

```
1 using SwinAdventure;
2 using NUnit.Framework;
3 namespace LookCommandTest
4 {
5     public class LookCommandTest
6     {
7         LookCommand look;
8         Player player;
9         Bag bag;
10        Item gem;
11
12        [SetUp]
13        public void Setup()
14        {
15            look = new LookCommand();
16            player = new Player("Fred", "the mighty programmer");
17            bag = new Bag(new string[] { "bag" }, "a bag", "This is a tote ➤
            bag");
18            gem = new Item(new string[] { "gem" }, "a gem", "This is a ➤
            bright red gemstone");
19
20            player.Inventory.Put(gem);
21        }
22
23        [Test]
24        public void TestLookAtMe()
25        {
26            Assert.That(look.Execute(player, new string[] { "look", "at", ➤
            "inventory" }), Is.EqualTo("the mighty programmer"));
27        }
28
29        [Test]
30        public void TestLookAtGem()
31        {
32            Assert.That(look.Execute(player, new string[] { "look", "at", ➤
            "gem" }), Is.EqualTo("This is a bright red gemstone"));
33        }
34
35        [Test]
36        public void TestLookAtUnknown()
37        {
38            player.Inventory.Take("gem");
39            Assert.That(look.Execute(player, new string[] { "look", "at", ➤
            "gem" }), Is.EqualTo("I cannot find the gem"));
40        }
41
42        [Test]
43        public void TestLookAtGemInMe()
44        {
```

```
45         Assert.That(look.Execute(player, new string[] { "look", "at",  
            "gem", "in", "inventory" }), Is.EqualTo("This is a bright red  
            gemstone"));  
46     }  
47  
48     [Test]  
49     public void TestLookAtGemInBag()  
50     {  
51         player.Inventory.Take("gem");  
52         bag.Inventory.Put(gem);  
53         player.Inventory.Put(bag);  
54         Assert.That(look.Execute(player, new string[] { "look", "at",  
            "gem", "in", "bag" }), Is.EqualTo("This is a bright red  
            gemstone"));  
55     }  
56  
57     [Test]  
58     public void TestLookAtGemInNoBag()  
59     {  
60         player.Inventory.Take("bag");  
61         Assert.That(look.Execute(player, new string[] { "look", "at",  
            "gem", "in", "bag" }), Is.EqualTo("I cannot find the bag"));  
62     }  
63  
64     [Test]  
65     public void TestLookAtNoGemInBag()  
66     {  
67         player.Inventory.Put(bag);  
68         bag.Inventory.Take("gem");  
69         Assert.That(look.Execute(player, new string[] { "look", "at",  
            "gem", "in", "bag" }), Is.EqualTo("I cannot find the gem"));  
70     }  
71  
72     [Test]  
73     public void TestInvalidLook()  
74     {  
75         Assert.That(look.Execute(player, new string[] { "look",  
            "around" }), Is.EqualTo("I don't know how to look like  
            that"));  
76         Assert.That(look.Execute(player, new string[] { "hello",  
            "it's", "me" }), Is.EqualTo("Error in look input"));  
77         Assert.That(look.Execute(player, new string[] { "look", "at",  
            "a", "at", "b" }), Is.EqualTo("What do you want to look  
            in?"));  
78         Assert.That(look.Execute(player, new string[] { "look", "in",  
            "bag", "for", "gem" }), Is.EqualTo("What do you want to look  
            at?"));  
79     }  
80
```

81 }

82 }