```
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             GameObject container;
20
21             player.Inventory.Put(gem);
22         }
23
24         [Test]
25         public void TestLookAtMe()
26         {
27             Assert.That(look.Execute(player, new string[] { "look", "at",
                 "inventory" }), Is.EqualTo(player.FullDescription));
28         }
29
30         [Test]
31         public void TestLookAtGem()
32         {
33             Assert.That(look.Execute(player, new string[] { "look", "at",
                 "gem" }), Is.EqualTo("This is a bright red gemstone"));
34         }
35
36         [Test]
37         public void TestLookAtUnknown()
38         {
39             player.Inventory.Take("gem");
40             Assert.That(look.Execute(player, new string[] { "look", "at",
                 "gem" }), Is.EqualTo($"I cannot find the gem in Player"));
41         }
42
43         [Test]
44         public void TestLookAtGemInMe()
```

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

```
80          }
81
82      }
83  }
```