CS535: Assignment #0 – Cook it!

Out: August 23, 2024
Back/Due: August 30, 2024

Objective:
This objective of this assignment is a simple warm-up program to help setup your programming and graphics environment. This assignment will require you to setup a shell programming environment for this and future assignments, using OpenGL with FreeGLUT on Windows or Linux. It is to your benefit to write the program modularly and with a clean setup so as to facilitate subsequent assignments. You have one week but it should take you much less time. The FreeGLUT framework is bare-bones, but easy to setup. If you want to use a different framework (e.g., Qt, GLFW, etc.), that is up to you (note that you still must use OpenGL within your framework).

Summary:
The assignment is to implement a program which draws a simple screen saver like program. The "screen-saver" consists of bouncing a loaded 3D object inside an imaginary box. You may use whatever OpenGL commands you wish but probably you just need to add some simple matrix math operations. The object should start at a random location and move at a "reasonable speed". You do not need to make the collision nor the response of the object with the walls of the imaginary box be perfectly on the boundary nor physically correct, just plausible.

Specifics:

(0) Start with the template from the course website. You may develop under Windows or Linux. For Windows, the template contains a Visual Studio 2019 solution (which is a guess to the oldest VS version you might have, so compatibility should not be an issue). For Linux, there is a Makefile. See the README for details on building and running the code. The included FreeGLUT.dll and .lib files are pre-built. If you use a different compiler, you may need to download or build FreeGLUT for your environment.

(1) Compile and run the template. Use the mouse to manipulate the view, and right-click to switch to different loaded objects. Read the code to understand how it works. Don't hesitate to ask questions! Then, make the changes below.

(2) When the program starts, a random initial location and velocity for the loaded object should be assigned. You may use the GUI to add options, including the name of the file to load. The templates come with a simple .OBJ loader. You may also hardcode to load one object file.

(3) One way to implement the bouncing is to modify the when the object centroid passes a wall, snap it to the wall and reflect the velocity vector about the wall (e.g., if a "vertical wall in the YZ plane, then swap the sign of the x-component of the velocity).

Turn-in:
To give in the assignment, please use Brightspace. GIVE IN A VIDEO OF YOUR RUNNING PROGRAM. Also, give in a zip file with your complete project (project files, source code, and precompiled executable). It is your responsibility to make sure the assignment is delivered/dated before it is due.

Don’t wait until the last moment to hand in the assignment!

For grading, the program will be run with no command line parameters and the code will be inspected. If the program does not compile, zero points will be given.

```
C:\ <your program>
<window pops up, points bounce, etc>
<ctrl-c to end>
```

If you have more questions, please ask!