```
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 PathTest
10 {
11
12     [TestFixture]
13     public class Tests
14     {
15         private Location _mountain1;
16         private Location _mountain2;
17         private Player _player;
18         private Item _sword;
19         private MoveCommand _moveCommand;
20         private Paths _pathToMountain1;
21         private Paths _pathToMountain2;
22         [SetUp]
23         public void Setup()
24         {
25
26             _mountain1 = new Location(new string[] { "mountain1" },
                 "Mountain 1", "first mountain");
27             _mountain2 = new Location(new string[] { "mountain2" },
                 "Mountain 2", "second mountain");
28             _sword = new Item(new string[] { "sword" }, "Excalibur", "a
                 strong sword");
29             _mountain1.Inventory.Put(_sword);
30
31
32             _pathToMountain1 = new Paths(new string[] { "west" },
                 "Journey to the West", "path leading West", _mountain1);
33             _pathToMountain2 = new Paths(new string[] { "east" },
                 "Journey to the East", "path leading East", _mountain2);
34
35             _mountain1.AddPath(_pathToMountain2);
36             _mountain2.AddPath(_pathToMountain1);
37             _player = new Player("Wukong", "The monkey");
38             _player.Location = _mountain1;
39
40
41             _moveCommand = new MoveCommand();
42         }
43
44         [Test]
45         public void TestPathCanMovePlayer()
46         {
47             string result = _moveCommand.Execute(_player, new string[]
                 { "move", "east" });
```

```
48              Assert.AreEqual(_mountain2, _player.Location);
49
50          }
51
52          [Test]
53          public void TestGetPathFromLocation()
54          {
55              Paths path = _mountain1.GetPath("east");
56              Assert.IsNotNull(path);
57              Assert.AreEqual(_mountain2, path.Destination);
58          }
59
60          [Test]
61          public void TestPlayerCanLeaveLocation()
62          {
63              _moveCommand.Execute(_player, new string[] { "head",
                  "east" });
64              Assert.AreEqual(_mountain2, _player.Location);
65
66
67          }
68
69          [Test]
70          public void TestPlayerRemainInLocation()
71          {
72              _moveCommand.Execute(_player, new string[] { "head",
                  "north" });
73              Assert.AreNotEqual(_mountain2, _player.Location);
74          }
75      }
76  }
```