

Test Case ID	Test Cases	Actual Result
T1	Verify that the game can be launched successfully on different platforms (Windows, Mac, etc).	The game was successfully launched on Windows, Mac platforms.
T2	Verify that the game can be played in multiplayer mode with at least two players.	The game was played in multiplayer mode with two players, and it worked correctly.
T3	Verify that the game's graphics and sound effects are displayed correctly.	The game's graphics and sound effects were displayed correctly.
T4	Verify that the game's controls are responsive and intuitive.	The game's controls were responsive and intuitive.
T5	Verify that the game's scoring system is working correctly.	The game's scoring system was working correctly.
T6	Verify that the game's collision detection system is working correctly.	The game's collision detection system was working correctly.
T7	Verify that the game's power-ups are working correctly.	The game's power-ups were working correctly.
T8	Verify that the game's network connectivity is stable and reliable.	The game's network connectivity was stable and reliable.

## Testing with NUnit in Unity:

NUnit is an open-source unit testing framework for the .NET Framework and Mono. It serves the same purpose as JUnit does in the Java world, and is one of many programs in the xUnit family.

NUnit provides a console runner (nunit3-console.exe), which is used for batch execution of tests. The console runner works through the NUnit Test Engine, which provides it with the ability to load, explore and execute tests. When tests are to be run in a separate process, the engine makes use of the nunit-agent program to run them.

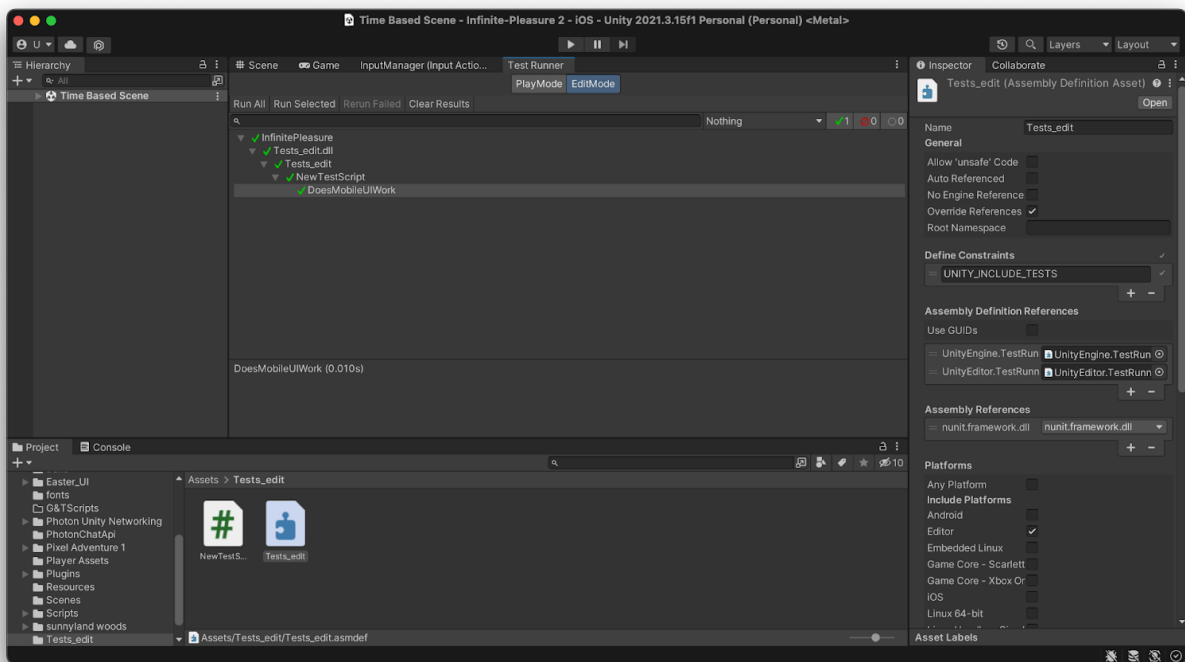
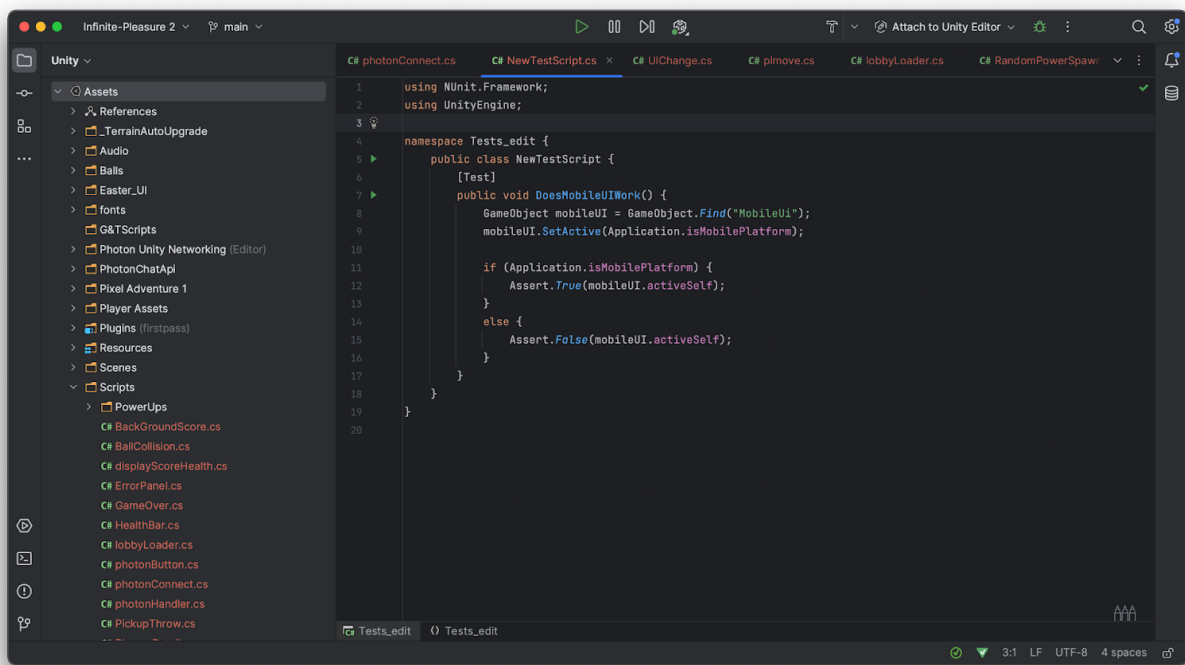
Unit provides a rich set of assertions as static methods of the Assert class. If an assertion fails, the method call does not return and an error is reported. If a test contains multiple assertions, any that follow the one that failed will not be executed. For this reason, it's usually best to try for one assertion per test.[citation needed]

Nunit 3.x is supporting multiple assertions.

```
[Test]
public void ComplexNumberTest()
{
    ComplexNumber result = SomeCalculation();

    Assert.Multiple(() =>
    {
        Assert.AreEqual(5.2, result.RealPart, "Real part");
        Assert.AreEqual(3.9, result.ImaginaryPart, "Imaginary part");
    });
}
```

## Edit mode tests



## Play mode tests

