CSE 425: Computer Graphics Fall 2020
Programming Assignment

1. Getting Started

Let's start off by looking at main.cpp. It contains a fully functional application that displays a teapot. Other than that, it's not very interesting.

I could compile the code using **Visual Studio 10** without any trouble. After you successfully compile and run the given code, it should display a teapot. Yes, that's all it does. It's now your job to make this application a bit more interesting by modifying the code.

2. Requirements

2.1. Task 1: Color Changes (10)

Add the ability to change the color of the displayed model. Right now, the color is set to [0.5, 0.5,0.9] (RGB), which is a boring light blue.

a) Your task is to use the **c** key to toggle through several other colors (feel free to choose which colors you want).
b) Modify the code so that the **t** key smoothly transitions between different colors (rather than just toggling it).

Note: How do you handle keyboard events? Notice that, when you press keys while the application is running, the console says something like this:

```
Unhandled key press h.
Unhandled key press e.
Unhandled key press l.
Unhandled key press l.
Unhandled key press o.
```

The code that prints these messages is in the **keyboardFunc** function. Modify the code to handle the **c** key appropriately. A reasonable way to do this might be to have the c key increment some sort of global counter variable and then use that variable to select a color in the **drawScene** function. Note that GLUT will not immediately redraw the scene after it has called keyboardFunc. The end of the function contains a call, **glutPostRedisplay()** that updates the display.

2.2. Task 2: Light Position Changes (5)

Add the ability to change the position of the light. In the code, the light is placed at [1.0,1.0,5.0]. Wire the arrow keys to change the position of the light. More specifically, the left/right arrow keys should decrement/increment the first value of the position by 0.5, and the up/down arrow keys should do the same for the second value. This can be done quite similarly to the suggested method for the previous requirement.

2.3. Task 3: Spin the Model (5)

a. use **r** key to spin the model. Implement this functionality in your code (look up glutTimerFunc).

2.4. Task 4: Mouse based camera control (10)

Implement a mouse-based camera control to allow the user to rotate and zoom in on the object.

3. Submission

As a final step, write a README.txt that answers the following questions:

a. Did you collaborate with anyone in the class? If so, let us know who you talked to and what sort of help you gave or received.

b. Were there any references (books, papers, websites, etc.) that you found particularly helpful for completing your assignment? Please provide a list.

c. Are there any known problems with your code? If so, please provide a list and, if possible, describe what you think the cause is and how you might fix them if you had more time or motivation. **This is very important, as we're much more likely to assign partial credit if you help us understand what's going on.**

d. Did you do any extra work? If so, let us know how to use the additional features. If there was a substantial amount of work involved, describe what and how you did it.

e. Got any comments about this assignment that you'd like to share? Was it too long? Too hard? were the requirements unclear? Did you have fun, or did you hate it? Did you learn something, or was it a total waste of your time?

Submit the following in Google Classroom in a Zip file

- Your code (probably just main.cpp for this assignment).
- A compiled executable built from your code.
- The aforementioned README.txt file.
- Any additional files necessary to run your program.