

```
1 using _10._1C;
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
10 namespace CommandProcessorTest
11 {
12     [TestFixture]
13     public class CommandProcessorTests
14     {
15
16         private CommandProcessor _commandProcessor;
17         private Player _player;
18
19
20         [SetUp]
21         public void SetUp()
22         {
23             _commandProcessor = new CommandProcessor();
24
25             _player = new Player("Nguyen", "A programmer");
26
27             Location mountain1 = new Location(new string[] { "mountain1" }, "Mountain 1", "first mountain");
28             Location mountain2 = new Location(new string[] { "mountain2" }, "Mountain 2", "second mountain");
29             Item sword = new Item(new string[] { "sword" }, "Excalibur", "a strong sword");
30             _player.Inventory.Put(sword);
31
32
33             Paths pathToMountain1 = new Paths(new string[] { "west" }, "Journey to the West", "path leading West", mountain1);
34             Paths pathToMountain2 = new Paths(new string[] { "east" }, "Journey to the East", "path leading East", mountain2);
35
36             _player.Location = mountain1;
37             _player.Location = mountain2;
38             mountain2.AddPath(pathToMountain1);
39             mountain1.AddPath(pathToMountain2);
40         }
41
42         [Test]
43         public void TestValidLookCommand()
44         {
45             // Arrange
46             string input = "look at sword in inventory";
47             string expectedResponse = "a strong sword";
48
```

```
49         // Act
50         string response = _commandProcessor.ExecuteCommand(input,
51             _player);
52
53         // Assert
54         Assert.AreEqual(expectedResponse, response);
55     }
56
57     [Test]
58     public void TestValidMoveCommand()
59     {
60         // Arrange
61         string input = "move west";
62         string expectedResponse = "You move west to Mountain 1.";
63
64         // Act
65         string response = _commandProcessor.ExecuteCommand(input,
66             _player);
67
68         // Assert
69         Assert.AreEqual(expectedResponse, response);
70         Assert.AreEqual("Mountain 1", _player.Location.Name);
71     }
72
73     [Test]
74     public void TestUnknownCommand()
75     {
76         // Arrange
77         string input = "fly";
78         string expectedResponse = "Unknown command: fly";
79
80         // Act
81         string response = _commandProcessor.ExecuteCommand(input,
82             _player);
83
84         // Assert
85         Assert.AreEqual(expectedResponse, response);
86     }
87
88     [Test]
89     public void TestEmptyCommand()
90     {
91         // Arrange
92         string input = ""; // Empty command string
93         string expectedResponse = "Invalid command.";
94
95         // Act
96         string response = _commandProcessor.ExecuteCommand(input,
97             _player);
98
99         // Assert
100        Assert.AreEqual(expectedResponse, response);
101    }
```

```
98     }  
99  
100 }
```