COMP2004 Computer Graphics Assignment 2 Report

Zachary Kidd-Smith/20205377

THEME:
The theme for this scene is a homage to an area called Firelink Shrine from the video game Dark Souls by From Software. I felt that it perfectly fit the theme of ‘recreation’ since not only is it from a video game (of which playing is a recreational activity), but it is also one of the very few areas in the game that give you a proper chance to rest and feel safe.

All the controls and features described in the assignment have been implemented, with a few notes to be made.

1 OBJECTS AND TEXTURES:

There are a total of ten objects in the scene, of which eight are composite objects made up of varying numbers of boxes.

Single Box:

1: Grassy plain: A single box stretched into a plain with a grassy ground texture applied. It has a dullish shine to it since wet grass can be a little reflective.

2: Skybox: A giant box that encompasses the entire playable area, everything is spawned and drawn inside of it. It has a pretty cool sky texture applied to it, however because it isn’t a proper cube map, it looks a little weird if stare too close. It has almost no shine to it since you don’t see reflections in the sky.

Composite:

1: Tree: A lone tree with a large trunk and a few branches poking out at different angles, all of its leaves have wilted off (maybe because creating leafy branches with just boxes was a bit too hard). It is made of a total of 4 boxes, 1 for the trunk and 3 for the branches, but the branches pierce through it at odd angles giving the impression of 6 unique branches. It has a mossy tree bark texture applied to it with very little shine since wood doesn’t reflect but wet moss does a bit.

2: Stone Bench: A stone bench propped up on a couple legs, it has 3 boxes, one for the top and one for each leg. It has a brick stone texture with a slight shine.

3: Crestfallen Knight: A shady guy who refuses to leave the shrine out of apathy, he is slowly going mad from the boredom and has devolved to just sitting around on the bench with his shaking head in his hands. He’s the most complex object in the scene, made up of 8 boxes, one for his torso, two for his legs, four for his arms (They are double jointed so he can properly hold his head), and one for his head. He has two textures, all of his body is covered in a chainmail texture except for a single side on his head which has an opening for his face. His chain armor has a shiny finish but his face has only a slight shine.

4: Archway: An archway made of mossy stone, in the actual map there are a bunch of these leading to different parts of the area. It's made of 4 boxes, two for the base legs and two more forming the arch. It has a mossy stone texture with a slightly shiny finish.

5: Bonfire: The Dark Souls equivalent of a save point, the one in Firelink Shrine is one that you always come back to. It has a pile of bone ash at the bottom with a rusted sword poking into it. It is mad up of only 3 boxes, 1 for the pile of bone ash, 1 for the sword blade, and 1 more for the sword handle. The bone ash has a texture made of bones and a very dull finish, the blade and handle have a rust texture and are a bit shinier.

6: Chosen Undead: The main character, they can wear a lot of different armor in the game, but here they are wearing some simple platemail that one of the classes can start with. They are made of 6 boxes, two for the legs, one for the body, two for the arms, and one for the head. They have an animation that triggers on a key press in which they sit down at the bonfire to rest, this animation involves the legs and arms rotating and moving into place while the whole body moves down. They have three textures, one for their plate armour with a shiny finish, one for their hair and one for their face both of which have a slightly shiny finish.

7: Grave Marker: A simple grave marker for someone unknown. It is made of two boxes, one for the base and another for the head. The base has a stone texture and the marker has a wooden texture, both with very little shine.

8: Wooden Chest: A wooden chest that has recently been emptied and left with the lid ajar. It is made of 6 boxes, one for each side of the chest and the lid. It has a wooden texture with very little shine.

ANIMATIONS:

Looping Animation:

The Crestfallen Knight has been animated with a subtle breathing animation by modifying the scale of his torso over time with sin:

```
model = glm::scale(model, glm::vec3(crestfallen_base_scale.x, crestfallen_base_scale.y + ((float)sin(time) * 0.02), crestfallen_base_scale.z));
// ^ Small breathing animation by adding a small amount to the body's y scale over time.
```

And a shaking head in his hands with a rotation applied over time with sin:

```
model = glm::rotate(model, (float)(sin(time) * 0.5), glm::vec3(0.0, 1.0, 0.0));
// ^ Head shaking animation, common behaviour for human's experiencing hollowing.
```

User Triggered Animation:

The Chosen Undead has been animated to sit down at the Bonfire on a press of the R key and stay there until it is pressed again and they are reset.

The animation for this is a lot more complex as it requires the torso and legs to moved independently, the other body parts follow the body though the arms have to be rotated and slightly moved to stay in position.

There is a modifier that increases over time to a limit once the key is pressed.

```
// Modify values for animation.
if (sitAnimation)
{
        sitModifier = sitModifier - (0.1 * deltaTime);
}
else
{
        sitModifier = 0.0f;
}
if (sitModifier < -0.4f)
{
        sitModifier = -0.4f;
}
```

The body's position is modified by this value.

```
glm::vec3 chud_body_position = glm::vec3(bonfire_mound_position.x, chud_legs_positions[0].y + (chud_legs_scale.y / 1.5) + sitModifier , chud_legs_positions[0].z
```

The values for the legs and arms are lerped into position:

```
        // Draw chud arms.
for (int i = 0; i < 2; i++)
{
        model = glm::mat4(1.0f);
        if (sitAnimation) // When R is pressed, the arms slowly relax over time.
        {
                // Normalize the sitModifier value.
                float normed_sit = (sitModifier - 0.0f) / (-0.4f - 0.0f);
                // Lerp the rotation of the arms
                float rotationOverTime = 0.0f * (1.0f - normed_sit) + 45.0f * normed_sit;
                float scaleOverTime = 1.0f * (1.0f - normed_sit) + 2.0f * normed_sit;

                // Apply animation transformations.
                model = glm::translate(model, glm::vec3(chud_arms_positions[i].x, chud_arms_positions[i].y, chud_arms_positions[i].z + sitModifier / 4));
                model = glm::rotate(model, glm::radians(rotationOverTime), glm::vec3(1.0f, 0.0f, 0.0f));
                model = glm::scale(model, glm::vec3(chud_arms_scale.x, chud_arms_scale.y / scaleOverTime, chud_arms_scale.z));
        }
        else // No animations necessary, arms at their side.
        {
                model = glm::translate(model, chud_arms_positions[i]);
                model = glm::scale(model, chud_arms_scale);
        }
        defaultShader.setMat4("model", model);
        glDrawArrays(GL_TRIANGLES, 0, 36);
}
```

```
        // Draw chud arms.
for (int i = 0; i < 2; i++)
{
        model = glm::mat4(1.0f);
        if (sitAnimation) // When R is pressed, the arms slowly relax over time.
        {
                // Normalize the sitModifier value.
                float normed_sit = (sitModifier - 0.0f) / (-0.4f - 0.0f);
                // Lerp the rotation of the arms
                float rotationOverTime = 0.0f * (1.0f - normed_sit) + 45.0f * normed_sit;
                float scaleOverTime = 1.0f * (1.0f - normed_sit) + 2.0f * normed_sit;

                // Apply animation transformations.
                model = glm::translate(model, glm::vec3(chud_arms_positions[i].x, chud_arms_positions[i].y, chud_arms_positions[i].z + sitModifier / 4));
                model = glm::rotate(model, glm::radians(rotationOverTime), glm::vec3(1.0f, 0.0f, 0.0f));
                model = glm::scale(model, glm::vec3(chud_arms_scale.x, chud_arms_scale.y / scaleOverTime, chud_arms_scale.z));
        }
        else // No animations necessary, arms at their side.
        {
                model = glm::translate(model, chud_arms_positions[i]);
                model = glm::scale(model, chud_arms_scale);
        }
        defaultShader.setMat4("model", model);
        glDrawArrays(GL_TRIANGLES, 0, 36);
}
```