

**main**

[**UploadedCodeToPrintAgain**](https://github.com/KingSchlock/UploadedCodeToPrintAgain) / **IHaveInventory.cs** / Jump to



**1** contributor

[**KingSchlock**](https://github.com/KingSchlock) [Add files via upload](https://github.com/KingSchlock/UploadedCodeToPrintAgain/commit/7d7c8ad6d3cf32ea91842b4a3bf83da93481846c)

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

using System;

using System.Collections.Generic; using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace SwinAdventure

{

public interface IHaveInventory

{

//! Everything that has an inventory has to have the ability to

//! locate items.

GameObject Locate(string id);

public string Name

{

get;

}

}

}

20 lines (18 sloc) 417 Bytes

https://github.com/KingSchlock/UploadedCodeToPrintAgain/blob/main/IHaveInventory.cs 1/1

# [KingSchlock](https://github.com/KingSchlock) / [**UploadedCodeToPrintAgain**](https://github.com/KingSchlock/UploadedCodeToPrintAgain)

**Private**

[**Code**](https://github.com/KingSchlock/UploadedCodeToPrintAgain)[Issues](https://github.com/KingSchlock/UploadedCodeToPrintAgain/issues) [Pull requests](https://github.com/KingSchlock/UploadedCodeToPrintAgain/pulls) [Actions](https://github.com/KingSchlock/UploadedCodeToPrintAgain/actions) [Projects](https://github.com/KingSchlock/UploadedCodeToPrintAgain/projects) [Security](https://github.com/KingSchlock/UploadedCodeToPrintAgain/security) [Insights](https://github.com/KingSchlock/UploadedCodeToPrintAgain/network/dependencies) [Settings](https://github.com/KingSchlock/UploadedCodeToPrintAgain/settings)



**main**

[**UploadedCodeToPrintAgain**](https://github.com/KingSchlock/UploadedCodeToPrintAgain) / **Bag.cs** / Jump to



**1** contributor

[**KingSchlock**](https://github.com/KingSchlock) [Add files via upload](https://github.com/KingSchlock/UploadedCodeToPrintAgain/commit/7d7c8ad6d3cf32ea91842b4a3bf83da93481846c)

31 lines (26 sloc) 727 Bytes

1 namespace SwinAdventure 2 {

3 public class Bag : Item, IHaveInventory 4 {

5 Inventory \_inventory = new(); 6

7 public Bag(string[] idents, string name, string description) : base(idents, name, desc 8 {

9

10 }

11

12 public Inventory Inventory

13 {

14 get { return \_inventory; } 15 }

16

17 public override string FullDescription

18 {

19 get { return $"In the {Name} you can see\n{\_inventory.ItemList}"; } 20 }

21

22 public GameObject Locate(string id)

23 {

24 if(this.AreYou(id) == true)

25 {

26 return this;

27 }

28 return \_inventory.Fetch(id); 29 }

30 }

31 }

https://github.com/KingSchlock/UploadedCodeToPrintAgain/blob/main/Bag.cs 1/1

# [KingSchlock](https://github.com/KingSchlock) / [**UploadedCodeToPrintAgain**](https://github.com/KingSchlock/UploadedCodeToPrintAgain)

**Private**

[**Code**](https://github.com/KingSchlock/UploadedCodeToPrintAgain)[Issues](https://github.com/KingSchlock/UploadedCodeToPrintAgain/issues) [Pull requests](https://github.com/KingSchlock/UploadedCodeToPrintAgain/pulls) [Actions](https://github.com/KingSchlock/UploadedCodeToPrintAgain/actions) [Projects](https://github.com/KingSchlock/UploadedCodeToPrintAgain/projects) [Security](https://github.com/KingSchlock/UploadedCodeToPrintAgain/security) [Insights](https://github.com/KingSchlock/UploadedCodeToPrintAgain/network/dependencies) [Settings](https://github.com/KingSchlock/UploadedCodeToPrintAgain/settings)



**main**

[**UploadedCodeToPrintAgain**](https://github.com/KingSchlock/UploadedCodeToPrintAgain) / **Player.cs** / Jump to



**1** contributor

[**KingSchlock**](https://github.com/KingSchlock) [Add files via upload](https://github.com/KingSchlock/UploadedCodeToPrintAgain/commit/7d7c8ad6d3cf32ea91842b4a3bf83da93481846c)

|  |  |
| --- | --- |
| 39 lines (35 sloc) 1.31 KB | |
| 1 | using System; |
| 2 | using System.Collections.Generic; |
| 3 | using System.Linq; |
| 4 | using System.Text; |
| 5 | using System.Threading.Tasks; |
| 6 |  |
| 7 | namespace SwinAdventure |
| 8 | { |
| 9 | public class Player : GameObject, IHaveInventory //x TODO Implement Inventory field, prope |
| 10 | { |
| 11 | Inventory \_inventory = new Inventory(); |
| 12 | public Player(string name, string description) : base(new string[] {"me", "inventory"} |
| 13 | { |
| 14 |  |
| 15 | } |
| 16 |  |
| 17 | public Inventory Inventory |
| 18 | { |
| 19 | get { return \_inventory; } |
| 20 | } |
| 21 |  |
| 22 | public override string FullDescription //! Can only override virtual properties |
| 23 | { |
| 24 | get { return $"You are {Name} {base.FullDescription}.\nYou are carrying\n{Inventor |
| 25 | } |
| 26 | public GameObject Locate(string id) //! Checks if the player holds an object with id |
| 27 | { |
| 28 | if (this.AreYou(id) == true) |
| 29 | { |
| 30 | return this; // returns this object |
| 31 | } |
| 32 | return \_inventory.Fetch(id); // if the object isn't around then check our inventor |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 33 | | } | } | } | /\*! NOTE:   * The Locate operation should return null if no objects match id as the defa * Fetch is null.   \*/ | |
| 34 | |
| 35 | |
| 36 | |
| 37 | |
| 38 | |
| 39 | |
|  |  | | | | |  |

# [KingSchlock](https://github.com/KingSchlock) / [**UploadedCodeToPrintAgain**](https://github.com/KingSchlock/UploadedCodeToPrintAgain)

**Private**

[**Code**](https://github.com/KingSchlock/UploadedCodeToPrintAgain)[Issues](https://github.com/KingSchlock/UploadedCodeToPrintAgain/issues) [Pull requests](https://github.com/KingSchlock/UploadedCodeToPrintAgain/pulls) [Actions](https://github.com/KingSchlock/UploadedCodeToPrintAgain/actions) [Projects](https://github.com/KingSchlock/UploadedCodeToPrintAgain/projects) [Security](https://github.com/KingSchlock/UploadedCodeToPrintAgain/security) [Insights](https://github.com/KingSchlock/UploadedCodeToPrintAgain/network/dependencies) [Settings](https://github.com/KingSchlock/UploadedCodeToPrintAgain/settings)



**main**

[**UploadedCodeToPrintAgain**](https://github.com/KingSchlock/UploadedCodeToPrintAgain) / **LookCommand.cs** / Jump to



**1** contributor

[**KingSchlock**](https://github.com/KingSchlock) [Add files via upload](https://github.com/KingSchlock/UploadedCodeToPrintAgain/commit/7d7c8ad6d3cf32ea91842b4a3bf83da93481846c)

|  |  |
| --- | --- |
| 81 lines (68 sloc) 2.16 KB | |
| 1 | using System; |
| 2 | using System.Collections.Generic; |
| 3 | using System.Linq; |
| 4 | using System.Text; |
| 5 | using System.Threading.Tasks; |
| 6 |  |
| 7 | namespace SwinAdventure |
| 8 | { |
| 9 | public class LookCommand : Command |
| 10 | { |
| 11 | public LookCommand() : base(new string[] { "look" }) |
| 12 | { |
| 13 |  |
| 14 | } |
| 15 |  |
| 16 | //! A series of checks which run when the look command is used |
| 17 | //! Returns the same as LookAtIn |
| 18 | public override string Execute(Player player, string[] text) |
| 19 | { |
| 20 | IHaveInventory container; |
| 21 | string thingId; |
| 22 |  |
| 23 | if (text.Length != 3 && text.Length != 5) |
| 24 | { |
| 25 | return "I don't know how to look for that."; |
| 26 | } |
| 27 |  |
| 28 | if (text[0] != "look") |
| 29 | { |
| 30 | return "Error in look input"; |
| 31 | } |
| 32 |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 33 | |  |  | if (text[1] != "at") | |
| 34 | |  |  | { | |
| 35 | |  |  | return "What do you want to look at?"; | |
| 36 | |  |  | } | |
| 37 | |  |  |  | |
| 38 | |  |  | if (text.Length == 5 && text[3] != "in") | |
| 39 | |  |  | { | |
| 40 | |  |  | return "What do you want to look in?"; | |
| 41 | |  |  | } | |
| 42 | |  |  |  | |
| 43 | |  |  |  | |
| 44 | |  |  | if (text.Length == 3) | |
| 45 | |  |  | { | |
| 46 | |  |  | container = player; | |
| 47 | |  |  | } | |
| 48 | |  |  | else | |
| 49 | |  |  | { | |
| 50 | |  |  | container = FetchContainer(player, text[4]); | |
| 51 | |  |  | } | |
| 52 | |  |  |  | |
| 53 | |  |  | if (container == null) | |
| 54 | |  |  | { | |
| 55 | |  |  | return $"I can't find the {text[4]}"; | |
| 56 | |  |  | } | |
| 57 | |  |  |  | |
| 58 | |  |  | thingId = text[2]; | |
| 59 | |  |  | return LookAtIn(thingId, container); | |
| 60 | |  |  | } | |
| 61 | |  |  |  | |
| 62 | |  |  | //! Grabs a container based on a string | |
| 63 | |  |  | private IHaveInventory FetchContainer(Player player, string containerId) | |
| 64 | |  |  | { | |
| 65 | |  |  | return player.Locate(containerId) as IHaveInventory; | |
| 66 | |  |  | } | |
| 67 | |  |  |  | |
| 68 | |  |  | //! checks if the thing requested exists inside a container, if so return it's full de | |
| 69 | |  |  | private string LookAtIn(string thingId, IHaveInventory container) | |
| 70 | |  |  | { | |
| 71 | |  |  | if(container.Locate(thingId) == null) | |
| 72 | |  |  | { | |
| 73 | |  |  | return $"I can't find the {thingId}"; | |
| 74 | |  |  | } | |
| 75 | |  |  | else | |
| 76 | |  |  | { | |
| 77 | |  |  | return container.Locate(thingId).FullDescription; | |
| 78 | |  |  | } | |
| 79 | |  |  | } | |
| 80 | |  | } |  | |
| 81 | | } |  |  | |
|  |  | | | |  |

# [KingSchlock](https://github.com/KingSchlock) / [**UploadedCodeToPrintAgain**](https://github.com/KingSchlock/UploadedCodeToPrintAgain)

**Private**

[**Code**](https://github.com/KingSchlock/UploadedCodeToPrintAgain)[Issues](https://github.com/KingSchlock/UploadedCodeToPrintAgain/issues) [Pull requests](https://github.com/KingSchlock/UploadedCodeToPrintAgain/pulls) [Actions](https://github.com/KingSchlock/UploadedCodeToPrintAgain/actions) [Projects](https://github.com/KingSchlock/UploadedCodeToPrintAgain/projects) [Security](https://github.com/KingSchlock/UploadedCodeToPrintAgain/security) [Insights](https://github.com/KingSchlock/UploadedCodeToPrintAgain/network/dependencies) [Settings](https://github.com/KingSchlock/UploadedCodeToPrintAgain/settings)



**main**

[**UploadedCodeToPrintAgain**](https://github.com/KingSchlock/UploadedCodeToPrintAgain) / **TestLookCommand.cs** / Jump to



**1** contributor

[**KingSchlock**](https://github.com/KingSchlock) [Add files via upload](https://github.com/KingSchlock/UploadedCodeToPrintAgain/commit/7d7c8ad6d3cf32ea91842b4a3bf83da93481846c)

|  |  |
| --- | --- |
| 116 lines (98 sloc) 4.19 KB | |
| 1 | using SwinAdventure; |
| 2 |  |
| 3 | namespace SwinAdventureTests |
| 4 | { |
| 5 | [TestFixture()] |
| 6 | public class TestLookCommand |
| 7 | { |
| 8 | LookCommand lookTest; |
| 9 | Player playerTest; |
| 10 | Bag bagTest; |
| 11 | Item swordTest; |
| 12 |  |
| 13 | string unknown, noBag, |
| 14 | badLength, badLook, |
| 15 | badAt, badIn; |
| 16 |  |
| 17 | [SetUp()] |
| 18 | public void Setup() |
| 19 | { |
| 20 | lookTest = new(); |
| 21 | playerTest = new("thomas", "The mighty keyboard warrior"); |
| 22 | bagTest = new(new string[] { "satchel" }, "satchel", "it's smol"); |
| 23 | swordTest = new(new string[] { "sword" }, "sword", "lil poker"); |
| 24 |  |
| 25 | unknown = "I can't find the sword"; |
| 26 | noBag = $"I can't find the {bagTest.Name}"; |
| 27 |  |
| 28 | badLength = "I don't know how to look for that."; |
| 29 | badLook = "Error in look input"; |
| 30 | badAt = "What do you want to look at?"; |
| 31 | badIn = "What do you want to look in?"; |
| 32 |  |

|  |  |
| --- | --- |
| 33 | playerTest.Inventory.Put(swordTest); |
| 34 | } |
| 35 |  |
| 36 | //! Return players descript when looking at the inventory |
| 37 | [Test()] |
| 38 | public void TestLookAtMe() |
| 39 | { |
| 40 | Assert.That(lookTest.Execute(playerTest, new string[] {"look", "at", "me"}), |
| 41 | Is.EqualTo(playerTest.FullDescription)); |
| 42 | } |
| 43 |  |
| 44 | //! Returns item description when looking for an item in players invent |
| 45 | [Test()] |
| 46 | public void TestLookAtItem() |
| 47 | { |
| 48 | Assert.That(lookTest.Execute(playerTest, new string[] {"look", "at", "sword"}), |
| 49 | Is.EqualTo(swordTest.FullDescription)); |
| 50 | } |
| 51 |  |
| 52 | //! Responds unknown when item isn't in inventory |
| 53 | [Test()] |
| 54 | public void TestLookAtUnkn() |
| 55 | { |
| 56 | playerTest.Inventory.Take("sword"); |
| 57 |  |
| 58 | Assert.That(lookTest.Execute(playerTest, new string[] { "look", "at", "sword", "in |
| 59 | Is.EqualTo(unknown)); |
| 60 | } |
| 61 |  |
| 62 | //! Returns item description when searching for item specifically in invent |
| 63 | [Test()] |
| 64 | public void TestLookAtItemInInventory() |
| 65 | { |
| 66 | Assert.That(lookTest.Execute(playerTest, new string[] {"look", "at", "sword", "in" |
| 67 | Is.EqualTo(swordTest.FullDescription)); |
| 68 | } |
| 69 |  |
| 70 | //! Returns item description when searching for it in a bag in players invent |
| 71 | [Test()] |
| 72 | public void TestLookAtItemInBag() |
| 73 | { |
| 74 | playerTest.Inventory.Take("sword"); |
| 75 |  |
| 76 | bagTest.Inventory.Put(swordTest); |
| 77 | playerTest.Inventory.Put(bagTest); |
| 78 |  |
| 79 | Assert.That(lookTest.Execute(playerTest, new string[] {"look","at","sword","in","s |
| 80 | Is.EqualTo(swordTest.FullDescription)); |
| 81 | } |
| 82 |  |
| 83 | //! Returns noBag when there's no container in players invent |
| 84 | [Test()] |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 85 | |  |  | public void TestLookAtItemInNoBag() | |
| 86 | |  |  | { | |
| 87 | |  |  | Assert.That(lookTest.Execute(playerTest, new string[] { "look", "at", "sword", "in | |
| 88 | |  |  | Is.EqualTo(noBag)); | |
| 89 | |  |  | } | |
| 90 | |  |  |  | |
| 91 | |  |  | //! Returns unknown when requested item isn't in bag | |
| 92 | |  |  | [Test()] | |
| 93 | |  |  | public void TestLookAtNoItemInBag() | |
| 94 | |  |  | { | |
| 95 | |  |  | playerTest.Inventory.Put(bagTest); | |
| 96 | |  |  |  | |
| 97 | |  |  | Assert.That(lookTest.Execute(playerTest, new string[] { "look", "at", "sword", "in | |
| 98 | |  |  | Is.EqualTo(unknown)); | |
| 99 | |  |  | } | |
| 100 | |  |  |  | |
| 101 | |  |  | //! Tests all error conditions | |
| 102 | |  |  | public void TestInvalidLook(string look, string result) | |
| 103 | |  |  | { | |
| 104 | |  |  | Assert.Multiple(() => { | |
| 105 | |  |  | Assert.That(lookTest.Execute(playerTest, new string[] {"aaaaa"}), | |
| 106 | |  |  | Is.EqualTo(badLength)); | |
| 107 | |  |  | Assert.That(lookTest.Execute(playerTest, new string[] { "search", "at", "sword | |
| 108 | |  |  | Is.EqualTo(badLook)); | |
| 109 | |  |  | Assert.That(lookTest.Execute(playerTest, new string[] { "look", "for", "sword" | |
| 110 | |  |  | Is.EqualTo(badAt)); | |
| 111 | |  |  | Assert.That(lookTest.Execute(playerTest, new string[] { "look", "for", "sword" | |
| 112 | |  |  | Is.EqualTo(badIn)); | |
| 113 | |  |  | }); //? can i use testcases here? | |
| 114 | |  |  | } | |
| 115 | |  | } |  | |
| 116 | | } |  |  | |
|  |  | | | |  |