ISTANBUL TECHNICAL UNIVERSITY

COMPUTER ENGINEERING DEPARTMENT

BLG 223E
DATA STRUCTURES AND ALGORITHMS
HOMEWORK 1 REPORT
Silent Knight Game

HOMEWORK NO	:	1
SUBMISSION DATE	:	04.04.2024
NAME	:	İrem Kalay
STUDENT ID	:	820220334

SPRING 2024

Contents

1 INTRODUCTION

In this assignment we were expected to perform the solution of Silent Knight Game from benefiting Stack data structure and DFS algorithm without taking any user input.

1.1 About The Game

In the game there were 3 rooms and in the each room there were different amount of objects. By selecting an action and an object, a new state was being created and if an unneccesary action took place, the advance function was returning 0. If a loosing state took place, it was returning -1 and lastly, if it returned a positive number, it meant that ,that state is necessary for continious of the game.

2 Methadology of Solution

2.1 Creating Stack Struct

Firstly, I created a struct of Stack which has a GameState typed Doubly Linked List in it and I declared operations of stack: pop and push. I used front of the linked list to have less complexity while every time performing pop and push operations ,not used the back of the list for operations. For pop operation, I used removeFront function of DoublyList struct and returned the first element of list. For push operation, I used addFront function of DoublyList sturct to add new state to the stack.

2.2 The Solution

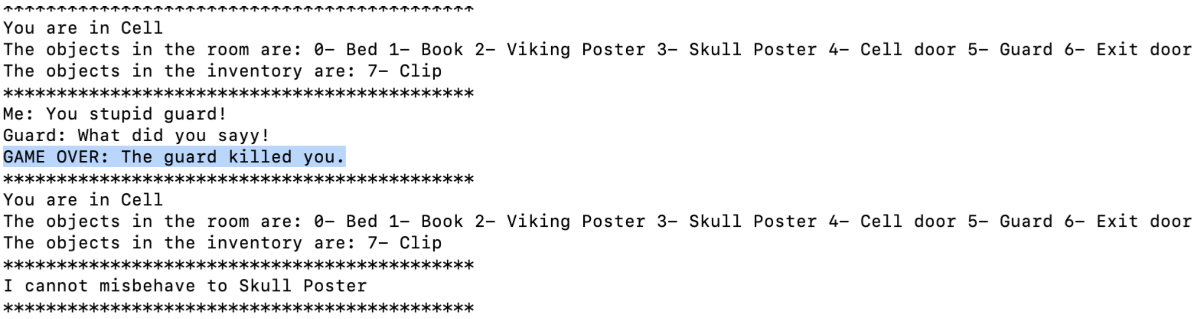
In the most initial of the homework, I tried to solve it by nested 2 for loops, however ,it entered to infinite loop between toilet and cell and could not able to find the solution. Then I began to search for other methods and recognised that I could use rand() function to created my states instead of using nested 2 for loops.

At the beginning of my code, first state is being created and pushed to my Gamestate typed stack(stack that was created with a struct by myself). Then, I started to produce new states by random integers which are : for action between 1-5 and for object between 0 and room object amount since in each room the object amount is different. I checked the result of advance function and if the result is greater than 0 , I pushed that state to my stack because the advance function is producing positive integer number which means that state is needed for solving the solution. If the result of advance function is -1, I popped the previous state and instead of the dead state, I continued the game with the state from stack which is not unsuccessful then continued the game with random action, object pair. Afterwards, when it finds out that game is won, it stops the “while” loop and we are able to see the message WELL DONE!!! in terminal.

3 Results

3.1 Game Over

In the below picture, the state terminates unsuccessful then it continues the game from the stored stack top gamestate at the same time pops it from the stack.

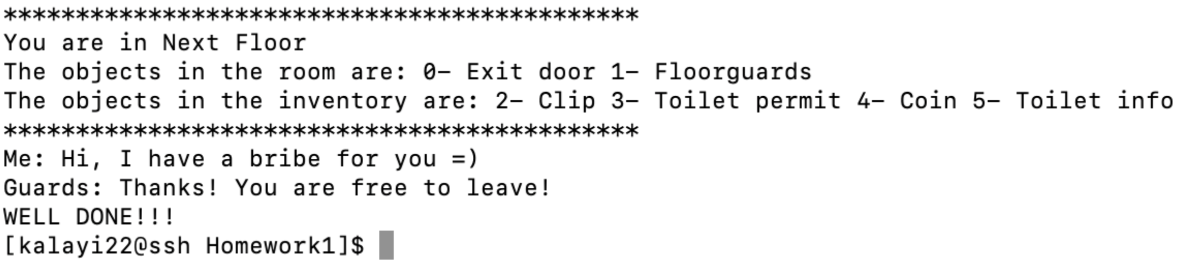


```
********************************************
You are in Cell
The objects in the room are: 0- Bed 1- Book 2- Viking Poster 3- Skull Poster 4- Cell door 5- Guard 6- Exit door
The objects in the inventory are: 7- Clip
********************************************
Me: You stupid guard!
Guard: What did you sayy!
GAME OVER: The guard killed you.
********************************************
You are in Cell
The objects in the room are: 0- Bed 1- Book 2- Viking Poster 3- Skull Poster 4- Cell door 5- Guard 6- Exit door
The objects in the inventory are: 7- Clip
********************************************
I cannot misbehave to Skull Poster
********************************************
```

Figure 1: Game Over State

3.2 Game is won

When the game is won, the loop stops and we see the result at terminal as follows in the Figure 2.



```
********************************************
You are in Next Floor
The objects in the room are: 0- Exit door 1- Floorguards
The objects in the inventory are: 2- Clip 3- Toilet permit 4- Coin 5- Toilet info
********************************************
Me: Hi, I have a bribe for you =)
Guards: Thanks! You are free to leave!
WELL DONE!!!
[kalayi22@ssh Homework1]$
```

Figure 2: When Game is won

4 Conclusion

In conclusion, stack structure and DFS algorithm was used while solving the Game. It checks until it reaches the win state or lost state. If it is lost state, it turns back to the successful states in the stack then again it searches way to find the solution.