Swinburne University of Technology

COS20007 Object Oriented Programming

Case Study - Iteration 4 - Look Command

PDF generated at 02:10 on Monday 16th October, 2023

Produced by OOP Submission Formatter

```
using CaseStudy_Iteration4;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace CaseStudy_Iteration4
{
    public interface IHaveInventory
    {

        GameObject Locate(string id);
        string Name
        {
            get;
        }
    }
}
```

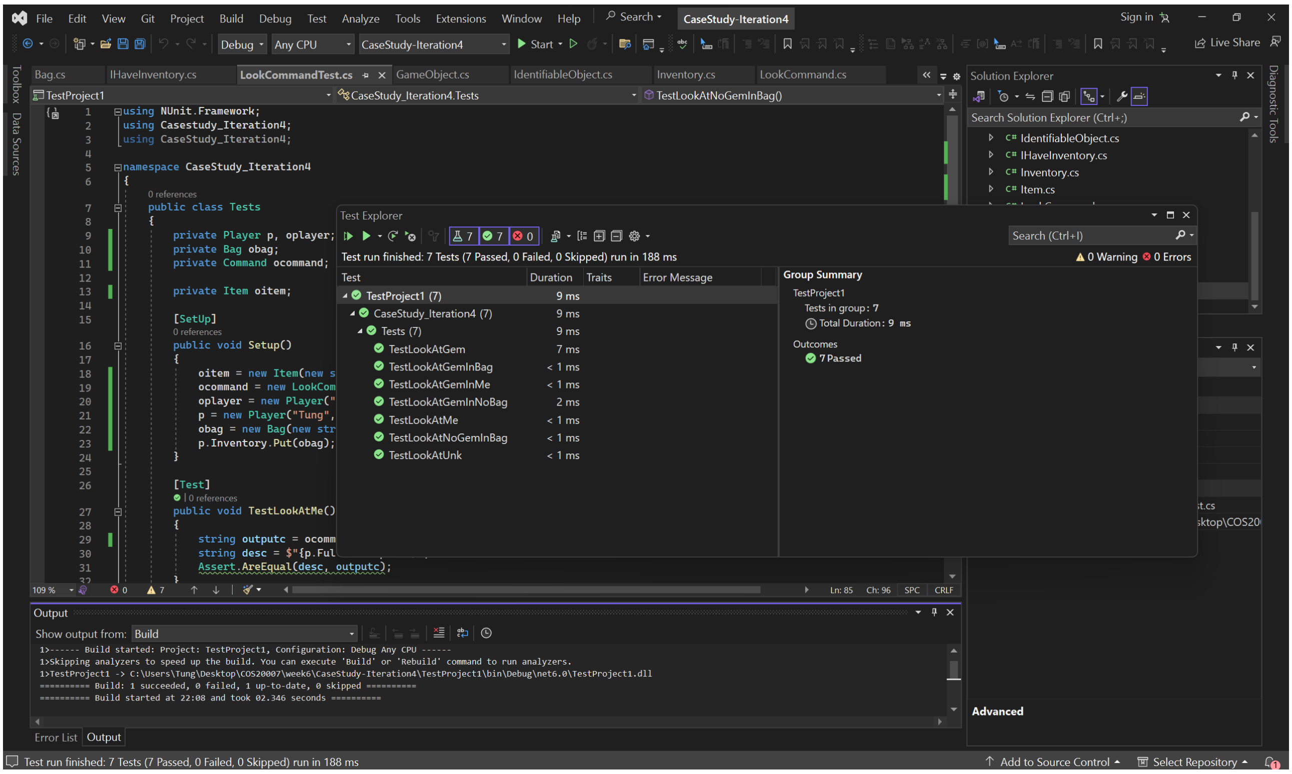
```
using CaseStudy_Iteration4;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace CaseStudy_Iteration4
{
    public class Player:GameObject , IHaveInventory
    {
        private Inventory oinventory;
        public Player(string name, string desc) : base(new string[] { "me", "inventory" }, name, desc)
        {
            oinventory = new Inventory();
        }
        public GameObject Locate(string id)
        {
            if (AreYou(id) == true)

            {
                return this;
            }
            return oinventory.Fetch(id);
        }
        public override string FullDescription
        {
            get { return $"You are {Name} {base.FullDescription}\nYou are carrying:\n{oinventory.Itemlist}"; }
        }
        public Inventory Inventory
        {
            get { return oinventory; }
        }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace CaseStudy_Iteration4
{
    public class Bag : Item , IHaveInventory
    {
        private Inventory oinventory;

        public Bag(string[] id, string name, string desc) : base(id, name, desc)
        {
            oinventory = new Inventory();
        }
        public GameObject Locate(string id)
        {
            if (AreYou(id) == true)
            {
                return this;
            }
            return oinventory.Fetch(id);

        }
        public override string FullDescription
        {
            get
            {
                return "In the " + Name + " you can see " + oinventory.Itemlist;
            }
        }
        public Inventory Inventory
        {
            get { return oinventory; }
        }
    }
}
```

```
using CaseStudy_Iteration4;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace CaseStudy_Iteration4
{
    public abstract class Command : IdentifiableObject
    {
        public Command(string[] ids) : base(ids)
        {

        }
        public abstract string Execute(Player p, string[] text);
    }
}
```

```
using CaseStudy_Iteration4;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Linq;
using System.Net.Http.Headers;
using System.Runtime.InteropServices.WindowsRuntime;
using System.Security.Policy;
using System.Text;
using System.Threading;
using System.Threading.Tasks;
using System.Xml.Schema;

namespace Casestudy_Iteration4
{
    public class LookCommand : Command
    {
        public LookCommand() : base(new string[] { "look ", "look" })
        {

        }
        public override string Execute(Player p, string[] text)
        {
            IHaveInventory container = null;
            if (text.Count() != 3 && text.Count() != 5)
            {
                return "I don't know how to look like that.";
            }
            else if (text[0] != "look")
            {
                return "Error in look input.";
            }
            else if (text[1] != "at")
            {
                return "What do you want to look at.";
            }
            if (text.Length == 3)
            {
                container = p;
            }
            if (text.Length == 5)
            {
                if (text[3] != "in")
                {
                    return "What do you want to look in";
                }
                else
                {
                    container = FetchContainer(p, text[4]);
                    if (container == null)
                    {
                        return $"I cannot find the {text[4]}";
                    }
                }
            }
            return LookAtln(text[2], container);
        }
        public IHaveInventory FetchContainer(Player p, string containerld)
        {
            return p.Locate(containerld) as IHaveInventory;
        }
        public string LookAtln(string thingId, IHaveInventory container)
        {
            if (container.Locate(thingId) != null)
            {
                return container.Locate(thingId).FullDescription;
            }
            return $"I cannot find the {thingId}";
        }

    }

}

```

```
using NUnit.Framework;
using Casestudy_Iteration4;
using CaseStudy_Iteration4;

namespace CaseStudy_Iteration4
{
    public class Tests
    {
        private Player p, oplayer;
        private Bag obag;
        private Command ocommand;

        private Item oitem;

        [SetUp]
        public void Setup()
        {
            oitem = new Item(new string[] { "b52" }, "a b52", "This is a good plane");
            ocommand = new LookCommand();
            oplayer = new Player("Tung", "best player");
            p = new Player("Tung", "professional");
            obag = new Bag(new string[] { "bag" }, "backpack", $"This is {p.FirstId} backpack");
            p.Inventory.Put(obag);
        }

        [Test]
        public void TestLookAtMe()
        {
            string outputc = ocommand.Execute(p, new string[] { "look", "at", "inventory" });
            string desc = $"{p.FullDescription}";
            Assert.AreEqual(desc, outputc);
        }

        [Test]
        public void TestLookAtGem()
        {
            p.Inventory.Put(oitem);

            string outputc = ocommand.Execute(p, new string[] { "look", "at", "b52" });
            string desc = $"{oitem.FullDescription}";
            Assert.AreEqual(desc, outputc);
        }

        [Test]
        public void TestLookAtUnk()
        {
            string outputc = ocommand.Execute(p, new string[] { "look", "at", "b52" });
            string desc = "I cannot find the b52";
            Assert.AreEqual(desc, outputc);
        }

        [Test]
        public void TestLookAtGemInMe()
        {
            p.Inventory.Put(oitem);
            string outputc = ocommand.Execute(p, new string[] { "look", "at", "b52", "in", "me" });
            string desc = $"{oitem.FullDescription}";
            Assert.AreEqual(desc, outputc);
        }

        [Test]
        public void TestLookAtGemInBag()
        {
            obag.Inventory.Put(oitem);
            string outputc = ocommand.Execute(p, new string[] { "look", "at", "b52", "in", "bag" });
            string desc = $"{oitem.FullDescription}";
            Assert.AreEqual(desc, outputc);
        }


        [Test]
        public void TestLookAtGemInNoBag()
        {
            obag.Inventory.Put(oitem);
            oplayer.Inventory.Put(obag);
            string outputc = ocommand.Execute(oplayer, new string[] { "look", "at", "b52", "in", $"{p.FirstId}" });
            string desc = $"I cannot find the b52";
            Assert.AreEqual(desc, outputc);
        }

        [Test]
        public void TestLookAtNoGemInBag()
        {
            obag.Inventory.Put(oitem);
            string outputc = ocommand.Execute(p, new string[] { "look", "at", "bag", "in", "b52" });
            string desc = "I cannot find the b52";
            Assert.AreEqual(desc, outputc);
        }
    }
}
```



```
using NUnit.Framework;
using Casestudy_Iteration4;
using CaseStudy_Iteration4;

namespace CaseStudy_Iteration4
{
    0 references
    public class Tests
    {
        private Player p, oplayer;
        private Bag obag;
        private Command ocommand;

        private Item oitem;

        [SetUp]
        0 references
        public void Setup()
        {
            oitem = new Item(new s
            ocommand = new LookCom
            oplayer = new Player("
            p = new Player("Tung",
            obag = new Bag(new str
            p.Inventory.Put(obag);
        }

        [Test]
        | 0 references
        public void TestLookAtMe()
        {
            string outputc = ocomm
            string desc = $"{p.Ful
            Assert.AreEqual(desc, outputc);
        }
```

Test run finished: 7 Tests (7 Passed, 0 Failed, 0 Skipped) run in 188 ms

0 Warning 0 Errors

Test	Duration	Traits	Error Message
TestProject1 (7)	9 ms		
CaseStudy_Iteration4 (7)	9 ms		
Tests (7)	9 ms		
TestLookAtGem	7 ms		
TestLookAtGemInBag	< 1 ms		
TestLookAtGemInMe	< 1 ms		
TestLookAtGemInNoBag	2 ms		
TestLookAtMe	< 1 ms		
TestLookAtNoGemInBag	< 1 ms		
TestLookAtUnk	< 1 ms		

Group Summary

TestProject1
Tests in group: 7
Total Duration: 9 ms

Outcomes
7 Passed

Show output from: Build

```
1>------ Build started: Project: TestProject1, Configuration: Debug Any CPU ------
1>Skipping analyzers to speed up the build. You can execute 'Build' or 'Rebuild' command to run analyzers.
1>TestProject1 -> C:\Users\Tung\Desktop\COS20007\week6\CaseStudy-Iteration4\TestProject1\bin\Debug\net6.0\TestProject1.dll
========== Build: 1 succeeded, 0 failed, 1 up-to-date, 0 skipped ==========
========== Build started at 22:08 and took 02.346 seconds ==========
```

Test run finished: 7 Tests (7 Passed, 0 Failed, 0 Skipped) run in 188 ms