CS/SE 3GC3 Lab 2

September 23, 2019

1 Resources

1. Red Book Chapter 3 `http://www.glprogramming.com/red/chapter03.html` (particularly "Manipulating the Matrix Stacks", "Examples of Composing Several Transformations")
2. GLUT documentation (e.g., `glutInitWindowSize`) `https://www.opengl.org/resources/libraries/glut/spec3/spec3.html`

2 Practice Exercises

These are ungraded exercises for the first hour of the tutorial. This is your time to get 1-on-1 help from the TA!

1. Read "Using glTranslate*() and glRotate*()" from Chapter 3 of the Red Book.
2. Run `make` to compile and run `tut2.c`. Read all the numbered comments carefully.
3. Change the `glTranslatef` call from comment 4 to move the square somewhere in the bottom left quadrant.
4. Use `glRotatef` to rotate the square 60 degrees about the z-axis (check out the documentation for `glRotatef`). Experiment with calling this before *xor* after the call to `glTranslatef`. What happens in each case?
5. Use `glScalef` to double the horizontal size of the "square" and quadruple the vertical size of the "square". You need to be careful about the order in which you call the transformations. It's all matrix multiplication under the hood — this is not commutative.
6. You should have a result that looks similar to the first image. Next, achieve a result that looks like the second image. You should use precisely two sets of calls to `glutSolidTeapot`, `glPushMatrix`, `glPopMatrix`, and `glTranslatef`. Push a new matrix, translate, draw the teapot, then pop

Figure 1: Before

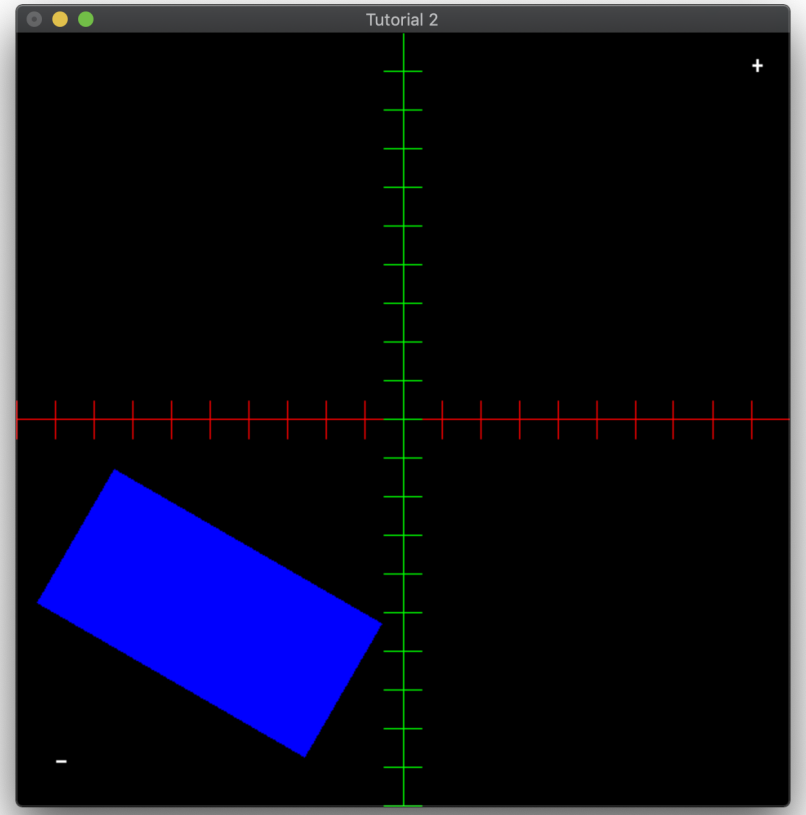
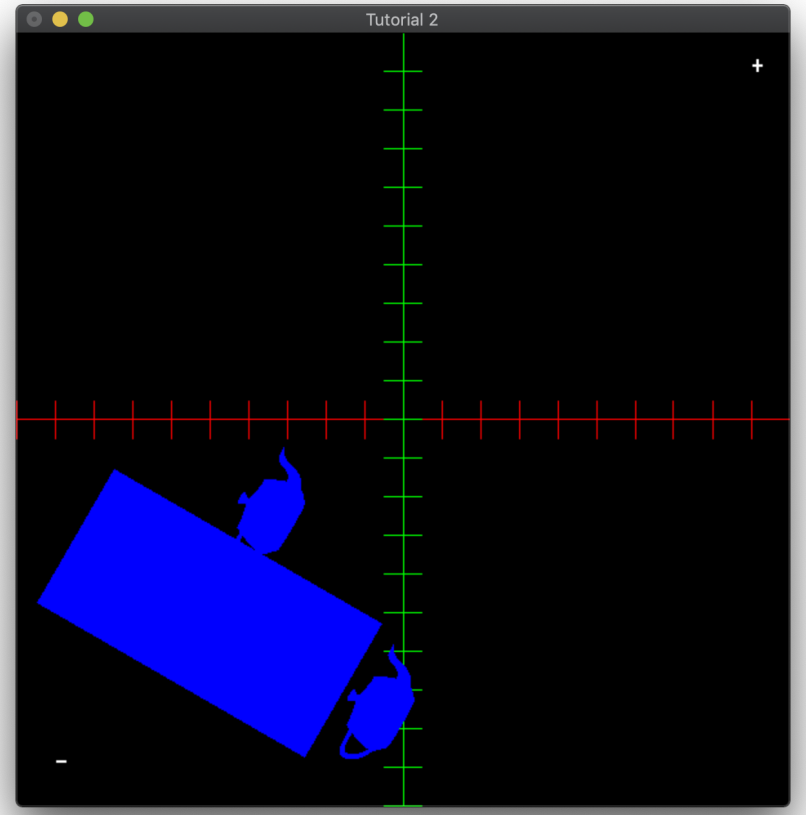


Figure 2: After



the matrix. Do it again for the next teapot. You should have one teapot that is to the right of the "square" and one teapot to the bottom. Each translation should only affect one axis! Take advantage of the matrix stack, do not be too clever!

3 Troubleshooting

1. I added a transformation and now my square isn't on the screen. What?

   It is likely that you need to change the order of your transformations. Calling `glScalef` before `glTranslatef` will have a very different effect than calling it afterwards!