SWINBURNE UNIVERSITY OF TECHNOLOGY

COS20007 OBJECT ORIENTED PROGRAMMING

Case Study - Iteration 6 - Locations

PDF generated at 10:39 on Friday $3^{\rm rd}$ November, 2023

File 1 of 10 Location class

```
namespace CaseStudy
2
3
        public class Location : GameObject, IHaveInventory
        {
5
            private Inventory _container;
6
            public Location(string[] idents, string name, string desc) : base(idents,
       name, desc)
                _container = new();
            }
10
11
            public GameObject Locate(string id)
12
13
                if (this.AreYou(id)) return this;
                GameObject locateResult = _container.Fetch(id);
                return locateResult;
16
            }
17
18
            public override string FullDescription
19
                get
21
                {
22
                     return "You are at: " + base.ShortDescription + "\nItems at this
23
       location:\n" + _container.ItemList;
                }
24
            }
25
26
            public Inventory Container
27
28
                get
29
                {
30
                     return _container;
32
            }
33
        }
34
   }
35
```

File 2 of 10 Location tests

```
using CaseStudy;
   namespace CaseStudyTest
3
   {
       public class LocationTest
5
6
            private Player _p;
            private Location _loca;
            private Item _sword;
10
            [SetUp]
11
            public void Setup()
12
13
                _p = new Player("Tran", "This is Vu Duc Tran");
14
                _loca = new Location(new string[] { "place1" }, "University", "Swinburne
15
       University");
                _sword = new Item(new string[] { "sword" }, "a sword", "This is a
16
       sword");
17
                _p.Location = _loca;
18
                _loca.Container.Put(_sword);
            }
20
21
            [Test]
22
            public void IdentifyLocation()
23
                Assert.That(_loca.AreYou("place1"), Is.EqualTo(true), "Test Identify
25
       Location");
            }
26
27
            [Test]
28
            public void TestLocationLocateItem()
29
30
                Assert.That(_loca.Locate("sword"), Is.EqualTo(_sword), "Test Identify
31
       Location");
            }
32
33
            [Test]
34
            public void PlayerLocateItemsInLocation()
35
36
                Assert.That(_p.Location.Locate("sword"), Is.EqualTo(_sword), "Test Player
37
       Locate Items In Location");
38
        }
39
   }
40
```

File 3 of 10 Player class

```
namespace CaseStudy
2
        public class Player : GameObject, IHaveInventory
            private Inventory _inventory;
5
            private Location _location;
6
            public Player(string name, string desc) : base(new string[] { "me",
        "inventory" }, name, desc)
            {
                 _inventory = new Inventory();
10
11
12
            public GameObject Locate(string id)
13
            {
14
                 if (AreYou(id))
                 {
16
                     return this;
17
18
                 GameObject obj = _inventory.Fetch(id);
19
                 if (obj != null)
                 {
21
                     return obj;
22
23
                 if (_location != null)
24
25
                     obj = _location.Locate(id);
26
                     return obj;
                 }
28
                 else
29
30
                     return null;
31
            }
33
34
            public override string FullDescription
35
36
                 get
38
                     return $"You are {Name}, {base.FullDescription}.\nYou are
39
        carrying:\n{_inventory.ItemList}";
40
            }
41
42
            public Inventory Inventory
44
                 get
45
46
                     return _inventory;
47
                 }
            }
49
50
            public Location Location
51
```

File 3 of 10 Player class

```
{
52
                  get
53
                  {
54
                       return _location;
55
                  }
56
                  set
57
                  {
58
                       _location = value;
59
                  }
60
             }
61
         }
62
    }
63
```

File 4 of 10 Player tests

```
using CaseStudy;
1
2
   namespace CaseStudyTest
3
   {
        public class PlayerTest
5
6
            private Player _player;
            private Item _shovel;
            private Item _sword;
            private Location _loc;
10
11
            [SetUp]
12
            public void Setup()
13
                _player = new Player("Vu Duc Tran", "Swinburne Student");
15
                _shovel = new Item(new string[] { "shovel" }, "a shovel", "This is a
        shovel");
                 _sword = new Item(new string[] { "sword" }, "a sword", "This is a
17
        Sword");
                 _loc = new Location(new string[] { "school" }, "School", "Swinburne
18
        University");
19
                _player.Inventory.Put(_sword);
20
                _player.Location = _loc;
21
                _loc.Container.Put(_shovel);
22
            }
23
24
            [Test]
25
            public void TestPlayerIdentifiable()
26
27
                Assert.That(_player.AreYou("me"), Is.EqualTo(true), "Test player
28
        identifiable");
                Assert.That(_player.AreYou("inventory"), Is.EqualTo(true), "Test player
        identifiable");
            }
30
31
            [Test]
32
            public void TestPlayerLocateItems()
33
34
                Assert.That(_player.Locate("sword"), Is.EqualTo(_sword), "Test player
35
       locate items");
            }
36
37
            [Test]
38
            public void TestPlayerLocateItself()
40
                Assert.That(_player.Locate("me"), Is.EqualTo(_player), "Test player
41
        locate it self");
                Assert.That(_player.Locate("inventory"), Is.EqualTo(_player), "Test
42
        player locate it self");
            }
43
44
            [Test]
45
```

File 4 of 10 Player tests

```
public void TestPlayerLocateNothing()
46
            {
47
                Assert.That(_player.Locate("mirror"), Is.EqualTo(null), "Test player
48
       locate nothing");
            }
49
50
            [Test]
51
            public void TestPlayerFullDescription()
52
53
                Assert.That(_player.FullDescription, Is.EqualTo($"You are Vu Duc Tran,
       Swinburne Student.\nYou are carrying:\n{_player.Inventory.ItemList}"), "Test
       player full descrition");
            }
55
56
            [Test]
57
            public void PlayerLocateItemsInLocation()
59
                Assert.That(_player.Location.FullDescription, Is.EqualTo("You are at:
60
       School (school)\nItems at this location:\na shovel (shovel)\n"));
61
        }
62
   }
63
```

File 5 of 10 LookCommand class

```
using System;
   namespace CaseStudy
        public class LookCommand : Command
        {
5
            public LookCommand(string[] ids) : base(ids)
6
                AddIdentifier("look");
            }
10
            public override string Execute(Player p, string[] text)
11
12
                 string ItemToFind;
13
                 string PlaceToLookIn;
                 string ItemString;
15
                 IHaveInventory Container;
17
                 if (text.Length == 1 && text[0] == "look")
18
19
20
                     ItemString = p.Location.FullDescription;
                     return ItemString;
22
                }
23
24
                 if (text.Length != 3 && text.Length != 5)
25
26
                     return "I don't know how to look like that\n";
27
                }
29
                 if (text[0] != "look")
30
31
                     return "Error in look input\n";
32
                 }
34
                   (text[1] != "at")
                 if
35
36
                     return "What do you want to look at?\n";
37
                 }
38
39
                 if (text.Length == 5)
40
41
                     if (text[3] != "in")
42
43
                         return "What do you want to look in?\n";
                     }
                }
46
47
                    (text.Length == 3)
48
                 {
49
                     ItemToFind = text[2];
50
                     return LookAtIn(ItemToFind, p as IHaveInventory);
51
                 }
52
53
```

File 5 of 10 LookCommand class

```
if (text.Length == 5)
54
55
                    ItemToFind = text[2];
56
                    PlaceToLookIn = text[4];
                    Container = FetchContainer(p, PlaceToLookIn);
58
                    if (Container is null)
60
                         return $"I can't find the {PlaceToLookIn}\n";
61
                    return LookAtIn(ItemToFind, Container);
63
                return "Vu Duc Tran - Swinburne Student";
65
            }
66
67
68
            private IHaveInventory FetchContainer(Player p, string containerId)
70
                return p.Locate(containerId) as IHaveInventory;
72
73
            private string LookAtIn(string thingId, IHaveInventory container)
                if (container.Locate(thingId) != null)
76
77
                    return $"{container.Locate(thingId).FullDescription}\n";
78
79
                return $"I can't find the {thingId}\n";
            }
81
        }
82
   }
83
84
```

File 6 of 10 LookCommand tests

```
using System;
   using CaseStudy;
2
   namespace CaseStudyTest
   {
5
       public class LookCommandTest
6
            private LookCommand _look;
            private Player _player1, _player2;
            private Bag _bag;
            private Location _myhouse;
12
            private Item _gem;
13
            private Item _pen;
            private string _output;
15
            [SetUp]
17
            public void Setup()
18
19
                _gem = new Item(new string[] { "gem" }, "a gem", "This is a gem");
20
                _pen = new Item(new string[] { "pen" }, "a pen", "This is a pen");
                _look = new LookCommand(new string[] { "look" });
22
                _player1 = new Player("Vu Duc Tran 104175614 ", "Swinburne Student");
23
                _player1.Inventory.Put(_bag);
24
                _player2 = new Player("Dylan Tran", "student");
25
26
                _bag = new Bag(new string[] { "bag" }, "Duc's bag", $"This is
27
        {_player1.FirstID} bag");
                _myhouse = new Location(new string[] { "house" }, "My House", "My
28
       House");
29
                _player1.Inventory.Put(_gem);
30
                _bag.Inventory.Put(_gem);
                _player1.Location = _myhouse;
32
                _player1.Location.Container.Put(_pen);
33
            }
34
35
            [Test]
36
            public void TestLookAtMe()
37
            {
38
                _output = _look.Execute(_player1, new string[] { "look", "at",
39
        "inventory" });
                Assert.That(_output, Is.EqualTo($"You are {_player1.Name}, Swinburne
40
       Student.\nYou are carrying:\n{_player1.Inventory.ItemList}\n"), "Test Look At
       Me");
            }
41
42
            [Test]
43
            public void TestLook()
                _output = _look.Execute(_player1, new string[] { "look" });
46
                Assert.That(_output, Is.EqualTo($"You are at: My House (house)\nItems at
47
       this location:\na pen (pen)\n"), "Test Look");
```

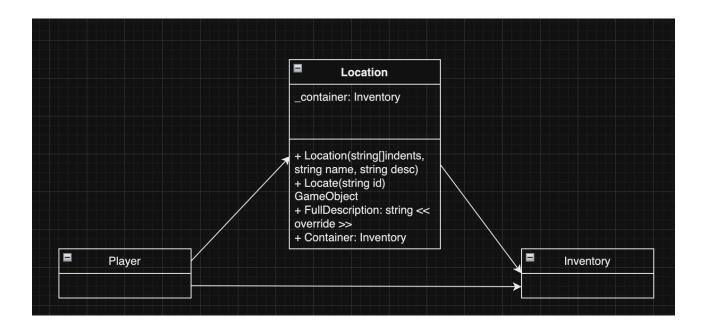
File 6 of 10 LookCommand tests

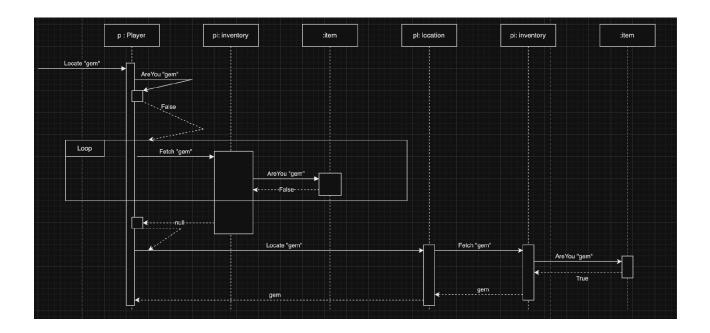
```
}
48
49
            [Test]
50
            public void TestLookAtGem()
            {
52
                _output = _look.Execute(_player1, new string[] { "look", "at", "gem",
53
       "in", "inventory" });
                Assert.That(_output, Is.EqualTo("This is a gem\n"), "Test Look At Gem");
54
            }
55
56
            [Test]
            public void TestLookAtUnk()
58
59
                _output = _look.Execute(_player1, new string[] { "look", "at", "gem2",
60
       "in", "inventory" });
                Assert.That(_output, Is.EqualTo("I can't find the gem2\n"), "Test Look At
61
       Unk");
            }
62
63
            [Test]
64
            public void TestLookAtGemInMe()
66
                _output = _look.Execute(_player1, new string[] { "look", "at", "gem",
        "in", "me" });
                Assert.That(_output, Is.EqualTo("This is a gem\n"), "Test Look At Gem in
68
       Me");
            }
69
70
            [Test]
71
            public void TestLookAtGemInBag()
72
73
                _output = _look.Execute(_player1, new string[] { "look", "at", "gem",
74
        "in", "bag" });
                Assert.That(_output, Is.EqualTo("This is a gem\n"), "Test Look At Gem in
75
       Bag");
            }
76
77
            [Test]
78
            public void TestLookAtGemInNoBag()
79
            {
80
                _output = _look.Execute(_player2, new string[] { "look", "at", "gem",
81
        "in", "bag" });
                Assert.That(_output, Is.EqualTo("I can't find the bag\n"), "Test Look At
82
       Gem in No Bag");
            }
84
            [Test]
85
            public void TestLookAtNoGemInBag()
86
87
                _output = _look.Execute(_player1, new string[] { "look", "at", "gem2",
88
        "in", "bag" });
                Assert.That(_output, Is.EqualTo("I can't find the gem2\n"), "Test Look At
89
       No Gem in Bag");
```

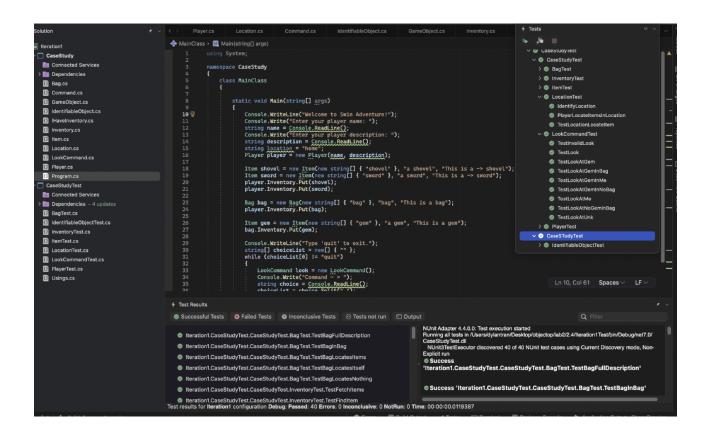
File 6 of 10 LookCommand tests

```
}
90
91
            [Test]
92
            public void TestInvalidLook()
            {
94
                 _output = _look.Execute(_player1, new string[] { "look", "around" });
95
                 Assert.That(_output, Is.EqualTo("I don't know how to look like that\n"),
96
        "Test Invalid Look");
97
                 _output = _look.Execute(_player1, new string[] { "hello", "hello",
98
        "hello" });
                Assert.That(_output, Is.EqualTo("Error in look input\n"), "Test Invalid
99
        Look");
100
                 _output = _look.Execute(_player1, new string[] { "look", "in", "a", "in",
101
        "b" });
                 Assert.That(_output, Is.EqualTo("What do you want to look at?\n"), "Test
102
        Invalid Look");
103
                 _output = _look.Execute(_player1, new string[] { "look", "at", "a", "at",
104
        "b" });
                Assert.That(_output, Is.EqualTo("What do you want to look in?\n"), "Test
105
        Invalid Look");
106
        }
107
    }
108
109
```

File 7 of 10 UML class diagram







```
using System;
using System.Collections.Generic;
using System.Ling;
using System.Ling;
using System.Met.Http.Headers;
using System.Reflection.Metadata.Ecma335;
using System.Text;
using System.Text;
            CaseStudy
Connected Services
Connected Services
Dependencies
Dependencies
Dependencies
Dependencies
Command.cs
Command.cs
Command.cs
Dependencies
Depen
                                                                                                                                                                                                                                                                                                                                                             namespace CaseStudy
                                                                                                                                                                                                                                                                                                                                                                                        public class Location : GameObject, IHaveInventory
{
                                                                                                                                                                                                                                                                                                                                                                                                                 private Inventory _container;
private List<Path> _pathList = new List<Path>(){ };
public Location(string[] idents, string name, string desc) : base(idents, name, desc)
{
                                                                                                                                                                                                                                                                                                                                                                                                                   public GameObject Locate(string id)
{
                                                                                                                                                                                                                                                                                                                                                                                                                                              if (this.AreYou(id)) return this;
GameObject locateResult = _container.Fetch(id);
return locateResult;
        CaseStudyTest
Connected Services
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             Ln 1, Col 1 Spaces V CRLF
    Connected Services

Dependencies - 4 updates

BagTest.cs
CommandProcessorTest.cs
IdentifiableObjectTest.cs
                                                                                                                                                                                                                                                                                                                                                                                                                   public void AddPath(Path path)
                                                                                                                                                                                                                                                                               Welcome to Swin Adventure!
Enter your player name: duc
Enter your player description: yo
Command: loook
Error input.
Command: look
You are at: Hy classroom (classroom)
Items at this location:
a pen (pen)

identifiableObjectTest.c
inventoryTest.cs
                                                                                                                                                                                                                                                                               Command:
      PlayerTest.cs
Usings.cs
```