Pass Task 6 – Unit Testing Shape

Related Learning Outcomes

ULO1 – Explain the OO Principles

This exercise demonstrated object Testing with NUnit. It taught me how to add new NUnit files for testing features for the shape game. The exercise taught me about attributes which are annotations to the class or method used my NUnit to run tests. An assertion means that the statement must be true otherwise the test would fail.

ULO2 – Use OO Language and Library

This exercise also introduced the Swingame API, showing me some exclusive methods to perform specific tasks that I had to look up to find information about them. Aswell as NUnit specific commands.

ULO3 – Design, Develop and Test using an IDE

The code was developed using Xamarin Studio and the Swingame API to build and run the program, as well as integrated debugging features to step and inspect values.

ULO4 – Communicate using UML Diagrams

I learned how to interpret a UML class diagram and write the related code. Understanding what part of the UML diagram is for fields, methods and properties.

ULO5 – Describe Elements of Good OO Design

The exercise demonstrated correct use of C# coding conventions.

Screenshots

[code running]

[use of IDE]

MyGame - ShapeTest.cs - Xamarin Studio

File   Edit   View   Search   Project   Build   Run   Version Control   Tools   Window   Help

Debug-Windows   Default   Build successful.   Press 'Control+,' to search

Solution

MyGame
  MyGame
    References
    Packages
    lib
    Resources
    src
      GameMain.cs
      packages.config
      Shape.cs
      ShapeTest.cs

GameMain.cs   Shape.cs   ShapeTest.cs

ShapeTest ▸ TestShapeWhenResized ()

```
36
37              s.X = 50;
38              s.Y = 50;
39
40              Assert.IsFalse (s.IsAt (SwinGameSDK.SwinGame.PointAt (25, 25)));
41
42          }
43
44          [Test ()]
45          public void TestShapeWhenResized ()
46          {
47              Shape s = new Shape ();
48
49              s.X = 0;
50              s.Y = 0;
51              s.Width = 50;
52              s.Height = 50;
53
54              Assert.IsTrue (s.IsAt (SwinGameSDK.SwinGame.PointAt (25, 25)));
55
56              s.Width = 5;
57              s.Height = 5;
58
59              Assert.IsFalse (s.IsAt (SwinGameSDK.SwinGame.PointAt (25, 25)));
60
61          }
62      }
63 }
64
65
```

F# Interactive

> Microsoft (R) F# Interactive version 12.0.21005.1
Copyright (c) Microsoft Corporation. All Rights Reserved.
For help type #help;;

Unit Tests

Run All

MyGame
  MyGame
    MyGame
      ShapeTest
        TestShapeAt
        TestShapeWhenMov
        TestShapeWhenResi

Test Results   Errors   Tasks   Application Output   Package Console