CS/SE 3GC3 Lab 2

September 26, 2020

1 Resources

1. Red Book Chapter 1 `http://www.glprogramming.com/red/chapter01.html` (particularly "A Smidgen of OpenGL Code")
2. GLUT documentation (e.g., `glutInitWindowSize`) `https://www.opengl.org/resources/libraries/glut/spec3/spec3.html`

2 Exercises

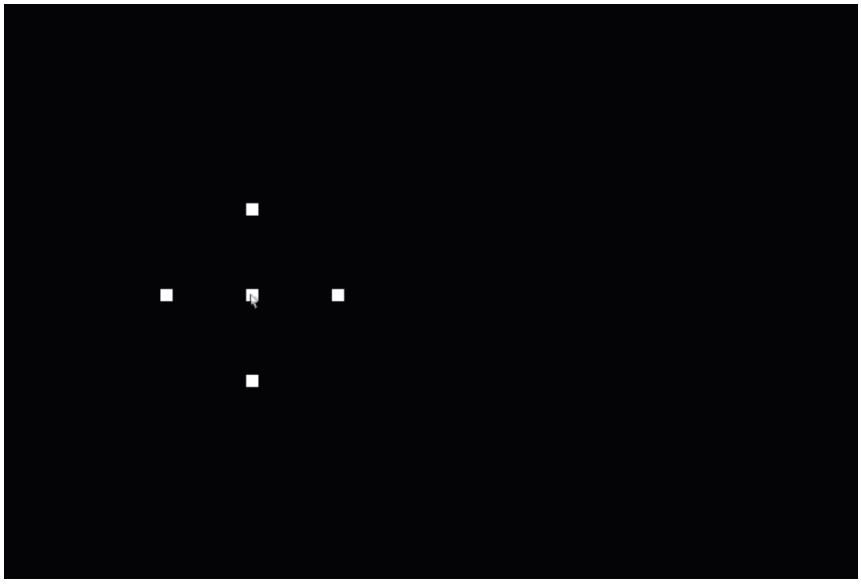
These exercises are graded. You must push your work to your GitLab repo by the end of the tutorial. You will not have any more time to work on these exercises after the tutorial is over. We will only mark the contents of your GitLab repo as they are at the end of the tutorial. This lab is to be completed **only** by you and your lab partner.

This lab will build up on your knowledge of assignment 1, and will help further familiarize you with callback functions, along with an introduction to animating with OpenGL and GLUT.

1. Pull the provided `main.cpp` file for lab 2 from your gitlab repo. This file has basic OpenGL and GLUT environment set up, along with basic functionalities prepared.
2. Write a C++ program which upon clicking on the *left-mouse button*, it will draw a point at that location.
   (a) Use `glutMouseFunc` to handle mouse input. Consult "handling inputs" from Chapter 1 of the Red Book.
   (b) Update the location clicked by the mouse as a global variable for easy access during drawing for later steps.
   (c) Modify the display function to use `drawPoint` to draw a point with the global variables of the mouse click — make sure it draws at the correct location!

3. Draw four additional points — one on top, bottom, left, and right of the point drawn. These will be drawn in the display function.

4. Animate the four points such that they fly outwards from the point (the top point will fly up, the right point will fly right, etc.).



   (a) The animation can be achieved by incrementing a global offset by an animation function.

   (b) Add the callback function `glutTimerFunc` to use this animation function. Consult the GLUT documentation (`https://www.opengl.org/resources/libraries/glut/spec3/node64.html#SECTION00081900000000000000`), or consult lecture's code demo for details on how to implement the timer function.

5. Finally, you should implement a keyboard function, such that whenever you press the 'P' key, the animation will pause. Pressing 'P' a second time should unpause the animation. Note, that this should only affect current animated points. In other words, pressing the left mouse button again will always follow the same behaviour as defined in 4

Make sure you push and commit your changes! Check the GitLab website to ensure you pushed successfully. Your lab can be submitted to either of the lab partners repository.

3 Bonus

The previous exercises are essential for your understanding of the course material and future problems you will be expected to solve. You may implement this

bonus if you have finished the regular exercises. **Make sure you submit the regular exercise before moving onto the bonus.**

1. Press the space bar which will toggle rainbow mode. This will continuously randomize the colours of the the points.