**Real-Time 3D Coursework Report**

**B00317267**

**Team:**

**B00317267, B00287629, B00324457**

# Objectives

The main aim of this project was to create a tech demo where the player can control a person in a 3D, lit, world. My group and I decided to create a coin collector game which would have the player moving around a 3D area collecting coins and adding to an overall score. To take this from a tech demo to a game our aim was to add a timer which the player would race against to collect as many coins as possible before the time ran out.

# Design

We used OpenGL for rendering as we had previously worked with OpenGL in other modules and as a group we were comfortable using this for the rendering of objects within out project.

Due to OpenGL's limitations of only being a renderer the group and I chose to use SDL2 for creating windows as we had used it throughout the labs of Real-Time 3D and were becoming accustomed to using it and wanted to show our abilities in using SDL2.

As we were using SDL2 for creating windows this opened us up to more possibilities, such as using the SDL2\_TTF library for creating labels which we used for a basic HUD system.

We also used the GLM (OpenGL mathematics) library for any mathematics which was done within our code, this was code we had learned about from other modules and became useful to us, this is shown in calls to matrices & converting degrees to radians by calling the glm::radians() function, which replaced the macro from the labs.

The sound files were loaded and played using the BASS library which was a library giving to us during the 9th lab of Real-Time 3D and became useful to our group when looking for a way to play sound effects.

We chose to use a 3rd person camera for this project as the 1st person camera was given to us during the labs and we wanted to show that we had learned something more than just the basics. This also allowed us to show off our 3D model and animations easily.

Since we were tasked to create a tech demo in 3D we used the RT3D Library which provided the group and I a lot of useful code for working in 3D, from lighting to loading meshes for objects and drawing everything in 3D space.

For the 3D models we used the MD3 model format as we were given a 3D model which uses the MD2 model format and we did not have enough time to learn how to implement a different type of model and instead stuck with what we were comfortable programming with, I.e; the MD2 model format.

# User Interaction

**Turning the camera:** Pressing the **RIGHT MOUSE BUTTON** down & **rotating** the mouse allows the player to move the camera on the x axis which in turn also turns the 3D model.

**Jumping:** Pressing the SPACE key down allows the players jump animation to begin playing, holding it down will allow the players jump animation to play on loop until the key is released.

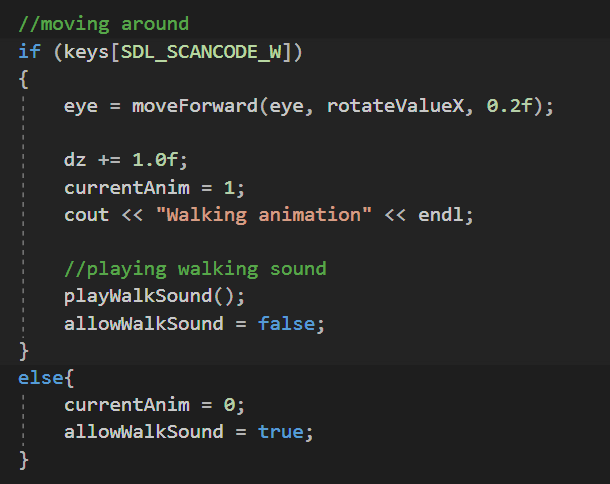
**W:** move the character & camera forward

**S:** move the character & camera backwards

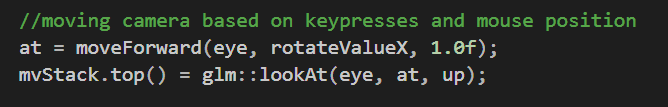
**A:** move the character & camera to the left

**D:** move the character & camera to the left

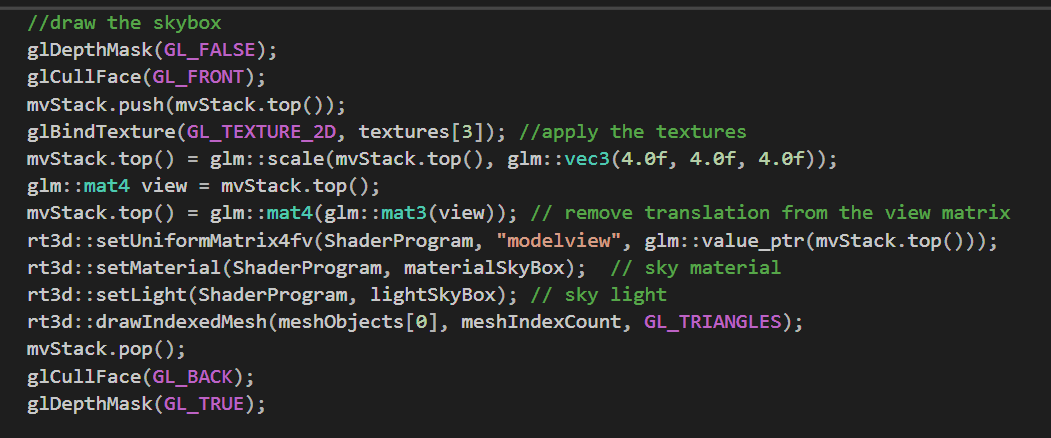
# Implementation



This code snippet shows how the user moves the camera and, in turn, the 3D model in the Update method. Firstly, the keyboard is checked to see if the "W" key has been pressed, if it has then we make the eye (where the camera is positioned) move forward based on where the camera is pointing. To make the players model move with the camera I have Implemented some basic code to get the character always moving positive along the z axis when W is pressed. Then I set the current animation to the running animation to give the game a more relatable feeling. For debugging purposes, I write some words to the console window while this code is running to make sure that it is being triggered when the key is pressed then I allow the walking sound effect method to be played only once per keypress using the next 2 lines of code. Finally, I have a check for is the "W" key is not pressed, if this is so then the current animation is set to its idle and allow the sound effect to be played again if the key is pressed again.



This code shows how the camera works taken from the Draw method. Firstly, the cameras "at" value is set to the move forward method to allow the camera to look at the correct position and not lock to the first "at" value which is set at run time. I also implemented a "rotateValueX" integer in the move forward method, this is a number which is updated when the mouse is pressed down and dragged across the screen which in turn updates the cameras rotation on the X axis. Next, we set the LookAt function to be the top of the stack and pass in 3 important vectors: "eye" which is the cameras position in the scene, "at" which is where in the scene the camera is currently looking at and the "up" vector which is used to make sure the camera isn't upside-down.



This is the code for drawing the skybox from the Draw method, I felt this was important to include as it was not easy to get working and I am proud of the work we did to get it working. Firstly, the depth mask is disabled as we want to draw the skybox before we enable this. Then we chose which texture we would like to use for the skybox. Next the skybox is scaled up as we want the scene to be inside the skybox. Then we remove the possibility of moving the skybox by taking it out of the matrix. Next, we set the shaders and put it into the model view, then we set the lights and materials for the skybox and draw it to the screen before re-enabling the depth buffer and drawing the rest of the scene

# Results / critical reflection

I believe the greatest result of working on this project and one of the best parts of our group project would be learning about skyboxes. After implementing this the project felt a lot more complete and I was much more satisfied by the final outcome. This was probably one of the hardest parts to implement, however it was something we weren't willing to leave out and therefor worked the hardest on this section to make sure it was completed.

Another good aspect of the code would be the label for the HUD. I found this to be one of the more challenging aspects to code (mainly due to the fact I hadn't noticed until it was later pointed out to me that there was a lab which explained how to do it.) however after a few failed attempts I managed to eventually work out how to create and use the labels. The one thing I am disappointed about with the labels is that I was unable to get numbers displaying properly, I assume the issue is that "char" can't hold integer/ float values, however I ran out of time working on other aspects of the code and was unable to fix this issue.

A section of this project I am disappointed with is the 3rd person camera. Even though I managed to get the camera to follow the 3D character when you first move left, right, forwards & backwards, after turning the player model using the mouse to face a different direction the camera and 3D character do not move forward in the expected way, this is a very annoying problem and I know what is causing the issue, however after running out of time working on other aspects of the code to improve the overall project, this element was pushed to one side and as a group we eventually ran out of time.

# What would I do next time?

If I was to do this again I would spend more time working on a better collision detection system as I feel like this could have drastically improved our project and give the tech demo a more complete feel.

I would also split my time up better, possibly using a timetable and a deadline system as I felt other projects and exams got in the way and some projects got in the way of others which made progressing in any of the projects difficult.

I would also investigate creating a 3D terrain for the ground as I believe implementing a 3D terrain would have greatly improved our project and would have set us apart from all the other groups, this was a huge opportunity to learn which was regrettably avoided to allow for work on other, smaller, aspects of the project.

# Conclusion

In conclusion, I believe the overall project was a success, as a group we managed to create a 3D tech demo after never working with 3D code or 3D animations before. Even though we did not fully succeed in turning our tech demo into a game by adding a timer for the player to race against or designing a better collision detection system, I believe we managed to create something better than we expected at the beginning as we implemented animations, skyboxes & 3D model textures which was all completely new to me and I was pleasantly surprised I was able to succeed in finishing a project which contained these elements.

# References

Angel. E (2011) "Interactive Computer Graphics – A top-down approach with shader based OpenGL." 6th Edition. Pearson.

Used when implementing BASS sound library: <http://www.un4seen.com/>

Jump: https://freesound.org/people/Lefty\_Studios/sounds/369515/

Dr. James Riordan's Real-Time 3D lecturers & labs 1-10.

# Video

<https://www.youtube.com/watch?v=NTczXQf4jQQ>