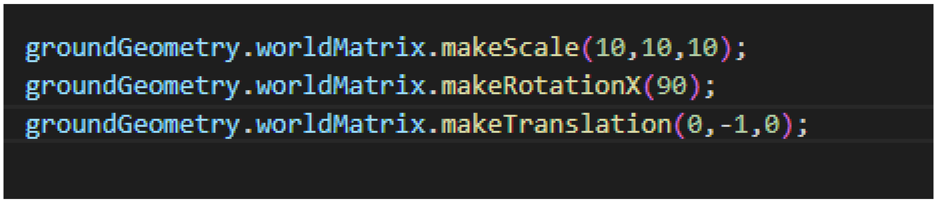
Concept:

Most concepts used in this week were everything we have seen before. It was fascinating to see the concepts being used to create a first person camera basically. But the previous lectures and homework really made it “eye-opening” on how the matrices play into the camera and for it to move. I say eye-opening because it wasn’t easy but not understanding the previous assignments. Would make this assignment much more difficult than it should be.

Also an important concept I noticed to understand when working on this assignment was what two properties we mostly use in 3D graphics. These are MATRIX x MATRIX and VECTOR x MATRIX. Understanding this, made this assignment much smoother.

Experience:

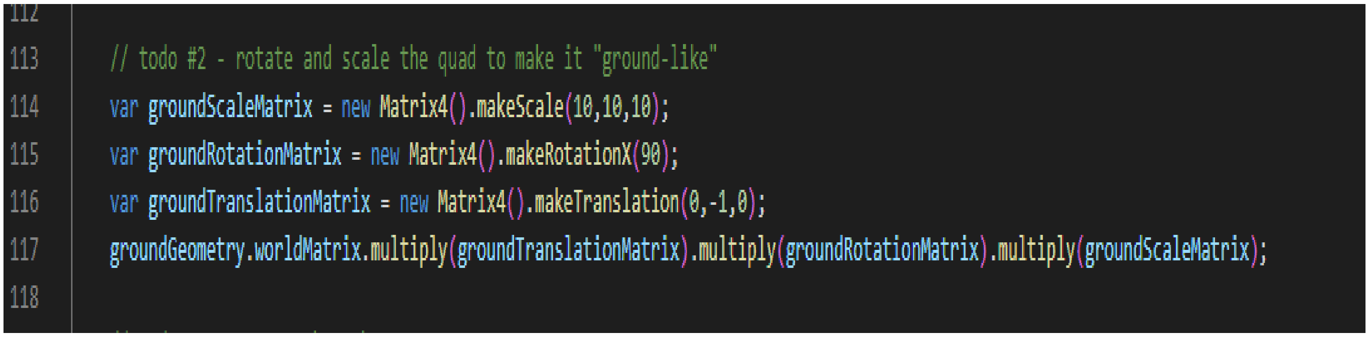
In my experience of working on this assignment, I most struggled with todo #2. I created my matrices. I need to transform the groundGeometry. But I wasn’t sure how to apply the matrices. I tried just doing this,



```
groundGeometry.worldMatrix.makeScale(10,10,10);
groundGeometry.worldMatrix.makeRotationX(90);
groundGeometry.worldMatrix.makeTranslation(0,-1,0);
```

With it “worked” but gave back some weird results and not what I was looking for. But from the concept I talked about above, I need to apply the changes by multiplying the groundGeometry.worldMatrix by all of the transformations I am doing.

So I would make it like this,



```
    // todo #2 - rotate and scale the quad to make it "ground-like"
    var groundScaleMatrix = new Matrix4().makeScale(10,10,10);
    var groundRotationMatrix = new Matrix4().makeRotationX(90);
    var groundTranslationMatrix = new Matrix4().makeTranslation(0,-1,0);
    groundGeometry.worldMatrix.multiply(groundTranslationMatrix).multiply(groundRotationMatrix).multiply(groundScaleMatrix);
```

Soon as I did this it worked and my quad was now “ground-like”. And after reviewing this, I understood why it worked. Since we followed our matrix logic/properties. It will transform our main matrix properly. It was interesting to see, and understand, why we only use mXm and vXm in 3D graphics, well least in this class.