Journal Report 5
9/30/19-10/4/19
Ajit Kadaveru
Computer Systems Research Lab
Period 4, White

Daily Log

Monday September 30

Coded the rotation moves x, y, z, x′, y′, z′. These moves simply change the orientation of the cube. However, after testing them, I realized that the centers wouldn't move with the other stickers on the cube. I forgot that the cube state originally was a 6x8 array instead of a 6x9 array, since I assumed the centers would stay where they were.

Tuesday October 1

I created a new array for the orientation of the centers so that I can implement all the moves that change the orientation of the centers. Now the x, y, z, x′, y′, z′ all work. Then I began coding the actual moves. I coded r, r′, r2, l, l′, l2, d, d′, d2, u, u′, u2, f, f′, f2, b, b′, b2 as compositions of the capital letter moves (one layer moves), and the cube rotations .

Thursday October 3

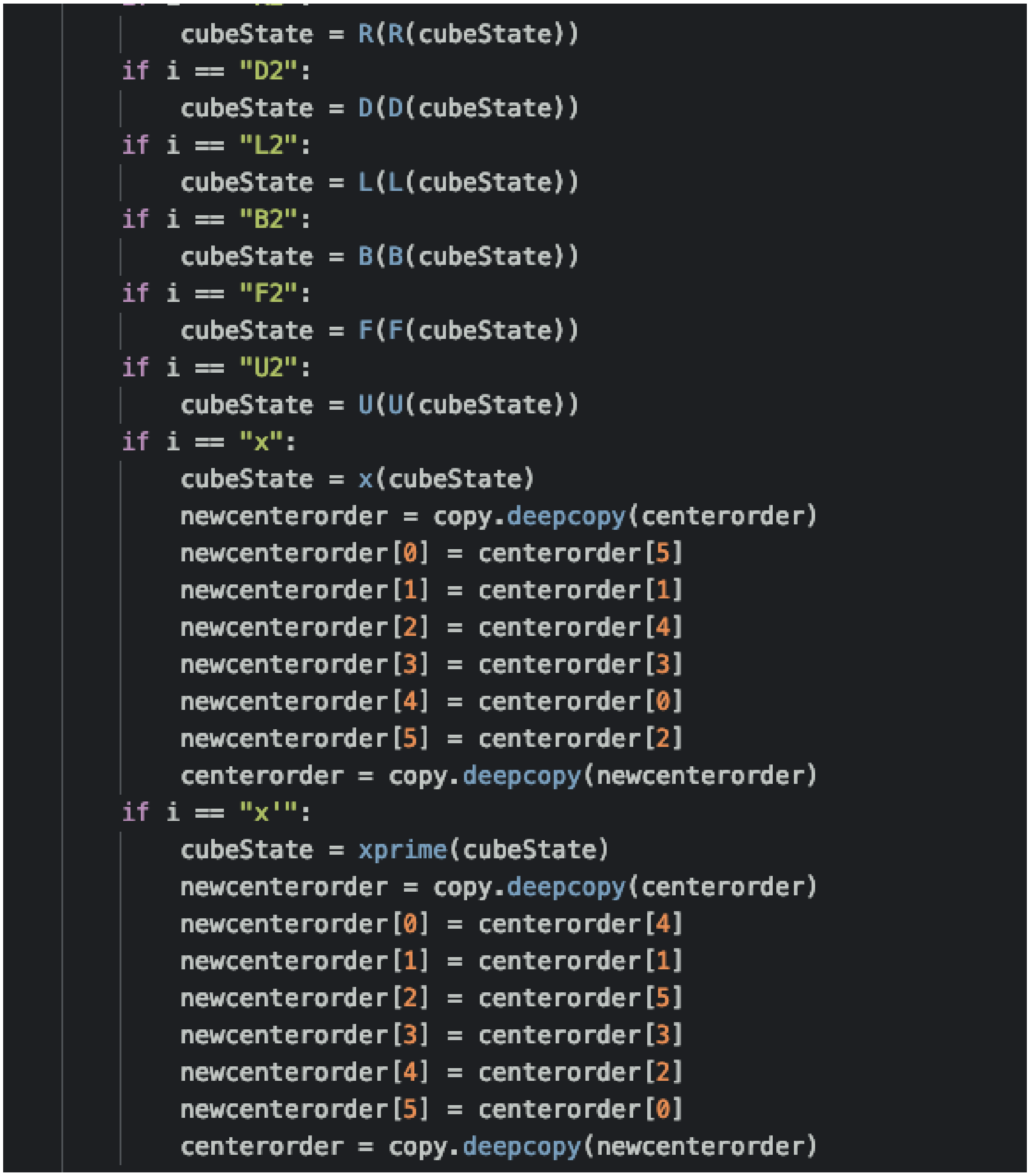
I finished coding all the moves, adding M, M′, M2, S, S′, S2, E, E′, E2. However, when I tested these moves, they didn't work for some reason. For example, when I typed any of these moves 4 times, the cube didn't return its original position. I started debugging this problem, but didn't find out what was wrong yet.

Timeline

Date	Goal	Met
Today minus 2 week	Make the T-display of the cube with colors and change live as you enter move	No, encountered some problems with live display, as PyGame didn't work. But now it's all good since Tkinter seems to work well. I coded the white side to work (see picture below)
Today minus 1 week	Finish updating the GUI for the cube moves so that the live T-display works for the whole cube. Review edge detection in opencv, and upload a sample image of me holding a cube onto the python program to test the edge detection.	Yes, but realized that there are more moves that change the orientation of the centers. These should be a quick add next week though.
Today	Begin coding edge detection to eventually determine which frames of the video the cube are in the shape of a cube (not in the middle of turning),	No, didn't start this since I realized that there are more moves I need to code, so was coding those.
Today plus 1 weeks	Finish the fully working cube GUI with all the moves, and begin coding edge detection to eventually determine which frames of the video the cube are in the shape of a cube (not in the middle of turning),	
Today plus 2 weeks	Upload a sample image to program, and be able to get a picture of just the edges on the image. Possibly, zone out everything but the cube outline.	

Reflection

After realizing that I just need to code cube rotations, and the rest of the moves would follow easily as just a combination of previously coded moves, I encountered some difficulty when dealing with center orientation. When I originally coded the data structure for the cube, I assumed that the orientation of the centers would always be constant. So, I now have to add in a new data structure for the orientation of the centers, which makes dealing with the centers a little harder for these new moves that change their orientation. For each of these moves, I have to not only change the cube data structure, but also the centerorder data structure. It shouldn't take much longer to figure out the problem, fix it, and get the fully working cube GUI done.



```
            cubeState = R(R(cubeState))
        if i == "D2":
            cubeState = D(D(cubeState))
        if i == "L2":
            cubeState = L(L(cubeState))
        if i == "B2":
            cubeState = B(B(cubeState))
        if i == "F2":
            cubeState = F(F(cubeState))
        if i == "U2":
            cubeState = U(U(cubeState))
        if i == "x":
            cubeState = x(cubeState)
            newcenterorder = copy.deepcopy(centerorder)
            newcenterorder[0] = centerorder[5]
            newcenterorder[1] = centerorder[1]
            newcenterorder[2] = centerorder[4]
            newcenterorder[3] = centerorder[3]
            newcenterorder[4] = centerorder[0]
            newcenterorder[5] = centerorder[2]
            centerorder = copy.deepcopy(newcenterorder)
        if i == "x'":
            cubeState = xprime(cubeState)
            newcenterorder = copy.deepcopy(centerorder)
            newcenterorder[0] = centerorder[4]
            newcenterorder[1] = centerorder[1]
            newcenterorder[2] = centerorder[5]
            newcenterorder[3] = centerorder[3]
            newcenterorder[4] = centerorder[2]
            newcenterorder[5] = centerorder[0]
            centerorder = copy.deepcopy(newcenterorder)
```

Figure 1: An image of sample code showing the contrast between executing regular moves, vs executing moves that change orientation of centers