

Edwin Zheng, Lauren Richardson, Brandon Cheng

File

Object-Oriented Programming

03/24/17

Internal Design & Requirement

Description of Classes & Methods

1. TextAdventure - Only contains 1 method (the main). This class functions as an introduction to the game, sort of like start screen of a typical game. This class interacts only with AdventureModel class and is only used to start the game.
 - a. Main -
 - Prints welcoming statement
 - Asks if the user would like to play (if no, terminate. If yes, calls AdventureModel class.)
 - **/**@param:** user input/yes or no
2. AdventureModel - Contains all the methods needed to execute commands such as [look] to get a description of class. This class creates rooms, objects, and the Adventurer. Also displays the commands available to the player and allows the player to move from room to room.
 - a. Beginning -
 - Creates the Room object
 1. Room room = new room();
 - Creates the Adventurer
 1. Adventurer Ninja = new Adventurer();
 - Creates number command loop that displays available commands
 - Calls on other methods to execute command
 - **/**@param** integer input
 - b. Look -
 - Calls on Room class to print a description of what the room looks like.
 - **/**@param:** name
 - **/**@return:** object
 - c. Go -
 - Prints out number command of available rooms the player can enter.
 - Takes input of number command and calls the corresponding object from Room class.
 - **/**@param:** number command
 - d. Take -

- Method that calls Room class and returns the object in the room.
 - Adds said object into ArrayList in Adventurer.
 - `/**@param: number command`
 - `/**@return: object`
- e. Backpack -
 - One of the number command options for when people want to see what's in their backpack.
 - Method that calls method Display in Adventurer class.
 - `/**@param: number command`
- f. Map -
 - One of the number command options for when people want to see what rooms exist.
 - Print statement of all the rooms and how they connect to each other.
 - `/**@param: number command`
- g. End -
 - One of the number command options for when people want to quit the game.
 - Very straight forward `'System.exit(0)'`.
- 3. Adventurer: This class is basically a giant arraylist that also stores your location. When you need to store or retrieve information, you call this class and its methods.
 - a. Location -
 - String value location - stores name of current room, changes every time the adventurer moves into a different room
 - `/**@return: string location` to Room method to print out description of room.
 - b. Display -
 - When called, this method prints out the entire arraylist.
 - Prints arraylist
 - `/**@param: ArrayList`
 - c. AddItem -
 - When called, this method adds the item into the arraylist.
 - `/**@param: Item`
- 4. Room -
 - a. Description -
 - When this method is called, it prints out a string description of what this looks like. This method is called by command [look]
 - `/**@param: location`

■ /**@return: string

Interaction Between Classes

TextAdventure

- Can call AdventureModel class to start the game.
 - Method Beginning

AdventureModel

- Method Beginning can call:
 - Method Take - interacts with class Adventurer and Room
 - Method Backpack - interacts with class Adventurer
 - Method Look - interacts with classes Adventurer and Room
 - Method End - when player wants to quit
 - Method Go - when player wants to move, interacts with Room and Adventurer
 - Method Map - to display map of game,

Adventurer

- Can be called by AdventureModel
 - AddItem - adds item to arrayList
 - Backpack - prints out inventory arrayList
 - Go - sets String location new room

Rooms

- Can be called by AdventureModel
 - Look - returns string room to AdventureModel
 - Take - sets boolean item to false
 - Go - prints out description of new room

Program Control Flow

1. The game begins in the TextAdventure class, it displays a introduction to the game and asks the user if they would like to play. If the user responds yes then TextAdventure calls method beginning in AdventureModel class to begin the game.
2. Method beginning in AdventureModel will print out a description of the adventurer's starting location and a list of commands for the user, it will continue to do so after every instance of user input. What happens next depends on the command the user chooses.
 - a. The initial "Take Backpack" command will set the boolean value inventory to true, allowing the AdventureModel class to call and add items to the arrayList backpack in Adventurer.

- b. A “Look” command will call on the specified array object in the Room class. The Room class will print out a description of itself and any present objects (`item == true`).
- c. A “Take” command will call the display method in Adventurer class and add the item to the inventory arrayList backpack. It will then call on the corresponding array object in the Room class and will set the boolean value item to false.
- d. The “Backpack” command will call on the Adventurer class to print out the inventory arrayList backpack.
- e. A “Go” command will first call on the Adventurer class to verify that the boolean for the specified location is true. It will then call on the corresponding instance in the Room class to print out its description.
- f. The “Read Map” command calls the map method in AdventureModel and prints out a map statement.
- g. The “End” command will terminate the program.