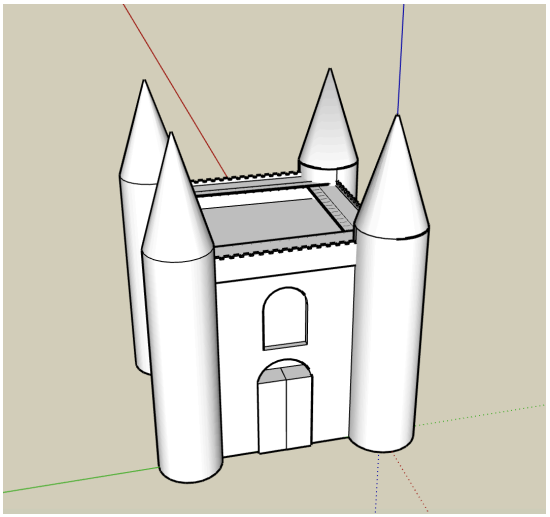
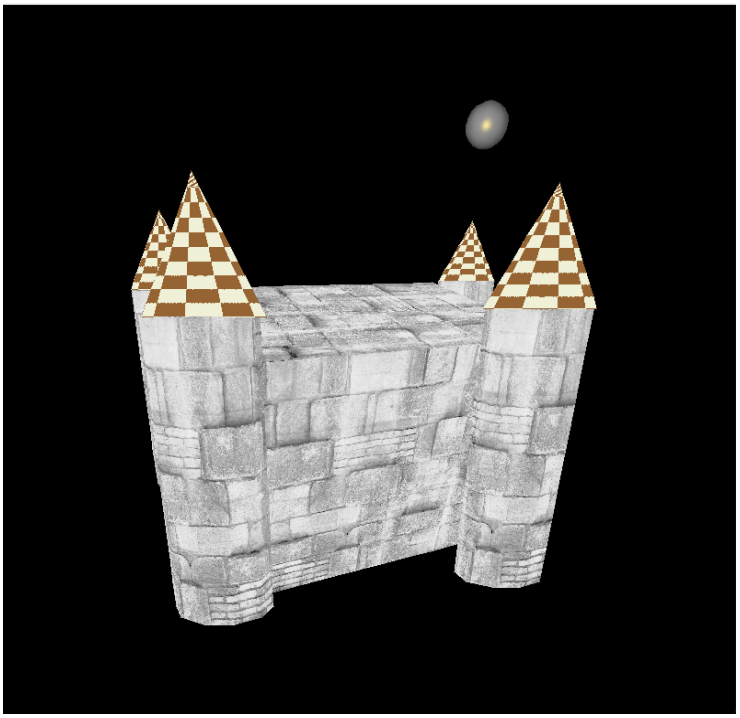
Approach Document

Pravina Bhalerao
CG Finals
pb5580

The artwork that I have modelled is a castle which is made of many geometrical objects: cylinder , cone ,cube which makes it ideal for the finals . I had submitted the same image during Activity 14 .



The Final Result :



The View Window is set up to (700, 0, 700, 0)

Camera Setting :

```
eye = glm.vec3(0.0, 2.0, -4.0)
lookat = glm.vec3(0.0, 0.0, -10.0)
up = glm.vec3(0.0, 1.0, 0.0)
```

I am using Perspective/Frustum Projection Transform:

```
projectionT = myEngine.frustum3D(-6, 6, -6, 6, 8, 15)
```

I am using hierarchical modelling (one of the style points)

Traversing down the hierarchy and push transform on the stack and pop the transform off the stack/. The code defines a class TransformStack, which is intended to manage transformation matrices. It has methods to push and pop matrices onto a stack, which enables hierarchical transformations.

The castle is made of placing Cube in the centre, cylinder placed around the cube and cones at the top of those cylinders which completes the criteria of using 3 or more objects . Then I applied texture using image tiledstones.jpg to the cylinder and cube which gives the brick look . For the cones I have applied a procedural texture CheckeredBox covering one of the style points. The Sphere at the top of the castle is the Moon and I have applied the sphere Phong shading .

The draw_cube, draw_cone, and draw_cylinder functions handle the rendering of different geometric shapes using the provided engine functions like myEngine.drawTrianglesTextures(for cube ,cylinder for brick texture) and myEngine.drawTrianglesCheckerboard(for Cone). Each of these functions takes parameters for transformation matrices, view transforms, projection transforms, and color parameters , uv values.

In the default_action function, a scene is defined by positioning and rendering various geometric shapes like cubes, cylinders, cones, and a sphere (moon). Each shape is transformed using translation, rotation, and scaling matrices before being rendered with different rendering functions provided by the engine.