Selected files

18 printable files

```
Swin-Adventure/IdentifiableObjectClass.cs
Swin-Adventure/GameObject.cs
Swin-Adventure/Item.cs
Swin-Adventure/Inventory.cs
Swin-Adventure/Player.cs
Swin-Adventure/Bag.cs
Swin-Adventure/Command.cs
Swin-Adventure/IHaveInventory.cs
Swin-Adventure/Location.cs
Swin-Adventure/LookCommand.cs
Swin-Adventure/Program.cs
IdentifiableObjectTest/BagTest.cs
IdentifiableObjectTest/IdentifiableObjectTest.cs
IdentifiableObjectTest/ItemTest.cs
IdentifiableObjectTest/InventoryTest.cs
IdentifiableObjectTest/LocationTest.cs
IdentifiableObjectTest/LookCommandTest.cs
IdentifiableObjectTest/PlayerTest.cs
```

Swin-Adventure/IdentifiableObjectClass.cs

```
using System;
2
   namespace Swin_Adventure
3
   {
4
         public class IdentifiableObject
5
6
              private List<string> _identifiers;
7
8
              public IdentifiableObject(string[] idents)
9
10
                    identifiers = new List<string>(idents);
                    identifiers.AddRange(idents);
11
12
              }
13
14
              public bool AreYou(string id)
15
16
                    return _identifiers.Contains(id.ToLower());
17
18
19
              public string FirstId
20
21
                    get
22
23
                         if (_identifiers.Count == 0)
24
25
                               return "";
26
27
                         return _identifiers[0];
28
                    }
29
              }
30
31
              public void AddIdentifier(string id)
32
33
                    _identifiers.Add(id.ToLower());
34
              }
35
         }
36
   }
```

27/05/2024, 13:25

Swin-Adventure/GameObject.cs

```
using System;
   using System.Xml.Linq;
 4
   namespace Swin_Adventure
 5
 6
         public class GameObject : IdentifiableObject
 7
 8
            private string _description;
 9
            private string _name;
10
            public GameObject(string[] ids, string name, string description) : base(ids)
11
12
13
                _description = description;
14
                _name = name;
15
16
17
            public string Name
18
              {
19
                    get { return _name.ToLower(); }
20
              }
21
22
              public string ShortDescription
23
24
                get { return $"a {_name.ToLower()} ({FirstId.ToLower()})"; }
25
26
27
            public virtual string FullDescription
28
29
                    get { return _description; }
30
              }
         }
31
32
   }
33
34
```

Swin-Adventure/Item.cs

```
1
   using System;
 2
 3
   namespace Swin_Adventure
 4
 5
        public class Item : GameObject
 6
 7
            public Item(string[] idents, string name, string description) : base(idents,
    name, description)
 8
 9
10
            }
11
        }
12
    }
13
14
```

Swin-Adventure/Inventory.cs

```
1 using System;
2 namespace Swin_Adventure
```

```
public class Inventory
              private List<Item> _items;
              public Inventory()
                    _items = new List<Item>();
              public bool HasItem(string id)
                    foreach (Item itm in _items)
                         if (itm.AreYou(id))
16
17
18
                               return true;
19
20
                    }
21
                    return false;
              }
22
23
              public void Put(Item itm)
24
25
26
                    _items.Add(itm);
              }
27
28
29
            public Item Take(string id)
30
31
                 Item itm = Fetch(id);
32
33
                if (itm != null)
34
35
                     _items.Remove(itm);
36
37
38
                 return itm;
39
            }
40
            public Item Fetch(string id)
41
42
              {
43
                    foreach (Item itm in _items)
44
45
                         if (itm.AreYou(id))
46
47
                               return itm;
48
49
50
                    return null;
51
              }
52
53
              public string ItemList
54
55
                    get
56
57
                         string list = "";
58
                     foreach (Item item in _items)
59
                         list += "\t" + "a " + item.Name + " (" + item.FirstId + ")\n";
60
61
62
                     return list;
```

Swin-Adventure/Player.cs

```
using System;
 2
   using Swin_Adventure;
 3
 4
   namespace Swin Adventure
 5
 6
        public class Player : GameObject, IHaveInventory
 7
 8
            private Inventory _inventory;
 9
            private Location _location;
10
11
            public Player(string name, string desc, Location location)
                : base(new string[] { "player", "me", "inventory", name }, name, desc)
12
            {
13
                _inventory = new Inventory();
14
                _location = location;
15
16
17
            public GameObject Locate(string id)
18
19
20
                if (AreYou(id))
21
                {
22
                    return this;
23
24
                GameObject item = _inventory.Fetch(id);
25
                if (item != null)
26
                {
27
                    return item;
28
29
                if (_location != null)
30
31
                    item = _location.Locate(id);
32
                    if (item != null)
33
34
                         return item;
35
36
37
                return null;
38
            }
39
40
            public override string FullDescription
41
                get { return "You are " + Name + ", " + base.FullDescription + ".\n" +
42
                            "You are carrying:\n" + Inventory.ItemList; ; }
43
44
            }
45
46
            public Inventory Inventory
47
48
                get { return _inventory; }
49
50
            public Location Location
51
```

```
27/05/2024, 13:25
              {
52
53
                   get { return _location; }
54
                   set { _location = value; }
55
56
         }
57
    }
58
```

Swin-Adventure/Bag.cs

```
using System;
 2
   namespace Swin_Adventure
 3
 4
        public class Bag : Item, IHaveInventory
 5
 6
            private Inventory _inventory;
 7
 8
            public Bag(string[] ids, string name, string description) : base(ids, name,
    description)
 9
            {
10
                _inventory = new Inventory();
11
            }
12
13
            public GameObject Locate(string id)
14
                if (this.AreYou(id))
15
16
17
                    return this;
18
19
                return _inventory.Fetch(id);
20
            }
21
22
            public override string FullDescription
23
24
                get { return $"In the {Name} you can see:\n" + _inventory.ItemList; }
25
26
27
            public Inventory Inventory
28
29
                get { return _inventory; }
30
31
        }
32
   }
33
34
```

Swin-Adventure/Command.cs

```
1
   using System;
 2
   namespace Swin_Adventure
 3
 4
        public abstract class Command : IdentifiableObject
 5
 6
            public Command(string[] ids) : base(ids)
 7
            {
            }
 8
 9
             public abstract string Execute(Player player, string[] text);
10
        }
11
   }
12
13
```

Swin-Adventure/IHaveInventory.cs

```
1
    using System;
 2
   namespace Swin_Adventure
 3
 4
        public interface IHaveInventory
 5
 6
            GameObject Locate(string id);
 7
            string Name { get; }
 8
        }
 9
    }
10
11
```

Swin-Adventure/Location.cs

```
1
   using System;
    using System.Collections.Generic;
 3
 4
   namespace Swin Adventure
 5
 6
        public class Location : GameObject, IHaveInventory
 7
 8
            private Inventory _inventory;
 9
            public Location(string [] idents, string name, string description) :
10
    base(idents, name, description)
11
            {
12
                _inventory = new Inventory();
            }
13
14
15
            public Inventory Inventory
16
17
                get { return _inventory; }
18
19
20
            public GameObject Locate(string id)
21
22
                if (AreYou(id))
23
                {
24
                     return this;
25
26
                return _inventory.Fetch(id);
27
28
            }
29
30
            public override string FullDescription
31
            {
32
                get
33
                 {
                     return "You are at the " + Name + ". " + base.FullDescription + "\n"
34
35
                            "You can see:\n" + Inventory.ItemList;
36
                }
37
            }
        }
38
39
    }
40
41
```

Swin-Adventure/LookCommand.cs

```
using System;
 2
 3
   namespace Swin_Adventure
 4
 5
        public class LookCommand : Command
 6
 7
            public LookCommand() : base(new string[] { "look" }) { }
 8
 9
            public override string Execute(Player player, string[] text)
10
11
12
                if (text.Length == 1 && text[0].ToLower() == "look")
13
                     return player.Location != null ? player.Location.FullDescription : "
14
   There is no location to look at.";
15
                }
16
17
18
                if (text.Length != 3 && text.Length != 5)
19
20
                     return "I don't know how to look like that";
21
                }
22
                if (text[0].ToLower() != "look")
23
24
25
                     return "Error in look input";
26
                }
27
28
                if (text[1].ToLower() != "at")
29
30
                     return "What do you want to look at?";
31
                }
32
33
                IHaveInventory container;
34
35
36
                if (text.Length == 3)
37
38
                    container = player;
39
40
                else if (text.Length == 5 && text[3].ToLower() == "in")
41
42
                    GameObject found = player.Locate(text[4].ToLower());
43
                    if (found == null)
44
45
                         return $"I can't find the {text[4]}";
46
                    }
47
48
                    container = found as IHaveInventory;
49
50
                    if (container == null)
51
52
                         return $"The {text[4]} does not contain items";
53
54
                }
55
                else
56
57
                     return "What do you want to look in?";
58
                }
59
```

```
Console.WriteLine("Welcome to SwinAdventure, designed by Thuan!");
12
            Console.WriteLine("Enter Player Name: ");
13
            string playerName = Console.ReadLine();
14
15
            Console.WriteLine("Enter your description: ");
16
            string playerDescription = Console.ReadLine();
17
18
19
            Location startingLocation = new Location(new string[] { "jungle" }, "Jungle",
   "Big wood jungle");
20
21
22
            Player player = new Player(playerName, playerDescription, startingLocation);
23
            Item item1 = new Item(new string[] { "weapon" }, "sword", "this is an
24
   Excalibur");
25
            Item item2 = new Item(new string[] { "armor" }, "shield", "this is a shield")
26
            player.Inventory.Put(item1);
27
            player.Inventory.Put(item2);
28
29
            Bag bag = new Bag(new string[] { "bag" }, "bag", "This is a bag.");
30
            player.Inventory.Put(bag);
31
            Item itemInBag = new Item(new string[] { "gem" }, "ruby", "This is a
32
   beautiful gem");
33
            bag.Inventory.Put(itemInBag);
34
35
            bool exitRequested = false;
36
37
            while (!exitRequested)
38
            {
```

```
Console.WriteLine("Enter a command (or type 'exit' to quit):");
39
40
                string input = Console.ReadLine();
41
                string[] inputArray = input.Split(' ');
42
43
                if (inputArray.Length > 0)
44
                {
45
                    string command = inputArray[0].ToLower();
46
                    if (command == "exit" || command == "quit")
47
48
49
                         exitRequested = true;
50
                    }
51
                    else
52
                    {
53
                        LookCommand lookCommand = new LookCommand();
54
                         string result = lookCommand.Execute(player, inputArray);
55
                         Console.WriteLine(result);
56
                    }
57
                }
            }
58
59
        }
60
61
62
```

IdentifiableObjectTest/BagTest.cs

```
using System;
 1
 2
   namespace Swin Adventure
 3
 4
         [TestFixture]
 5
         public class BagTest
 6
 7
              private Bag _bagTest1;
 8
            private Bag bagTest2;
 9
            private Item _weaponTest;
10
            private Item _armorTest;
11
12
            [SetUp]
13
            public void SetUp()
14
15
                   _bagTest1 = new Bag(new string[] { "bag1" }, "backpack", "It's
    spacious");
                _bagTest2 = new Bag(new string[] { "bag2" }, "suitcase", "It's compact");
16
                _weaponTest = new Item(new string[] { "weapon" }, "sword", "this is an
17
    Excalibur"):
                _armorTest = new Item(new string[] { "armor" }, "shield", "this is a
18
    shield");
19
20
                _bagTest1.Inventory.Put(_bagTest2);
21
                _bagTest1.Inventory.Put(_weaponTest);
                _bagTest2.Inventory.Put(_armorTest);
22
            }
23
24
25
            [Test]
26
            public void TestBagLocatesItems()
27
            {
28
               Assert.AreSame(_weaponTest, _bagTest1.Locate("weapon"));
29
            }
30
            [Test]
31
```

```
32
            public void TestBagLocatesitself()
33
34
               Assert.AreSame(_bagTest1, _bagTest1.Locate("bag1"));
35
            }
36
37
            [Test]
38
            public void TestBagLocatesnothing()
39
40
               Assert.IsNull( bagTest1.Locate("bag3"));
41
            }
42
43
            [Test]
44
            public void TestBagFullDescription()
45
46
               Assert.AreEqual("In the backpack you can see:\n\ta suitcase (bag2)\n\ta
    sword (weapon)\n", _bagTest1.FullDescription);
47
48
49
            [Test]
50
            public void TestBaginBag()
51
            {
52
                Assert.AreSame(_bagTest2, _bagTest1.Locate("bag2"));
53
                Assert.AreSame(_weaponTest, _bagTest1.Locate("weapon"));
54
                Assert.IsNull(_bagTest1.Locate("armor"));
55
            }
56
        }
57
   }
58
59
```

IdentifiableObjectTest/IdentifiableObjectTest.cs

```
using NUnit.Framework;
   using Swin_Adventure;
 3
 4
   namespace IdentifiableObjectTest
 5
   {
 6
 7
        internal class Tests
 8
 9
            private IdentifiableObject _test1;
10
            private IdentifiableObject _test2;
11
            private IdentifiableObject _test3;
12
            private IdentifiableObject _test4;
13
            private IdentifiableObject _test5;
14
            private IdentifiableObject _test6;
15
16
            [SetUp]
17
            public void Setup()
18
                _test1 = new IdentifiableObject(new string[] { "fred", "bob" });
19
                _test2 = new IdentifiableObject(new string[] { "fred", "bob" });
20
                _test3 = new IdentifiableObject(new string[] { "fred", "bob" });
21
                _test4 = new IdentifiableObject(new string[] { "fred", "bob" });
22
23
                _test5 = new IdentifiableObject(new string[] { });
24
                _test6 = new IdentifiableObject(new string[] { "fred", "bob" });
            }
25
26
27
            [Test]
28
            public void TestAreYou()
29
```

```
30
                Assert.IsTrue(_test1.AreYou("fred"));
31
                Assert.IsTrue(_test1.AreYou("bob"));
32
            }
33
34
            [Test]
            public void TestNotAreYou()
35
36
37
                Assert.IsFalse(_test2.AreYou("wilma"));
38
                Assert.IsFalse( test2.AreYou("boby"));
39
            }
40
41
            [Test]
42
            public void TestCaseSensitive()
43
44
                Assert.IsTrue( test3.AreYou("FRED"));
45
                Assert.IsTrue(_test3.AreYou("b0B"));
            }
46
47
48
            [Test]
            public void TestFirstID()
49
50
51
                Assert.AreEqual("fred", _test4.FirstId);
52
            }
53
54
            [Test]
55
            public void TestFirstIdWithNoIDs()
56
                Assert.AreEqual("", _test5.FirstId);
57
            }
58
59
60
            [Test]
            public void TestAddID()
61
62
                _test6.AddIdentifier("wilma");
63
64
                Assert.IsTrue(_test6.AreYou("fred"));
                Assert.IsTrue(_test6.AreYou("bob"));
65
                Assert.IsTrue(_test6.AreYou("wilma"));
66
            }
67
        }
68
69
   }
70
71
```

IdentifiableObjectTest/ItemTest.cs

```
using System;
 2
   using Swin_Adventure;
 3
 4
   namespace SwinAdventureTest
 5
 6
        [TestFixture]
 7
        public class ItemTest
 8
         {
 9
              private Item _itemTest;
10
11
12
            [SetUp]
13
            public void Setup()
14
   _itemTest = new Item(new string[] { "weapon" }, "sword", "This is an Excalibur");
15
```

```
27/05/2024, 13:25
16
             }
17
18
19
             [Test]
20
             public void TestItemIsIdentifiable()
21
                 Assert.IsTrue(_itemTest.AreYou("weapon"));
22
23
24
25
             [Test]
             public void TestShortDescription()
26
27
28
                 Assert.IsTrue(_itemTest.ShortDescription == "a sword (weapon)");
29
30
             [Test]
31
32
             public void TestFullDescription()
33
34
                 Assert.IsTrue(_itemTest.FullDescription == "This is an Excalibur");
             }
35
36
         }
37
38
39
```

IdentifiableObjectTest/InventoryTest.cs

```
using System;
 1
 2
   using Swin Adventure;
 3
 4
   namespace SwinAdventureTest
 5
 6
        [TestFixture]
 7
        public class InventoryTest
 8
         {
 9
              private Inventory _inventoryTest;
10
              private Item _weaponTest;
              private Item _armorTest;
11
12
13
              [SetUp]
14
              public void SetUp()
15
                   _inventoryTest = new Inventory();
16
17
                   _weaponTest = new Item(new string[] { "weapon" }, "sword", "this is an
   Excalibur");
                _armorTest = new Item(new string[] { "armor" }, "shield", "this is a
18
    shield");
19
                _inventoryTest.Put(_weaponTest);
20
21
                _inventoryTest.Put(_armorTest);
            }
22
23
24
            [Test]
25
            public void TestFindItem()
26
            {
27
                Assert.IsTrue(_inventoryTest.HasItem("weapon"));
28
                Assert.IsTrue(_inventoryTest.HasItem("armor"));
29
            }
30
31
            [Test]
32
            public void TestNoItemFind()
```

```
27/05/2024, 13:25
            {
33
34
                 Assert.IsFalse(_inventoryTest.HasItem("axe"));
35
                 Assert.IsFalse(_inventoryTest.HasItem("helmet"));
            }
36
37
38
            [Test]
39
            public void TestFetchItem()
40
41
                 Assert.IsTrue( weaponTest == inventoryTest.Fetch("weapon"));
42
                 Assert.IsTrue( inventoryTest.HasItem("weapon"));
43
44
                 Assert.IsTrue( armorTest == inventoryTest.Fetch("armor"));
45
                 Assert.IsTrue(_inventoryTest.HasItem("armor"));
            }
46
47
            [Test]
48
            public void TestTakeItem()
49
50
            {
51
                 Assert.IsTrue(_weaponTest == _inventoryTest.Take("weapon"));
52
                 Assert.IsFalse( inventoryTest.HasItem("weapon"));
53
54
                 Assert.IsTrue(_armorTest == _inventoryTest.Take("armor"));
55
                 Assert.IsFalse(_inventoryTest.HasItem("armor"));
56
            }
57
58
            [Test]
59
            public void TestItemList()
60
                 Assert.IsTrue(_inventoryTest.ItemList.Replace("\t", "") == "a sword
61
    (weapon)\na shield (armor)\n");
62
        }
63
64
    }
65
66
```

IdentifiableObjectTest/LocationTest.cs

```
1 using NUnit.Framework;
   using Swin_Adventure;
   using System;
   namespace SwinAdventureTest
 4
 5
 6
        public class LocationTest
 7
 8
            public Player _playerTest;
 9
            public Location _locationTest;
10
            public Item _itemTest;
11
12
            [SetUp]
13
            public void Setup()
14
                 locationTest = new Location(new string[] { "jungle" }, "Jungle", "Big
15
   wood jungle"):
                _playerTest = new Player("thuan", "dan choi", _locationTest);
16
                _itemTest = new Item(new string[] { "sword" }, "a sword", "This is a
17
    sword");
                _playerTest.Location = _locationTest;
18
19
                _locationTest.Inventory.Put(_itemTest);
20
            }
21
```

```
22
            [Test]
            public void TestLocationItSelf()
23
24
25
26
                Assert.IsTrue( locationTest.AreYou("location"));
            }
27
28
29
            [Test]
            public void TestLocationNotItSelf()
30
31
32
33
                Assert.IsFalse( locationTest.AreYou("NotForest"));
            }
34
35
            [Test]
36
37
            public void TestPlayerHasLocation()
38
39
40
                Assert.AreEqual(_playerTest.Locate("location"), _locationTest);
            }
41
42
43
            [Test]
44
            public void TestLocationLocateItem()
45
46
47
                Assert.AreEqual(_itemTest, _locationTest.Locate("sword"));
48
49
50
            [Test]
51
            public void TestLocationLocateNoItem()
52
53
54
                Assert.AreNotEqual(_itemTest, _locationTest.Locate("spear"));
            }
55
56
57
        }
58 }
```

IdentifiableObjectTest/LookCommandTest.cs

27/05/2024, 13:25

```
1
   using NUnit.Framework;
   using System.Numerics;
 3
   using Swin_Adventure;
 4
 5
   namespace SwinAdventureTest
 6
 7
        [TestFixture]
 8
        public class TestLookCommand
 9
10
            private LookCommand _lookCommandTest;
11
            private Player _playerTest;
12
            private Bag _bagTest;
13
            private Item _gemTest;
14
            private Location _locationTest;
15
            [SetUp]
16
17
            public void Setup()
18
   locationTest = new Location(new string[] { "jungle" }, "Jungle", "Big
wood jungle");
19
20
                _lookCommandTest = new LookCommand();
```

27/05/2024, 13:25 >a href="htt

```
_playerTest = new Player("thuan", "dan choi",
21
                                                                _locationTest);
                 bagTest = new Bag(new string[] { "duffelbag" }, "duffelbag", "it's
22
    small-sized");
                _gemTest = new Item(new string[] { "gem" }, "gem", "a beautiful gem");
23
24
            }
25
26
            [Test]
27
            public void TestLookAtMe()
28
29
                Assert.That(_lookCommandTest.Execute(_playerTest, new string[] { "look",
    "at", "inventory" }), Is EqualTo("You are thuan, dan choi.\nYou are carrying:\n"));
30
31
32
            [Test]
33
            public void TestLookAtGem()
34
35
                _playerTest.Inventory.Put(_gemTest);
36
37
                Assert.That(_lookCommandTest.Execute(_playerTest, new string[] { "look",
          "gem"
                }), Is.EqualTo("a beautiful gem"));
38
39
40
            [Test]
41
            public void TestLookAtUnk()
42
43
                Assert.That(_lookCommandTest.Execute(_playerTest, new string[] { "look",
    "at", "unknown" }), Is.EqualTo("I can't find the unknown"));
44
45
46
            [Test]
47
            public void TestLookAtGemInMe()
48
49
                _playerTest.Inventory.Put(_gemTest);
50
51
                Assert.That(_lookCommandTest.Execute(_playerTest, new string[] { "look",
          "gem",
                 "in", "inventory" }), Is.EqualTo("a beautiful gem"));
52
            }
53
54
            [Test]
55
            public void TestLookAtGemInBag()
56
57
                _bagTest.Inventory.Put(_gemTest);
58
                _playerTest.Inventory.Put(_bagTest);
59
60
                Assert.That(_lookCommandTest.Execute(_playerTest, new string[] { "look",
          "gem",
                 "in", "duffelbag" }), Is.EqualTo("a beautiful gem"));
            }
61
62
63
            [Test]
64
            public void TestLookAtGemInNoBag()
65
                66
          "gem",
            }
67
68
69
            [Test]
70
            public void TestLookAtNoGemInBag()
71
72
                _playerTest.Inventory.Put(_bagTest);
73
                Assert.That(_lookCommandTest.Execute(_playerTest, new string[] { "look",
   "in", "duffelbag" }), Is.EqualTo("I can't find the gem"));
74
```

```
27/05/2024, 13:25
75
                 }
76
77
                 [Test]
78
                 public void TestInvalidLook()
79
     Assert.That(_lookCommandTest.Execute(_playerTest, new string[] { "look", "around" }), Is.EqualTo("I don't know how to look like that"));
80
                      Assert.That(_lookCommandTest.Execute(_playerTest, new string[] { "hello"
81
      }), Is.EqualTo("I don't know how to look like that"));
                    Assert.That(_lookCommandTest.Execute(_playerTest, new string[] { "look",
"at", "b" }), Is.EqualTo("What do you want to look in?"));
82
     Assert.That(_lookCommandTest.Execute(_playerTest, new string[] { "hello", "at", "a" }), Is.EqualTo("Error in look input"));
83
     Assert.That(_lookCommandTest.Execute(_playerTest, new string[] { "look", "by", "a" }), Is.EqualTo("What do you want to look at?"));
84
85
           }
86
87
     }
88
```

IdentifiableObjectTest/PlayerTest.cs

```
using System;
 2
   using System.Numerics;
 3
   using Swin Adventure;
 4
 5
   namespace SwinAdventureTest
 6
    {
 7
        [TestFixture]
 8
        public class PlayerTest
 9
            public Player _playerTest;
10
11
            public Item _weaponTest;
12
            public Item _armorTest;
13
            public Location _locationTest;
14
15
            [SetUp]
16
            public void Setup()
17
18
19
                  locationTest = new Location(new string[] { "jungle" }, "Jungle", "Big
   wood jungle");
                _playerTest = new Player("thuan", "dan choi", _locationTest);
20
   _weaponTest = new Item(new string[] { "weapon" }, "sword", "this is an Excalibur");
21
                _armorTest = new Item(new string[] { "armor" }, "shield", "this is a
22
    shield");
23
24
25
26
27
                _playerTest.Inventory.Put(_weaponTest);
28
                _playerTest.Inventory.Put(_armorTest);
29
            }
30
31
            [Test]
32
            public void TestPlayerIsIdentifiable()
33
            {
34
                Assert.IsTrue(_playerTest.AreYou("me"));
35
                Assert.IsTrue(_playerTest.AreYou("inventory"));
36
            }
37
```

```
38
             [Test]
             public void TestPlayerLocateItems()
39
40
41
                 Assert.IsTrue( playerTest.Locate("weapon") == weaponTest);
42
                 Assert.IsTrue(_playerTest.Locate("armor") == _armorTest);
43
44
                 Assert.IsTrue(_playerTest.Inventory.HasItem("weapon"));
45
                 Assert.IsTrue(_playerTest.Inventory.HasItem("armor"));
             }
46
47
48
             [Test]
49
             public void TestPlayerLocateItself()
50
                 Assert.IsTrue(_playerTest == _playerTest.Locate("me"));
51
52
                 Assert.IsTrue( playerTest == playerTest.Locate("inventory"));
53
             }
54
55
             [Test]
56
             public void TestPlayerLocateNothing()
57
58
                 Assert.IsTrue( playerTest.Locate("helmet") == null);
59
             }
60
61
             [Test]
62
             public void TestPlayerFullDescription()
63
    Assert.IsTrue(_playerTest.FullDescription == "You are thuan, dan choi.\nYou are carrying:\n\ta sword (weapon)\n\ta shield (armor)\n");
64
65
66
67
             [Test]
68
             public void TestPlayerLocateItemInLocation()
69
70
71
                 Assert.AreSame(_weaponTest, _playerTest.Locate("sword"));
             }
72
73
74
             [Test]
75
             public void TestPlayerLocateNoItemInLocation()
76
             {
77
78
                 Assert.IsNull(_playerTest.Locate("helmet"));
79
             }
80
        }
81
    }
82
83
```