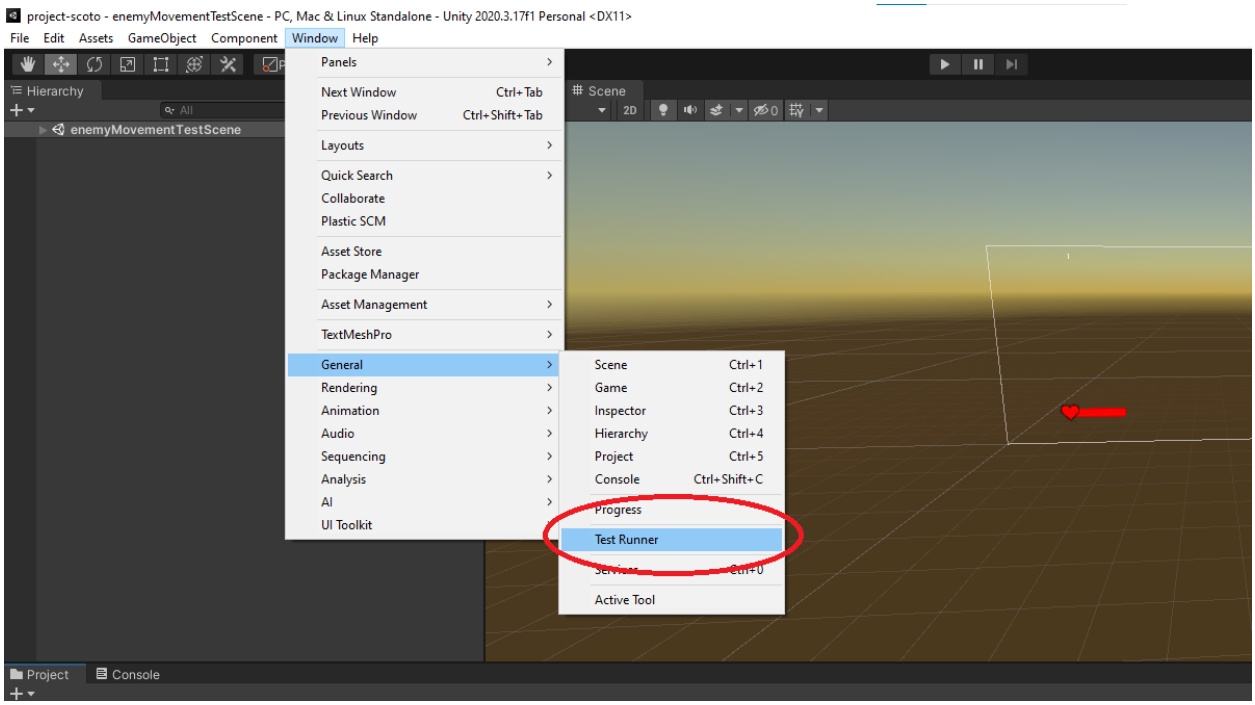
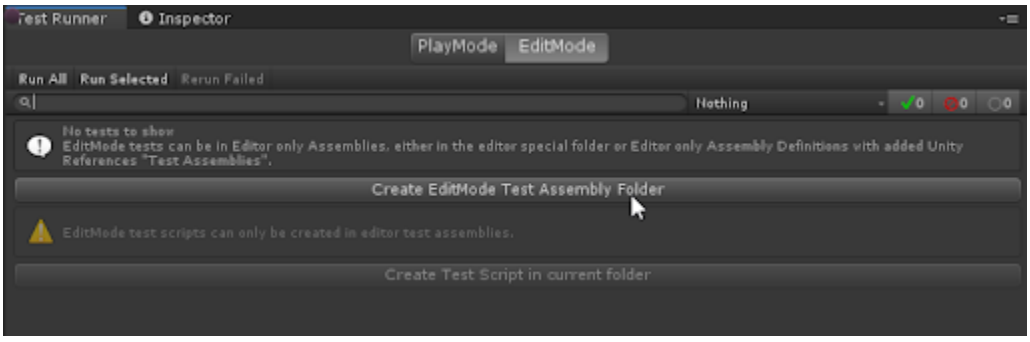
Prerequisites

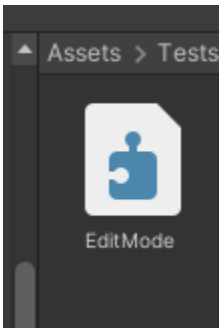
1. Create a Test folder to put all your tests in, and navigate to that folder in the Unity Project File Explorer.
2. Navigate to Window->General->Test Runner and open up the Test Runner



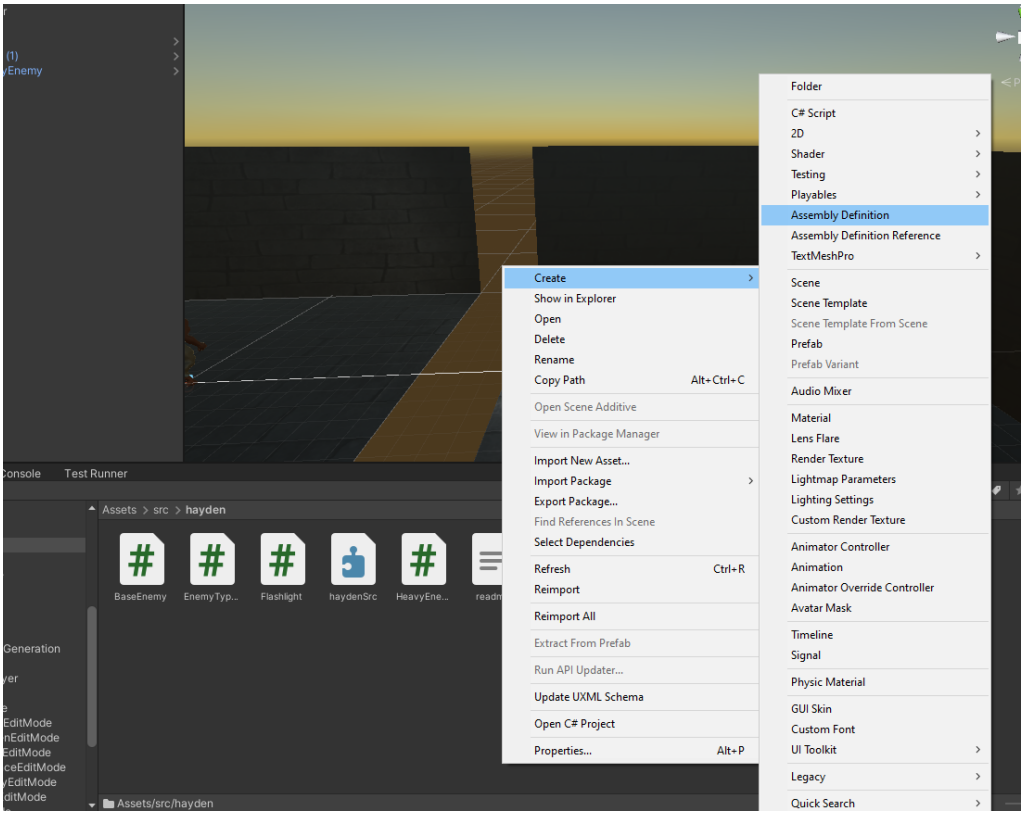
3. You should receive a prompt that looks like the following. Make sure to select either Play Mode or Edit mode at the top, depending on what test folder you would like to create. The highlighted option will be a darker gray. Hit the Create Test Assembly Folder as shown in the screenshot.



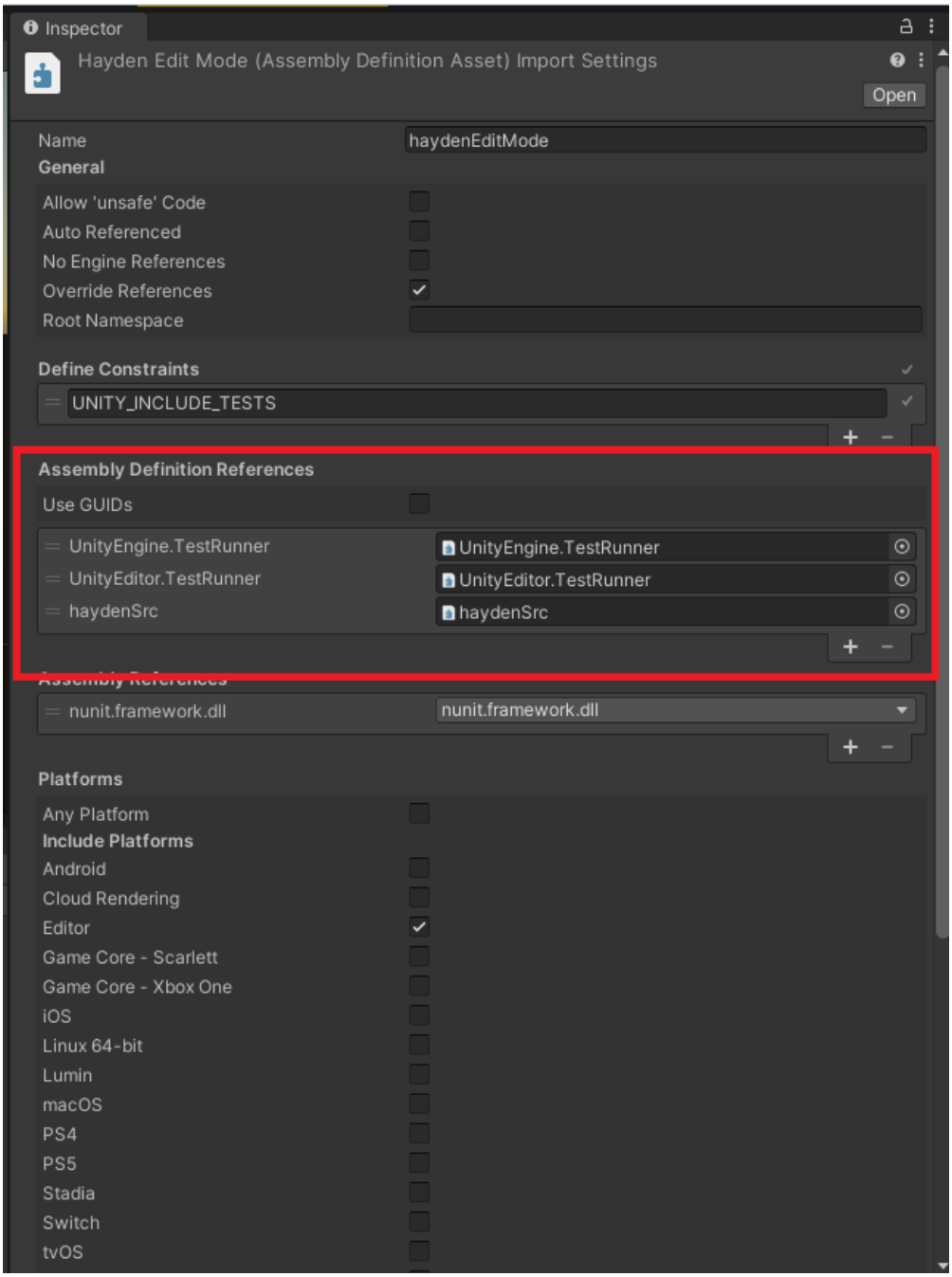
4. Rename the new folder you just created to something appropriate, either PlayMode or EditMode, depending on which option you selected.
5. Inside that folder there should be an assembly definition like so



6. Click this and add an assembly definition reference to your src folder containing all the c# scripts relevant to your tests.
    a. Before you do this, you will need to have an assembly definition reference already in your src folder. To create one, navigate to the source folder, right click to open up the create asset menu, and select assembly definition.



b. Once you have done this, navigate back to the testing folder and add to the testing assembly definition to include a reference to your source folder.



7. At this point, the setup for creating tests is complete. Note that this process is the same for both EditMode tests and PlayMode tests, so you will need to do this twice.

For any more information on setup, check this wonderful youtube video that explains everything above. https://www.youtube.com/watch?v=PDYB32qAsLU

Creating Edit Mode Tests

1. Now that you have everything set up, create a new C# script in your EditTest folder. Make sure to name the file with respect to what you are testing exactly.
2. Replace the starter code with the following code:

```
using System.Collections;
using System.Collections.Generic;
using NUnit.Framework;
using UnityEngine;
using UnityEngine.TestTools;

public class EditModeTestExample
{
    [Test]
    public void Test1()
    {
        int number1 = 5;
        int number2 = 2;
        Assert.AreEqual(number1, number3);
    }

    [Test]
    public void Test2()
    {
        Assert.IsTrue(0 <= 1);
    }

}
```

3. Take a look at different Assert statements here: https://docs.unity3d.com/ScriptReference/Assertions.Assert.html Some common ones that may help you create tests are Assert.IsTrue and Assert.AreEqual. These Assert statements will make the test either pass or fail whether they are true or not.

Creating Play Mode Tests

1. Look at creating Edit Mode tests up above. The process is the exact same, except use this starter code instead:

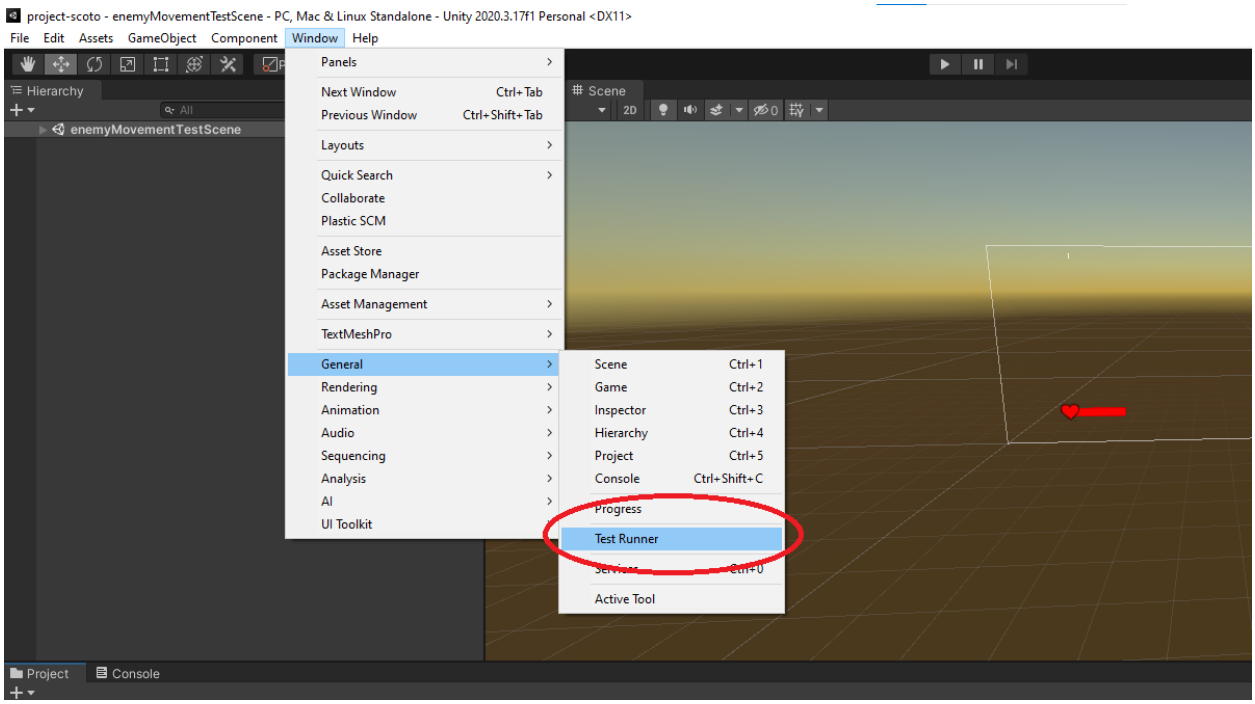
```
using System.Collections;
using System.Collections.Generic;
using NUnit.Framework;
using UnityEngine;
using UnityEngine.TestTools;
using UnityEditor;
using UnityEngine.SceneManagement;
public class PlayModeTestExample
{
    // checks if enemy moves to player properly
    [UnityTest]
    public IEnumerator EnemyCloseToPlayer()
    {
        SceneManager.LoadScene("enemyMovementTestScene"); // loads a
test scene
        yield return new WaitForSeconds(2); // wait for scene to
load

        GameObject player = GameObject.Find("Player"); // find the
Player in the scene
        GameObject enemy = GameObject.Find("HeavyEnemy(Clone)"); //
find the enemy in the scene

        yield return new WaitForSeconds(5); // wait for enemy to
move

        Vector3 playerCoords = player.transform.position;
        Vector3 enemyCoords = enemy.transform.position;
```

```
        float expectedEnemyDistance =
enemy.GetComponent<HeavyEnemy>().GetAttackRange();
        float expectedActualDiff = Mathf.Abs(expectedEnemyDistance -

Vector3.Distance(playerCoords, enemyCoords));

        Assert.IsTrue(expectedActualDiff < 1); // allowing for some
discrepancy

        yield return null;

    }
}
```

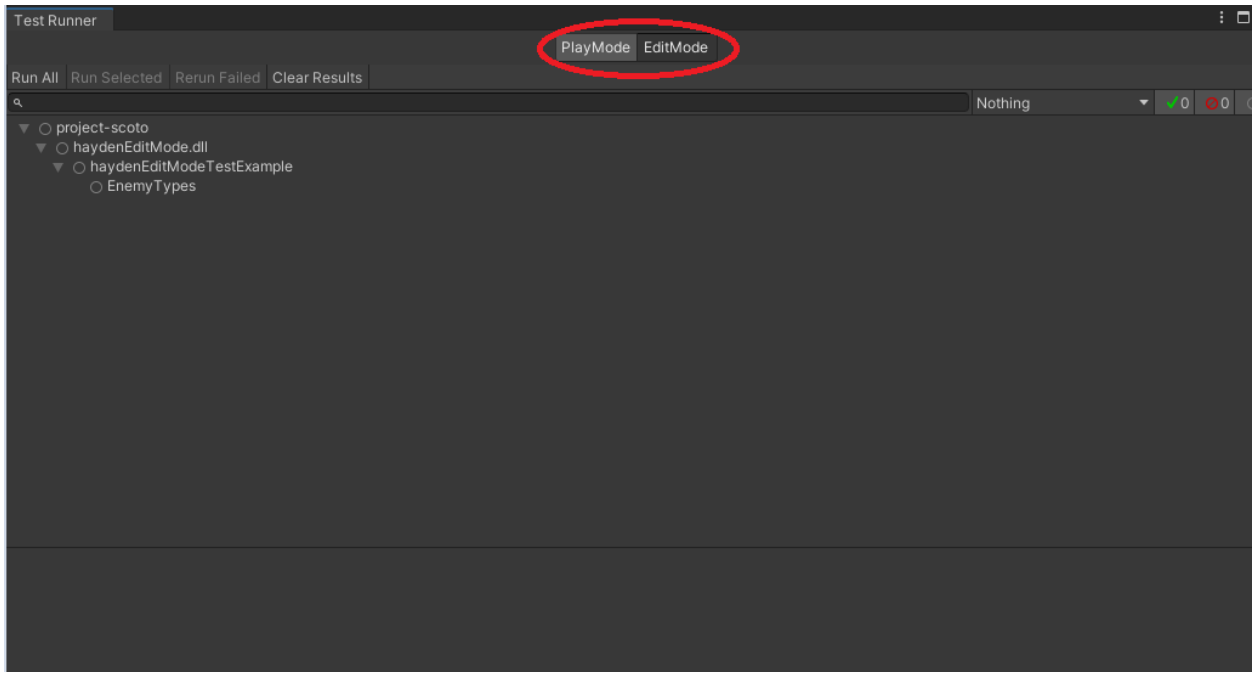
This example tests if an enemy is within bounds of the player. One very helpful way of quickly creating play mode tests is to create a test scene, specifically for testing your playmode test.

Running Edit Mode and Play Mode Tests

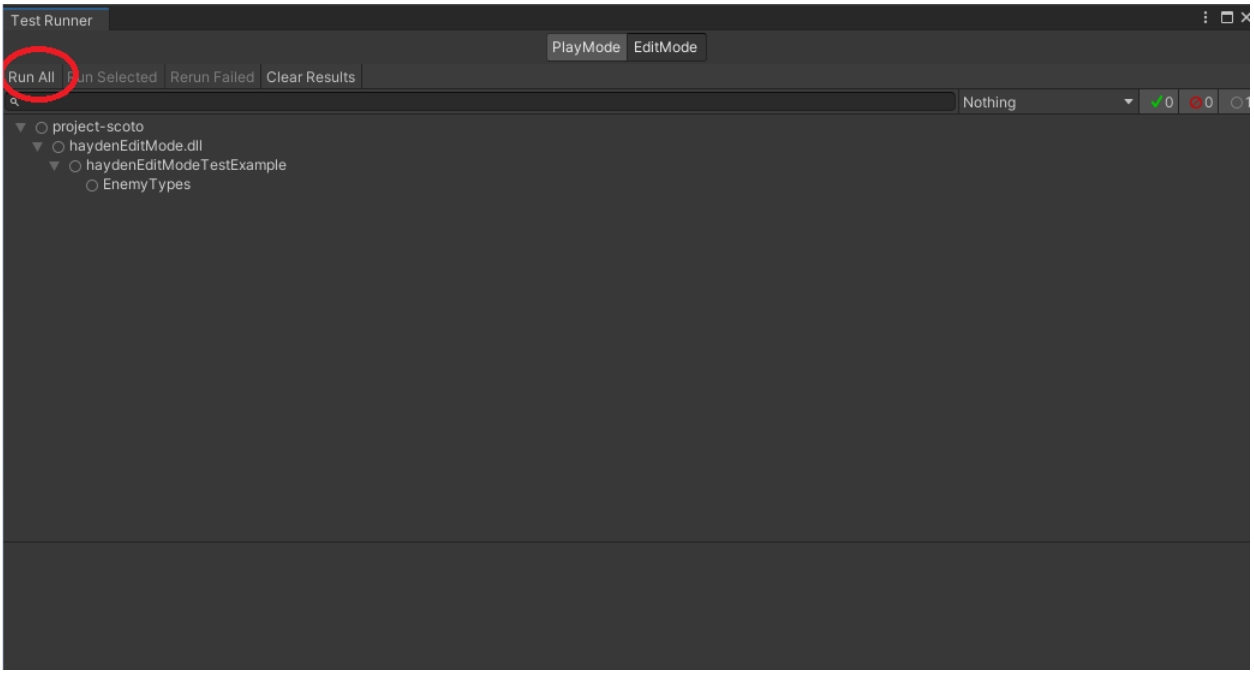
1. Open the test runner window by navigating to Window->General->Test Runner



2. To run Edit Mode Tests, make sure Edit Mode is selected. It should be a dark grey color if selected. To run Play Mode tests, make sure Play Mode is selected.



3. Once you have clicked either Play Mode or Edit Mode, click Run All in the left hand corner of the Test Runner.



GOF Design Patterns

For information about design patterns refer to the Team Lead 3 presentation or go to https://sourcemaking.com/design_patterns