

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using SwinAdventure;
7
8 namespace SwinAdventureTest
9 {
10     public class LookCommandtest //Change from internal to public
11     {
12         Item gem;
13         Bag bag;
14         Player player;
15         LookCommand look;
16         [SetUp]
17         public void SetUp()
18         {
19             look = new LookCommand();
20
21             player = new Player("Truong Ngoc Gia Hieu", "A brave Swinburne Warrior");
22             bag = new Bag(new string[] { "bag" }, "Hieu's Bag", "A small Bag");
23
24             gem = new Item(new string[] { "gem" }, "a gem", "A bright red gem");
25
26             player.Inventory.Put(bag);
27         }
28         [Test]
29         public void TestLookAtGem()
30         {
31             player.Inventory.Put(gem);
32
33             string output = look.Execute(player, new string[] { "look", "at", "gem" });
34             string expected = "A bright red gem";
35             Assert.AreEqual(expected, output);
36         }
37
38         [Test]
39         public void TestLookAtMe()
40         {
41             string output = look.Execute(player, new string[] { "look", "at", "inventory" });
42             string expected = "Truong Ngoc Gia Hieu, A brave Swinburne Warrior\nList of Items that you have:\nHieu's Bag: bag\n";
43             Assert.AreEqual(expected, output);
```

```
44     }
45
46     [Test]
47     public void TestLookAtGemInMe()
48     {
49         player.Inventory.Put(gem);
50
51         string output = look.Execute(player, new string[] { "look",
52             "at", "gem", "in", "inventory" });
53         string expected = "A bright red gem";
54         Assert.AreEqual(expected, output);
55     }
56
57     [Test]
58     public void TestLookAtUnk()
59     {
60         // Test looking at an unknown item
61         string output = look.Execute(player, new string[] { "look",
62             "at", "gem" });
63         string expected = "I cannot find the gem in the Truong Ngoc
64             Gia Hieu";
65         Assert.AreEqual(expected, output);
66     }
67
68     [Test]
69     public void TestLookatGemInNoBag()
70     {
71         Player player1 = new Player("NPC", "Robot Warrior");
72
73         string output = look.Execute(player1, new string[] { "look",
74             "at", "gem", "in", "bag" });
75         string expected = "I cannot find the gem";
76         Assert.AreEqual(expected, output);
77     }
78
79     [Test]
80     public void TestLookAtGemInBag()
81     {
82         // Put the gem in the bag
83         bag.Inventory.Put(gem);
84
85         // Test looking at the gem in the bag
86         string output = look.Execute(player, new string[] { "look",
87             "at", "gem", "in", "bag" });
88         string expected = "A bright red gem";
89         Assert.AreEqual(expected, output);
90     }
91
92     [Test]
```

```
88     public void TestInvalidLookCommands()
89     {
90         string output1 = look.Execute(player, new string[] { "look",
91             "around" });
92         string output2 = look.Execute(player, new string[]
93             { "hello" });
94         string expected = "I don't know how to look like that";
95
96         Assert.AreEqual(expected, output1);
97         Assert.AreEqual(expected, output2);
98     }
99
100     [Test]
101     public void TestLookAtNoGemInBag()
102     {
103         // Create an empty bag
104         Bag emptyBag = new Bag(new string[] { "emptybag" }, "Empty
105             Bag", "An empty bag");
106
107         // Put the empty bag in the player's inventory
108         player.Inventory.Put(emptyBag);
109
110         // Test looking at the gem in the empty bag
111         string output = look.Execute(player, new string[] { "look",
112             "at", "gem", "in", "emptybag" });
113         string expected = "I cannot find the gem in the Empty Bag";
114         Assert.AreEqual(expected, output);
115     }
116 }
```