The version of Unity that we will be using for this part of the project is 2018.2.5f1.

Programming Guidelines for KCU

1. Try to keep file length small and programs concise
    a. Instead of having one large frankenfile, break them down into purposes (I.e. EnemyMovement, CharacterMovement, etc.)
    b. Have files call functions in other files instead of duplicating the functions
2. Be consistent in formatting and syntax.
    a. Use descriptive names. No abcdef, instead enemySlimeHealth
    b. Use camel case for variables and Pascal case for script and file names.
        i. i.e. camelVariableName and ScriptFileName
    c. If a line would extend past the edge of the screen, try breaking it up into multiple lines.
    d. Do not be afraid of w h i t e s p a c e – legibility is key
        i. Put space in between functions, in between variables, anywhere it makes logical sense.
        ii. Indent within blocks of code.
3. Comment your code!
    a. If the code you're writing does not have a clear purpose, it's best to comment it.
    b. You don't have to comment the actual process if necessary, just the results.
        i. i.e. 'This function calculates the velocity and returns it as a variable v'
        ii. This loop counts out how many times something does something.
    c. Functions should have a basic description of input/output, and the beginning of files.
4. Check out the Microsoft coding guide:
    a. https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/inside-a-program/coding-conventions

```
/*
 * This is an example of what to strive for for coding standards, kind of.
 */

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class ExampleScript : MonoBehaviour {

    private GameObject globalVariable;
    public GameObject publicVariable;

    int exampleVar = 5;
    int exampleNum = 10;


    // Prints a message to console
    public void HelloWorld()
    {
        Debug.Log("Hello, World! " + exampleVar);
    }


    // Use this for initialization
    void Start () {
        int i;
        if (exampleNum > 10)
        {
            exampleNum--;
        }
        else
        {
            exampleNum++;
            for (i = 0; i < 100; i++)
            {
                exampleNum++;
            }
        }

        HelloWorld();
    }

```