

CS324 Coursework Report

The AMG Experience

(Word count - 971)

Student ID - 1913904

16th January 2022

1 Introduction

The focus and style of the game I have created is themed around a racing-type video game, titled as ‘The AMG Experience’. In particular, existing franchises including the likes of Need for Speed [6], Gran Turismo [3] etc. have provided motivation for me to implement a similar game-style. The player drives a ‘Majestic-Benz’, modelled around a SLS AMG Roadster by Mercedes-Benz [5], crafted from scratch in Blender (more on this discussed in section 2.2), racing against the clock and setting impressive lap-times.

2 Design Aspects

The majority of the design of the game has been made possible using libraries, Three.JS, Cannon.JS, Blender and HTML/CSS.

2.1 Game levels

To provide a sense of challenge to the game, the player will be faced with an increase in difficulty with each level passed. The player is tasked with completing lap times under an increasingly tighter time-frame e.g. completing a lap in under 30 seconds, 25 seconds, 20 seconds etc.

Along with stricter time constraints, environmental changes also occur including increasing levels of fog to affect player vision to navigate the car through the racetrack. This also forces the player to switch camera view, namely the ‘third-person’ camera view, otherwise visibility is significantly reduced.

2.2 Blender Model

The figures listed below, Figures 1, 2, 3, 4 and 5, illustrated the process of the blender model creation of the Majestic-Benz car from scratch. The first step was to create the dimension of the car with a cube, stretched and extended to match that of the blueprint image. This was completed by alternating viewpoints top, front and left and also switching between shading views solid and wireframe.

Moreover, acute details such as grill, wing mirrors, headlights etc. could now be added. The finishing touches was adding colours - The ‘Material Properties’ tab, allowed us to add materials, adding a deep blue colour to the car and black for wheels, roof and spoiler. We could also add image textures - a carbon fibre style image was used to colour the bonnet. Figure 5 displays this.

This was all implemented using symmetry where we altered the ‘Modifier Properties’ and mirrored the result in the x-axis - We simply change one half of the car and the other half is completed. Other blender

models that were used in-game that we did not create from scratch are discussed in section 2.4.1.

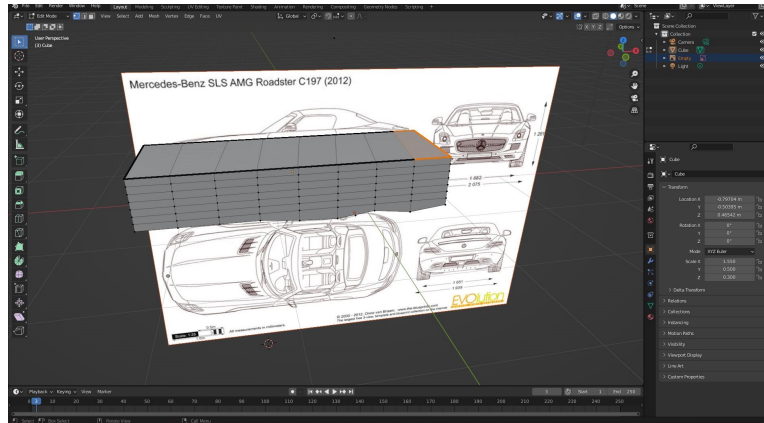


Figure 1: Initial body shape for the car, starting off from a cube and extruded

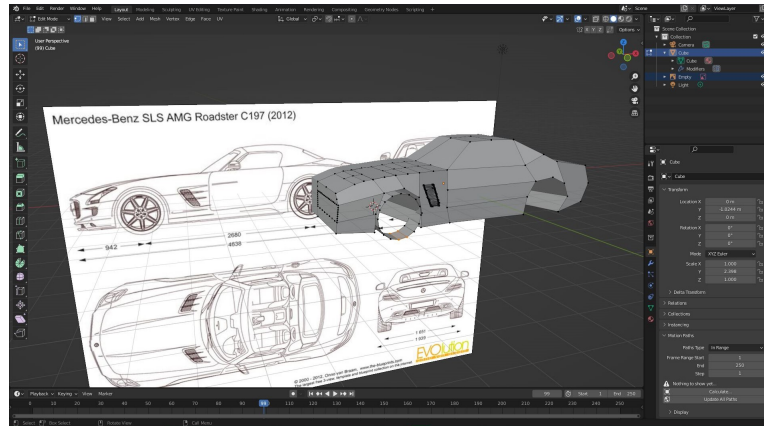


Figure 2: Translation of vertices & edges to fit the car blueprint outline

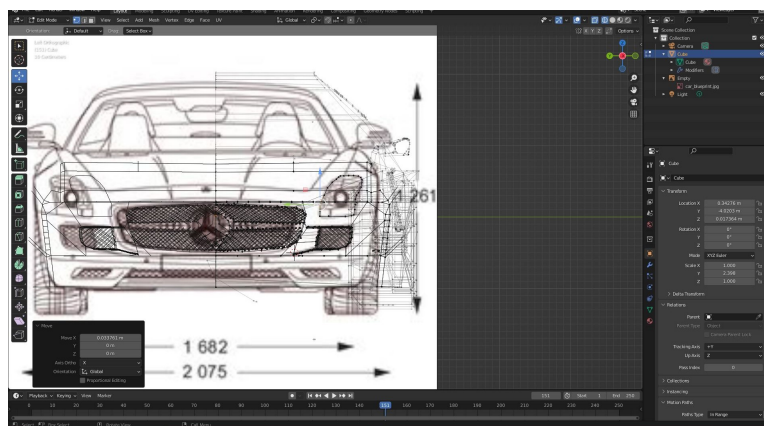


Figure 3: 'Wireframe' view to outline and trace the shape for frontal details like grill and headlights

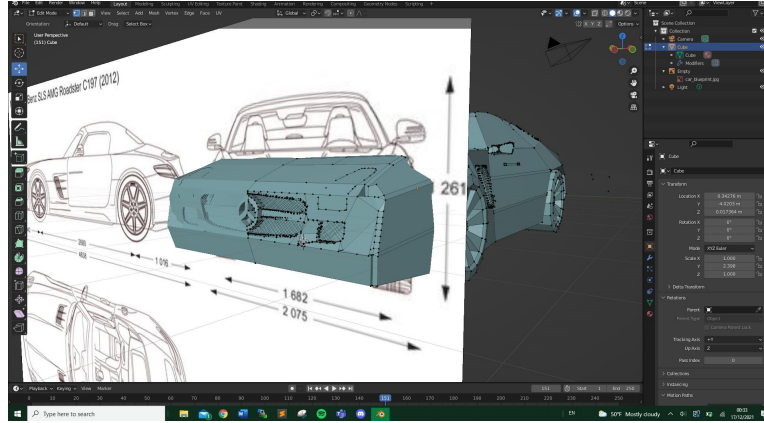


Figure 4: Adding car's features including grill and headlights

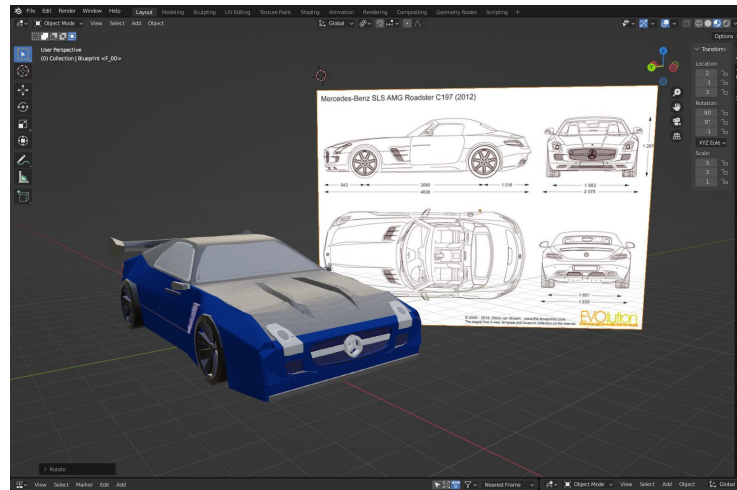


Figure 5: Final shape of the Majestic-Benz car model - Added carbon-fibre texturing to bonnet and spoiler

2.3 Animation

The animations produced in the game involve the turning of wheels for the car model when moving forward/s/backwards and steering left/right. These were implemented in Blender and exported to Three.js. Figure 6 involves capturing the initial position of the car, then onto Figure 7 that captures the position after wheel rotation, plotted at later frames. This was repeated for the other animations.

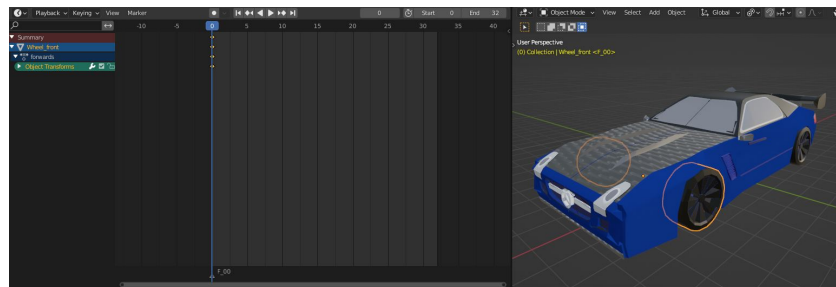


Figure 6: Insertion of initial keyframes

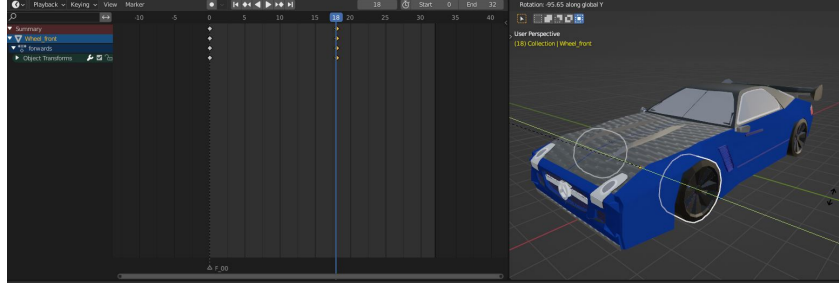


Figure 7: Insertion of keyframes at later frames, after rotation of wheels along y-axis

2.4 Visuals

2.4.1 Game Environment

To set the scene for the game, a racetrack layout had to be created. Firstly, the plane was created and was wrapped with a racetrack texture [8] before blender objects may be placed in the scene. These included concrete barriers [2], trees [4], street lamps [9] and tyres [11].

With the use of library Cannon.js, all blender models used (except trees) were wrapped in a bounded box, together with the Majestic-Benz car model whose bounded box position follows the car's position at each frame, updated in the 'updatePhysics()' function. These were to simulate physics in the real-world, using collision detection, adjusting their respective positions accordingly.

For the street lamp model, a THREE PointLight was used to illuminate the scene - turned on when the game is ran at night (day/night chosen on page load). This acts as one of the two light sources, the other being the sun chosen to be a DirectionalLight source, illuminating the entire plane.

Finally, a sky-box was added to set the background for the scene - six textures assigned to each face of a cube with the plane inside to provide environmental realism.

2.4.2 Main Menu Interface

The main menu acts as the entry-point and contains a series of headings for the player to choose from. The design choices in mind were user-friendliness, providing a real "feel" for the game and importantly sets the theme of the game. Players may see the brief description of the game and how to play it under the relevant headings.

2.4.3 Head-Up Display (HUD)

Whilst the game is being played, live information is displayed including lap number and lap times on the left and on the bottom dashboard, the speedometer of the car along with the current time of the lap and what type of camera view the player is currently in. Players may access different views using keys 1, 2 and 3 for follow, fixed and top-down views respectively (controls may be found in the corresponding heading in the main menu).

2.5 Music/Sound Effects

It was evident that from a choosing a racing-style game, the game had to include car sounds including acceleration, deceleration, idling, pops and bangs etc. for full user experience. With every lap completed either successfully or not, a successful ding sound is played if the level is satisfied. Otherwise, a loud buzzer to indicate failure is played.

With every game, they usually have a theme song - The chosen song is 'Workshop' from creators Jagex

renowned for the game, RuneScape [7]. Their music is royalty-free and non-copyrighted, made readily available to everyone to use.

3 External Code Sources

For the purposes of learning and implementing physics, using Cannon.js, alongside the code in Three.js, I used Mr Guo's online article on how to initialise a Cannon world [1]. This initially involved creating a rigid Cannon body overlapping the Three plane, wrapping meshes around objects in the scene and adjusting their data according to data by the physics engine.

Moreover, implementation for the third-person camera (using key 1) to follow the car model smoothly was inspired by examples by CodePen.io [10]. The solution being to 'lerp' a camera to the goal, being the car, and keeping a constant distance away from it.

4 Setting Up

As seen in labs to run the WebGL application in the browser, run command `'python3 -m http.server'` in the home directory and navigate to 'index.html' to access the game. Any errors concerning 'address in-use', run on a different port e.g. `'python3 -m http.server 7000'`.

References

- [1] Cannon.js article - Physics engine of Three.js. Mr Guo, 2021-01-18. URL: <https://www.mrguo.link/article?id=50> (accessed on 26/12/2021).
- [2] Concrete Street Barrier 3D Model. Free3D, 25/12/2020. URL: <https://free3d.com/3d-model/concrete-street-barrier-917223.html> (accessed on 23/12/2021).
- [3] Gran Turismo Video Game. Gran Turismo, 2022. URL: <https://www.gran-turismo.com/gb/> (accessed on 16/01/2022).
- [4] Low Poly Tree 3D Model. Free3D, 12/09/2017. URL: <https://free3d.com/3d-model/lowpolytree-54187.html> (accessed on 21/12/2021).
- [5] Mercedes-Benz SLS AMG Roadster review. AutoExpress, 2013-01-13. URL: <https://www.autoexpress.co.uk/mercedes/sls-amg/roadster> (accessed on 01/12/2021).
- [6] Need For Speed Video Game. EA, 2022. URL: <https://www.ea.com/en-gb/games/need-for-speed> (accessed on 16/01/2022).
- [7] Old Style RuneScape. Jagex, 2001. URL: <https://oldschool.runescape.com/> (accessed on 05/01/2022).
- [8] Racetrack Image. Unsplash, 2001. URL: <https://unsplash.com/photos/MekX0hHQVY4> (accessed on 21/12/2021).
- [9] Street Light (Lamp) 3D Model. Free3D, 20/11/2019. URL: <https://free3d.com/3d-model/street-light-lamp-61903.html> (accessed on 21/12/2021).
- [10] Third-Person Camera(pulling) + Keyboard. Fyrestar, CodePen.io. URL: <https://codepen.io/Fyrestar/pen/WNwMWZV> (accessed on 20/12/2021).
- [11] Wheel 3D Model. Free3D, 30/07/2017. URL: <https://free3d.com/3d-model/wheel-13956.html> (accessed on 21/12/2021).