**Game of rooms**

**Problem Description**

You are going to create a game like the old fashioned MUDs (<https://en.wikipedia.org/wiki/MUD>).

To get some inspiration for what MUDs are, check out this video of a dramatic reading of Cleft of Dimensions (<https://www.youtube.com/watch?v=0s2ukNqLEE0>).

Your game needs to have a series of interconnected rooms that the user can explore and interact with. For our game, the basic premise is that you prompt (=solicitar, requerir) the user for movement commands to move between rooms. Rooms can have their own special action commands to trigger (=desencadenar) special behavior (=comportamiento).

The mechanics (=mecánica, mecanismo) are as follows:

* The character (=personaje) starts (=empieza, comienza) in a room.
* The description of the room is printed. The description should give the user a hint (=pista) about what to do next.
* A > symbol is shown to prompt the user for input.
* There are 4 options for movement input: N for north, S for south, E for east and W for west.
* If the character (=personaje) is able (=puede) to move in the direction (s)he chose (=elige), a description of the new room is printed and the prompt (=solicitud) is shown again for inputting (=introducir) additional movement commands or special room-specific commands.
* If the character (=personaje) is not able (=no puede) to move in the direction (s)he chose, a game-appropriate error message must be shown and the description of the current room along with the prompt must be repeated.
* A room's special commands can trigger (=desencadenar) any behavior (=comportamiento) desired. For example, if there are objects in the room, the user could pick up the objects and use them. If the room has another person in it the user could interact with the person. Use your imagination!

**Sample game**

You are in a dark room. There is a door to the north.

>

E

There is no exit there.

You are in a dark room. There is a door at the north.

>

N

You are in the forest. There is a lot of light. There is a bear sleeping.

>

S

You are in a dark room. There is a door at the north.

>

N

You are in the forest. There is a lot of light. There is a bear sleeping.

>

Cry

I don't understand

You are in the forest. There is a lot of light. There is a bear sleeping

>

Sing

The bear wakes up and kills you. Game over.

**Iteration 1: Create a simple game**

* Create a game with at least 5 spaces. They don’t necessarily have to be rooms. Each space can have more than one exit/entry, for example an exit to the East and an exit to the West.
* Write descriptions of each of the spaces and error messages for when the user tries to go in a direction where there is no exit, or when they enter a command that is not recognized.
* The user should have four directional commands: N, E, S, W.
* Write some global action commands that the user can enter in any space, like “look for food” and write what the consequences are in the game (the same action could have different consequences in different spaces).

class Game

def initialize

@player\_position = [0, 0]

@rooms = []

@lifes = 3

end

def add\_room(@room)

@rooms.push(@room)

end

def control\_direction

actual\_room = @rooms.find do |rm|

rm.@room\_position = @player\_position

end

puts actual\_room.@description

entry = gets.chomp

case entry

when “n”

next\_positon = @player\_position[0] + 1

when “e”

next\_positon = @player\_position[1] + 1

when “s”

next\_positon = @player\_position[0] - 1

when “w”

next\_positon = @player\_position[1] - 1

when “look for food”

puts “You have won. Congratulations!”

exit

else

next\_positon = nil

end

next\_room = @rooms.find do |rm|

rm.@room\_position = next\_positon

end

if next\_room == nil

puts “There is no exit there.”

@lifes -= 1

else

@player\_position = next\_room.room\_position

end

def play

while lifes > 0

# game1.control\_direction

control\_direction

end

puts “You have lost. Bye, bye!”

exit

end

end

class Room

def initialize(name, room\_position, description)

@name = name

@room\_positon = room\_position

@description = description

end

end

room1 = Room.new(“Room1”, [0, 0], “you are in room 1”)

room2 = Room.new(“Room2”, [1, 0], “you are in room 2”)

room3 = Room.new(“Room3”, [2, 0], “you are in room 3”)

room4 = Room.new(“Room4”, [0, 1], “you are in room 4”)

room5 = Room.new(“Room5”, [1, 1], “you are in room 5”)

room6 = Room.new(“Room6”, [2, 1], “you are in room 6”)

room7 = Room.new(“Room7”, [0, 2], “you are in room 7”)

room8 = Room.new(“Room8”, [1, 2], “you are in room 8”)

room9 = Room.new(“Room9”, [2, 3], “you are in room 9”)

game1 = Game.new

game1.add\_room(room1)

game1.add\_room(room2)

game1.add\_room(room3)

game1.add\_room(room4)

game1.add\_room(room5)

game1.add\_room(room6)

game1.add\_room(room7)

game1.add\_room(room8)

game1.add\_room(room9)

game1.play

**Iteration 2: Auto print exits**

You can shorten the descriptions if in each room a message like the following is automatically printed. For each room, the user is given a list of possible exits.

Don’t just add the auto exits to the room descriptions, that’s not scalable! The list of exits should be dynamically generated.

You are in a dark room. There is a door.

Exits: N

>

**Iteration 3: Inventory**

Can you add objects and inventory to the game? This means that you can pick up and drop objects. You can also carry objects from one room to another. For example:

You are in a dark room. There is a door at the north

>

N

You are in the forest. There is a lot of light. There is a bear sleeping

You can see a sword

>

Pick up sword

You pick up a sword

You are in the forest. There is a lot of light. There is a bear sleeping

>

S

You are in a dark room. There is a door at the north

>

Inventory

You have:

- A sword

You are in a dark room. There is a door at the north

>

Drop sword

You are in a dark room. There is a door at the north

You can see a sword

>

**Iteration 4: File persistence**

Store the state of the game in a file so you can save game with save command and restore it with a load command.