

SWINBURNE UNIVERSITY OF TECHNOLOGY

COS20007 OBJECT ORIENTED PROGRAMMING

Case Study - Iteration 6 - Locations

PDF generated at 10:39 on Friday 3rd November, 2023

```
1
2 namespace CaseStudy
3 {
4     public class Location : GameObject, IHaveInventory
5     {
6         private Inventory _container;
7         public Location(string[] idents, string name, string desc) : base(idents,
8 ↪ name, desc)
9         {
10             _container = new();
11         }
12         public GameObject Locate(string id)
13         {
14             if (this.AreYou(id)) return this;
15             GameObject locateResult = _container.Fetch(id);
16             return locateResult;
17         }
18
19         public override string FullDescription
20         {
21             get
22             {
23                 return "You are at: " + base.ShortDescription + "\nItems at this
24 ↪ location:\n" + _container.ItemList;
25             }
26         }
27         public Inventory Container
28         {
29             get
30             {
31                 return _container;
32             }
33         }
34     }
35 }
```

```
1  using CaseStudy;
2
3  namespace CaseStudyTest
4  {
5      public class LocationTest
6      {
7          private Player _p;
8          private Location _loca;
9          private Item _sword;
10
11         [SetUp]
12         public void Setup()
13         {
14             _p = new Player("Tran", "This is Vu Duc Tran");
15             _loca = new Location(new string[] { "place1" }, "University", "Swinburne
↵ University");
16             _sword = new Item(new string[] { "sword" }, "a sword", "This is a
↵ sword");
17
18             _p.Location = _loca;
19             _loca.Container.Put(_sword);
20         }
21
22         [Test]
23         public void IdentifyLocation()
24         {
25             Assert.That(_loca.AreYou("place1"), Is.EqualTo(true), "Test Identify
↵ Location");
26         }
27
28         [Test]
29         public void TestLocationLocateItem()
30         {
31             Assert.That(_loca.Locate("sword"), Is.EqualTo(_sword), "Test Identify
↵ Location");
32         }
33
34         [Test]
35         public void PlayerLocateItemsInLocation()
36         {
37             Assert.That(_p.Location.Locate("sword"), Is.EqualTo(_sword), "Test Player
↵ Locate Items In Location");
38         }
39     }
40 }
```

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

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

```
1 using CaseStudy;
2
3 namespace CaseStudyTest
4 {
5     public class PlayerTest
6     {
7         private Player _player;
8         private Item _shovel;
9         private Item _sword;
10        private Location _loc;
11
12        [SetUp]
13        public void Setup()
14        {
15            _player = new Player("Vu Duc Tran", "Swinburne Student");
16            _shovel = new Item(new string[] { "shovel" }, "a shovel", "This is a
↵ shovel");
17            _sword = new Item(new string[] { "sword" }, "a sword", "This is a
↵ Sword");
18            _loc = new Location(new string[] { "school" }, "School", "Swinburne
↵ University");
19
20            _player.Inventory.Put(_sword);
21            _player.Location = _loc;
22            _loc.Container.Put(_shovel);
23        }
24
25        [Test]
26        public void TestPlayerIdentifiable()
27        {
28            Assert.That(_player.AreYou("me"), Is.EqualTo(true), "Test player
↵ identifiable");
29            Assert.That(_player.AreYou("inventory"), Is.EqualTo(true), "Test player
↵ identifiable");
30        }
31
32        [Test]
33        public void TestPlayerLocateItems()
34        {
35            Assert.That(_player.Locate("sword"), Is.EqualTo(_sword), "Test player
↵ locate items");
36        }
37
38        [Test]
39        public void TestPlayerLocateItself()
40        {
41            Assert.That(_player.Locate("me"), Is.EqualTo(_player), "Test player
↵ locate it self");
42            Assert.That(_player.Locate("inventory"), Is.EqualTo(_player), "Test
↵ player locate it self");
43        }
44
45        [Test]
```

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

```
1  using System;
2  namespace CaseStudy
3  {
4      public class LookCommand : Command
5      {
6          public LookCommand(string[] ids) : base(ids)
7          {
8              AddIdentifier("look");
9          }
10
11         public override string Execute(Player p, string[] text)
12         {
13             string ItemToFind;
14             string PlaceToLookIn;
15             string ItemString;
16             IHaveInventory Container;
17
18             if (text.Length == 1 && text[0] == "look")
19             {
20
21                 ItemString = p.Location.FullDescription;
22                 return ItemString;
23             }
24
25             if (text.Length != 3 && text.Length != 5)
26             {
27                 return "I don't know how to look like that\n";
28             }
29
30             if (text[0] != "look")
31             {
32                 return "Error in look input\n";
33             }
34
35             if (text[1] != "at")
36             {
37                 return "What do you want to look at?\n";
38             }
39
40             if (text.Length == 5)
41             {
42                 if (text[3] != "in")
43                 {
44                     return "What do you want to look in?\n";
45                 }
46             }
47
48             if (text.Length == 3)
49             {
50                 ItemToFind = text[2];
51                 return LookAtIn(ItemToFind, p as IHaveInventory);
52             }
53
```



```
54         if (text.Length == 5)
55         {
56             ItemToFind = text[2];
57             PlaceToLookIn = text[4];
58             Container = FetchContainer(p, PlaceToLookIn);
59             if (Container is null)
60             {
61                 return $"I can't find the {PlaceToLookIn}\n";
62             }
63             return LookAtIn(ItemToFind, Container);
64         }
65         return "Vu Duc Tran - Swinburne Student";
66     }
67
68
69     private IHaveInventory FetchContainer(Player p, string containerId)
70     {
71         return p.Locate(containerId) as IHaveInventory;
72     }
73
74     private string LookAtIn(string thingId, IHaveInventory container)
75     {
76         if (container.Locate(thingId) != null)
77         {
78             return $"{container.Locate(thingId).FullDescription}\n";
79         }
80         return $"I can't find the {thingId}\n";
81     }
82 }
83 }
84
```

```

1  using System;
2  using CaseStudy;
3
4  namespace CaseStudyTest
5  {
6      public class LookCommandTest
7      {
8          private LookCommand _look;
9          private Player _player1, _player2;
10         private Bag _bag;
11         private Location _myhouse;
12
13         private Item _gem;
14         private Item _pen;
15         private string _output;
16
17         [SetUp]
18         public void Setup()
19         {
20             _gem = new Item(new string[] { "gem" }, "a gem", "This is a gem");
21             _pen = new Item(new string[] { "pen" }, "a pen", "This is a pen");
22             _look = new LookCommand(new string[] { "look" });
23             _player1 = new Player("Vu Duc Tran 104175614 ", "Swinburne Student");
24             _player1.Inventory.Put(_bag);
25             _player2 = new Player("Dylan Tran", "student");
26
27             _bag = new Bag(new string[] { "bag" }, "Duc's bag", $"This is
↪ {_player1.FirstID} bag");
28             _myhouse = new Location(new string[] { "house" }, "My House", "My
↪ House");
29
30             _player1.Inventory.Put(_gem);
31             _bag.Inventory.Put(_gem);
32             _player1.Location = _myhouse;
33             _player1.Location.Container.Put(_pen);
34         }
35
36         [Test]
37         public void TestLookAtMe()
38         {
39             _output = _look.Execute(_player1, new string[] { "look", "at",
↪ "inventory" });
40             Assert.That(_output, Is.EqualTo($"You are {_player1.Name}, Swinburne
↪ Student.\nYou are carrying:\n{_player1.Inventory.ItemList}\n"), "Test Look At
↪ Me");
41         }
42
43         [Test]
44         public void TestLook()
45         {
46             _output = _look.Execute(_player1, new string[] { "look" });
47             Assert.That(_output, Is.EqualTo($"You are at: My House (house)\nItems at
↪ this location:\na pen (pen)\n"), "Test Look");

```

```
48     }
49
50     [Test]
51     public void TestLookAtGem()
52     {
53         _output = _look.Execute(_player1, new string[] { "look", "at", "gem",
↪ "in", "inventory" });
54         Assert.That(_output, Is.EqualTo("This is a gem\n"), "Test Look At Gem");
55     }
56
57     [Test]
58     public void TestLookAtUnk()
59     {
60         _output = _look.Execute(_player1, new string[] { "look", "at", "gem2",
↪ "in", "inventory" });
61         Assert.That(_output, Is.EqualTo("I can't find the gem2\n"), "Test Look At
↪ Unk");
62     }
63
64     [Test]
65     public void TestLookAtGemInMe()
66     {
67         _output = _look.Execute(_player1, new string[] { "look", "at", "gem",
↪ "in", "me" });
68         Assert.That(_output, Is.EqualTo("This is a gem\n"), "Test Look At Gem in
↪ Me");
69     }
70
71     [Test]
72     public void TestLookAtGemInBag()
73     {
74         _output = _look.Execute(_player1, new string[] { "look", "at", "gem",
↪ "in", "bag" });
75         Assert.That(_output, Is.EqualTo("This is a gem\n"), "Test Look At Gem in
↪ Bag");
76     }
77
78     [Test]
79     public void TestLookAtGemInNoBag()
80     {
81         _output = _look.Execute(_player2, new string[] { "look", "at", "gem",
↪ "in", "bag" });
82         Assert.That(_output, Is.EqualTo("I can't find the bag\n"), "Test Look At
↪ Gem in No Bag");
83     }
84
85     [Test]
86     public void TestLookAtNoGemInBag()
87     {
88         _output = _look.Execute(_player1, new string[] { "look", "at", "gem2",
↪ "in", "bag" });
89         Assert.That(_output, Is.EqualTo("I can't find the gem2\n"), "Test Look At
↪ No Gem in Bag");
```

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







