

Computer Programming

Course Project: Text-Based Adventure Game

(This is purely a work of fiction and any resemblance to an actual person or monster is purely coincidental.)

Game Description

The princess of the kingdom of CPeria has been captured by an evil wizard and placed in a castle guarded by monsters. Your job is to set her free and bring her back outside the castle. The castle has 9 rooms. The game starts with you standing in room 1. Room 1 had the exit to the castle on its west. Some of the rooms are linked together. Each room had either a wall on its north, south, east or west or another room (see maze below). The princess is locked in room 9. There are two monsters in the castle that need to be killed in order to reach the princess. Each monster is guarding a magical path and whenever you kill a monster, this path appears and two rooms get linked together. Fortunately, the monsters are rather docile and do not attack you unless you attack them.

The first monster that you need to kill is the snake headed lady from Greek myth affectionately called Medusa by some. She is so beautiful that whenever someone sees her, he or she turns to stone. The trick to kill her is to attack her using a magic shield. This shield can be picked up from room 3. Medusa will see her own reflection in the shield and turn to stone. As soon as she dies, a pathway will appear which will let you explore room 7 and 8.

The second monster is Mr. Dracula. To finish him off, you must attack him using a silver dagger right through his heart. You can find the dagger in room 7 while Dracula is in room 6. He guards a door which opens up in room 9 when he is killed. On killing him, you have to get to the princess and must lead her out to the exit once you do that.

There are treasure-items scattered throughout the castle for the financially inclined ones. Since you carry a bag with you, you can put all the items that you pick during the game (the treasure, the dagger and the magical shield) in that bag. The bag can only contain at the most 10 items at a time. The treasure is of three types. There is a golden egg in room 2 and a golden chalice in room 4. In room 8, there's a piece of paper with the proof of the famous unsolved problem $P=NP$ on it¹. The golden egg and the golden chalice are worth half a million dollar each; where as the proof paper is worth a million dollar. Unfortunately there are no buyers of the shield and dagger so they are of no particular value in the OOPTopia black market.

Once an item is picked, you can drop it in any room if you want. If a monster is attacked without the weapon that he can be killed with in your bag, you'll die. If you get out of the castle without freeing the princess, you'll lose the game.

¹ Unlike all the other things in this story, the million dollar $P=NP$ problem is real. Use wikipedia for more info.



Maze: Dotted lines show the paths that are guarded by the monsters

Project Requirements

Your project is to implement this game. Here are a few requirements that need to be implemented.

- The game should start by stating the story in two or three lines. This story should be present in a file called Start.txt.
- The user should input all moves as text commands. Valid commands are **MOVE, PICK, DROP, EXIT, ATTACK, LOOK**, LOAD and SAVE.
- The MOVE command should take the direction in which the user wants to move as the second parameter. The directions can be NORTH, SOUTH, EAST or WEST. For example, on entering the MOVE EAST command in room 1, the user should move to the room 2.
- If there is a wall on the side, a message should be displayed telling the user that the move is not possible. For example, MOVE EAST is an invalid command.
- Whenever the user moves to the new room, a one line description of the room should be displayed. The description of all the rooms should be present in a file called Rooms.txt which should be read at the beginning of the program and stored in memory.
- After the room description, all the possible exits from the room should be displayed.
- If there are any items in the room, they should be displayed after the exit path list. There cannot be more than 5 items in a room.
- If there is a monster in the room, its name should be displayed after the item list. A room can contain at the most one monster. e.g. if a user enters a room with a door to east and west, the dagger and **Medusa (alive)**, the description will be.

This room has weird shapes drawn all over the walls and flickering torches make the faces even scarier. There is a door to your east. There is a door to you west. The dagger is lying on the floor. Medusa is waiting to kill you beside a locked door.

- The PICK command should accept the object to be picked as the second parameter. e.g. In the above example, if the user type the command PICK DAGGER, it should remove the dagger from the room, add the dagger to the user's bag and display a message on the screen. Something that is not in the room should not be allowed to be picked.
- The DROP command should also accept the object to be dropped from the user's bag into the current room.
- The EXIT command should exit the program.
- The ATTACK command should attack the monster in the room (if there is one). If the weapon that the monster can be killed with is in the bag, the monster should die. Other wise it should kill the user. In case of a death by the hands of monster, the game should end after displaying all the text in a file called EndDead.txt on the screen.
- A monster's death should create a **link** between two rooms and its body should remain on the floor. Medusa is guarding the link between room 5 and room 8. Dracula is guarding the link between room 6 and room 9.
- The LOOK command should print out the description of the room including all the items and doors in the room.
- If the user exits the castle after rescuing the princess, the text in a file called EndWin.txt should be displayed and the game should end.
- If the user exits the castle without rescuing the princess, the text in a file called EndLose.txt should be displayed and the game should end.