COSC363 Assignment 1 Report

Student Id: 79172462

Name: Ambrose Ledbrook

The Scene

The scene shows a castle within a snowy environment with a flying saucer hovering within the castle walls. The castle has two cannons sitting by the castle doors, the cannons sit in a horizontal position until they are going to be fired and then they raise up to a firing angle. The castle doors are constantly opening and closing. There are two robots moving through the scene outside of the castle walls. One of the robots patrols the perimeter of the scene, the other four-legged robot walks in a square out in front of the castle.

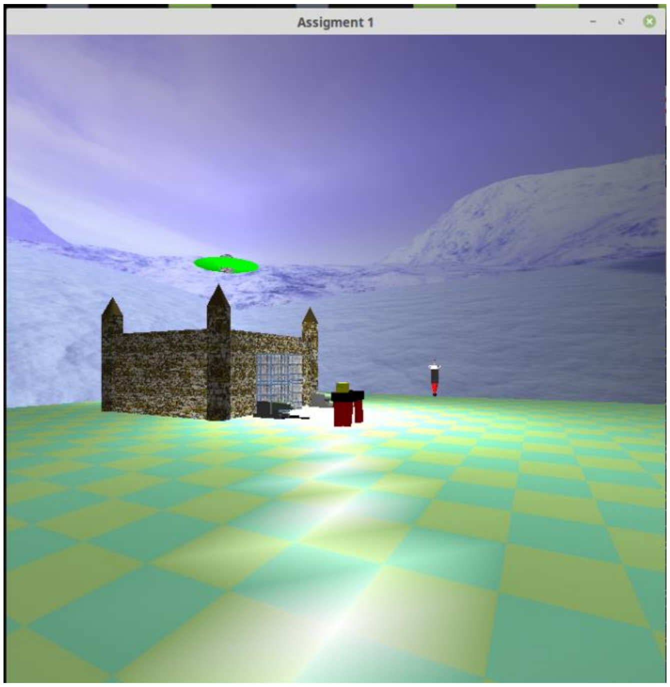


Figure 1 Default camera mode view of scene

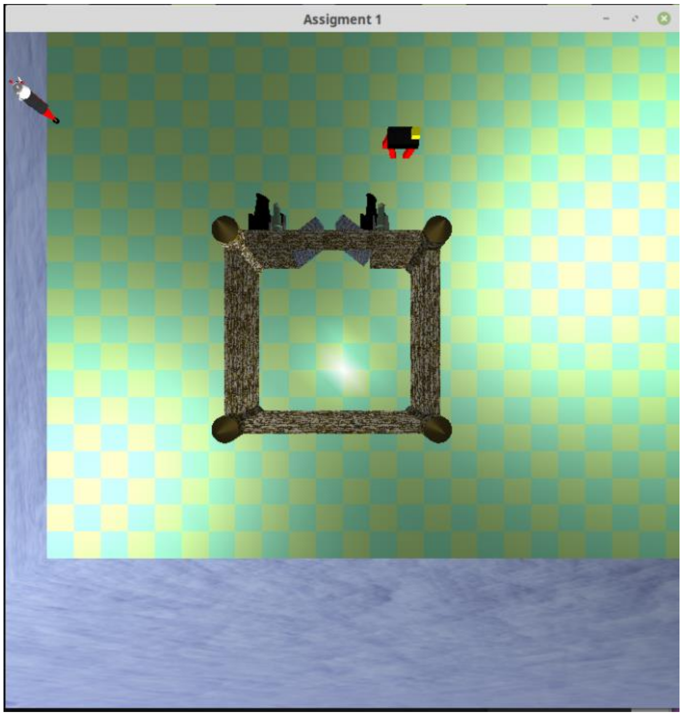
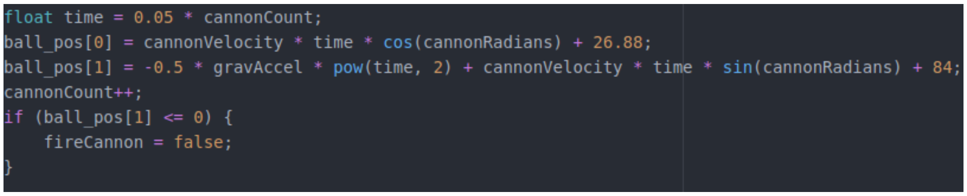


Figure 2 Ship view camera mode view of scene

Extra Features

- **Planar shadows:** There are planar shadows cast from the cannons in front of the castle and the cannon balls that are fired from the cannons.
- **Spotlight on a moving/rotating object:** There are two spotlights attached to the alien ship in the scene. The spotlights are attached to the rotating sphere within the ship, the spotlights can be seen rotating with the ship and then raising up as the ship fly's out of the scene.
- **Additional animated system within fortress:** The additional animated systems within the fortress are the fortress doors and the animation on the cannons. The doors continuously open and close, the cannons raise from horizontal to firing position before they are fired.
- **Two camera modes:** The scene has two camera modes, the default camera mode and the alien ships perspective camera mode. The default camera mode allows for the user to move throughout the scene freely. The ships camera mode is a fixed camera mode which gives the user a perspective from the ships position.
- **Physics model:** The physics model used with in the scene is a projectile trajectory model used to fire the cannon ball. The equations used to model the trajectory are: $x = v_c * t * \cos(\theta_c) + 26.88$ and $y = -0.5 * a_g * t^2 * v_c * t * \sin(\theta_c) + 84$ where $x = horizontal\ distance,\ y = vertical\ distance,\ v_C = cannon\ ball\ velocity,\ t = time,$ $\theta_c = the\ laucnh\ angle$



```
float time = 0.05 * cannonCount;
ball_pos[0] = cannonVelocity * time * cos(cannonRadians) + 26.88;
ball_pos[1] = -0.5 * gravAccel * pow(time, 2) + cannonVelocity * time * sin(cannonRadians) + 84;
cannonCount++;
if (ball_pos[1] <= 0) {
    fireCannon = false;
}
```

Figure 3 Code for cannon ball trajectory calculation

- **Collison detection:** There is collision detection within the scene to keep the user within the boundaries of the skybox.
- **Skybox:** The scene has a 5-panel skybox that covers the 4 sides and top of the scene. The floor is not textured due to problems that I ran into, described below.

Challenges

- The main challenge that I faced throughout the assignment was to do with textured surfaces and spotlights. Originally, I had a 6-pannel skybox, so all 6 sides of the scene sides were textured. This became an issue when I added in the spotlights on my alien ship, the spotlights were not showing up on the textured floor. After struggling through this issue for a few days, as advised by one of the tutors I tried removing the texturing from the floor of the scene, this fixed the issue and made the spotlights show. This is why my skybox now only has 5 textured sides and the floor is made from the floor function given in labs.
- Another issue I faced was being able to texture the towers on the corners of my castle. To start with I made the towers out of gluCylinders, but I quickly realized that these objects could not be textured so another solution needed to be found. After reading through some of the courses notes and looking at models online, I made a function that draws a cylinder using a combination of GL_POLYGONs and GL_QUAD_STRIPs to construct a cylinder that can be textured.
- Another issue that I was running into at the start of the assignment was the scene flashing constantly. I resolved this issue after reading through some of the course notes by changing the scene to use a double buffer.

Controls

- **Up Arrow:** The up arrow moves default camera forward.
- **Down Arrow:** the down arrow moves the default camera backwards.
- **Left Arrow:** The left arrow rotates the scene left by 5 degrees.
- **Right Arrow:** The right arrow rotates the scene right by 5 degrees.
- **‘c’ Key:** The ‘c’ key raises the cannon from horizontal to firing angle and then fires the cannon once firing angle is reached.
- **‘home’ Key**: The home key toggles the camera mode between default and ship view.
- **‘s’ Key:** The ‘s’ key launches the rocket

References

- **Skybox Textures:** http://www.custommapmakers.org/skyboxes.php
- **All other textures within scene:** https://www.textures.com/