

```
1 using _9._2C;
2 using System;
3 using System.Collections.Generic;
4 using System.Linq;
5 using System.Threading.Tasks;
6 using NUnit.Framework;
7 using System.Numerics;
8
9 namespace LookCommandTest
10 {
11     [TestFixture]
12     public class Tests
13     {
14         private Bag _bag;
15         private Item _item;
16         private Player _player;
17
18         [SetUp]
19         public void Setup()
20         {
21             _bag = new Bag(new string[] { "backpack" }, "Backpack",
22                             "very gud backpack");
23             _item = new Item(new string[] { "diamond" }, "Diamond",
24                             "shiny diamond");
25             _player = new Player("Nevan", "a desparate programmer");
26         }
27
28         [TestCase]
29         public void TestLookAtMe()
30         {
31             LookCommand look = new LookCommand();
32             string expectedDescription = look.Execute(_player, new
33                 string[] { "look", "at", "inventory" });
34             Assert.AreEqual(_player.FullDescription,
35                             expectedDescription);
36         }
37
38         [TestCase]
39         public void TestLookAtGem()
40         {
41             LookCommand look = new LookCommand();
42             _player.Inventory.Put(_item);
43             string result = look.Execute(_player, new string[]
44                 { "look", "at", "diamond" });
45             Assert.AreEqual(_item.FullDescription, result);
46         }
47
48         [TestCase]
49         public void TestLookAtUnk()
50         {
51             LookCommand look = new LookCommand();
52             string result = look.Execute(_player, new string[]
53                 { "look", "at", "diamond" });
```

```
48         Assert.AreEqual("I cannot find the diamond in Nevan",  
49             result);  
50     }  
51     [TestCase]  
52     public void TestLookAtGemInMe()  
53     {  
54         _player.Inventory.Put(_item);  
55         LookCommand look = new LookCommand();  
56         string expectedDescription = look.Execute(_player, new  
57             string[] { "look", "at", "diamond", "in", "inventory" });  
58         Assert.AreEqual(_item.FullDescription,  
59             expectedDescription);  
60     }  
61     [TestCase]  
62     public void TestLookAtGemInBag()  
63     {  
64         _player.Inventory.Put(_bag);  
65         _bag.Inventory.Put(_item);  
66         LookCommand look = new LookCommand();  
67         string expectedDescription = look.Execute(_player, new  
68             string[] { "look", "at", "diamond", "in", "backpack" });  
69         Assert.AreEqual(_item.FullDescription,  
70             expectedDescription);  
71     }  
72     [TestCase]  
73     public void TestLookAtGemInNoBag()  
74     {  
75         //_player.Inventory.Put(_bag);  
76         _bag.Inventory.Put(_item);  
77         LookCommand look = new LookCommand();  
78         string expectedDescription = look.Execute(_player, new  
79             string[] { "look", "at", "diamond", "in", "backpack" });  
80         Assert.AreEqual("I cannot find the backpack",  
81             expectedDescription);  
82     }  
83     [TestCase]  
84     public void TestLookAtNoGemInNBag()  
85     {  
86         _player.Inventory.Put(_bag);  
87         //_bag.Inventory.Put(_item);  
88         LookCommand look = new LookCommand();  
89         string expectedDescription = look.Execute(_player, new  
90             string[] { "look", "at", "diamond", "in", "backpack" });  
91         Assert.AreEqual("I cannot find the diamond in Backpack",  
92             expectedDescription);  
93     }  
94     [TestCase]  
95     public void TestInvalidLook() //
```

```
92     {
93         LookCommand look = new LookCommand();
94         string expectedDescription = look.Execute(_player, new string[] { "look", "around" });
95         Assert.AreEqual("I don't know how to look like that", expectedDescription);
96
97         expectedDescription = look.Execute(_player, new string[] { "hello", "104772183" });
98         Assert.AreEqual("I don't know how to look like that", expectedDescription);
99
100        expectedDescription = look.Execute(_player, new string[] { "look", "at", "Nguyen" });
101        Assert.AreEqual("I cannot find the Nguyen in Nevan", expectedDescription);
102
103
104    }
105
106
107 }
108 }
```