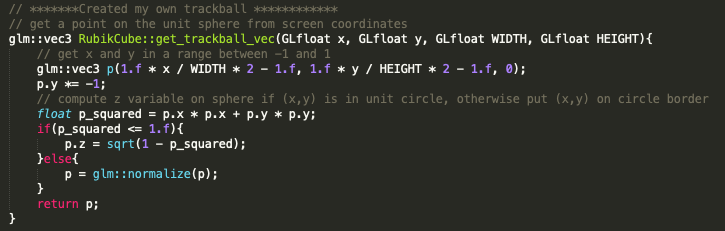
Rubik’s Cube

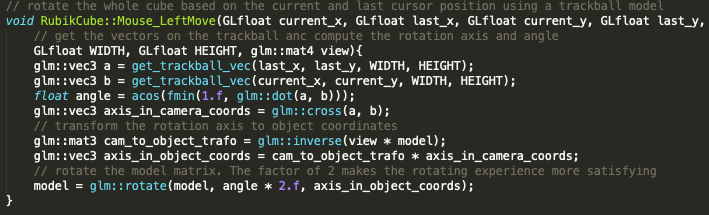
For this project it took me a really long time to understand how the code was structured but the hardest part was to configure the dependencies. I always used my MacOS and Cmake to build my projects and the first time I have managed to do it but I had an unfortunate event and my whole data was wiped out. The second time when I tried to build the project it wasn’t showing the cube when I built and ran the framework because I had a limited time I decided to use my old windows laptop instead to build my project and it worked.

In the given code it was stating or a hint was given that we could use trackball and I also noticed in the RubikCube.h file “ #include "trackball.h” was commented out. I haven’t created my own trackball.h but I have created my own vec3 trackball as shown below:

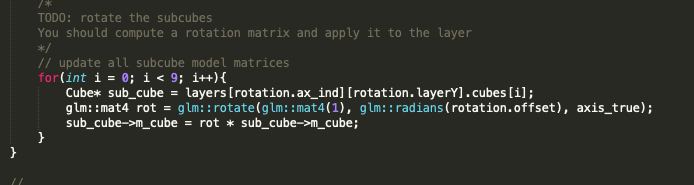
(The rough explanation has been commented). This allowed us to get the vectors on the trackball



The next function I have implemented was to rotate the whole cube using the trackball that I have created. This will allow us transform the rotation axis to object coordinates and rotate the model matrix.



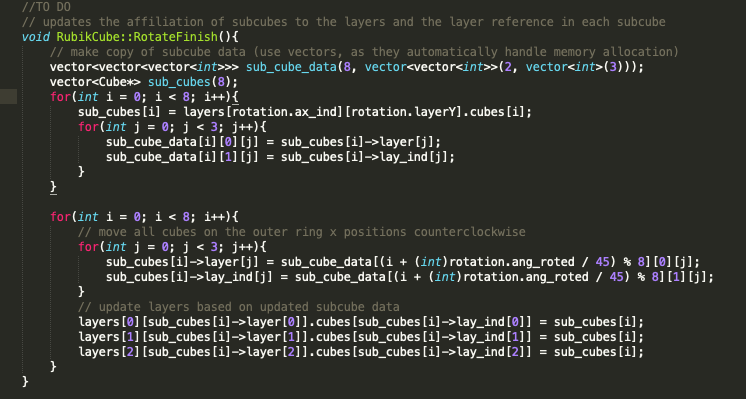
Next, I had to rotate and update all sub cube model matrices and applying it to the layer. This was a part of the rotation step function. I had to update all sub cube model matrices.



As stated on the comments, the positions of cubes have been updated but the connection between the cubes and the layers have not been updated and this causes problems.

For the sub-cubes of rotated layers, we had to update which cubes belonged to a layer.

This function included two cases: clockwise and counterclockwise. It updates the affiliation of sub cubes to the layers and the layer reference in each sub cube by making a copy of the sub cube data, moving all cubes on the outer ring positions counterclockwise and updates the layers based on our updated sub cube data that we have gathered.



Working Screenshots:

