Swinburne University of Technology

COS20007 Object Oriented Programming

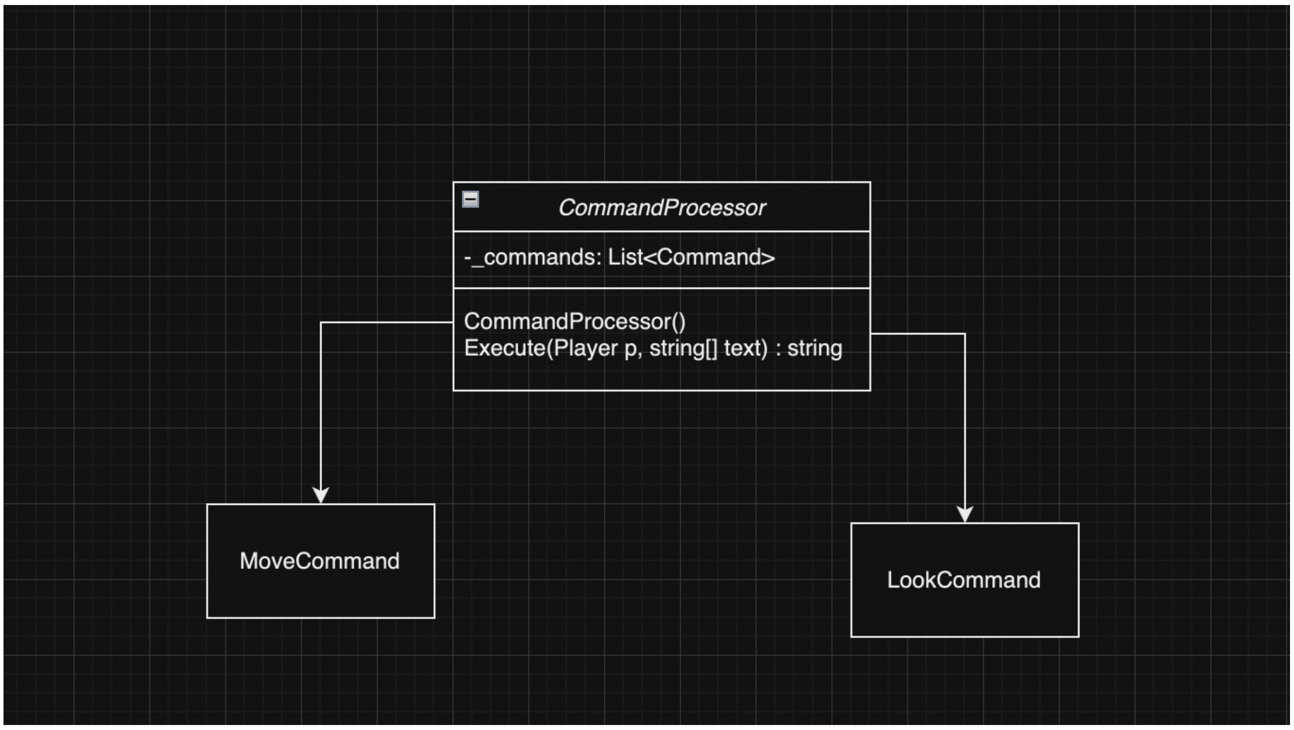
Case Study - Iteration 8 - Command Processor

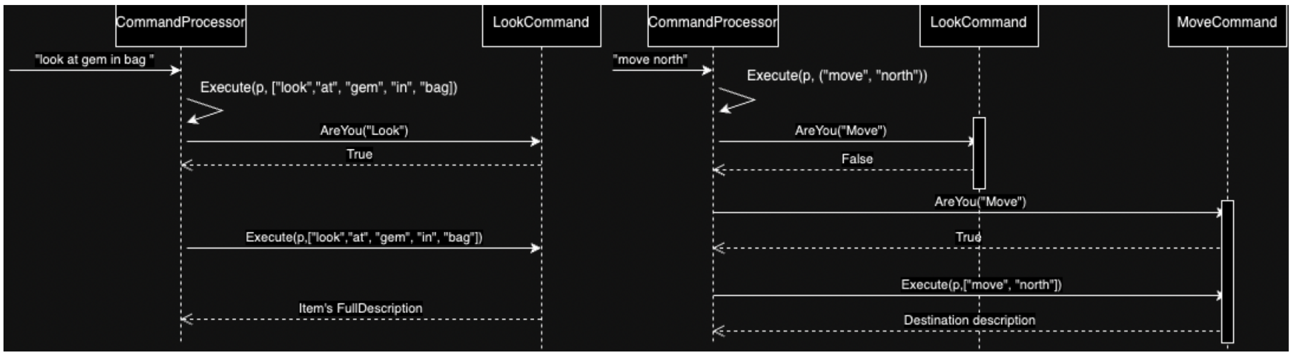
PDF generated at 22:20 on Thursday 26th October, 2023

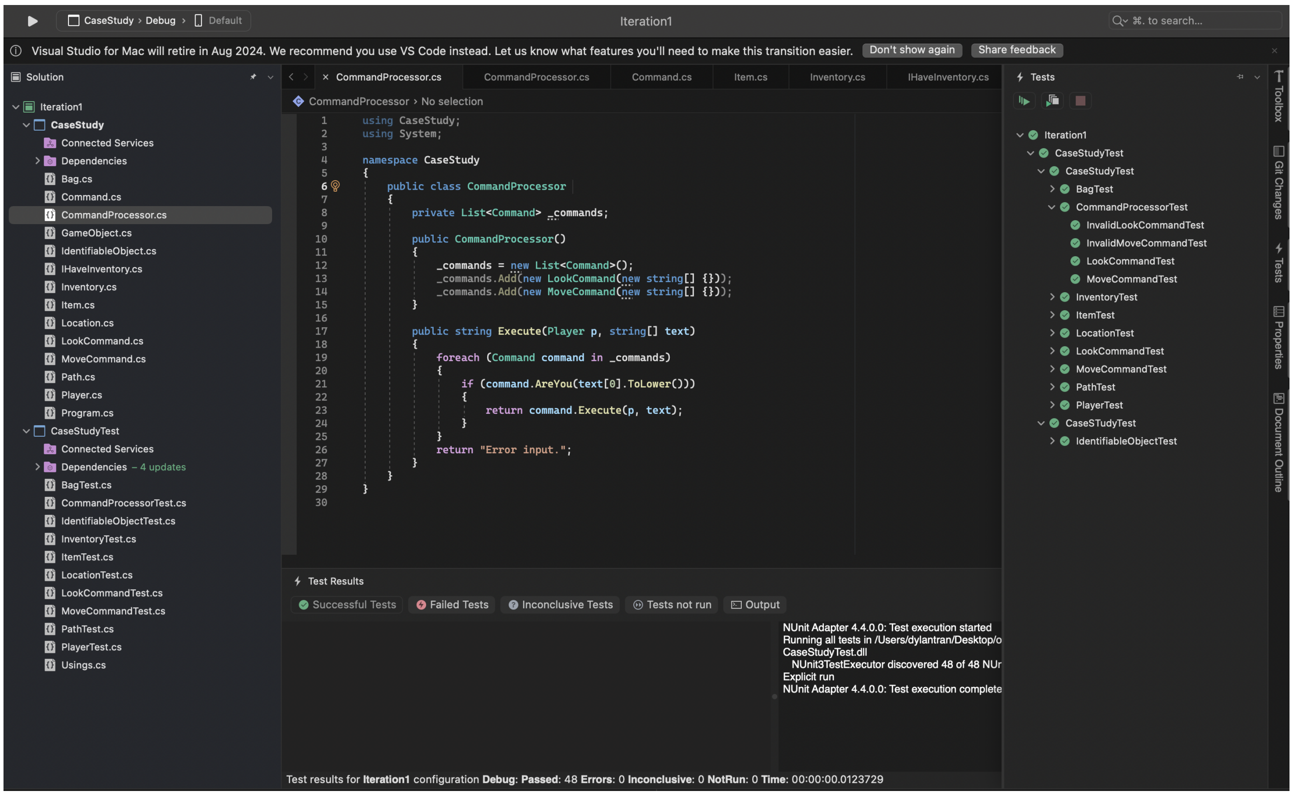
```
using System;

namespace CaseStudy
{
    class MainClass
    {

        static void Main(string[] args)
        {
            Console.WriteLine("Welcome to Swin Adventure!");
            Console.Write("Enter your player name: ");
            string name = Console.ReadLine();
            Console.Write("Enter your player description: ");
            string description = Console.ReadLine();
            string location = "home";
            Player player = new Player(name, description);

            Item shovel = new Item(new string[] { "shovel" }, "a shovel", "This is a:
↪   shovel");
            Item sword = new Item(new string[] { "sword" }, "a sword", "This is a:
↪   sword");
            player.Inventory.Put(shovel);
            player.Inventory.Put(sword);

            Bag bag = new Bag(new string[] { "bag" }, "bag", "This is a bag");
            player.Inventory.Put(bag);

            Item gem = new Item(new string[] { "gem" }, "a gem", "This is a gem");
            bag.Inventory.Put(gem);

            Item pen = new Item(new string[] { "pen" }, "a pen", "This is a pen");
            Location myclass = new Location(new string[] { "classroom" }, "My
↪   classroom", "My classroom");
            player.Location = myclass;
            player.Location.Container.Put(pen);

            Item pencil = new Item(new string[] { "pencil" }, "a pencil", "This is a
↪   pencil");
            Location oopclass = new Location(new string[] { "OOPclassroom" }, "OOP
↪   Class", "OOP Class");
            Path classtooop = new Path(new string[] { "right" }, "door", "travel
↪   through door", oopclass);
            Path ooptoclass = new Path(new string[] { "left" }, "door", "travel
↪   through door", myclass);
            myclass.AddPath(classtooop);
            oopclass.AddPath(ooptoclass);
            oopclass.Container.Put(pencil);

            // Console.WriteLine("Type 'quit' to exit.");

            string _input;
            CommandProcessor c = new CommandProcessor();
            while (true)
            {
                Console.Write("Command: ");
                _input = Console.ReadLine();

                if (_input.ToLower() != "quit")
                {
                    Console.WriteLine(c.Execute(player, _input.Split()));
                }
                else
                {
                    Console.WriteLine("Bye");
                    Console.ReadKey();
                    break;
                }
            }
        }
    }
}
```

```
using CaseStudy;
using System;

namespace CaseStudy
{
    public class CommandProcessor
    {
        private List<Command> _commands;

        public CommandProcessor()
        {
            _commands = new List<Command>();
            _commands.Add(new LookCommand(new string[] {}));
            _commands.Add(new MoveCommand(new string[] {}));
        }

        public string Execute(Player p, string[] text)
        {
            foreach (Command command in _commands)
            {
                if (command.AreYou(text[0].ToLower()))
                {
                    return command.Execute(p, text);
                }
            }
            return "Error input.";
        }
    }
}
```

```
using System;
using CaseStudy;

namespace CaseStudyTest
{
    public class CommandProcessorTest
    {
        private Player _p;
        private Location _loca;
        private Location _locb;
        private CaseStudy.Path _path;
        private CommandProcessor _command;
        private Item _gem;
        private string _output;

        [SetUp]
        public void Setup()
        {
            _p = new Player("Duc", "this is Vu Duc Tran");
            _gem = new Item(new string[] { "gem" }, "a gem", "This is a gem");
            _loca = new Location(new string[] { "" }, "Classroom", "Swinburne
↪   University");
            _locb = new Location(new string[] { "ENbuilding" }, "ENbuilding",
↪   "ENbuilding");
            _path = new CaseStudy.Path(new string[] { "north" }, "ENbuilding",
↪   "classroom to ENbuilding", _locb);
            _command = new CommandProcessor();

            _p.Location = _loca;
            _loca.AddPath(_path);
            _p.Inventory.Put(_gem);

        }
        [Test]
        public void MoveCommandTest()
        {
            string cmt = _command.Execute(_p, new string[] { "move", "north" });

            Assert.That(_p.Location, Is.EqualTo(_locb), cmt + new string[] {"move",
↪   "north"}.Length.ToString());
        }

        [Test]
        public void InvalidMoveCommandTest()
        {
            _output = _command.Execute(_p, new string[] { "move", "south" });

            Assert.That(_output, Is.EqualTo("Error in move input."), "Test Invalid
↪   Look At Me");
        }

        [Test]
        public void LookCommandTest()
        {
            _output = _command.Execute(_p, new string[] { "look", "at", "inventory" });
            Assert.That(_output, Is.EqualTo($"You are {_p.Name}, this is Vu Duc Tran.\nYou are carrying:\n{_p.Inventory.ItemList}\n"), "Test Look At Me");
        }

        [Test]
        public void InvalidLookCommandTest()
        {
            _output = _command.Execute(_p, new string[] { "see", "at", "inventory" });
            Assert.That(_output, Is.EqualTo("Error input."), "Test Invalid Look At Me");
        }
    }
}
```


CommandProcessor
-_commands: List<Command>
CommandProcessor()
Execute(Player p, string[] text) : string
MoveCommand
LookCommand


CommandProcessor
LookCommand
"look at gem in bag "
Execute(p, ["look","at", "gem", "in", "bag])
AreYou("Look")
True
Execute(p,["look","at", "gem", "in", "bag"])
Item's FullDescription
CommandProcessor
LookCommand
MoveCommand
"move north"
Execute(p, ("move", "north"))
AreYou("Move")
False
AreYou("Move")
True
Execute(p,["move", "north"])
Destination description



Visual Studio for Mac will retire in Aug 2024. We recommend you use VS Code instead. Let us know what features you'll need to make this transition easier.

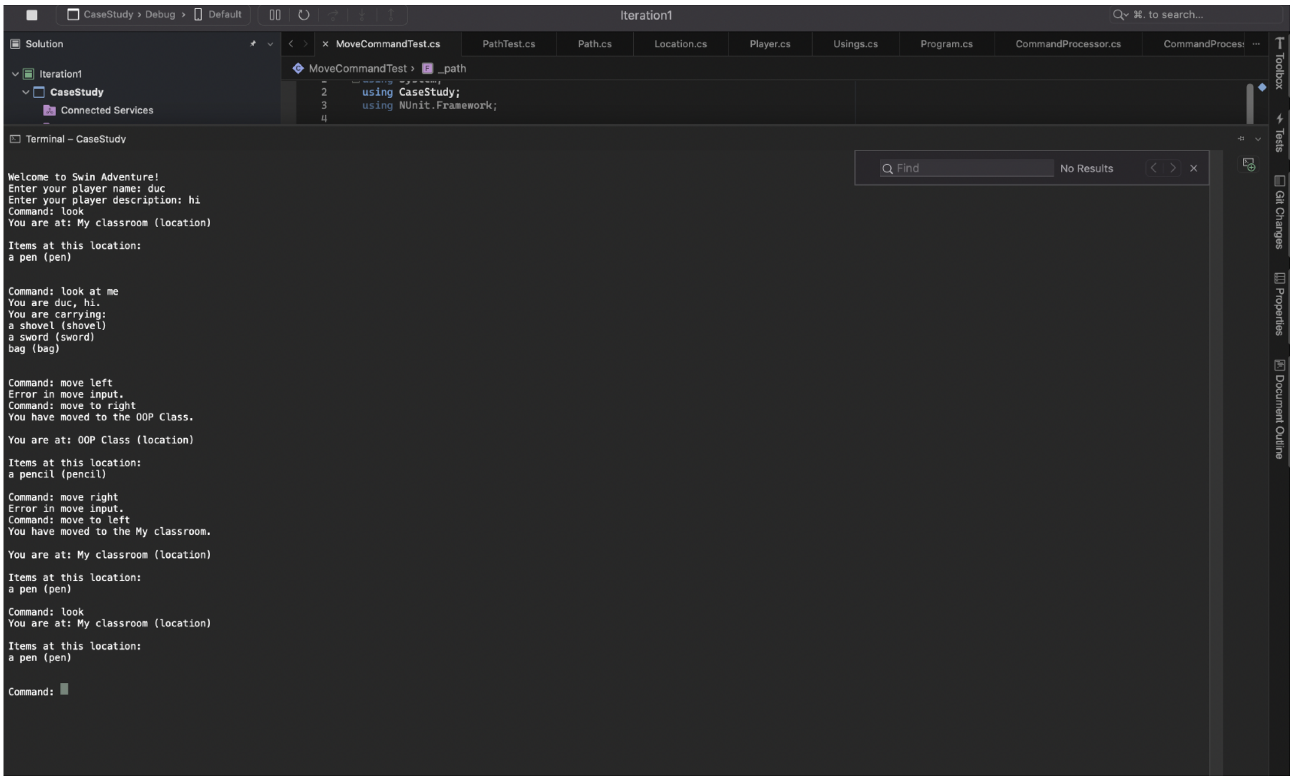
```
Iteration1
  CaseStudy
    Connected Services
    Dependencies
    Bag.cs
    Command.cs
    CommandProcessor.cs
    GameObject.cs
    IdentifiableObject.cs
    IHaveInventory.cs
    Inventory.cs
    Item.cs
    Location.cs
    LookCommand.cs
    MoveCommand.cs
    Path.cs
    Player.cs
    Program.cs
  CaseStudyTest
    Connected Services
    Dependencies – 4 updates
    BagTest.cs
    CommandProcessorTest.cs
    IdentifiableObjectTest.cs
    InventoryTest.cs
    ItemTest.cs
    LocationTest.cs
    LookCommandTest.cs
    MoveCommandTest.cs
    PathTest.cs
    PlayerTest.cs
    Usings.cs
```

```csharp
using CaseStudy;
using System;

namespace CaseStudy
{
    public class CommandProcessor
    {
        private List<Command> _commands;

        public CommandProcessor()
        {
            _commands = new List<Command>();
            _commands.Add(new LookCommand(new string[] {}));
            _commands.Add(new MoveCommand(new string[] {}));
        }

        public string Execute(Player p, string[] text)
        {
            foreach (Command command in _commands)
            {
                if (command.AreYou(text[0].ToLower()))
                {
                    return command.Execute(p, text);
                }
            }
            return "Error input.";
        }
    }
}
```

```
Iteration1
  CaseStudyTest
    CaseStudyTest
      BagTest
      CommandProcessorTest
        InvalidLookCommandTest
        InvalidMoveCommandTest
        LookCommandTest
        MoveCommandTest
      InventoryTest
      ItemTest
      LocationTest
      LookCommandTest
      MoveCommandTest
      PathTest
      PlayerTest
    CaseSTudyTest
      IdentifiableObjectTest
```

```
NUnit Adapter 4.4.0.0: Test execution started
Running all tests in /Users/dylantran/Desktop/o
CaseStudyTest.dll
  NUnit3TestExecutor discovered 48 of 48 NUr
Explicit run
NUnit Adapter 4.4.0.0: Test execution complete
```

Test results for **Iteration1** configuration **Debug**: **Passed**: 48 **Errors**: 0 **Inconclusive**: 0 **NotRun**: 0 **Time**: 00:00:00.0123729



```
using CaseStudy;
using NUnit.Framework;
```

```
Welcome to Swin Adventure!
Enter your player name: duc
Enter your player description: hi
Command: look
You are at: My classroom (location)

Items at this location:
a pen (pen)


Command: look at me
You are duc, hi.
You are carrying:
a shovel (shovel)
a sword (sword)
bag (bag)


Command: move left
Error in move input.
Command: move to right
You have moved to the OOP Class.

You are at: OOP Class (location)

Items at this location:
a pencil (pencil)

Command: move right
Error in move input.
Command: move to left
You have moved to the My classroom.

You are at: My classroom (location)

Items at this location:
a pen (pen)

Command: look
You are at: My classroom (location)

Items at this location:
a pen (pen)


Command:
```