Snakes and Ladders for N Players

January 15, 2021

1 Pritish Budhiraja

2 1805317

3 House Loki

[1]:
```
# Importing random function for random.randint() to generate numbers between 1 to 6(both inclusive)
import random
```

[2]:
```
# Position of Snakes.
# Stored in a dictionary format where key is the start position and value is the position where snake will end.
posSnakes={
    17:7,
    54:34,
    62:19,
    98:79
}
# Position of Ladders.
# Stored in a dictionary format where key is the start position and value is the position where Ladder will take.
posLadders={
    3:38,
    24:33,
    42:93,
    72:84
}
```

[3]:
```
"""
It's a function that will return the command that user wants.
If user wants to write a automatic input he/she will write roll, it will send a random number betweem 1 to 6.
If user wants to manually enter User has a span of 1 to 20. Between these it's a valid input.
If user wants to quit, he/she can write quit and the player turn will stop and rest player can play until only one player is left.
```

```
If Anything gibrish player has typed, it will print Incorrect Move and give␣
↪that player a second chance.
"""
def userCommand(playerState):
    while True:
        playerCommand=input("Enter roll for automatic input or for manual input␣
↪Enter a number between 1 to 20 for "+playerState["name"]+": ")
        if(playerCommand == "roll"):
            return random.randint(1,6)
        elif(playerCommand == "quit"):
            return playerCommand
        elif(playerCommand.isdigit()):
            try:
                value=int(playerCommand)
                if(value < 0 or value > 20):
                    raise Exception()
                return value
            except:
                print("Enter Move between 0 to 20")
                continue
        else:
            print("Incorrect Move entered.")
```

[4]:
```
"""
It's a function that will tell about the Player Move and Current Position(cp)
"""
def playerMove(playerState,value):
    if(playerState["cp"]+value<=100):
        playerState["cp"]=playerState["cp"]+value;
        if(checkWin(playerState["cp"])):
            return "Win"
        playerState["cp"]=checkSnake(playerState["cp"])
        playerState["cp"]=checkLadder(playerState["cp"])
    else:
        print("Oops!! Beyond the max limit. You gone too far. Try again in next␣
↪turn!!")
    return 0
```

[5]:
```
"""
This function is used to check whether the player has won the game or not.
If won it will return a True to playerMove function.
"""
def checkWin(value):
    if(value==100):
        return True
```

[6]:
```
"""
This function is used to check whether the player has stepped on a snake or not.
 ↪
If the player has stepped it will return the final position where that player␣
 ↪is taken else it will return the same position
that has been passed to it.
"""
def checkSnake(currentValue):
    if currentValue in posSnakes:
        print("Oops. You stepped on a Snake at " + str(currentValue))
        return posSnakes[currentValue];
    else:
        return currentValue;
```

[7]:
```
"""
This function is used to check whether the player has stepped on a ladder or␣
 ↪not.
If the player has stepped it will return the final position where that player␣
 ↪is taken else it will return the same position
that has been passed to it.
"""
def checkLadder(currentValue):
    if currentValue in posLadders:
        print("Great. You stepped onto a ladder at " + str(currentValue))
        return posLadders[currentValue];
    else:
        return currentValue;
```

[12]:
```
"""
This function is used to take number of players that are playing.
If a list of dictionary or we can say a list of objects. And each object store␣
 ↪three things.
1)playerName-To show which player has won the game or the move taken by which␣
 ↪player.
2)currentPosition(cp)-To store the current position of each player. Initially,␣
 ↪cp=0.
3)and isPlaying- It is a boolean variable which stores that the current player␣
 ↪is playing or not. Used to check whether
he/she has quited or not. If he/she Quit then isPlaying will change to False.␣
 ↪Initially, all are playing so it is True.
flag variable is used to get out of nested loops.
count is used to check how many players are still playing. If only one player␣
 ↪left, it will be declared as winner.

"""
def start():
```

```
    print("                                    Welcome to Snakes and Ladders      ␣
↪                              ")
    n=int(input("Enter number of players: "))
    gameState=[]
    for i in range(n):
        name=input("Enter player name: ")
        playerState={
            "name":name,
            "cp":0,
            "isPlaying":True
        }
        gameState.append(playerState)
    count=n
    flag=0
    while True:
        for i in range(n):
            if(gameState[i]["isPlaying"]):
                val=userCommand(gameState[i])
                if val=="quit":
                    count-=1;
                    gameState[i]["isPlaying"]=False
                    if(count==1):
                        for j in range(n):
                            if(gameState[j]["isPlaying"]==True):
                                print("All players Quit except␣
↪"+gameState[j]["name"])
                                print(gameState[j]["name"]+" Wins!!")
                                break
                        flag=1
                        break
                else:
                    print("You got a "+str(val))
                    output=playerMove(gameState[i],val)
                    if(output=="Win"):
                        print(gameState[i]["name"]+" Wins!!")
                        flag=1
                        break
                    print(gameState[i]["name"]+"'s Final position is:␣
↪"+str(gameState[i]["cp"]))
        if(flag==1):
            print("                                    #### Game Successfully␣
↪Finished ####                          ")
            break
```

[13]:
```
"""
It's a main function which will call the start function ,do and call all the␣
↪functionalities.
```

```
"""
start()
```

```
                                Welcome to Snakes and Ladders
Enter number of players: 2
Enter player name: P
Enter player name: M
Enter roll for automatic input or for manual input Enter a number between 1 to
20 for P: quit
All players Quit except M
M Wins!!
                                #### Game Successfully Finished ####
```

[ ]: