

---

## 4.2P - Case Study - Iteration 2 - Players Items and Inventory

---

PDF generated at 17:46 on Thursday 20<sup>th</sup> April, 2023

```
1  using System;
2  using System.Collections.Generic;
3  using Iteration_2;
4
5  namespace Iteration_2 {
6
7      public class GameObject : IdentifiableObject{
8
9          private string _name;
10
11         private string _description;
12
13         public GameObject(string[] identifiers, string name, string description) :
↪      base(identifiers) {
14
15             _name = name;
16             _description = description;
17
18         }
19
20         public String Name { get { return _name; } }
21
22         public string ShortDescription { get { return $"a {_name}
↪      ({Identifiers[0]}"; } }
23
24         public string LongDescription { get { return _description; } }
25
26     }
27
28 }
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using Iteration_2;
5
6  namespace Iteration_2 {
7
8      public class Player : GameObject {
9
10         private Inventory _inventory = new Inventory();
11
12         public Player(string name, string description) : base(new string[] {"me",
↵ "Inventory"}, name, description) {}
13
14         public String FullDescription()
15         {
16             return $"You are {Name}, {ShortDescription}. You are
↵ carrying:\n{_inventory.ItemList}";
17         }
18
19         public Inventory Inventory
20         {
21             get { return _inventory; }
22         }
23
24         public GameObject Locate(string check_id, Player self)
25         {
26             if (check_id == "me" || check_id == "inventory")
27             {
28                 return self;
29             }
30             else if (self.Inventory.HasItem(check_id))
31             {
32                 return self.Inventory.Fetch(check_id);
33             }
34             else
35             {
36                 return null;
37             }
38         }
39     }
40 }
41
42 }
43
44 }
```

```
1  using System.Numerics;
2  using Iteration_2;
3
4  namespace PlayerTests
5  {
6      public class Tests
7      {
8
9          [Test]
10
11         public void PlayerIdentifiable()
12         {
13             string test_name = "test_name";
14
15             string test_description = "test_description";
16             Player test_player = new Player(test_name, test_description);
17
18             if (test_player.Locate("me", test_player) == test_player)
19             {
20                 Assert.Pass();
21             }
22             else
23             {
24                 Assert.Fail();
25             }
26         }
27
28         [Test]
29
30         public void PlayerLocatesItems()
31         {
32             string test_name = "test_name";
33
34             string test_description = "test_description";
35             Player test_player = new Player(test_name, test_description);
36
37             Item test_item = new Item(new string[] { "id1", "id2" }, "item_name",
↵ "item_description");
38
39             test_player.Inventory.Put(test_item);
40
41             if (test_player.Locate("id1", test_player) == test_item)
42             {
43                 Assert.Pass();
44             }
45             else
46             {
47                 Assert.Fail();
48             }
49         }
50
51         [Test]
52
```

```
53     public void PlayerLocatesNothing()
54     {
55         string test_name = "test_name";
56
57         string test_description = "test_description";
58         Player test_player = new Player(test_name, test_description);
59
60         Item test_item = new Item(new string[] { "id1", "id2" }, "item_name",
↪ "item_description");
61
62         test_player.Inventory.Put(test_item);
63
64         if (test_player.Locate("id3", test_player) == null)
65         {
66             Assert.Pass();
67         }
68         else
69         {
70             Assert.Fail();
71         }
72     }
73
74     [Test]
75
76     public void PlayerFullDescription()
77     {
78         string test_name = "test_name";
79
80         string test_description = "test_description";
81         Player test_player = new Player(test_name, test_description);
82
83         Item test_item = new Item(new string[] { "id1", "id2" }, "item_name",
↪ "item_description");
84
85         test_player.Inventory.Put(test_item);
86
87         string test_full_description = "You are test_name, test_description. You
↪ are carrying:/n    a item_name (id1)/n";
88
89         if (test_player.FullDescription() == test_full_description)
90         {
91             Assert.Pass();
92         }
93         else
94         {
95             Assert.Fail();
96         }
97     }
98 }
99 }
```

```
1  using System;
2  using System.Collections.Generic;
3
4  namespace Iteration_2 {
5
6      public class Item : GameObject {
7
8          public Item(string[] identifiers, string name, string description) :
↪      base(identifiers, name, description) { }
9
10
11     }
12
13 }
```

```
1  using Iteration_2;
2
3  namespace Iteration_2_tests
4  {
5      public class Tests
6      {
7
8          [Test]
9          public void Item_Identifiable()
10         {
11
12             var test_identifiers = new string[] { "id1", "id2", "id3" };
13
14             string test_name = "test_item";
15
16             string test_description = "test_description";
17
18             Item test_item = new Item(test_identifiers, test_name, test_description);
19
20
21
22             bool test_are_you = test_item.Are_You("id2");
23
24
25             if (test_are_you)
26             {
27                 Assert.Pass();
28             }
29             else
30             {
31                 Assert.Fail();
32             }
33
34         }
35
36         [Test]
37
38         public void Item_ShortDescription()
39         {
40
41             var test_identifiers = new string[] { "sword", "id2", "id3" };
42
43             string test_name = "bronze sword";
44
45             string test_description = "test_description";
46
47             Item test_item = new Item(test_identifiers, test_name, test_description);
48
49
50             string test_short_description = test_item.ShortDescription;
51
52             if (test_short_description == "a bronze sword (sword)")
53             {
```

```
54         Assert.Pass();
55     }
56     else
57     {
58         Assert.Fail();
59     }
60
61 }
62
63 [Test]
64
65 public void Item_FullDescription()
66 {
67
68     var test_identifiers = new string[] { "id1", "id2", "id3" };
69
70     string test_name = "test_name";
71
72     string test_description = "test_description";
73
74     Item test_item = new Item(test_identifiers, test_name, test_description);
75
76
77     string test_full_description = test_item.LongDescription;
78
79     if (test_full_description == "test_description")
80     {
81         Assert.Pass();
82     }
83     else
84     {
85         Assert.Fail();
86     }
87 }
88 +
89
90 }
91 }
```



```
1  using System;
2  using System.Collections.Generic;
3  using Iteration_2;
4
5  namespace Iteration_2 {
6
7      public class Inventory {
8
9          private List<Item> _items = new List<Item> { };
10
11         public Inventory() { }
12
13         public bool HasItem(string test_id) {
14
15             bool hasitem = false;
16
17             foreach (Item item in _items) {
18
19                 if (item.Are_You(test_id))
20                 {
21                     hasitem = true;
22                 }
23             }
24
25             return hasitem;
26
27         }
28
29         public void Put(Item new_item) {
30
31             _items.Add(new_item);
32
33         }
34
35         public Item Take(string identifier) {
36
37             int index = 0;
38
39             int item_index = 0;
40
41             foreach (Item item in _items)
42             {
43                 if (item.Are_You(identifier))
44                 {
45
46                     item_index = index;
47
48                 }
49
50                 index++;
51             }
52
53         }
```

```
54         Item return_item = _items[item_index];
55
56         _items.RemoveAt(item_index);
57
58         return return_item;
59
60     }
61
62     public Item Fetch(string identifier) {
63
64         int index = 0;
65
66         int item_index = 0;
67
68         foreach (Item item in _items)
69         {
70             if (item.Are_You(identifier))
71             {
72                 item_index= index;
73             }
74
75             index++;
76         }
77
78         return _items[item_index];
79
80     }
81
82     public String ItemList
83     {
84
85         get
86         {
87
88             string itemlist = "";
89
90             foreach (Item item in _items)
91             {
92
93                 itemlist += $"    {item.ShortDescription}/n";
94
95             }
96
97             return itemlist;
98
99         }
100
101     }
102
103 }
104
105
106 }
```

```
1  using Iteration_2;
2
3  namespace InventoryTests
4  {
5      public class Tests
6      {
7
8          [Test]
9
10         public void InventoryFindItem()
11         {
12
13             Inventory test_inventory = new Inventory();
14
15             var test_identifiers = new string[] { "id1", "id2", "id3" };
16
17             string test_name = "test_name";
18
19             string test_description = "test_description";
20
21             Item new_item = new Item(test_identifiers, test_name, test_description);
22
23             test_inventory.Put(new_item);
24
25             if (test_inventory.HasItem("id1"))
26             {
27                 Assert.Pass();
28             }
29             else
30             {
31                 Assert.Fail();
32             }
33         }
34
35         [Test]
36
37         public void InventoryNoItemFind()
38         {
39
40             Inventory test_inventory = new Inventory();
41
42             var test_identifiers = new string[] { "id1", "id2", "id3" };
43
44             string test_name = "test_name";
45
46             string test_description = "test_description";
47
48             Item new_item = new Item(test_identifiers, test_name, test_description);
49
50             test_inventory.Put(new_item);
51
52             if (test_inventory.HasItem("id4"))
53             {
```

```
54         Assert.Fail();
55     }
56     else
57     {
58         Assert.Pass();
59     }
60 }
61
62 [Test]
63
64 public void InventoryFetch()
65 {
66
67     bool fetch_bool = false;
68
69     bool fetch_remains = false;
70
71
72
73     var test_identifiers = new string[] { "id1", "id2", "id3" };
74
75     string test_name = "test_name";
76
77     string test_description = "test_description";
78
79     Item test_item = new Item(test_identifiers, test_name, test_description);
80
81     Inventory test_inventory = new Inventory();
82
83     test_inventory.Put(test_item);
84
85     Item item_fetched = test_inventory.Fetch("id1");
86
87
88
89     if (item_fetched == test_item) { fetch_bool = true; }
90
91     if (test_inventory.HasItem("id1")) { fetch_remains = true; }
92
93     if (fetch_bool && fetch_remains)
94     {
95         Assert.Pass();
96     }
97     else
98     {
99         Assert.Fail();
100     }
101
102 }
103
104 [Test]
105
106 public void InventoryTake()
```

```
107     {
108         bool take_bool = false;
109
110         bool take_is_gone = false;
111
112
113
114         var test_identifiers = new string[] { "id1", "id2", "id3" };
115
116         string test_name = "test_name";
117
118         string test_description = "test_description";
119
120         Item test_item = new Item(test_identifiers, test_name, test_description);
121
122         Inventory test_inventory = new Inventory();
123
124         test_inventory.Put(test_item);
125
126
127
128         Item item_taken = test_inventory.Take("id1");
129
130
131
132         if (item_taken == test_item) { take_bool = true; }
133
134         if (!(test_inventory.HasItem("id1"))) { take_is_gone = true; }
135
136         if (take_bool && take_is_gone)
137         {
138             Assert.Pass();
139         }
140         else
141         {
142             Assert.Fail();
143         }
144     }
145
146     [Test]
147
148     public void InventoryItemList()
149     {
150         var test_identifiers = new string[] { "id1", "id2", "id3" };
151
152         string test_name = "test_name";
153
154         string test_description = "test_description";
155
156         Item test_item = new Item(test_identifiers, test_name, test_description);
157
158         Inventory test_inventory = new Inventory();
159
```

```
160         test_inventory.Put(test_item);
161
162
163         string item_string = test_inventory.ItemList;
164
165         if (item_string == $"    a test_name (id1)/n")
166         {
167             Assert.Pass();
168         }
169         else
170         {
171             Assert.Fail();
172         }
173     }
174 }
175 }
```

