separate module as I felt it is crucial for the workplace and in the use of any project development. This allowed me to use version control which was necessary in the case of any lost work or issues.

4.8 Jira

Jira is a proprietary issue tracking product developed by Atlassian that allows bug tracking and agile project management. [9] Jira was a tool I've never worked with before as on my work placement, Avaya used a separate project management tool which they created. I found the initial stages of working with Jira a struggle but after some YouTube tutorials, I realised how important the tool is for a project and hope to work with it in the future too.

5 Project Architecture

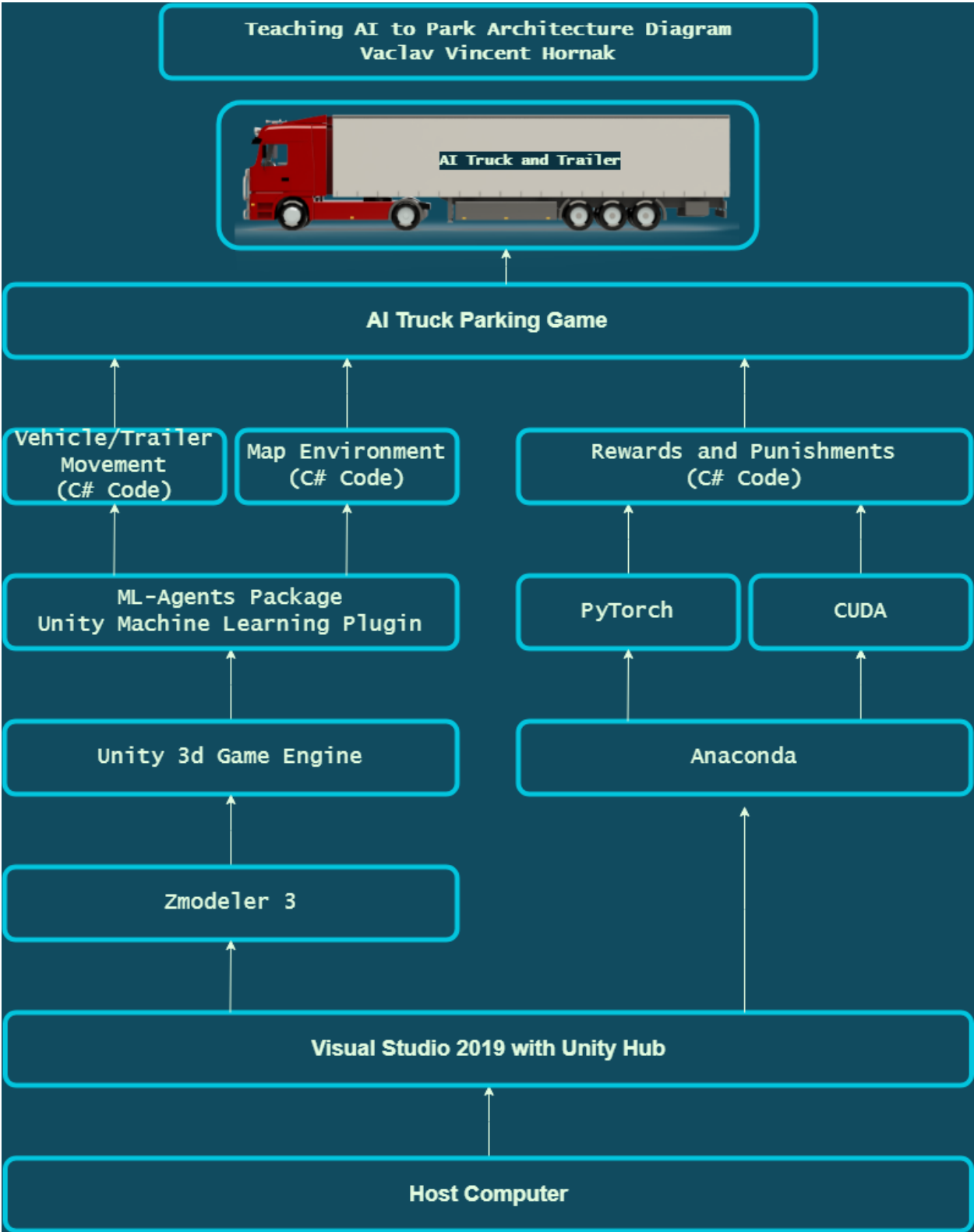


Figure 5-1 Architecture Diagram

6 Project Plan

6.1 Initial Timeline

My first stages of project planning consisted of coming up with a rough timeline of what I expect to occur. The timeline was created using diagrams.net and was a tool I used for planning before. I struggled with my project idea so getting some ideas and a plan was essential for the progression of my project. Machine learning was a new topic to me so figuring out how heavy the workload would be was a struggle and one I which I had addressed sooner.

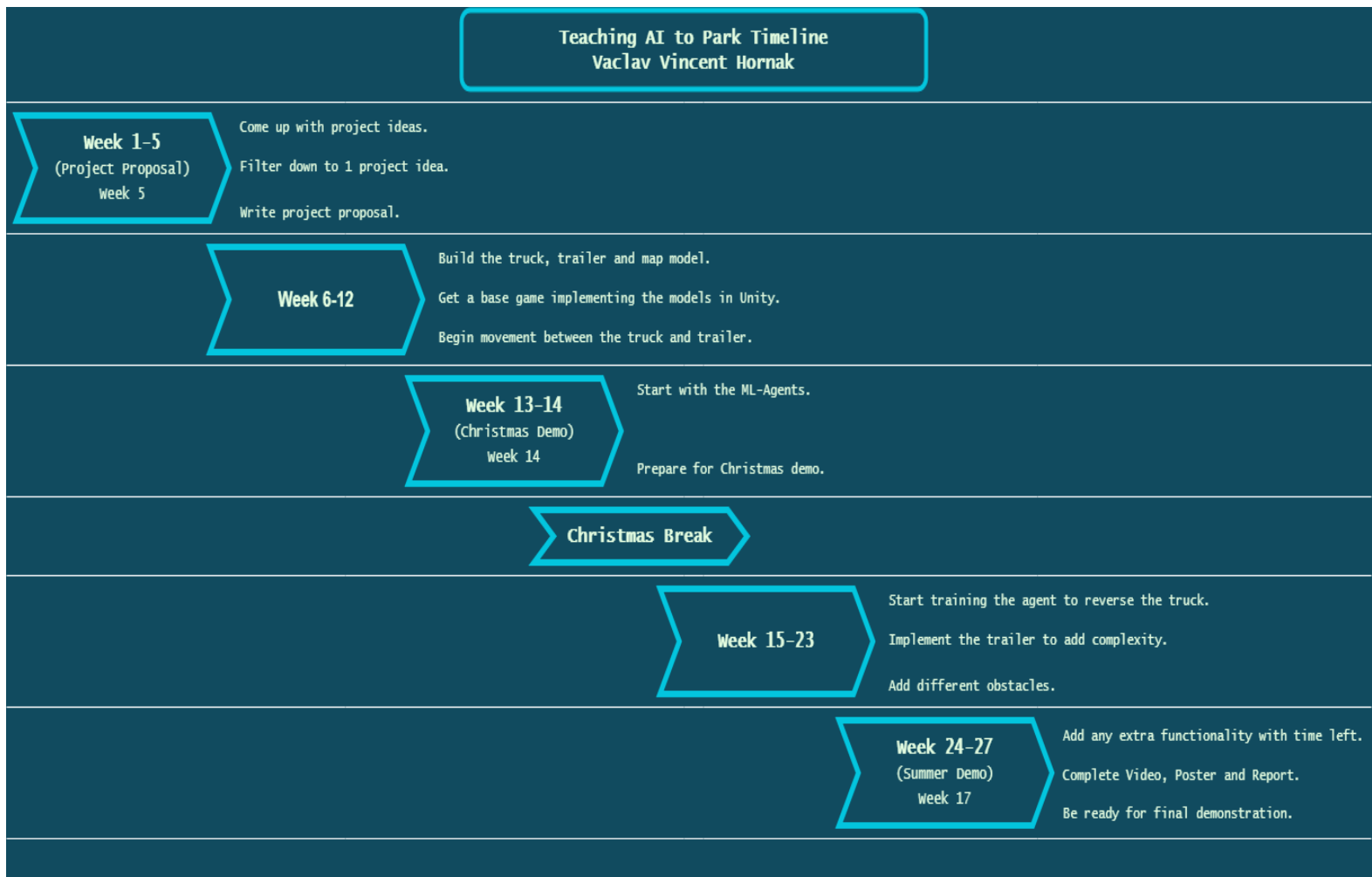


Figure 6-1 Initial Timeline

6.2 Revised Timeline

After the first semester I realised I needed to revise my timeline as I was starting to fall behind on my initial timeline. This time I decided to use Canva, and I feel a produced a better-looking result than my initial timeline. With a more accurate timeline that I thought I could achieve at the time. This timeline allowed me to feel like I am having a fresh start for the project.

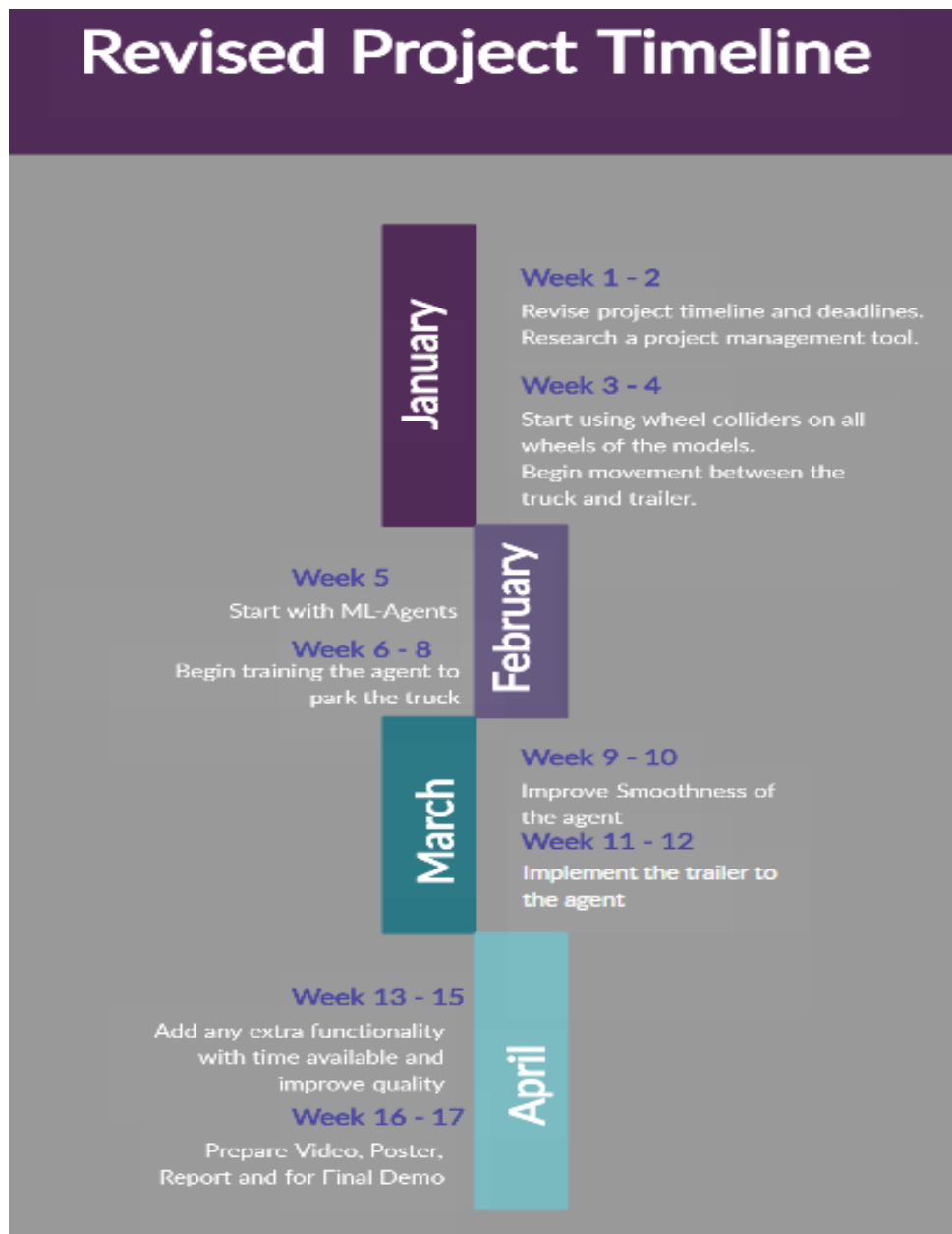


Figure 6-2 Revised Timeline

6.3 Jira Project Management

The importance of a project management tool was not something I considered till I realised the project timeline was not as interactive as I hoped and at some points wasn't descriptive enough. With the introduction of Jira, I was able to create a roadmap which is displayed below. Jira was very easy to operate and created a clear structure which I could showcase in my team meetings and see my progress.

I created Epic's for the main topic such as Vehicle Movement and divided it into sub tasks which were more specific but didn't clog the road map. This felt very rewarding and was nice to see the general progression of the project with this tool and it is one I will plan to use on my future projects.

I underestimated the sheer amount of work involved in machine learning and it was a part I struggled with. I learned a lot about the functionality but found it hard to implement it into my game which is shown on my roadmap.

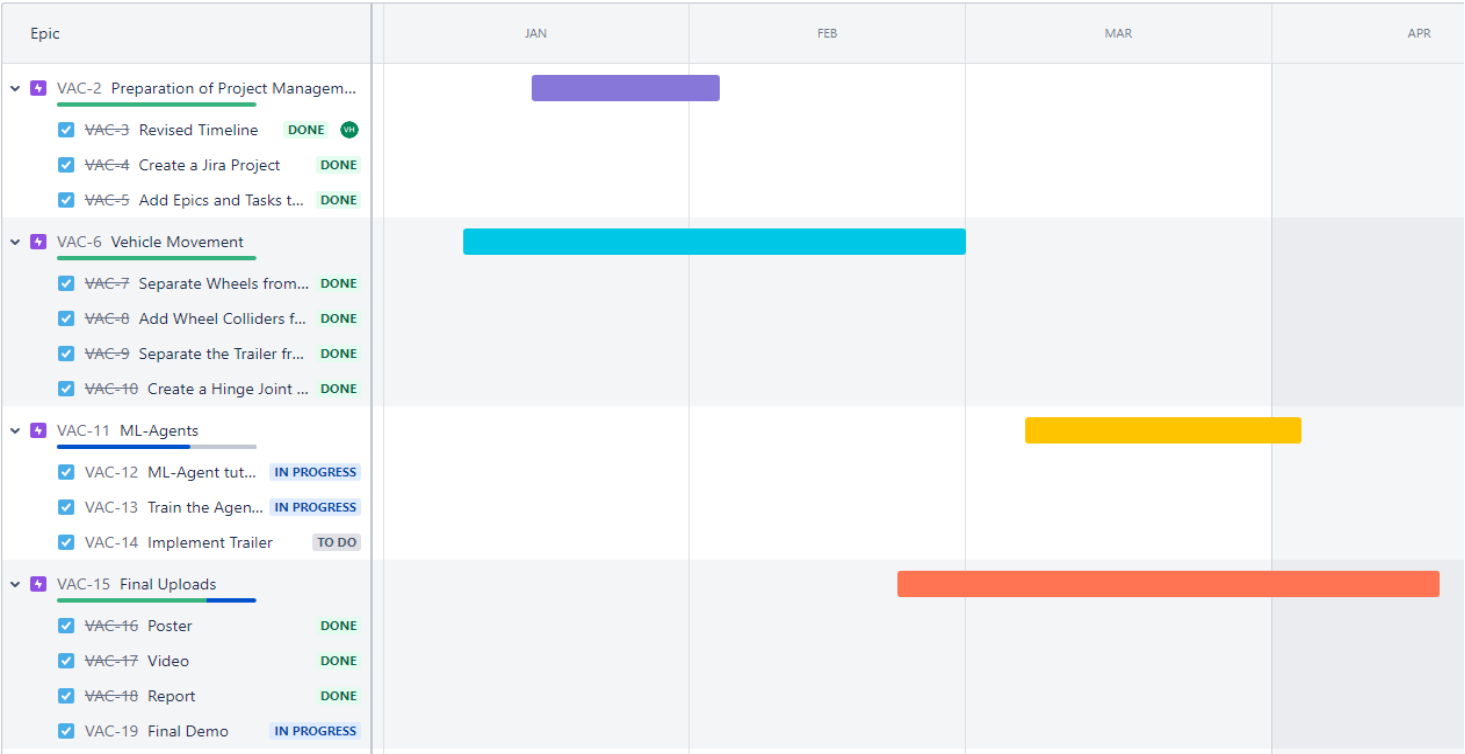


Figure 6-3 Jira Project Management

7 Project Description and Code

7.1 Models

The first part of my project consisted of creating some quality models that represented an accurate version of the vehicle in real life. I used Zmodeler 3 to work on these models and Blender to showcase some photos. For my unity game we did not need every detail as it is just a demo so to help performance, I removed any non-necessary parts to help with performance as the original model is very detailed.

The trailer is based on a real life Chereau trailer which was self-constructed in Zmodeler 3.

The map was the most fun part as this was a new experience for me creating a map, I created a yard with a basic building, shed and some parking spots which will be used later for my parking spots.

7.1.1 Truck



Figure 7-1 Truck 3D

7.1.2 Trailer

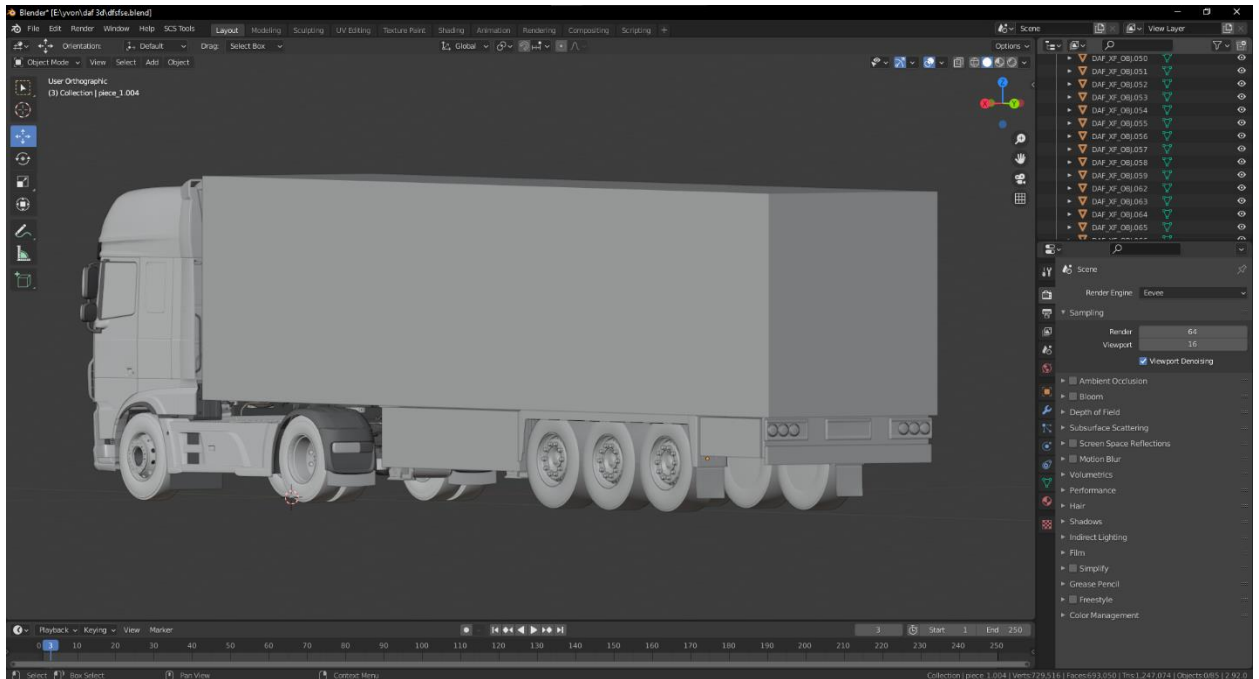


Figure 7-2 Trailer 3D

7.1.3 Map

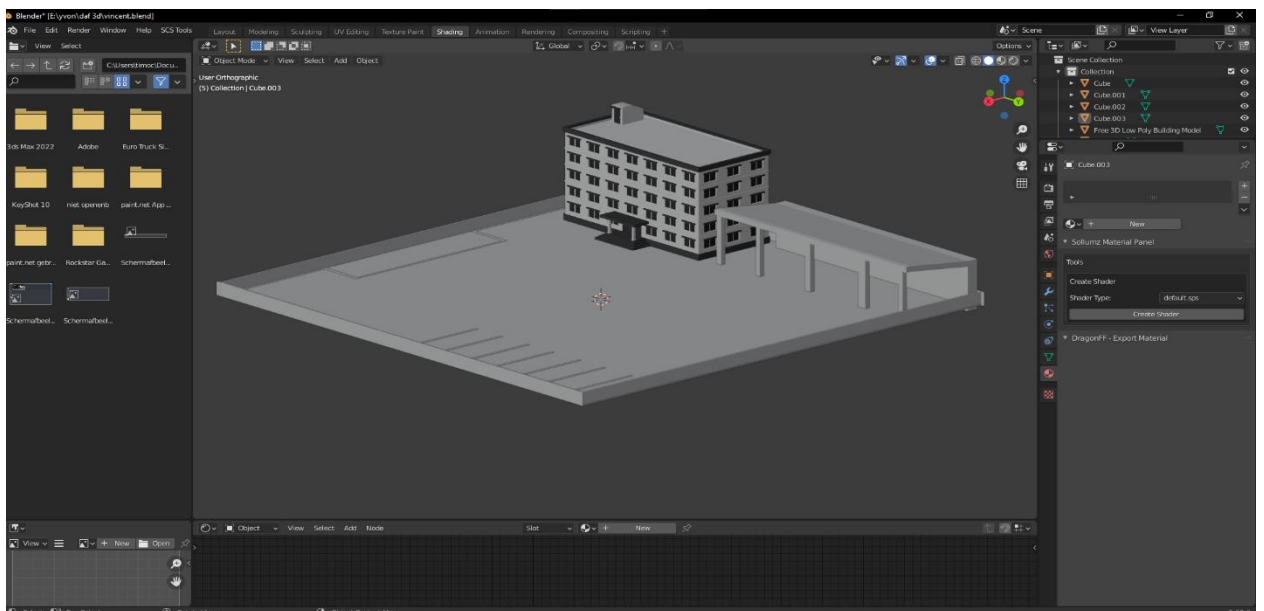


Figure 7-3 Map 3D

7.2 Rigidbody

Adding a Rigidbody component to an object will put its motion under the control of Unity's physics engine. Even without adding any code, a Rigidbody object will be pulled downward by gravity and will react to collisions with incoming objects if the right Collider component is also present. [10] The value I felt was best suited to the weight of the truck based on trial and error. I also tested a tutorial for a car which was using Mass 2500 which I felt was way too low for a truck this size, so I found this value worked well.

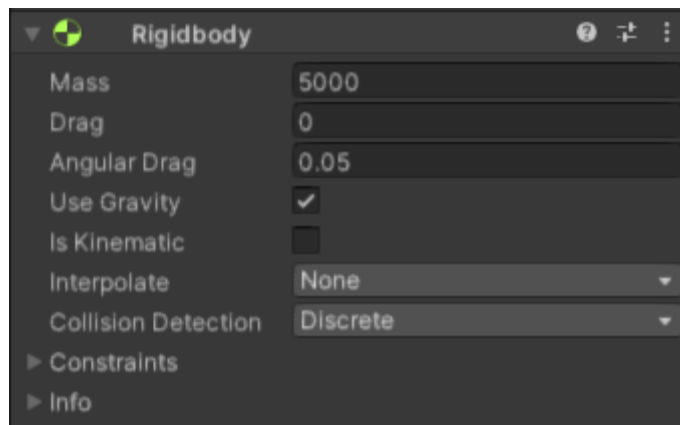


Figure 7-4 Rigidbody Config

7.3 Box Collider

The box collider creates a hitbox around the specified area which can cause a collision when interacting with another hitbox. This will later be used in the creation of the parking spot also.

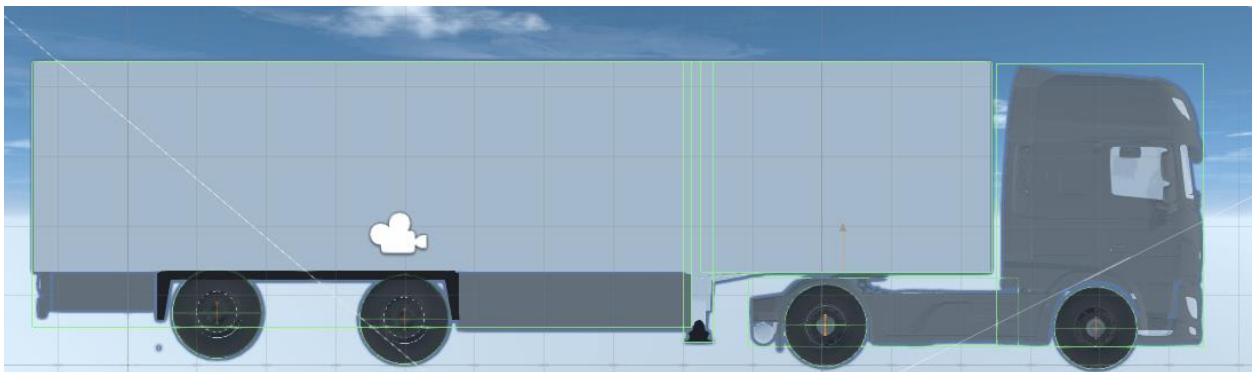


Figure 7-5 Box Collider Config

7.4 Wheel Collider

The wheel collider creates a hitbox around the wheels and is used in the Movement Controller Script to create the animation to turn the wheels. There was lots of trial and error involved with the mass of the wheel and getting the right suspension values so the vehicle wouldn't launch itself into the sky, but these are the values that ended up working the best for me.



Figure 7-6 Wheel Collider Config

7.5 Movement Controller Script

The main script running in my game is my Movement Controller script which basically controls all movement associated with the vehicle and trailer. SerializeField was something I've never used before and was very useful within unity for assigning certain objects to fields.

```
[SerializeField] private float motorForce;
[SerializeField] private float breakForce;
[SerializeField] private float maxSteerAngle;

[SerializeField] private WheelCollider frontLeftWheelCollider;
[SerializeField] private WheelCollider frontRightWheelCollider;
[SerializeField] private WheelCollider rearLeftWheelCollider;
[SerializeField] private WheelCollider rearRightWheelCollider;

[SerializeField] private Transform frontLeftWheelTransform;
[SerializeField] private Transform frontRightWheelTransform;
[SerializeField] private Transform rearLeftWheelTransform;
[SerializeField] private Transform rearRightWheelTransform;
```

Figure 7-7 SerializeField in C

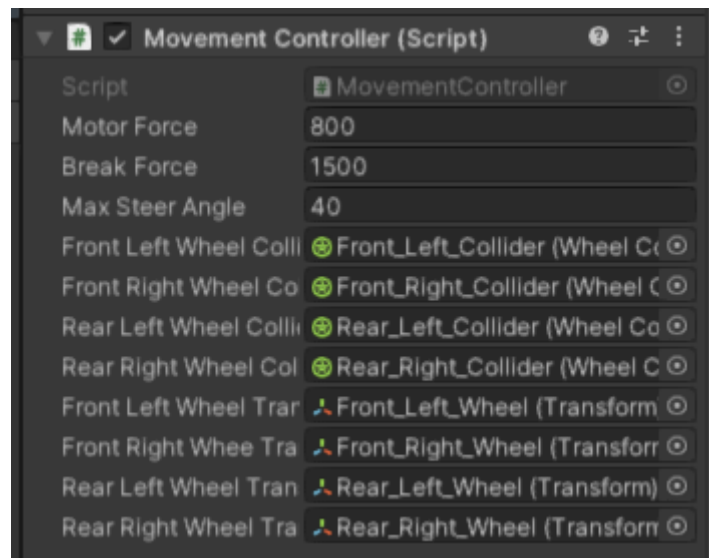


Figure 7-8 SerializeField in Unity

For the movement of the wheels we took the values initialised on the vehicles script and did some calculations to apply the torque to the vehicle. In this case all 4 vehicles were supplying power to the vehicle and using the breakforce to stop the vehicle. For the trailer wheels to move I had to create a fake motor on the wheels so they would still rotate while the truck pulled the trailer.

```
private void HandleMotor()
{
    frontLeftWheelCollider.motorTorque = verticalInput * motorForce;
    frontRightWheelCollider.motorTorque = verticalInput * motorForce;
    rearLeftWheelCollider.motorTorque = verticalInput * motorForce;
    rearRightWheelCollider.motorTorque = verticalInput * motorForce;
    currentBreakForce = isBreaking ? breakForce : 0f;
    ApplyBreaking();
}
```

Figure 7-9 Handle Motor Function

The update wheel function constantly updates the position of the wheel and creates a rotation animation while the vehicle is moving. Vector3 in unity allows us to send 3D positions which are necessary for the movement of the vehicle. Quaternion represents rotations which creates the animation of the wheels turning.

```
private void UpdateSingleWheel(WheelCollider wheelCollider, Transform wheelTransform)
{
    Vector3 pos;
    Quaternion rot;
    wheelCollider.GetWorldPose(out pos, out rot);
    wheelTransform.rotation = rot;
    wheelTransform.position = pos;
}
```

Figure 7-10 Update Wheel Position

7.6 Parking Spot Script

The Parking Spot script works as a box collider that has the onTrigger element enabled. Whenever two box colliders intersect the onTrigger is toggled it allows us to do an action which in my case was to display a message and restart the scene.

```
private void OnTriggerEnter(Collider other)
{
    Debug.Log("Parking Completed! Resetting Game");
    SceneManager.LoadScene(SceneManager.GetActiveScene().name);
}
```

Figure 7-11 Parking Spot Script

7.7 Sky Texture

I was able to locate a skybox set for free on the Unity Assets Store [11] to create a fairly realistic sky view and was able to shine a fake light to get the shadows in the right position I needed it in. This was very easy to implement once the skybox was installed and imported as an asset.



Figure 7-12 Sky Texture

8 Ethics

Ethics is a topic that I didn't ever consider until we started going through it in our modules. I majorly undervalued the topic and after extensive research in class and online I've realised the importance and feel it should always be considered when working in any field.

The first place I would like to discuss ethics is the deliverables side of the project. In the creation of these deliverables, we had to mention our research and reference any previous work. The use of work that is not your own without referencing is highly unethical as you are claiming other people's work as your own which fully defies the principles of ethics. With correct referencing you can use other people's work while still complying with the fundamentals of ethics and greatly enhances your credibility.

Another place I would like to discuss ethics in is the video game side. This is a highly debated topic over the last few years due to violence in video games and concerns of addiction to these video games. I am not here to argue against either side, but I feel both have brought up valuable comments and it is a topic I would like to delve into in the future.

9 Conclusion

In conclusion, I was happy with the progression and the knowledge I have gained from the topic of game development and machine learning. Although I didn't get to implement the machine learning fully in my project, the game development side was a success. The unity software was one that I really enjoyed working with and look forward to working with it again. I feel a lot better educated on the topic of reinforcement learning and have a greater understanding for neuralinks, after my progressive research.

I look forward to the development of my project in my own time and plan to get the reinforcement learning up on running and further my knowledge in game development as it's a path I would like to look at in my career.

10 References

- [1] T. M. Papers, "OpenAI Plays Hide and Seek," Two Minute Papers, 22 October 2019. [Online]. Available: <https://www.youtube.com/watch?v=Lu56xVIZ40M>. [Accessed 4 November 2021].
- [2] "Diagrams.net," Wikipedia, 19 April 2022. [Online]. Available: <https://en.wikipedia.org/wiki/Diagrams.net>. [Accessed 20 April 2022].
- [3] "Paint.NET," Wikipedia, 3 March 2022. [Online]. Available: <https://en.wikipedia.org/wiki/Paint.net>. [Accessed 19 April 2022].
- [4] "Zmodeler," Wikipedia, 14 June 2021. [Online]. Available: <https://en.wikipedia.org/wiki/Zmodeler>. [Accessed 19 April 2022].
- [5] "Microsoft Visual Studio," Wikipedia, 19 April 2022. [Online]. Available: https://en.wikipedia.org/wiki/Microsoft_Visual_Studio. [Accessed 19 April 2022].
- [6] "Unity Game Engine," Wikipedia, 14 April 2022. [Online]. Available: [https://en.wikipedia.org/wiki/Unity_\(game_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine)). [Accessed 2022 April 2022].
- [7] A. B. V. T. E. C. A. H. J. E. C. G. C. G. Y. H. H. M. M. L. D. Juliani, "Unity ML-Agents," Unity, 20 March 2020. [Online]. Available: <https://github.com/Unity-Technologies/ml-agents>. [Accessed 19 April 2022].
- [8] "GitHub," Wikipedia, 16 April 2022. [Online]. Available: <https://en.wikipedia.org/wiki/GitHub>. [Accessed 2019 April 2022].
- [9] "Jira," Wikipedia, 3 April 2022. [Online]. Available: [https://en.wikipedia.org/wiki/Jira_\(software\)](https://en.wikipedia.org/wiki/Jira_(software)). [Accessed 19 April 2022].

- [10] "RigidBody Documentation," Unity, 16 April 2022. [Online]. Available:
<https://docs.unity3d.com/ScriptReference/Rigidbody.html>. [Accessed 19 April 2022].
- [11] Avionx, "Skybox Series Unity Asset Store," 4 January 2022. [Online]. Available:
<https://assetstore.unity.com/packages/2d/textures-materials/sky/skybox-series-free-103633>. [Accessed 15 Decemeber 2021].