**Concept:**

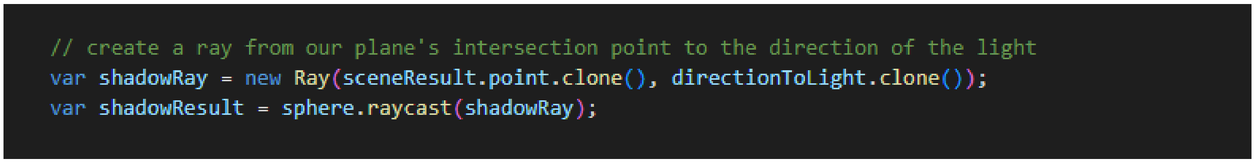
One concept I had to grasp was understanding what a pixel's width is. I am pretty sure the pixel width is the pixel's scale and then how many pixels are on the screen. This makes sense because if you scale a pixel, you have to account for that. Hence why I did pixelScale / pixelsAcross.

The second concept I had a weird time grasping was the eyeDistance. I know the eyeDistance is the distance from the camera to the scene objects. I know I had to use the tangent of the fovRadians. But at first I was confused on what to use for the area of the grid. I obviously knew I had to use pixelWidth or pixelHalfWidth and then how many pixels were on the screen horizontally. After messing around with those variables, it started to make sense on why those were needed and the way they needed to be used.

Third concept I thought was interesting was creating the shadow shaders. I understood the shadow shaders completely, because it was basically simple vector math. But I was confused at first on what intersection point to use (mostly because I didn't fully read the todo part). But after re-reading the todo section for it, it made me do a "aaaah" moment. Since I want to check any object that is able to reach the light or not. This was super interesting/awesome to understand since this can be applied to making some cool shadows in a game. This has helped me understand how shadows could work, when dealing with objects in a scene.

**Experience:**

My partner and I were an effective team. Like other weeks, it is a joy to be working with Julian. This time around we mostly had debugging problems. Or trying to solve bugs. We used the debugger efficiently but could not figure out this one bug with the shadow shaders (it was completely my fault with it).



```
// create a ray from our plane's intersection point to the direction of the light
var shadowRay = new Ray(sceneResult.point.clone(), directionToLight.clone());
var shadowResult = sphere.raycast(shadowRay);
```

Before creating the code a bit more universal, I had coded
**new Ray(sphereResult.point.clone(), directionToLight.clone());**
At first I was like "alright this seems good, correlates with what the document says." But it was wrong since I was taking the intersection point of the sphere and not the plane. I "soon" (3 hours later) realized this was the issue on why our shadow shader wasn't working. All because I mis-read the first picture, where it clearly states "take the PLANE's INTERSECTION…". As soon as I read this, I changed the sphereResult.point to planeResult.point. And boom, our shadow shader worked. I definitely just need to slow down and fully read everything.

This had to be the biggest challenge for us to overcome. Everything before and after this was pretty smooth sailing. It all made sense, even making it a bit more universal for multiple objects in the scene. I tried to implement the simple Anti-aliasing but it didn't come out so well. I am excited to see how I can implement this knowledge into future games I’ll be making. Let alone expand my knowledge on 3D graphics for game development.