MineCityAPI

Challenge

Create an API in form of a Class for a 2D mining game.

Game description:

There is an infinite 2D map where many players are mining for treasure. Every player starts at (0,0) facing Y+ axis. Players can move only in straight line backward or forwards and can turn left or right by 90 degrees. Players can mine items at any point, and it gets added to their inventory. We want the ability to track distance between players, distances between player and any point on the map. Another game feature is to be find inventory of each player, and group of players who have mined same items.

Specifications:

You will consume a stream of API calls from a text file, process them one at a time, and generate outputs as required.

Following is the description of the APIs:

1. **movePlayer(player: str, distance: int)**: moves the player forward(if distance>0) or backward(distance<0) by a certain distance. distance is in range [-1000, 1000]

2. **turnPlayer(player: str, direction: str)**: turns player in a certain direction. direction can be (left, right)

3. **mineItem(player:str, item: str)**: adds the mined item into player's inventory

4. **lookupInventory(player: str)**: lookup inventory of player and return count of each unique element

5. **lookupItemOwners(item: str)**: return all players that own the specified item

6. **lookupDistance(player1: str, player2: str)**: return Manhattan distance between players

7. **lookupDistance(player: str, x_coord: int, y_coord: int)**: return Manhattan distance between player and the point located at (x_coord, y_coord)

All lookup APIs should return values as well as log/print the output.

In case of ambiguity, you can make reasonable assumptions in your design and state them clearly in your solution.

Example:

(Values starting from # are comments to demonstrate output generated by the api, they will not be part of actual input file)

Sample input:

```
movePlayer:A,2
turnPlayer:B,right
movePlayer:B,-3
movePlayer:C,5
mineItem:C,diamond
mineItem:A,coin
mineItem:B,diamond
lookupInventory:A             # coin
lookupInventory:B             # diamond
lookupItemOwners:diamond      # B,C
lookupDistance:A,B            # 5
lookupDistance:A,-1,1         # 4
```

Sample Output:

```
coin
diamond
B,C
5
4
```

Submission:

Submit your solution in a Github repository and provide it's link. Your solution should have all the Classes related to the API, and a driver class that processes input file and generates an output file by calling your APIs. Provide instructions for setup and execution if necessary. Provide Big-O time and space complexities for each API in your design.

Judgment Criteria:

1. Systematic use of Github
2. Efficient use of data structures
3. Well documented code