Project Report

# Group IDs

B00271629  
B00271249  
B00270035

# Aims

We aim to create a tech demo using the skills we learned via our labs from our course. We aim to use a skybox that sets itself to the player along with collision into objects. We are aiming to make the tech demo in the form of a simple game that will allow the user to immerse themselves into our tech demo so that they can see all the features that we have.

# Specification

Create a program using all the techniques and libraries we’ve learned over the course of the module to show what we’ve learned, and have it compile and run in the labs.

# How it will be made

To make this project we are going to use the “HUD label example” project on moodle as the base of our project since it has examples on how to implement an animated MD2 model and displaying text in a 2D and 3D environment.

We will be using SDL2, OpenGL, GLM and BASS to create the project.

# Gameplay Instructions

Our gameplay is intuitive to today’s standards so the button combinations are nice and easy.

W – Walk Forward

S – Walk Back

A – Strafe Left

D – Strafe Right

Mouse move left: Turn left

Mouse move right: Turn right

Alt F4 – close program

The player will have to collect the collectable squares around the map by walking into them causing collision, not only the boxes but walking into the walls will cause collision to keep the player locked inside of the level area. To allow the players to know their score we have boxes appear along the bottom of the screen for every object they pick up.

When the player spawns in there is a label that indicates their instructions on what to do for example it states “collect the boxes!”.

# Class Design

Our first priority is to refactor as much as the existing code as much as possible to allow us to create the environment much easier. Below is the proposed design for the animated MD2 model class and the Skybox class.



The purpose of the Model class is to implement the existing code used to render and animate the MD2 model so that it can be used to represent the player and avoid having all the variables the model uses being global variables. It should also allow us to load in our own models much easier, including models for the terrain.

The existing code for the skybox also uses global variables, and takes up a considerable amount of space in the main.cpp file with code that is mostly the same with variations on the texture being used and the position. Having the skybox refactored into a class will clear up a lot of space in main, and avoid the skybox being removed or changed accidently. It will also make it easier for us to edit the skybox and use it in later projects.

# Development Issues faced

We had a lot of issues making the game object orientated. First off we had issues creating a new MD2 model class to use for the player. We managed to have the mesh loaded and rendered with no issues. However when we tried to run the animation code, the vert data was somehow becoming unreadable which caused the program to break. After spending a lot of time on it and seeking help around it, we sadly could not fix this and had to change it to main.cpp coded.

Another issue is that when we created a static mesh class. In the original version of the class we were loading in the mesh for every object, but since we were using the same mesh for a lot of the terrain, it didn’t make much sense to keep loading in the same data repeatably. So we added a new constructor that would pass in an existing mesh and texture. Unfortunately when the class would attempt to render mesh that was loaded in main.cpp, nothing was being rendered at all. We spent a lot of time trying to figure out the issue, but we had already lost a considerable amount of time trouble shooting the issue with the MD2 model class we had before. Sadly, with this we had to create the meshes in main.cpp but the class now only holds the data for the meshes, such as positions rotations scales and textures used for each model.

# Contribution

**B00271629:** Designed and programmed the model map layout

**B00270035:** Wrote and designed object orientated classes and animation

**B00271249**: Wrote collision code and assisted with class creation

# Conclusion

We were able to get the demo to a playable state despite not being able to fully refactor the code like we had intended. But overall we are happy with the final result.

# Future Work

If we were to go back to this project in the future, we would attempt to refactor the code again like we originally planned to, which would make editing and creating terrains easier. We would also add new elements such as other animated characters into the game world, and have some sort of hazard that the player would have to avoid, such as spikes or fire.

# Links

Youtube link: <https://youtu.be/Fn9T3XuvK0c>

GitHub Repository: <https://github.com/Jaymzeh/RT3D-Project>

# Credits

Base project (HUD label example no leakage) provided by Dr. Pablo Casaseca.

Bass sound library: <http://www.un4seen.com/>

SDL2 library <https://www.libsdl.org/credits.php>

OpenGL <https://www.opengl.org/wiki/Main_Page>

Wooden Box texture : [Alekei on opengameart.org](http://opengameart.org/content/box-2)

Audio track: [Improbable Cadence on opengamert.org](http://opengameart.org/content/the-forgotten-age)

Building Texture: [textures.com](http://www.textures.com/download/buildingsderelict0094/82472)

Metal fence texture: [textures.com](http://www.textures.com/download/metalbare0171/98230?q=MetalBare0171_1_S&filter=all)

Barrel texture: truzipp on [opengameart.org](http://opengameart.org/content/pixel-barrel)