Journal Entry -Vectors

Sunday, September 3, 2023 6:18 PM

Concepts Learned

I was quite scared to look back into vectors or doing any math since it isn't my strong suit. However, since starting this class I have had a greater understanding of the different concept of trigonometry and vectors. I still am having trouble with some of the calculations. They can come off as confusing or lengthy. For example, When taking the quiz for vectors this week I was nervous but, I felt confident that I knew enough about the subject to take a quiz on it. After taking the quiz on vectors, I realized that I didn't really grasp as much as I thought I would have. Prior to taking the quiz I made sure to watch most of the videos presented (including the extras). I feel like I should have reviewed them more than once because sometimes it can be hard for me to have an understanding of the video watching it once even when taking notes.

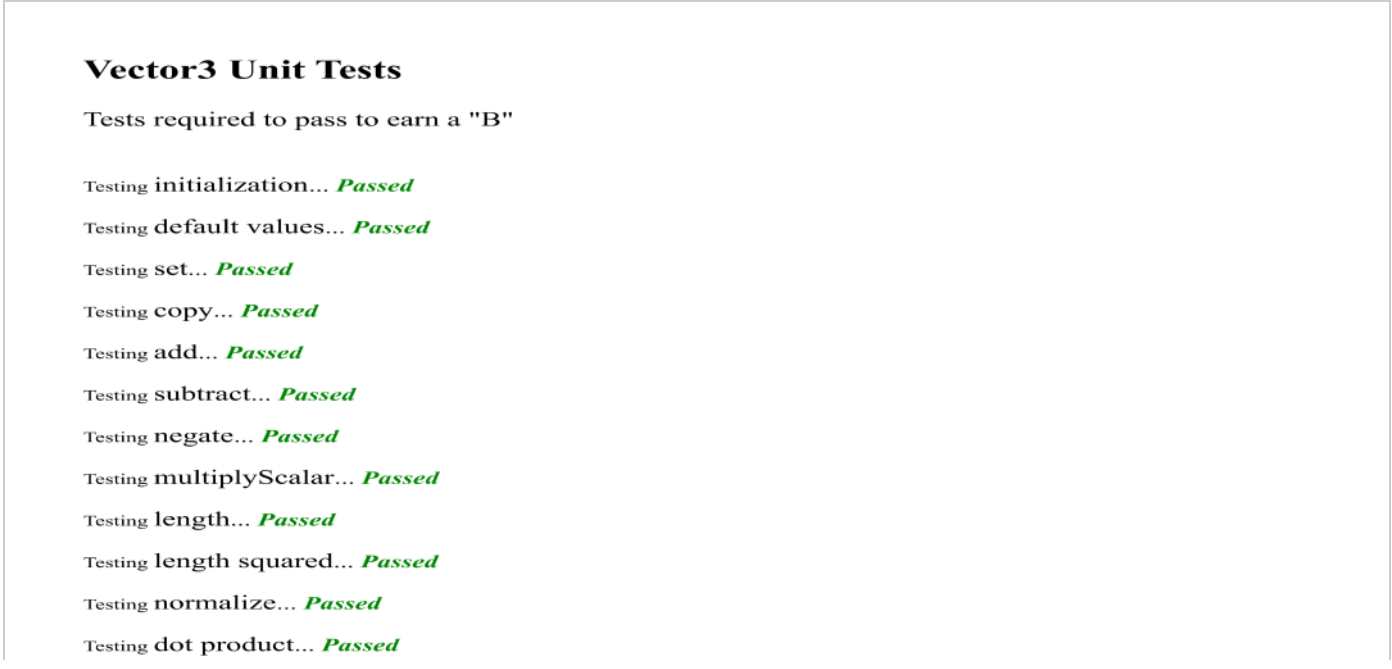
Some of the concepts that I found difficult was selecting the accurate depictions of a vector. I was able to select a few correct ones. However, I still have trouble understanding what I did wrong and what I was doing right. When doing some of the mathematical equations such as finding the length of a vector. It was a refresher. I took whatever I knew from past classes (which, to be honest, is not a lot) watched the videos and tried to follow along and some concepts where easy to remember. Other questions within the quiz did confuse me such as "All vectors can be normalized." I said that it was ***true*** but, a zero vector would have no direction which means it will not change when being multiplied or divided. When seeking additional sources for clarity, I would simply ask google, ChatGBT or just look at my provided resources within classroom.

Application of Concepts

When applying some of the concepts within the partnered programming assignment, I found the to-do areas easily understandable and straightforward. I did struggle with coding or finding different ways to code things in the mathematical sense. I felt a little uncomfortable when coding in this language but, that just means more room to grow. The way that I overcame this challenge was by challenging myself through debugging code, practicing and reading that one JavaScript webpage that introduced language formats. I believe my partner and I worked effectively as a team because we consistently remained in contact and answered each other's questions, concerns and ideas.

Visual Documentation & AI Utilization

https://chat.openai.com/share/8c2741ce-a51f-40a4-9df8-443cd63f9e3c



Testing length squared... *Passed*

Testing normalize... *Passed*

Testing dot product... *Passed*

Passed

Tests required to pass to earn an "A"

Testing from-to... *Passed*

Testing rescale... *Failed*

Testing angle... *Failed*

Testing projection... *Failed*

Passed 14 tests out of 17

Vector3 Unit Tests

Tests required to pass to earn a "B"

Testing initialization... *Passed*

Testing default values... *Passed*

Testing set... *Passed*

Testing copy... *Passed*

Testing add... *Passed*

Testing subtract... *Passed*

Testing negate... *Passed*

Testing multiplyScalar... *Passed*

Testing length... *Passed*

Testing length squared... *Passed*

Testing normalize... *Passed*

Testing dot product... *Passed*

Passed

Tests required to pass to earn an "A"

Testing from-to... *Passed*

Testing rescale... *Passed*

Testing angle... *Passed*

Testing projection... *Passed*

Passed 17 tests out of 17

```
10  /**
11   * Author: Melody Neely
12   * Partner: Luke Moore
13   * Date:09.03.2023
14   *
15   */
16
17  var Vector3 = function(x, y, z) {
18      this.x = x || 0;
```

```
/**
 * Author: Melody Neely
 * Partner: Luke Moore
 * Date:09.03.2023
 *
 */

var Vector3 = function(x, y, z) {
    this.x = x || 0;
    this.y = y || 0;
    this.z = z || 0;

    // Sanity check to prevent accidentally using this as a normal function call
    if (!(this instanceof Vector3)) {
        console.error("Vector3 constructor must be called with the 'new' operator");
    }

    // todo - make sure to set a default value in case x, y, or z is not passed in


}

Vector3.prototype = {
```

```
    //-------------------------------------------------------------------------
    normalize: function() {
        // todo - Change the components of this vector so that its magnitude will equal 1.
        // This SHOULD change the values of this.x, this.y, and this.z
        var mag = this.length();
        if (mag !== 0) {
            this.x /= mag;
            this.y /= mag;
            this.z /= mag;

        }
        return this;
    },

    //-------------------------------------------------------------------------
    dot: function(other) {
        // todo - return the dot product betweent this vector and "other"
        // This should NOT change the values of this.x, this.y, and this.z
        let dotProd = this.x * other.x + this.y * other.y + this.z * other.z;
        return dotProd;
    },
```

```
169
170     //-------------------------------------------------------------------------
171     angle: function(v1, v2) {
172         // todo - calculate the angle in degrees between vectors v1 and v2. Do NOT
173         //        change any values on the vectors themselves
174         var dotProd = v1.dot(v2);
175         var v1 = v1.length();
176         var v2 = v2.length();
177
178         if (v1 !== 0 && v2 !== 0) {
179             var cosTh = dotProd / (v1 * v2);
180             var rad = Math.acos(cosTh);
181             var deg = rad * (180 / Math.PI);
182
183             return deg;
184         }
185         return 0;
186     },
187
188     //-------------------------------------------------------------------------
189     project: function(vectorToProject, otherVector) {
190         // todo - return a vector that points in the same direction as "otherVector"
191         //        but whose length is the projection of "vectorToProject" onto "otherVector"
192         //        NOTE - "vectorToProject" and "otherVector" should NOT be altered (i.e. use clone)
193         //        See "Vector Projection SLides" under "Extras" for more info.
194
195         var dotProd = vectorToProject.dot(otherVector);
196         var v2 = otherVector.lengthSqr();
197         if (v2 !== 0) {
198             var scalar = dotProd / v2;
199             var re = otherVector.clone().multiplyScalar(scalar);
200             return re;
201         }
202         return new Vector3(0, 0, 0);
203     }
204 };
```