ACADEMIC YEAR: 2022-2023

Quiz 1

Quiz Date: July 13, 2022

Quiz Time: 24 hours from 3.00pm

Marks: 20

RULES:

When done, submit via Moodle:

- Submit all project, dependency, and source files (cpp, h, vcproj, sln, *etc.*)
- Do not delay the submission until the last minute! Even partial submission must still be uploaded.
- The submissions are allowed to be re-uploaded multiple times. The latest uploaded will be used for marking within the timeframe.

During the quiz you are allowed to have any online or other media resources about OpenGL and code samples you can lay your hands on, *etc.* BUT:

- You are not allowed to share any quiz related material anytime. At best, the penalties for giving and taking each other's quiz code will result in 0 grade by default. Later penalties may follow.
- You are not allowed to communicate with each other at all orally, electronically, or otherwise during quiz time. You are not allowed to seek any help from anywhere.
- Should you use some code from external resources, URLs to those resources must be present within the code that you submit as comments in the header.
- Your own code from A1 is allowed to be used as it is assumed to be well documented and referenced (if it is not so, this is another chance to fix that).

Any form of Plagiarism will be strictly penalized.

TASKS:

Comments in the code are of paramount importance.

• Briefly (but identifiably) cite all the resources used (online as URLs or offline as book or article titles, notes, own project *etc.*)

• Certain tasks need to be done in order, but some are not. Read through them all; do the sequences you absolutely need and the ones you are most comfortable with first, and then do as many as you can:

You will be building up on A1 and below mentioned requirements are already available to you through A1 code.

- Creates a 100x100 square grid (ground surface) in the XZ plane centered at the origin.
- Creates a set of three lines 5 grid units in length, in 3 different colors, representing each coordinate axis in virtual world space, centered at the origin.
- Places a virtual camera with the world space origin as the point of focus.
- For display and animation:
  - Create a GLFW window of size 1024x768 with double buffering support.
  - Render the coordinate axis, ground and all the models in the window.
  - The application should use a perspective view to display all the objects and enable hidden surface removal.

Below mentioned requirements are in addition to the baseline code mentioned above.

- Create a large enough cube to occupy the entire grid and set the color to sky blue. Your models as well as camera will be placed inside this cube which is your world. **(2 Pts)**
- For this quiz, you are required to design six alphabets from your last name. If your last name is shorter than 6 characters, borrow from your first name and your ID if the first name is also not long enough.
  1. Design one skateboard each for every character. Every character must be placed on its respective skateboard. You may create a basic model for the skateboard with four cubic wheels and a flattened cubic plank. **(5Pts)**
  2. Model the six characters and place them on the corresponding skateboards. **(5 Pts)**
  3. You may make use of cube utility functions/methods/classes from your assignment and design all the characters in the old Timex watch display style. Please refer to https://fontmeme.com/fonts/gau-font-cube-font/ for shape related help.
- The application should handle following input for each character.
  1. Each character should be selected by pressing a key from 1 to 6. **(1 Pt)**
  2. The user can rotate each character in clockwise or anti-clockwise direction around Y-axis which is pointing upwards. Assign apt keys for rotational purposes; however, each character needs to be independently rotatable and of distinct color. **(2 Pts)**
  3. The user can control camera position while looking at each character using keys 1 to 6. You must adjust the camera position as well as look-at-point to the center of the character you are looking at. **(2 Pts)**
- The application must use OpenGL 4.1 and onwards and must include brief comments explaining each step. Grader will ask for a small modification to understand your grasp of your code. **(3 Pts)**

Submission:

Quiz must be submitted through Moodle. No other form of submission will be considered. Please create a zip file containing your C/C++ code, vertex shader, fragment shader, a readme file (.txt). The zip file should be named Assignment#_YourStudetID. In the readme file document, the features and functionality of the application, and anything else you want the grader to know *i.e.* control keys, keyboard/mouse shortcuts, *etc.*

Bonus Features:

You can achieve an extra 5 bonus points if you create a detailed model for the skateboard and if you model the characters in a Rubik's cube style. You may refer to the link below for character shapes. This bonus will be given at the Grader's discretion.)
https://2.bp.blogspot.com/_zqFoq3qej2c/SWlTkZexp0I/AAAAAAAAjDY/9yNhtwH0EhU/s1600/rubiks-alphabet.jpg

Evaluation Procedure

You MUST demonstrate your program to the grader during a pre-scheduled session. You must run your code submitted before the deadline, demonstrate its full functionality, and answer questions about the OpenGL programming aspects of your solution. Major marking is done on the spot during the demo. Your code will be further checked for structure, non-plagiarism, *etc.* However, ONLY demonstrated submissions will receive marks. Other submissions will receive a zero score.