Computer Graphics

Programming Labs 4, 5 - Report

By: Rahul Bethi

Submission: March 21, 2016

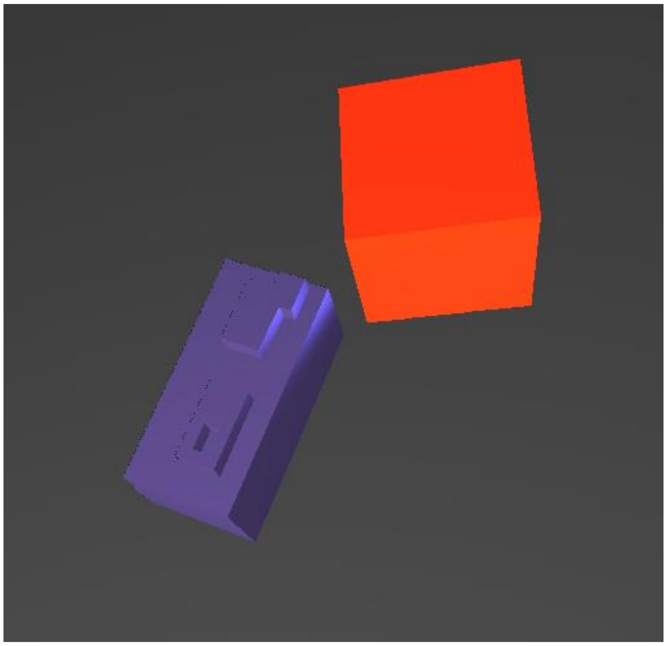
Objective: To learn how to draw with different buffers and methods to save the storage space. And also to apply transformations to them.

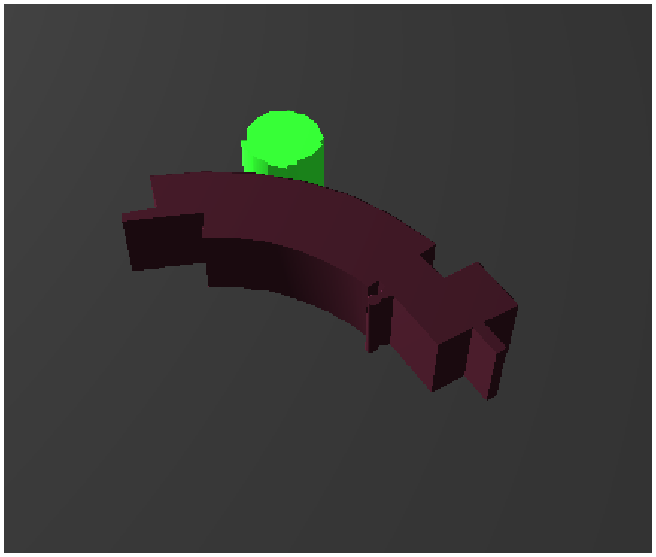
Duration: 3 weeks

Part – I, II

My objective was to create two campus buildings ECDC and BH.

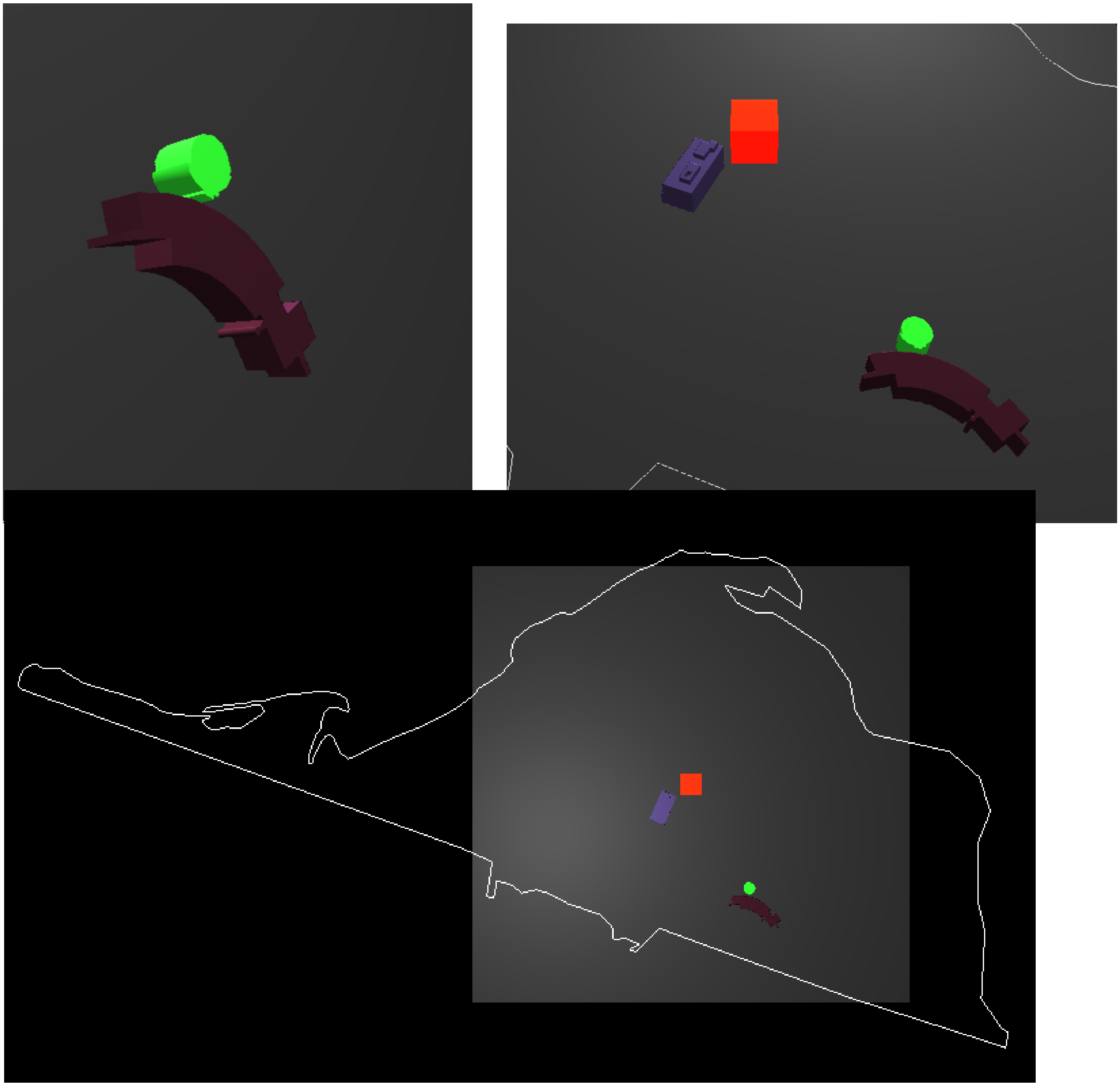
- First, I have adjusted the co-ordinates of the island given in the instructions to the center of the world.
- Then I drew the outline of the island.
- Third was the ground plane with two triangles.
- Fourth was the cube which is used as a test bed to test all the next improvements to the project.
- Now, I have hand written the co-ordinates of both the buildings.
- I have created an index for the vertices to draw triangles.
    - ✓ Each object is created using class, so everything has its own VAO, ABO, Ibo, shader Objects, variables, functions, etc.
- I adjusted the camera by giving keyboard inputs for the movement and preset modes.
    - ✓ Keys W S A D to move the camera in X, Y plane in world space.
    - ✓ Keys L.Shift, L.Ctrl to move the camera along Z axis in world space.
    - ✓ Arrow Keys to move the point of Interest of the camera in X. Y plane in world space.
    - ✓ Keys R.Shift, R.Ctrl to move the point of Interest along Z axis in world space.
    - ✓ Spacebar to cycles the camera position and point of interest to better view the buildings. (4 presets).
- ECDC building had two parts, and is also rounded, which I considered as an extra rounded object.
- A different vertex shader is used for drawing the island as it only needs to output the outline, not triangles.





ECDC

BayHall and Cube



Total Map

Part – III (Light)

Normals

- First step is to calculate normals. To do this, we need to have normal for each vertex per surface. We get different more than one normal per vertex at corners or edges which are not smooth.

  - To address this problem, I created different per surface, so that each vertex will have only one normal, and it is easy to calculate normal for each vertex in this case.
- I made another index file and repeated the vertices according to this new index. Index for IBO is adjusted accordingly with the new vertices order.
- Normals are calculated according to the triangle it belongs to and the order in which vertices are given for that triangle.
- If a vertex has more than one normal (curved surfaces), then the normals are added up and normalized to get a single normal per vertex.

Materials

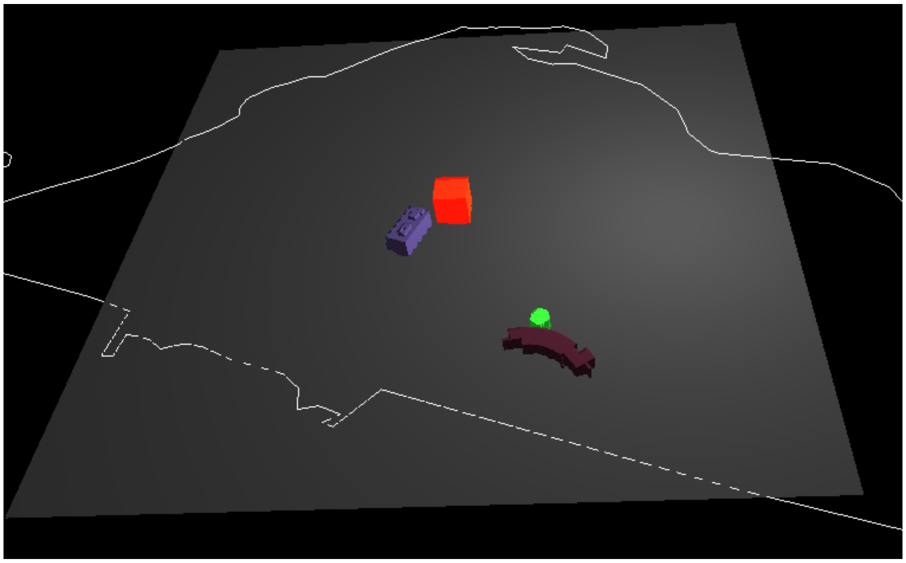
- I used three different materials. Copper, Silver, Gold.
  - Ground plane – silver
  - Cube – copper
  - ECDC – copper, silver (more than one material)
  - BayHall – gold

Illumination

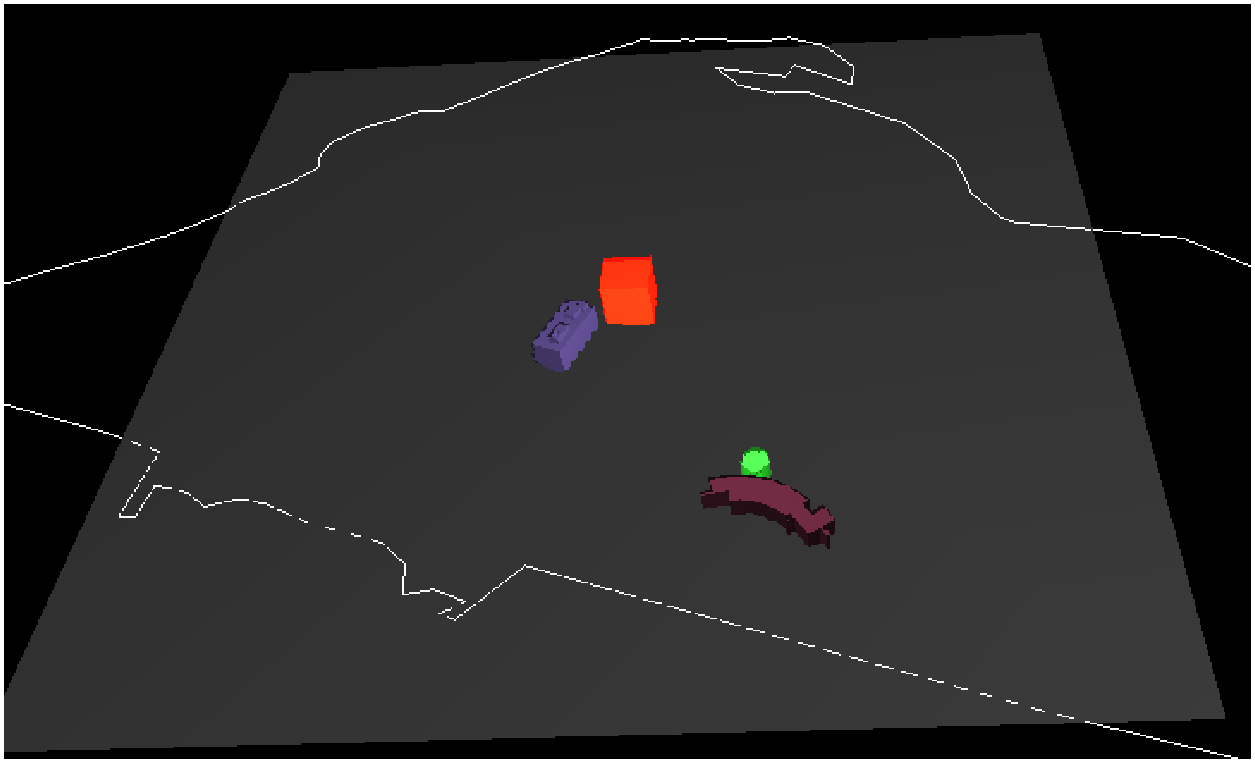
- I created two illumination models Phong and Gouraud with two vertex shaders and fragment shaders each with two shader programs.
  - Press TAB key to change between these two models, as the shader programs change with key release.

Lights

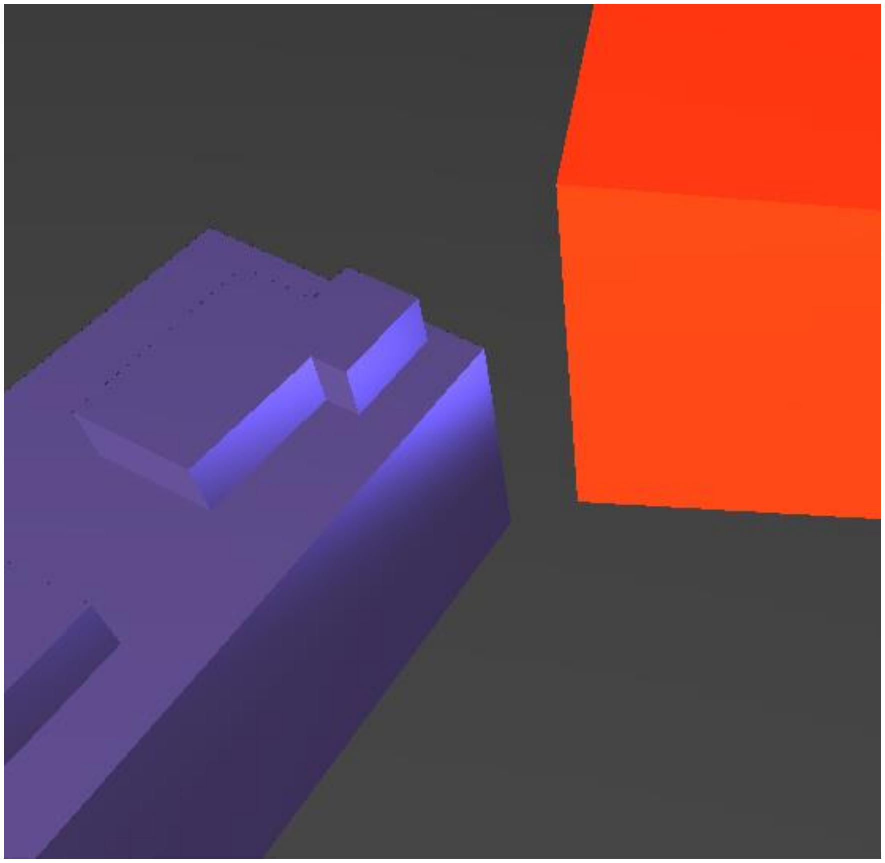
- I used two lights, one is point light, which rotates in a circle path in XY plane of world space.
- The second is directional light like the sun, which is stationary.



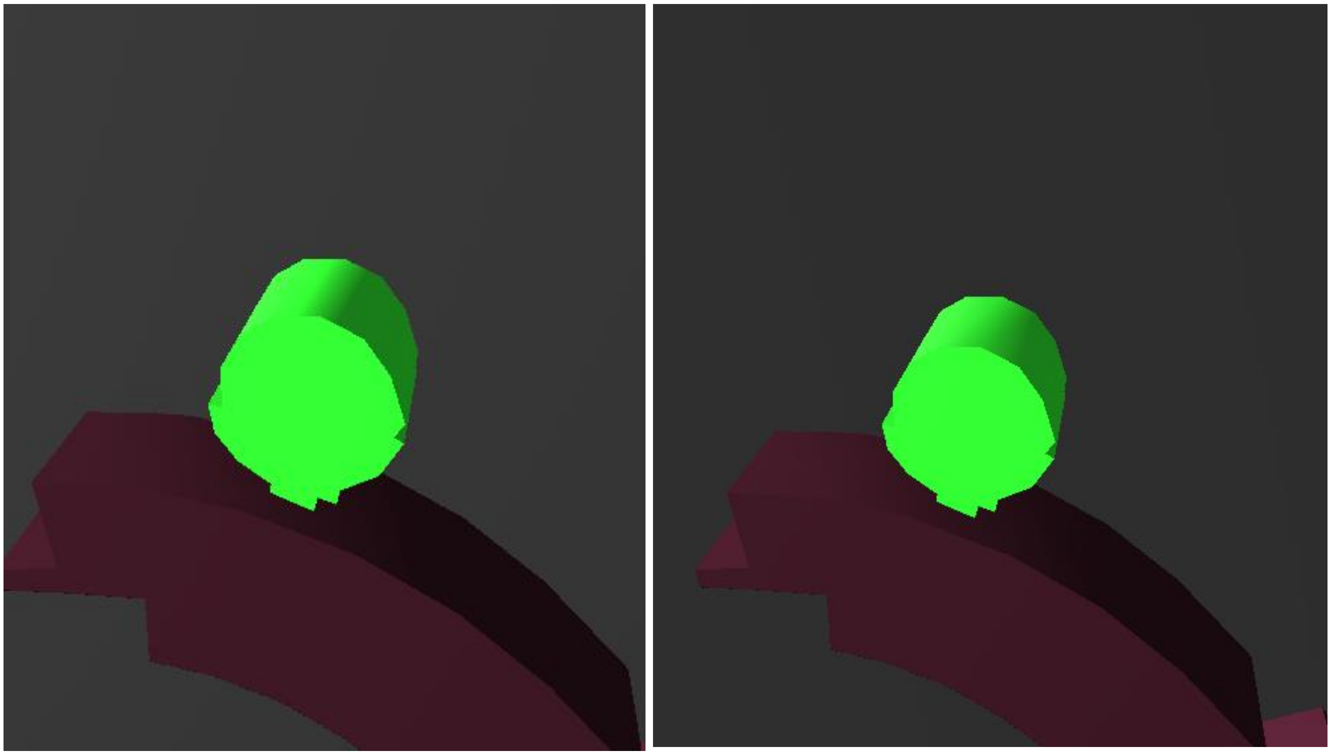
Phong model, intensity of the light can be seen on the ground plane.



Gouraud model – intensity of light can be seen on ground plane. Depends on the vertex of the ground plane.



Specular highlights can also be seen. (phong model)



Phong model is smooth on curves, but we can see the edges in the Gouraud model.

Problems Faced:

1. I got stuck in implementing objects in classes. With my poor experience in classes, it was tuff to get it working, as I was getting black screen with no errors.
    a. It finally worked after I re-wrote the code.
2. I tried to everything dynamic to use this for future, and in the process got stuck with many errors, which I had to re-write again fresh to make it neat without errors.
3. Didn't get the diffuse and specular working until I met Dr. King, and found out that I was replacing the normals with empty data. This took me a week, as I was about to give up.
4. I also made an error with MVP – model x view x projection. Which actually has to be reverse.
5. Finally, many optimizations.