**Spike:** Spike 12

**Title:** Command Pattern

**Author:** Bernardo Fitzmaurice Acevedo 105297603

**Goals / deliverables:**

create a modular and robust text command parser and command manager for the game suitable for the Zorkish game. Allow access to the player’s inventory, to some locations’ entities with its details, and to allow remapping of some commands.

Besides this report, what else was created?

For example: UML diagram, code, reports

* VS Community code project for the spike.
* Text file
* UML Diagram

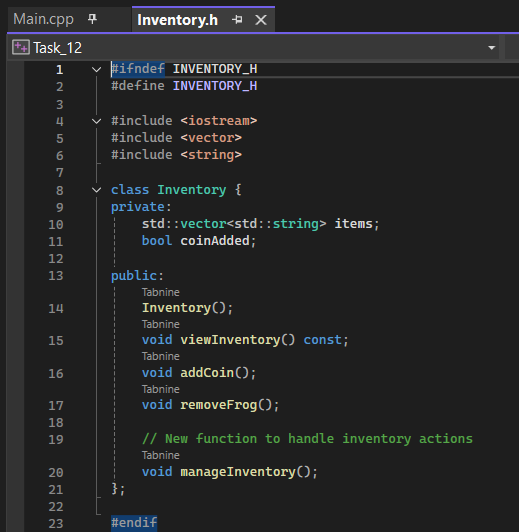
**Technologies, Tools, and Resources used:**

List of information needed by someone trying to reproduce this work

* Visual Studio Community 2022
* Notepad
* Microsoft Word
* Lucid Chart

**Tasks undertaken:**

* Download and install VS community 2022.
* Open the file of the previous task.
* Modify the text file adding entities on the locations with their own descriptions.
* Create an “Inventory” class, allowing the user to store items, adding one item and removing another one.

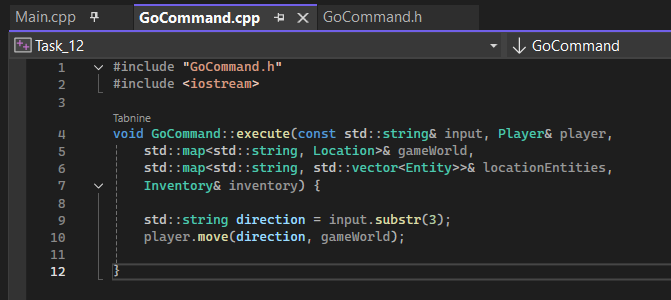


* Create an entity class that gathers the names and descriptions present on the text file.

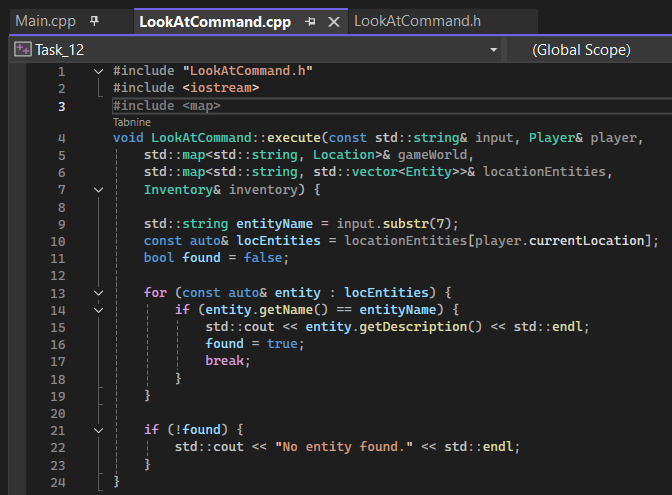
A screen shot of a computer

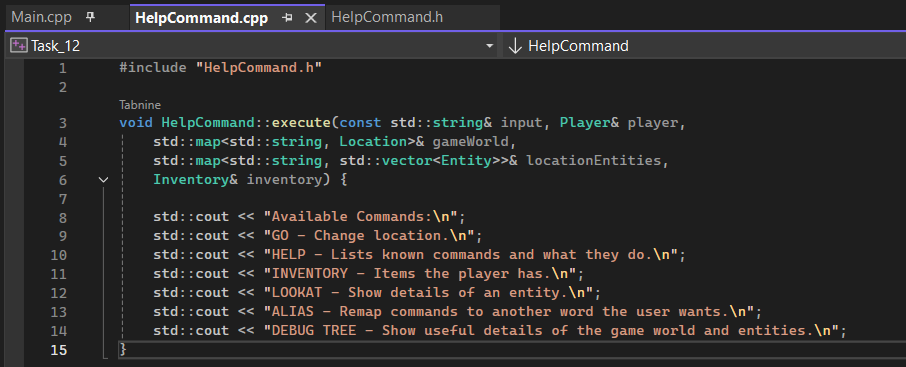
Description automatically generated

* Create a ‘GoCommand’ class that allows the user to move between locations.



* Create a ‘LookAtCommand’ class that allows the user to obtain the details of an entity.



* Create a ‘HelpCommand’ class that allows the user to see what each command do.  
  
* Create an ‘OpenInventoryCommand’ class that allows the user to open and see the inventory they have.  
  A computer screen shot of a program

  Description automatically generated
* Create an ‘AliasCommand’ class that allows the user to remap any of the commands they want, can use the original and the new,  
  A screen shot of a computer

  Description automatically generated
* Create a ‘Debug tree’ class that shows the user important information of the location they’re in.  
  A computer screen shot of a program code

  Description automatically generated
* Create the ‘Command’ class that facilitates the implementation of the previous classes/commands.  
  A computer screen shot of a program

  Description automatically generated
* Create the ‘CMD’ class that inserts the previous classes into a list and process the commands.  
  A computer screen shot of a program

  Description automatically generated
* Create the ‘CMD\_Manager’ class that inherits the CMD list of classes and loops it.  
  A computer screen shot of a program

  Description automatically generated
* Create a UML diagram of all the code’s classes.  
  A diagram of a computer

  Description automatically generated
* Compile code.

A screenshot of a computer program

Description automatically generated

**What we found out:**

During this spike I learned how to create a spike manager, I also understood even more how classes work and how to make inheritance work between multiple classes.