COSC363 Assignment 1

Yu Duan

45239028

Build and Run

The program can be built and ran the same way as the labs on Geany.

Compile: g++ -Wall -c "%f"

Build: g++ -Wall -o "%e" "%f" -lm -lGL -lGLU -lglut

Models

Newton's Cradle

This model is the Newton's Cradle. Let the balls from left to right be ball 1, 2, 3, 4, 5. This model has a continuous animation of a real Newton's Cradle with an exception that the balls never stop in this model. When ball 1 hits ball 2, ball 2, 3, 4 will swing towards the right and hit ball 5, where ball 5 will start swing towards the right and swing back to hit ball 4, which then ball 4, 3, 2 will swing towards the left and hit ball 1, which will then swing to the left and swing back to hit ball 2, and the animation will continue.

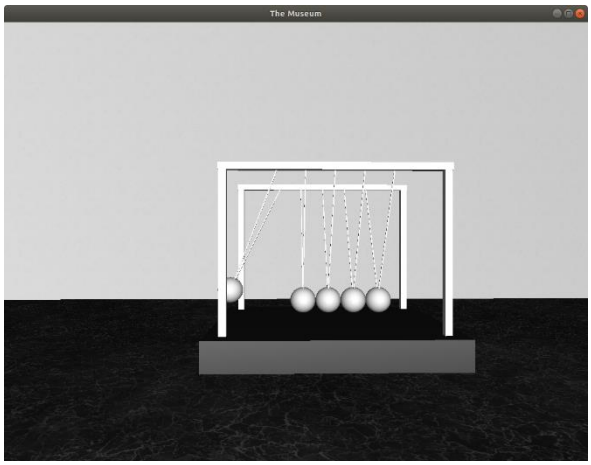

Figure 1 – Newton's Cradle

Earth and Moon

This model contains Earth and Moon, where the Earth orbits around its own axis and the Moon will orbit its own axis and around Earth. The Moon's orbit and rotation around the Earth is the same, thus the same side will always face the Earth. The Moon will also do a full rotation around the Earth when the Earth has

made 28 rotations about its axis. Extra features I have added in this model are planar shadows, and a spotlight on a moving/rotating object. The planar shadow on the moon also moves with the spotlight.

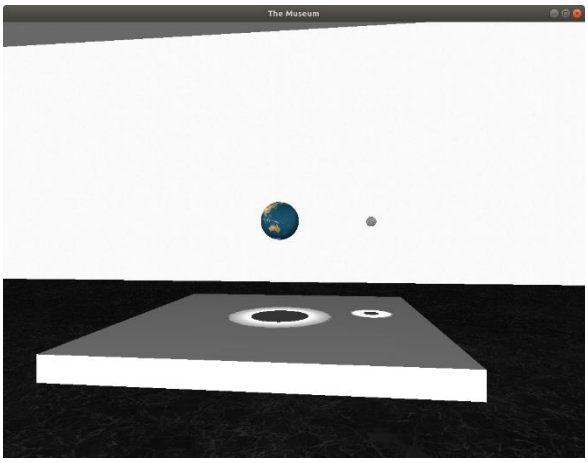

Figure 2 – One spotlight is placed above the Earth and one above the Moon, shadows are beneath them.

Cannon

This model uses the Cannon.off file and the code to load it from Lab 2. However, significant changes have been made from Lab 2. I created a cart to hold the cannon, which was quite a complex design process. The two sides of the carts were made using GL_QUAD_STRIP from the design shown in the Figure 3. Two of the shape in the figure are required to construct one side, and the two shapes was connected by another shape created using GL_QUAD_STRIP. The wheels and various rods were made using gluCyclidner and gluDisk.

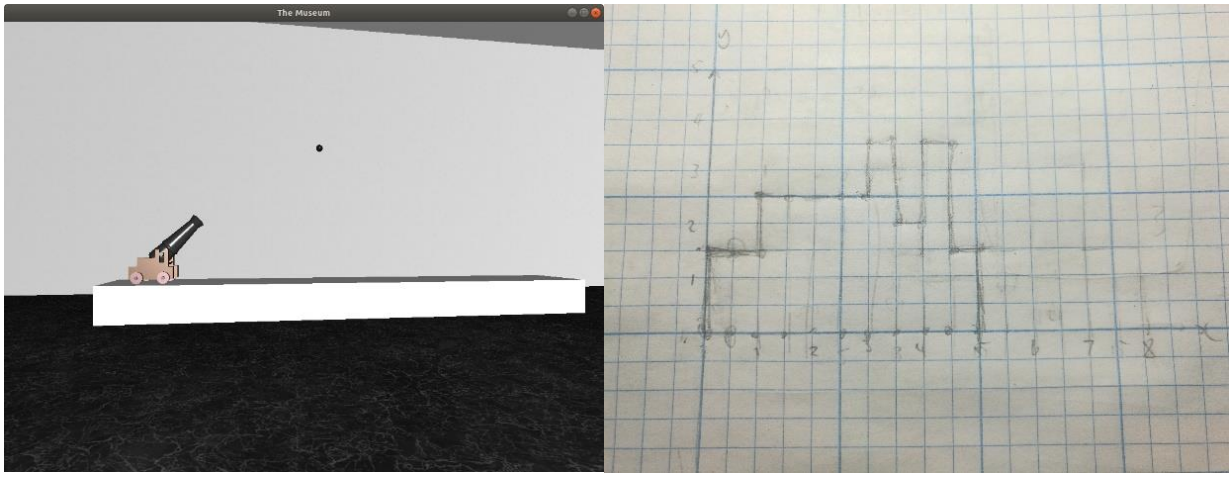

Figure 3 – Left shows the cannon firing the cannon ball, right shows the design for the side of the cannon's cart.

The animation of the cannon is shown in Figure 3, which is improved from Lab 2 where the ball did not fire in a projectile motion based on gravity. The controls for this animation are in the controls section of the report. The cannon can be angled up to a maximum of 60 degrees and angled down to a minimum of 5 degrees. The cannon angle cannot be changed while the ball is firing, and the cannon may fire again

once the ball has landed. Changing the angle of the cannon will move the ball with it and change the trajectory of the cannon ball based on the displacement projectile motion equations when fired.

$$x = v_0 t \cos(\theta)$$

$$y = v_0 t \sin(\theta) - \frac{1}{2} g t^2$$

Where v_0 is the initial velocity, t is the time that has passed since firing, θ is the cannon's angle of fire, and g is the gravity. The ball in the animation will move at a slow speed, this is to ensure that the ball will not fire out of the platform's range at the maximum firing angle of 45 degrees. The equation was modified so that it also adds the initial x and initial y coordinates of the ball so that x and y will represent the current coordinate of the ball. The initial location of the ball was calculated using the parametric equations of a circle, so whenever the cannon's angle is changed, the location of the ball will change correctly.

Hourglass

This model is an hourglass made using sweep surfaces, the points used to create the sweep surface is shown in Figure 4. The glass part of the hourglass was made transparent so that it looks like glass. The support poles and the top and bottom of the hourglass are made using gluCyclinder and gluDisk.

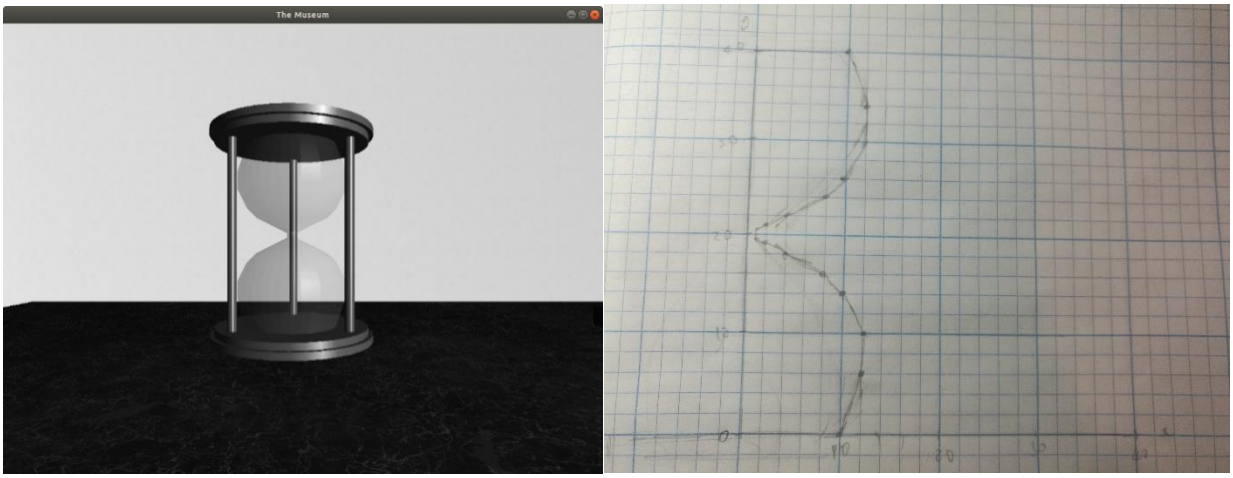

Figure 4 – Left shows the hourglass model, right shows the design of the hourglass for the points to plot for the sweep surface.

Extra Features

Extra features that were implemented and explained above were planar shadows on the Earth and the Moon, a rotating spotlight on the moon, physics-based animation on the cannon firing and a complex design process of creating the cannon's cart.

Apart from those extra features, a skybox was implemented. The skybox is a cube of size 1000 in x, y, and z direction. The museum and player are placed inside the cube. The player has a far plane view of 5000, this was decided based on what was suggested in a lecture, so that it reduces the distortion on the skybox. The floor of the skybox cannot be seen by the player, instead a floor is generated at y = -1.

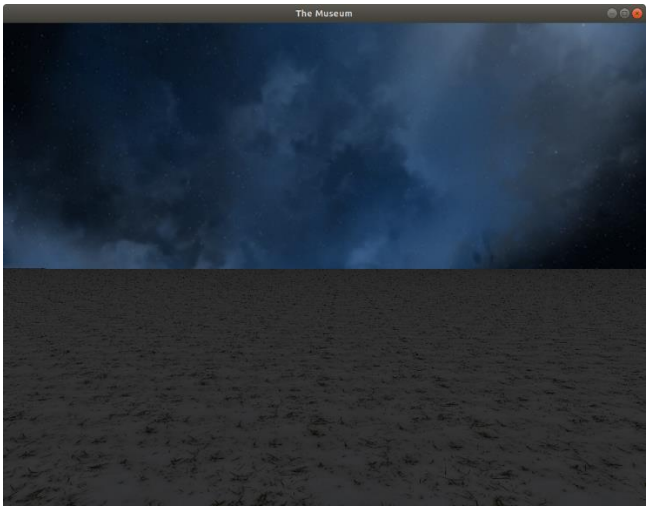

Figure 5 – The skybox is a night sky.

Controls

Up Arrow Key: Move camera forward

Left Arrow Key: Rotate camera towards the left

Down Arrow Key: Move camera backwards

Right Arrow Key: Rotate camera towards the right

'Z' Key: Angle the cannon upwards slightly

'X' Key: Angle the cannon downwards slightly

Space Bar: Fires the cannon ball

References

Earth and Moon textures - http://planetpixelemporium.com/earth.html

The outside wall of the museum - https://www.textures.com/download/bricksmallnew0114/72779

The inside wall of the museum - https://www.textures.com/download/pbr0088/133122

The floor of the museum - https://www.texturecan.com/details/106/

The texture for the skybox - https://www.cleanpng.com/png-space-skybox-texture-mapping-cube-mapping-night-sk-776480/

The texture for the floor outside the museum - https://3djungle.net/textures/snow/5126/

Projectile motion equation - https://en.wikipedia.org/wiki/Projectile_motion

Parametric equation of a Circle - https://www.mathopenref.com/coordparamcircle.html