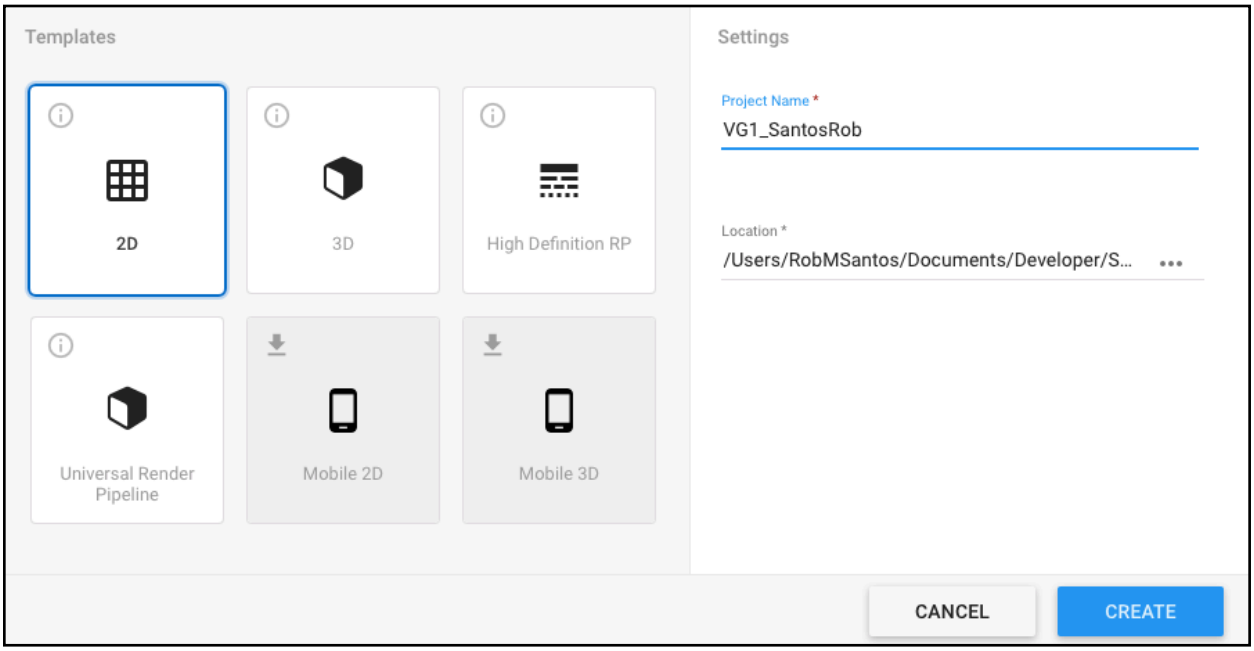
Quest 1 - Movement - Steps

1) Create Project

***Your Unity version may differ. You MUST use the Unity version required by the instructor for your particular course semester.**

You will use this one project for MULTIPLE assignments throughout the entire semester. Replace my name with your LastnameFirstName.

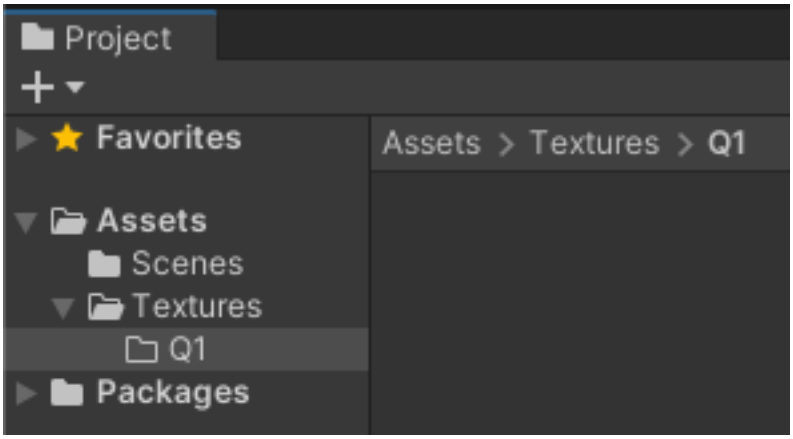
Unity provides different templates for starting your project. All of the projects we do this first semester will be 2D games.



2) Setup Folders

The Project tab portrays your file system and the assets that are within your project. You should always keep your projects organized with an appropriate folder structure. Clicking the + lets you create new assets including folders.

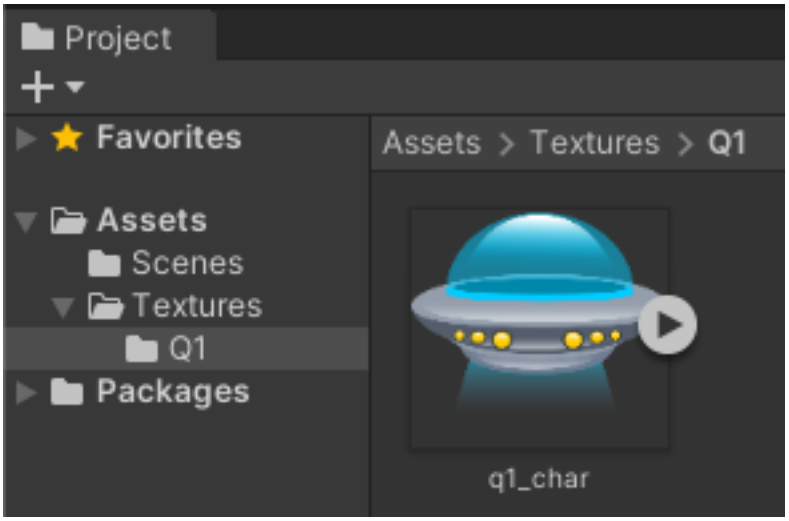
Because all of our assignments will share this one project, I will use folders to help organize content from different exercises.



3) Import Files

You can drag-and-drop files into the Project tab to add them to your game.

A texture is a graphic that can be applied to some aspect of your game.



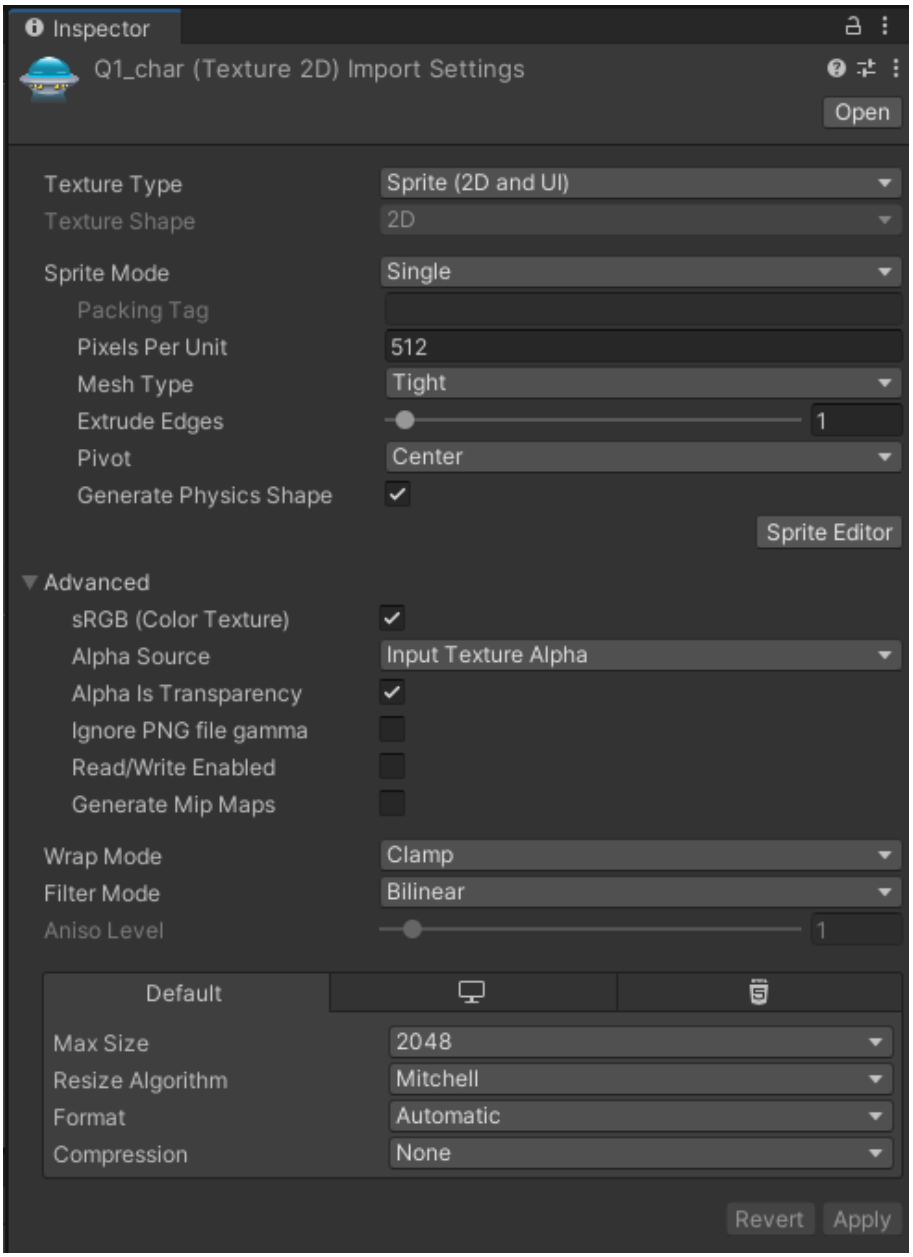
4) Graphics Import Settings

The Inspector tab is a contextual tool for examining and configuring elements of your game. Clicking different assets will show you different configuration options for that particular content.

Click your q1_char in the Project tab and match these settings for your newly imported video game graphic using the Inspector tab.

We will keep most of the default settings, but will specifically change the Pixels Per Unit and Compression. A Pixels Per Unit (PPU) of 512 means we will fit 512 pixels per grid length in our

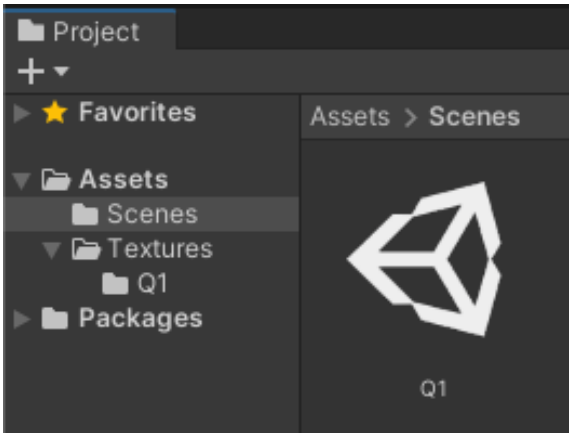
game's coordinate system. It ensures graphics are properly sized when they are brought into the game. Setting the Compression to None will keep our graphics at full quality.



5) Rename your Scene

To keep our project organized for the semester, we will rename our current scene to Q1 to represent this assignment.

Find your Scene in the Project->Assets->Scenes folder and rename SampleScene to Q1. You may be asked to reload the currently active scene.

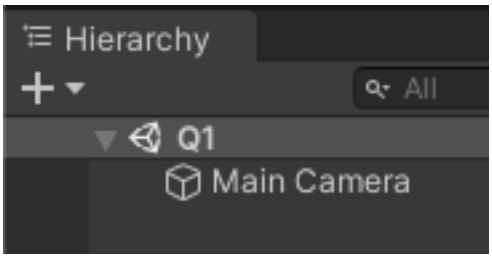


6) Create a GameObject

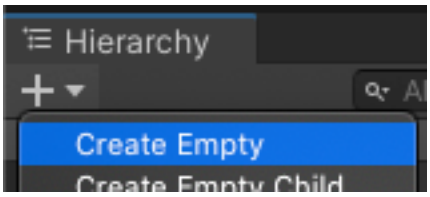
In Unity, nearly everything that appears in a game is “GameObject.” GameObjects are the fundamental building blocks of games in the Unity Engine.

While the Project tab shows you what assets are in your project, the Hierarchy tab shows you what GameObjects are in your Scene. A Scene is what is currently portrayed by your game. Like a movie, a game will progress through multiple Scenes, and you can use Scenes to organize your experience into different levels, screens, etc.

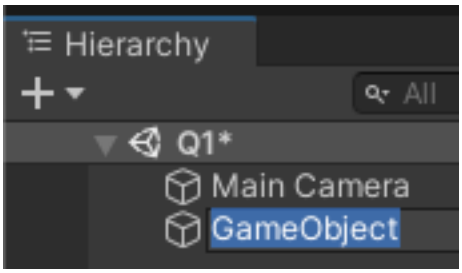
Right now, our Q1 Scene only has the default Camera that allows the engine to “see” the game.



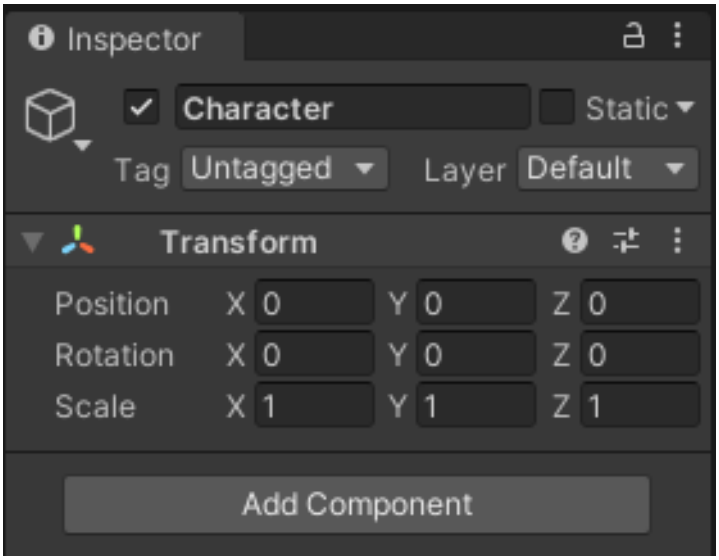
You can use the + to create an empty “blank” GameObject for us to configure.



Notice how it gives you the opportunity to rename the GameObject. Change the GameObject’s name to Character. We will eventually make this character move around in response to keyboard input.

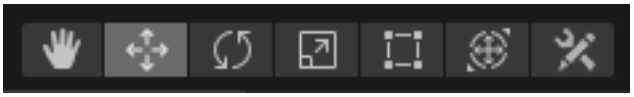


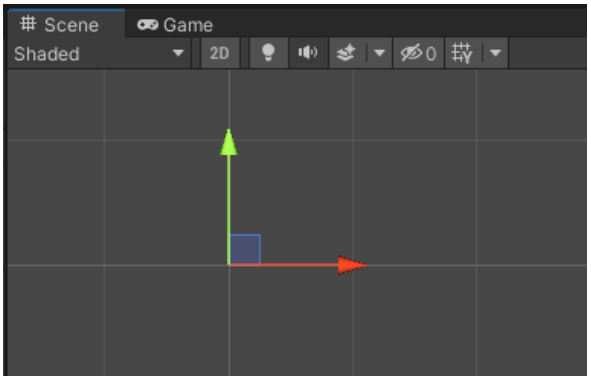
When you select Character in the Hierarchy tab, notice that the Inspector tab changes to describe that GameObject.



When you Inspect a GameObject, you will see that GameObject's list of Components. While GameObjects are the fundamental building blocks of the Unity Engine, Components are what define any GameObject. A GameObject is not a "Character" because we named it that, but rather because we attach specific Components that describe what a Character is.

Switch to the Move tool and look at the Scene tab.



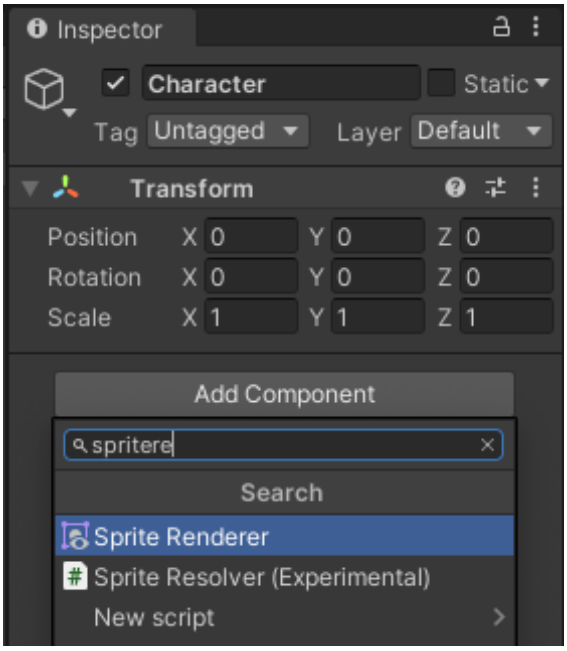


Notice that without any Components beyond the Transform Component, our Character GameObject simply exists at Position 0, 0, 0 somewhere in the game Scene. It has no graphics.

7) Add a SpriteRenderer Component

Our “Character” should have a graphic. This is an ability gained by having a specific kind of Component attached to our GameObject.

Specifically, the SpriteRenderer Component allows a GameObject to have 2D graphics. Attach one using the Add Component button.

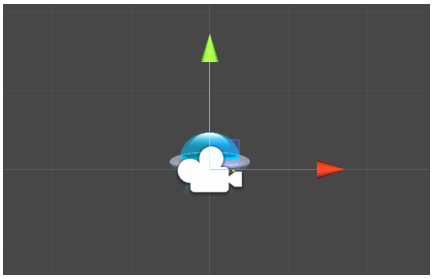


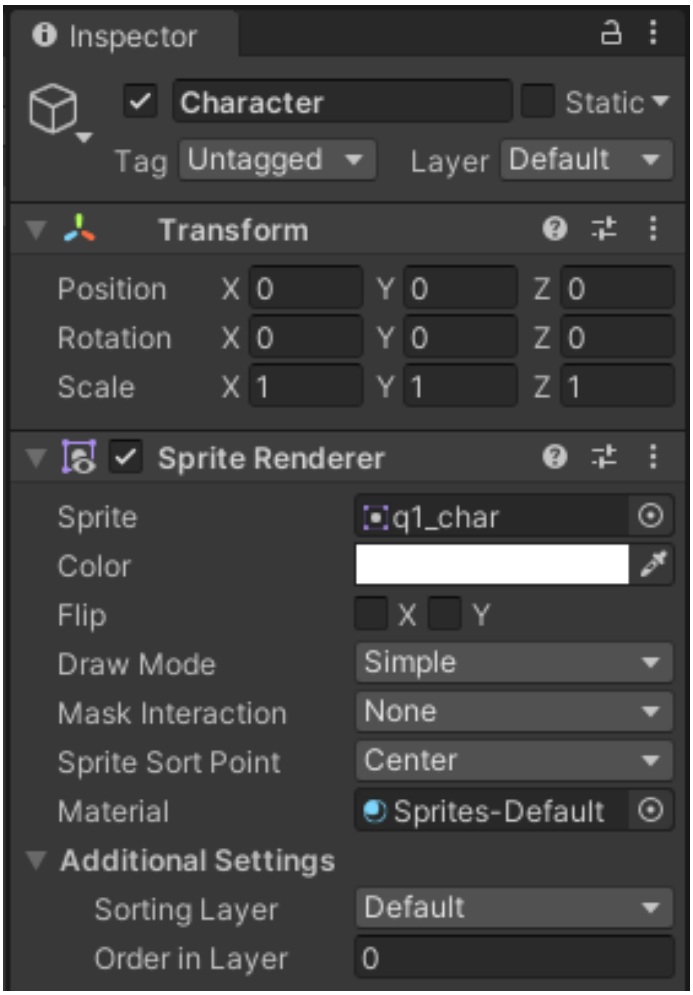
8) Configure SpriteRenderer Component

The Inspector will let us configure this SpriteRenderer Component.

In the Sprite property, you can drag-and-drop the spaceship graphic from the Project tab into the Sprite blank in the Inspector tab. The other default settings should be fine.

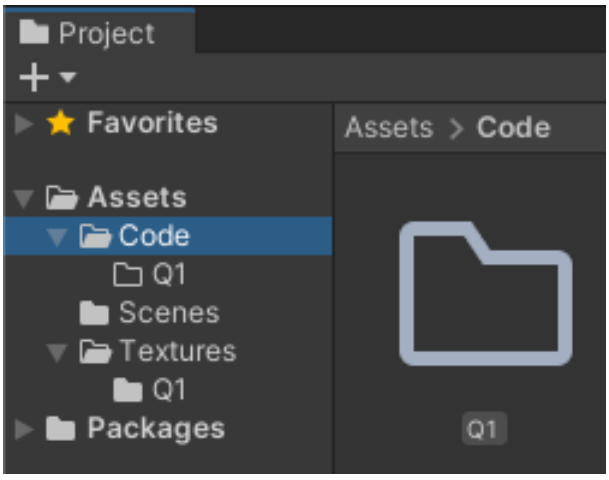
Our Character now has a graphic!





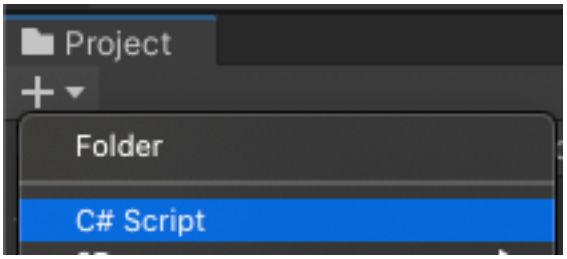
9) Organize More Folders

We will keep all of our C# code in dedicated folders for each exercise.

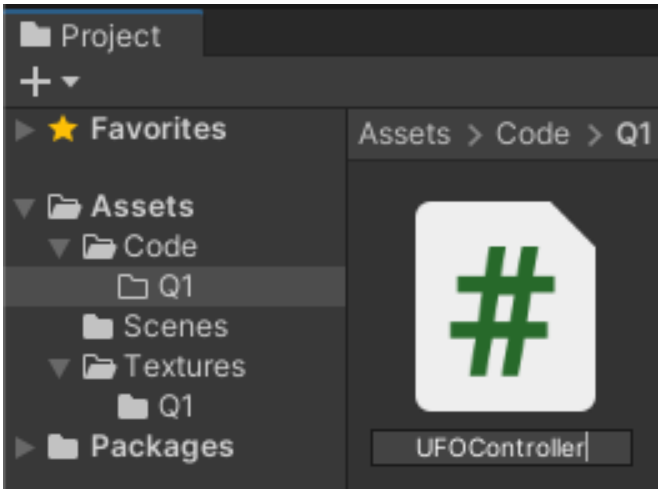


10) Create a C# File

Create a C# file, which we will use as a “Behavior Script.” (C# is not technically a scripting language, but because C# replaced Boo/UnityScript in the engine, the terminology is still commonly used to describe programming in Unity.)



Whenever you create a C# file through Unity, it will automatically generate it with a code template based on the name you provide. In Unity, the filename must ALWAYS match the C# Class name inside the file. If you ever mistype or rename your files, you must also rename the C# Class inside the code because the auto generator won’t know about the error. (This also means no spaces should be used in these filenames.)

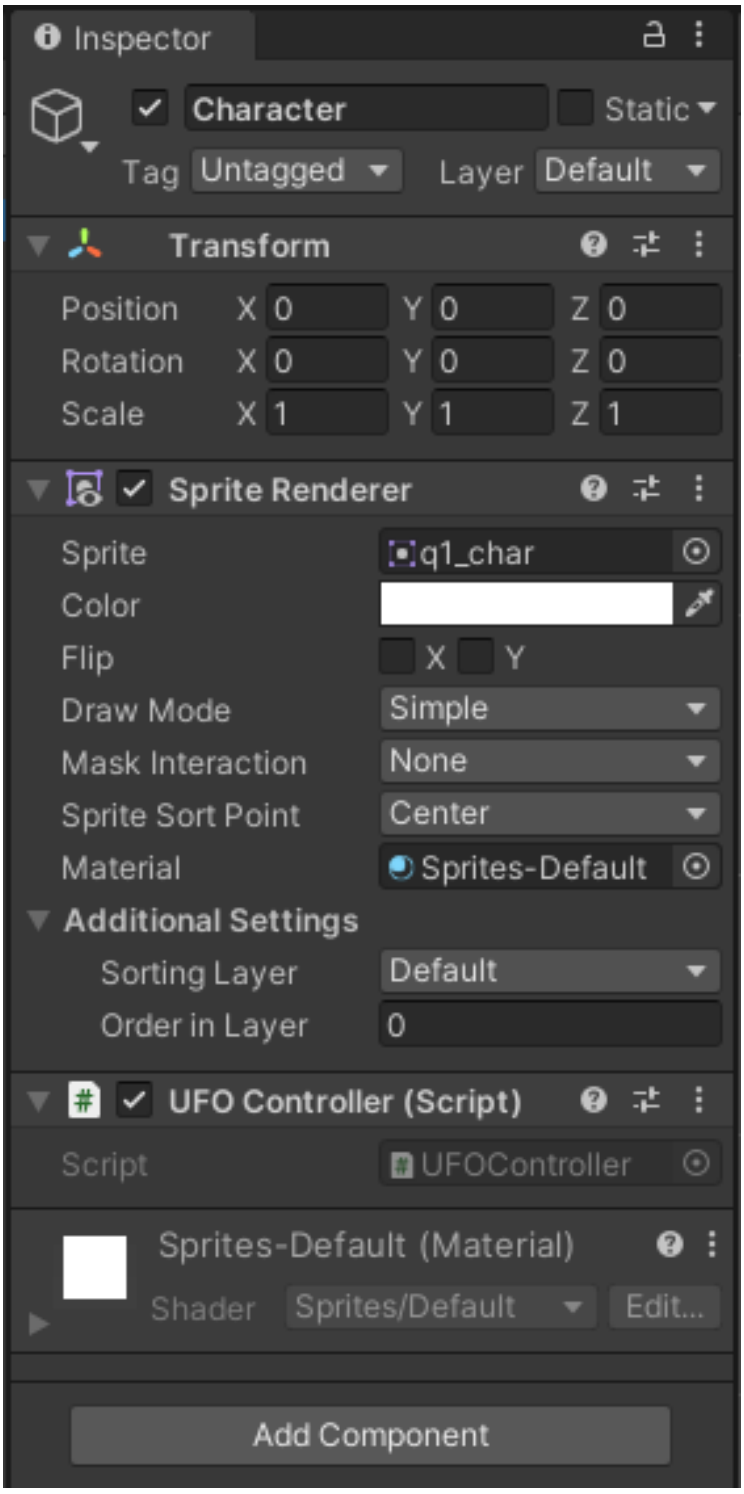



```
public class UFOController : MonoBehaviour
{
```

11) Attach the C# Behavior as a Component

In the Inspector when your Character GameObject is selected, click the “Add Component” button to search for and add UFOController. (You can also drag-and-drop the UFOController file from the Project tab onto the “Add Component” button in the Inspector tab.)

Whatever approach you use, your Behavior Script will now appear as a Component on the GameObject. Any Component you program can be just as functional as any of the Components that come built-into the Unity engine.



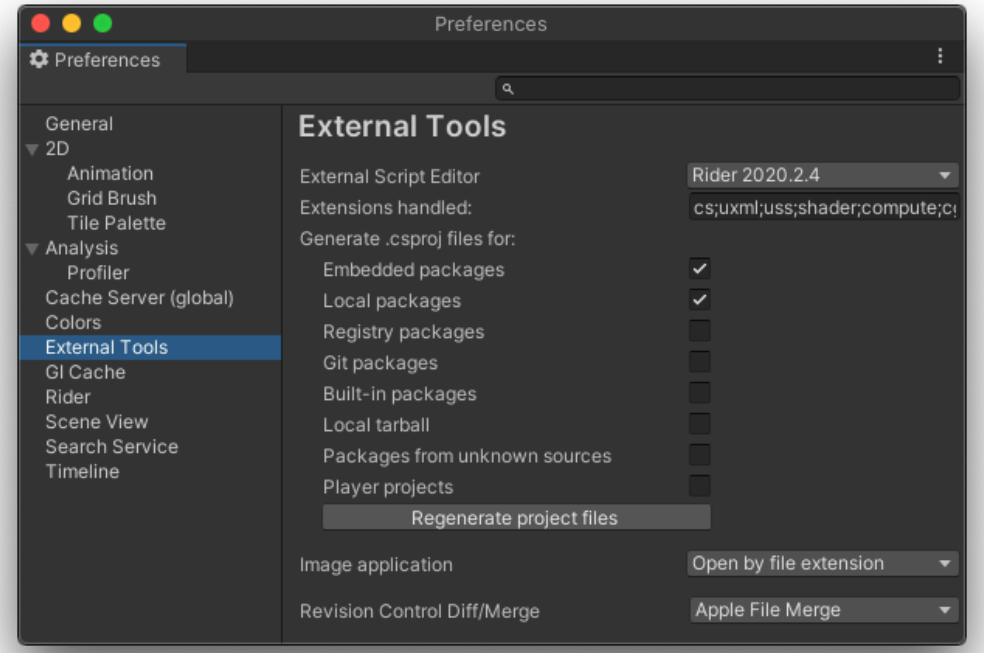
12) Program Basic Movement

As our “Hello World” exercise, our goal is simply to get a character to appear on the screen and move in response to player input. We will use the Arrow Keys to do this.

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

No asset usages
public class UFOController : MonoBehaviour {
    // Start is called before the first frame update
    Event function
    void Start() {

    }

    // Update is called once per frame
    Event function
    void Update() {
        // Move Up
        if(Input.GetKey(KeyCode.UpArrow)) {
            transform.position += new Vector3(0, 0.2f, 0);
        }

        // Move Down
        if(Input.GetKey(KeyCode.DownArrow)) {
            transform.position += new Vector3(0, -0.2f, 0);
        }

        // Move Left
        if(Input.GetKey(KeyCode.LeftArrow)) {
            transform.position += new Vector3(-0.2f, 0, 0);
        }

        // Move Right
        if(Input.GetKey(KeyCode.RightArrow)) {
            transform.position += new Vector3(0.2f, 0, 0);
        }
    }
}
```

Much of this code may be unfamiliar to you right now, but we will explore all of it more in-depth in future exercises.

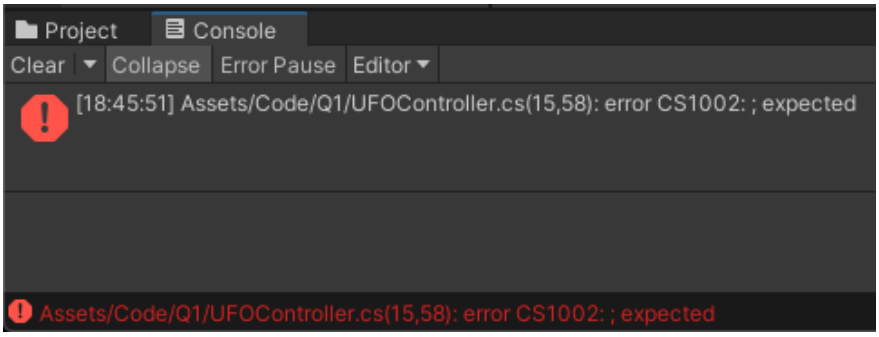
If your C# file doesn't open as expected, check your Unity->Preferences (Edit->Preferences on Windows) and make sure your code editor of choice is selected under External Tools.



13) Playtest

Click the Play button to test your game. Your character should move in the direction of each of the four arrow keys when pressed.

Pay attention to the bottom-left corner of the Unity Editor window for any Red Errors. You can view more details by opening up Window->General->Console. Assignments must never be submitted with Red Errors (which could occur during compilation or runtime.) Red Errors represent a video game crash. If you're ever unsure how to fix a Red Error, just ask.



You should ALWAYS PLAYTEST. Playtest frequently as you progress through your exercises to test for both expected and unexpected behavior in your projects. You are responsible for finding your own bugs. It is unacceptable in the video game industry to blame an unawareness of bugs on the end-user or player not telling you. Testing should be a part of your own process. There are severe penalties in this class for turning in assignments where it's clear that not even you could run your own game.