

## Concepts:

One major concept that caught my attention was projection. I found projection interesting because it reminds me of shadows from my understanding of it. When you project a vector onto another vector. You can imagine it as if the sun was above the two vectors. Then imagine the vector to project onto another vector was basically to be the shadow produced from the sun. So that is basically what “projection” is when it comes to projecting a vector onto another vector. This also got me thinking about shadows in video games. Maybe not entirely done this way, but I was wondering if lighting/shadows are somehow connected with the projection of two vectors. I did find some additional [information](#) to help clarify what projection is. And how it should look mathematically when dealing with 3D vectors.

Another concept I thought is interesting and good to understand is points/displacements. Points are positions in a defined space. Or points could be displacements from a fixed origin. This means you can use points to measure how far you are from an object, or area, in Unity or coding/programs in general.

Vectors on the other hand are just displacements in any given space. It doesn't really matter where you put them in space, they will still have the same magnitude and direction. Unless you subtract or do some sort of function to change the magnitude and direction of the vector. What I find most interesting about this topic is the operations points/displacements are capable of doing. You can subtract two points, add a point with a vector, add two vectors together, but you cannot add two points together.

## Experience:

To start with, my partner and I had no real difficulties working together. It was smooth and an enjoyable learning experience. For my understanding of converting vector operations into code, wasn't really difficult. I am not saying the material was easy to understand. I believe it all just made sense to me, from coding in Unity so much recently.

However, I was having trouble understanding projection. That is why I looked up some additional information to clarify it. At first I had some trouble implementing the projection functionality but that was mostly due to not cloning. I cloned the vectors passed in then performed the operations needed. Which also was a bit tricky, since I needed other vector operations to perform this. After a few minutes of visualizing what I needed to do first, I came to the terms that basically I am multiplying a vector by a scalar. At first that seemed kinda broad, but that scalar value you need to find is simply just finding the dot product of the two vectors and

that divides that by the magnitude squared of the vector you are projecting to the other vector.

```
    // ((u . v)/||v||) * v
    // this returns the projected vector of "u" onto "v"

    var cloneToProject = vectorToProject.clone();
    var cloneOtherProject = otherVector.clone();

    var sum = cloneToProject.dot(cloneOtherProject)/cloneOtherProject.lengthSqr();
    var vec = cloneOtherProject.multiplyScalar(sum);

    this.x = vec.x;
    this.y = vec.y;
    this.z = vec.y;

    return this;
  }
};
```