CITS3003 – GRAPHICS AND ANIMATION 2018

PROJECT REPORT

GROUP MEMBER:

HOANG TUAN ANH (ID: 21749914)

JOSHUA NG (ID: 20163079)

PROJECT PART 1 REPORT:

List of tasks that we have done:

- Task A: this task only requires us to add rotation around the X and Y-axis to the view matrix (following lecture 14: Computer Viewing).
- Task B: this task requires us to add the rotation matrix to the object's model matrix so that it can be rotated around X, Y and Z-axis (following lecture 9: Transformations).
- Task C: for this task, we just followed the structure of the already-implemented functions in the provided code framework and modify it so that it can modify the object's ambient, diffuse, specular and shine.
- Task D + E: we followed the lab 5 solution to do task E. And for task D, we only needed to change the nearDist variable to a very small value which increases the viewing volume.
- Task G: this task basically requires us to implement Blinn-Phong shading model from the lecture 17: Shading in OpenGL to replace the current Gouraud shading of the skeleton code.
- Task F: after implementing task G (as suggested by the description), from lecture 16: Shading II, we added a light distance variable to make our light's strength reduces with distance.
- Task H: for this task, we realised that to make the textures of object and the ground tend to white when we move the light close to them, we had to separate the specular calculation from the colour calculation. The specular is added at the final step when the fragment shader calculates gl_FragColor.

List of tasks that we are uncertain/not working:

- Task I: for this task, firstly, we modelled our second light after the first one. Then we tried setting its w component to 0.0 but it appeared not the same as in the video. We changed the position of our second light in the vertex shader so that it points to the origin. However, we are still unsure if this makes our light directional or not. That is why we list this task under here.

List of extra functionalities that we implemented (task J):

- Delete/Restore object: our scene editor can delete objects from the latest one to the first one (the ground) by pressing G. To restore the object, press ALT+G. It will restore the objects in the order that they were added in.
- Select Next/Previous object: our scene editor can select another object by pressing J which will select the next object or ALT+J to select the previous object.
- Duplication: our scene editor can duplicate an object by pressing U. The addObject function call was slightly changed so that it can takes a texture ID integer in. The change was necessary to make this function work. The overall program still works normally.
- Change colour over time: our scene editor can change the entire scene colour into random colour over time.

Outline of contribution:

- Josh did task A, B, D, E and Duplication and Change colour over time functions.
- Anh did task C, G, F and Delete/Restore objects and Select Next/Previous object functions.
- Task H and I were done together.

Reflection:

The code provided to us has a bug which causes the camera reverts to its previous position rather staying at where it currently is. A friend from the lab told us how to fix this and it is in the gnatidread.h. Also, to help those who want to do the project at home, we feel that there should be some instructions on how to install OpenGL correctly on our own machine.

PROJECT REPORT PART 2:

List of tasks that we have done:

- Task A: we added the texScale variable of the object to the gl_FragColor.
- Task B: we were able to export human models that can perform animation from MakeHumans.
- Task C: This task is straightforward. However, we needed some help from lab tutors to get the right export settings so that our human model can perform the animations correctly in our program.
- Task D: for this task, we had to first change the size of our human models in our scene editor so that they are large enough to see. We did have some troubles with how to make the animations start. Thanks to the lab tutors, we were able to make our human models animate. We added a new attribute to sceneObject structure, currPose, to know which frame of animation an object is at. Then, we implemented a timer to increment the currPose so that our models will perform their next frame of animation. Our human models are under the “Add object” menu, “51-60”. The models are “Human Model 2, “Human Model 3” and “Human Model 4”.

List of extra functionalities:

- Change the animation speed: we added a menu entry “Change animation speed”. Click the left mouse button and move the mouse to the left to increase animation speed and move to the right to decrease the animation speed.
- Change the animation order: we added a menu entry “Change animation forward/backward”. When click on the entry, the models will start doing their animations backward and click again they will do their animations forward normally.

Outline of contribution:

- Josh did task A, B and C and made model 58 and 59.
- Anh made model 57 and added the Change animation speed and Change animation order (Josh’s idea) functions.
- Task D was done together.

Reflection:

The model 56 from the unit web page appears to be bugged with the lab machines only. Many other students also report the same incident when they try to use model 56 in their codes on the lab machines. We were able to make it animate on our own machine. Perhaps the reason lies with the OpenGL version difference between the lab machines and the machine that model 56 was made on. When adding the code from the instructions to our program, our ground object was flipped 180 degrees. We had lab tutors' help to fix this issue. Generally, we think that the part 2 of this project was not too hard to figure out. However, most of the time was spent on trying to find out what was causing our program not to work, i.e. is our model not being exported correctly? Did we follow the instruction correctly? Etc.