COMP2004 Computer Graphics Assignment

Recreation Scene using OpenGL

1. REQUIREMENTS

In this assignment, you can unleash your imaginations and use OpenGL with C/C++ programming to produce a scene(s) that represents what you envision to be associated with the term *Recreation*. The definition of *Recreation* based on the Oxford Dictionary is: "*Any activity that refreshes, satisfies, and brings enjoyment to people, in which they engage on a voluntary basis during leisure time*". It is not necessarily the activity that you do personally. These are some examples:

- Any kind of physical sport/activity (e.g Football, Hiking, Surfing, Skydiving, etc).
- Digital Entertainment (e.g Video games, watching movies/youtube, etc)
- Music (e.g Playing piano, singing, etc)
- Outdoor activities (e.g Picnic at park, beach, riverside, etc)

Note: If you choose video games as your theme, you can create a scenery inside the video game instead. For example:

- If you play Skyrim, you can create a scenery where the main character is fighting a dragon.
- If you play Among Us, you can create a scenery where the crew is being chased by the imposter.
- If you play Dark Souls, you can create a scenery where the main character is rolling on the ground again and again to avoid enemy attack.
- If you play Minecraft, you can create a scenery where the main character digging the soil.
- And more… (unleash your imagination)

These are few key points regarding this assignment:

- This assignment is focused on the technical skills on OpenGL programming. Artistic creativity is still encouraged, but not required. Please spend reasonable time learning the OpenGL concepts itself.
- Interactivity exists in OpenGL assignment. For example, you can explore the world with <W>, <S>, <A>, <D> keys and mouse movement.
- It is possible to have multiple light sources in OpenGL. For simplicity, you only need to use one light source. However, the effect of ambient, diffuse, and specular has to be clearly visible.
- In order to make it more fun and test your creativity, the only shape you can use for this assignment is **"box"** shape. Yes, you read that right, ONLY "box" shape. NO sphere, cylinder, cone, pyramid, torus, and any other polyhedron. You can apply any geometrical transformation to it (enlargement, shrinking, stretching, rotation, even shearing) and combine with other boxes to create anything. You can think it as if you are playing toy block (but only box). If you are familiar with video games, it is similar to Minecraft (https://minecraft.net/en-us/). You create the scene only with boxes.

Program Requirements:

1. At least 8 unique composite objects and 8 different image textures. Since you can only use boxes, be creative on combining these boxes to create complex objects (*e.g.* a table). Using a skybox does not count (but, it can help to create beautiful scenery)
2. Only one light source is required. Please write the program so that the light source always follow the camera position by default (Imagine the camera as a firefly). However, if the **<F>** key is pressed, the light source should stay at its current location (stationary). Pressing the **<F>** key again should make the light source follow the camera position again.
3. Implementing a simple light source attenuation (See lecture 7). As the distance to the light source is increasing, the brightness should be reduced gradually. Pressing the key **<K>** will reduce the brightness radius of your light source, while key **<L>** will increase the brightness radius of your light source. You can derive your own formula for the light source attenuation, but describe it in your report.
4. The effect of the light source should be reasonable. For example, metal surface and waxed apple should be shiny, while wooden plate should be dull.
5. We should have the full control of the camera movement (Can fly around the scenery). Feel free to use the sample code from learnopengl.com for this implementation. Holding **<SHIFT>** key while moving should double the movement speed. The movement range of the camera should have a limit outside the virtual scenery you create (Imagine a 3D sandbox where the camera cannot travel outside it)
6. Pressing the key **<P>** shall toggle the switch between perspective and orthographic projection.
7. At least 2 animations should present in your created scenery. One should start playing the moment the program start rendering (For example, a person is chasing a butterfly in a circle nonstop). The other animation should only be triggered when the key **<R>** is pressed (For example, the person falls down and cries). Pressing the same key again will bring the scenery back to the original state (So, you can toggle the second animation as many as you want). Keep in mind that this animation should combine multiple geometrical transformations together to create a reasonably complex animation.
8. Please implement a functionality to toggle between the "darkish scenery" and "bright scenery" with the key **<O>**. When the "darkish scenery" is activated, the brightness of the objects/scenery should depends on the location and intensity of the light source (imagine night time with only one light source). On the other hand, "bright scenery" just simply disregards the existence of the light source and render everything with the color intensity as it is.
9. Please do NOT create any kind of jumpscare. There will be a heavy penalty for this.

T**he final version of your program must work on the lab environment OR VMware Horizon**. Please do not install any additional libraries that is not included in the setup guide.

You are allowed to use any of the sample code from www.learnopengl.com as the foundation of your assignment, and then expand the code. Please ensure that you have practiced OpenGL programming sufficiently before you attempt the assignment.

2. ASSESSMENT SUBMISSION

This assignment is due on the **15th October 2021, Friday at 11:59 PM**. You must zip the following items into a single zip file:

- The whole folder containing your source code directories, texture images, and relevant libraries;
- A simple readme.txt which briefly describes the way to compile and run your code(s) (should be similar to compiling your practical codes);
- The signed Declaration of Originality. By submitting the sheet, you declare that the work submitted are solely your own. Your assignment will not be marked if you forget this form;
- A short report of up to 5 pages. In the report, please describe the theme of your implementation. Please provide brief description on YOUR OWN work, not just a list of what are required. Please describe what simple or composite objects have been used to model the objects, and what kind of surface finishing you have employed for each of them. You are also required to describe the animation you have produced by providing the design ideas and their implementations. The report will contribute to 15% of Assignment marks.

This zip file should be uploaded to the Blackboard before the deadline. Just click on "OpenGL Assignment" in Blackboard and you'll see the uploading webpage. **You are allowed only 3 submissions, so please make sure the completeness and correctness of your files before your submission**. **All late submissions will not be accepted.**

NOTE: If your assignment cannot be compiled OR crashed immediately upon starting it, you will receive ZERO mark on the implementation part (which means failing this assignment). Remember, this unit is NOT an introduction unit to the programming anymore. If there is no Computer Graphic rendering you can show, then there is nothing to mark.