

CPT 205 Assessment2

3D Modelling Project

Module	CPT205 Computer Graphics
Degree Programme	Information and Computing Science
Name	Ruiyang Wu
Student ID	2257475

1. Brief Description

The title of my 3D Modelling Project is “**Five Hundred Miles ~ Way Back Home 66**”, players can control the vehicle to dash on US Route 66, collect fuel and avoid obstacles, and ultimately reach the finish line in time before running out of fuel to win.



Figure1. Game Screen Preview

I have created classic Western Highway landscapes, including road surfaces, deserts, sky, fog, power poles, wires, etc. During the gameplay, the lighting changes over time, and players can turn on their car lights for illumination or change their perspective to appreciate the scenery.

2. Features

2.1 Texture Mapping & Multisampling

Due to the inability to use other libraries, a self-made method is used in this project to read the colors of each pixel in the .bmp file and store them in the cache for mapping. “**GitexCoord2f**” binds the vertex positions of the texture and model, allowing the texture to be displayed in the correct position in space.

By reversing the normal and pasting the texture on the inside of the sphere, I successfully achieved a **Sky Dome**.

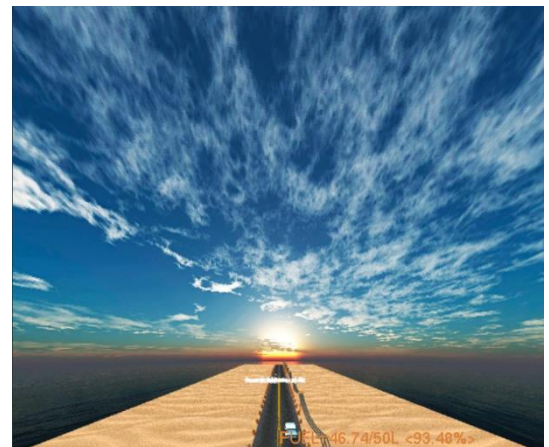


Figure2. Demonstration of Sky Dome

Due to the high frequency information contained in the Road and Desert textures in this project (the resolution is high as well), the movement of the texture may cause the texture to flicker. I used **Multisampling** instead of **mipmapping** to solve this problem. This is because in this circumstance, the perspective is relatively large, and mipmapping may cause significant resolution switching issues, while the effect of **Multisampling** is more natural.



Figure3. Texture of Desert, Glass and Road

2.2 Lighting and Materials

In this game, I used a **directional light from infinity** to simulate sunlight. Although the effect of creating shadows between different objects has not been achieved, changes of lighting angles and colors can still be observed from diffuse and specular reflections in different directions of the model.

When night falls, players can turn on the **car lights** and **dashboard lighting** to illuminate obstacles and oil drums on the road and see the speed on the dashboard. The position of the car lights is tied to the car, which can affect the model in the environment in real time, producing diffuse and specular reflections (which can be observed on wires).

For some objects, such as road and deserts, I apply textures to basic diffuse materials to represent the original colors of the textures. In the treatment of car window glass, I used **semi-transparent materials** and high reflectivity to reflect the characteristics of the glass. I used **emission materials** for the taillights and dashboard of the car. For other objects such as wires, wooden poles, and fuel tanks, I finely control the **specular, shininess, and color** of the materials to give them a rubber, wood, or metal feeling.

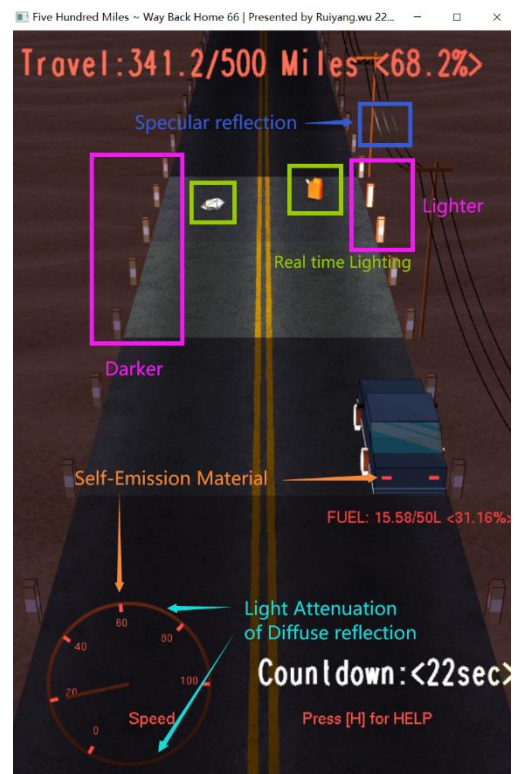


Figure4. Typical Lighting Effect at Night

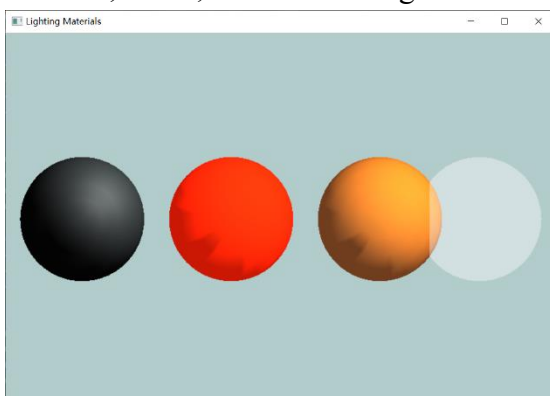


Figure5. Material Sphere of Rubber, Gas Tank, Telegraph Pole and Glass.

2.3 Hierarchical Modeling and Parametric Curve

I applied hierarchical modeling in modeling Cars, Telegraph Poles, Gas Tanks and Highway Guardrail. Local transformation only affects a portion and its sub items, while global transformation affects all its sub items. This allows for easy manipulation of models composed of multiple geometries, while also allowing for changes in the position and state of components without affecting the overall model. To simulate the natural shape of wires, I used calculated parabolas as lines and created wires in space using parametric equations, the thickness can be controlled by `glLineWidth()`.

2.4 Viewing and Projection

In the program, I designed two perspectives, and the default perspective imitates a recently popular game, which is the top-down perspective. The second perspective is a moving third person perspective, where the position and orientation of the perspective change as the vehicle turns, creating a sense of following and presence. In addition, I also achieved camera shake by changing the upward direction of the view matrix. The transformation between two perspectives has a linear animation.



Figure6. Animation of Perspective Transition

2.5 Dynamicity

`GlutTimerFunc()` is called in the main function to call `glutPostRedisplay()` after the specified time interval to achieve dynamic effects. It ensures smooth and stable movements, ensuring stable frame rates (60FPS in this project) on different environments.

2.6 Fog

The fog effect that comes with OpenGL can be activated through `glEnable(GL_FOG)`. Fog will intensify with increasing distance. The density of fog can be changed by setting `GL_FOG_DENSITY`.



Figure7. Fog

2.7 Draw Vector Fonts

The vector font rendering in OpenGL can be activated through `glutStrokeCharacter()`. Fonts are scalable and support transformations like rotation and scaling. The size of the font can be adjusted using `glScalef()`.



Figure7. Vector Fonts

3. Instructions

3.1 Keyboard interaction

- Change Lanes: Press 'a', 'A' to turn left; 'd', 'D' to turn right.
- Control Throttle: Press 'w', 'W' to accelerate; 's', 'S' to decelerate.
- Turn on Light: Press 'Space'.
- Change Game View: Press 'f', 'F'.
- Help: Press 'h', 'H'.
- Quit the game: Press 'q', 'Q'.

3.2 Game Rules

Players assume the role of a driver returning home and driving on Route 66. The **game goal** is to successfully drive 500 miles home.

There are **two constraints** in the game: time and fuel consumption. The faster the driving speed (the shorter the time), the higher the fuel consumption. Players must reach the finish line before the countdown ends; At the same time, it is necessary to ensure that the vehicle does not run out of fuel during driving.

Enjoy the Game! Screenshots are on the next page.

Typical Screenshots

