1ST LAB PRACTICE

In this practice you would develop the basic elements to stablish environments, like mazes, and later use it to interact with a being. Some possibilities are:

1. Define mazes and later put inside a being to search for exit.
2. Define a board (like chessboard) and later make a being or more of them to play

ENVIRONMENT DEFINITION

For simplicity a discrete space defined by a matrix of R Rows and C Columns is used, and rows identified by numbers and columns by letters.

So for example a maze of 15 rows and 15 columns, would be like:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1															
2							e								
3		e													
4															
5															
6															
7															
8															
9															
10		S2													
11															
12															
13															
14															
15															

This kind of maze is accompanied with a definition and a set of rules, like the white cells are road cells which one are the only ones where we can travel while the gray one are wall cells and we cant cross them. Additionally, we can stablish a coded value for this terrain types and use this codification for a easy and less computational complexity manipulation.

Color	Means	Coded Value
	Wall	0
	Road	1

Using this we can save the previous map in a coma separated values file text:

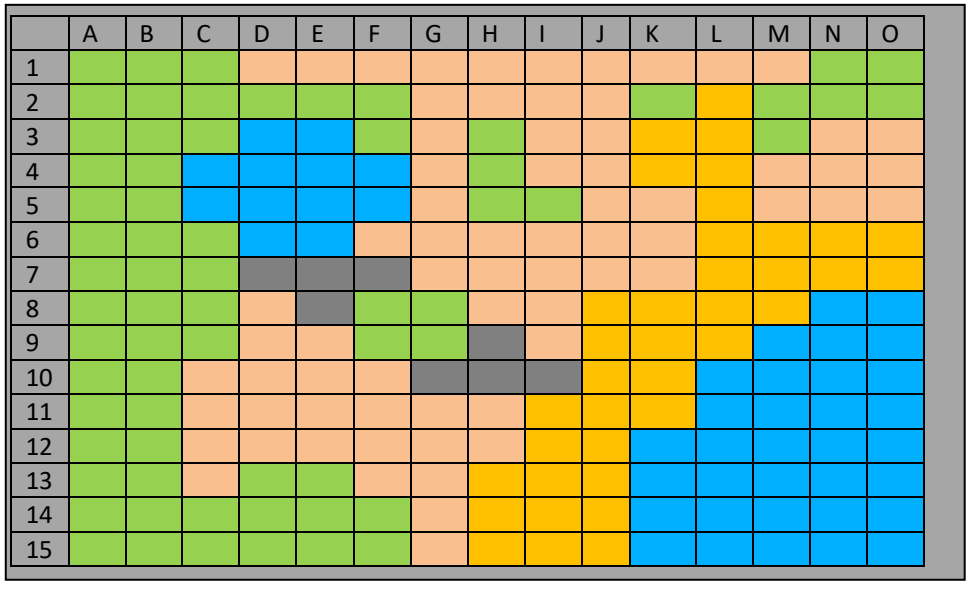
```
0,0,0,0,0,0,0,0,1,0,0,0,0,0,0
0,1,1,1,1,0,1,1,1,1,1,1,1,1,1
0,1,0,1,0,0,1,0,1,0,0,0,0,0,0
0,0,1,1,1,1,1,0,1,0,0,0,0,0,0
0,0,0,1,0,1,0,0,1,0,0,0,0,0,0
0,0,0,1,0,0,0,0,1,0,0,0,0,0,0
0,1,1,1,1,1,1,0,1,0,0,0,0,0,0
0,1,0,1,0,0,0,0,0,0,0,0,0,0,0
0,1,0,1,1,1,1,1,1,0,0,0,0,0,0
1,1,0,1,0,0,0,1,0,0,0,0,0,0,0
0,0,0,1,0,0,0,1,0,0,0,0,0,0,0
0,0,0,1,0,0,0,1,0,0,0,0,0,0,0
0,0,0,1,0,0,0,1,0,0,0,0,0,0,0
0,0,0,1,0,0,0,1,0,0,0,0,0,0,0
0,0,0,1,0,0,0,1,0,0,0,0,0,0,0
```

Later load into an array in a programing language in order to obtain an indexed data access

We can aggregate more values to represent more different terrains, so considering the following values:

Color	Means	Coded Value
	Mountain	0
	Land	1
	Water	2
	Sand	3
	Forest	4

We can create terrain maps as the following one:



Or make this codification means other things, like chess pieces.

White pieces		
Piece	Coded value Black	Coded value White
Pawn	11	21
Rook	12	22
Knight	13	23
Bishop	14	24
Queen	15	25
King	16	26

1 Loading Maps

Develop a system who allows the user the following

1.1 Load a map

Load a file (filename.txt), which inside it can have a map codified using any of the two previous showed. And show it in a graphic interface

1.2 Ask the value of a given cell

Put in the interface a way to ask for the terrain type of a given coordinate of the map, and show what is in it

For example in or Labyrinth map, in the coordinate (7,A) is wall and in the (7,B) is road. In our terrain map, (7,E) is mountain, (15,K) is water and (6, E) is sand.

1.2 Change the value of a given coordinate

Add to the interface a way to change the value of specific coordinate

For example we can change in the labyrinth map the terrain type of (7,A) from wall to road, or in the terrain map, the terrain map of (7,E) from mountain to water.

1.3 Mark positions

Additionally to the terrain map, the coordinates would have more information, for example initial point, already visited position, final point, specific objective, etc. Add a way to show this information in the map.

So For example, in the next map we establish the following marks

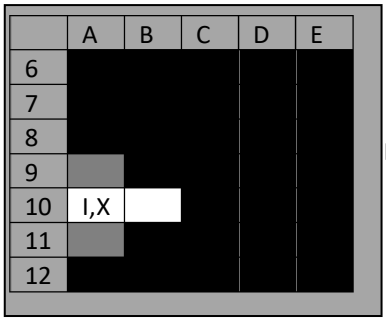
- V the road already visited
- O the places where a decision was make
- I the initial point
- X the actual position.

	A	B	C	D	E
6					
7		V	V	O,V	
8		V		V	
9		V		V,X,O	
10	I,V	V			
11				F	

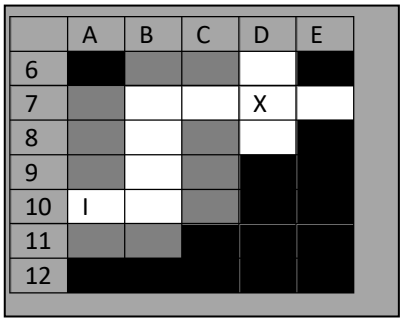
2 MASKING MAPS

One thing is the map made by creator and another one is the knowledge that the being inside have about it, actually the knowledge of this being about the map is only the part that their sensors allow to know and the memory of this perceptions.

Also consider that a being X is collocated inside the labyrinth, is able to memorize the labyrinth the map of the labyrinth already visited and is able to see only one cell of distance (up, down, left and right). Initially the being inside doesn't know the entire map and construct it travelling over it, discovering only what he can sense. For example, in the following labyrinth the unknown map is marked with black:



Initial



After 6 steps

allow to the system to:

1. Mask the entire map
2. Discover a given position of the map
3. Mask a given position of the map

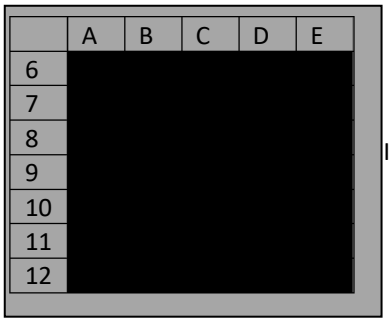
3 CREATE AGENTS THAT TRAVEL THE MAPS

In order to design an agent we need to specify what it can sense, and what he can do, among other proper abilities,

3.1 Sensing ability

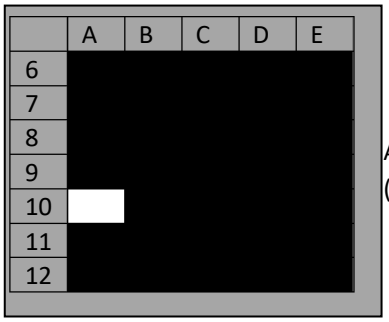
Environment discovery is made using sensing abilities, this mainly allows a being to know what terrain type is in a given cell, and another desired information, like if there is another being or some resource, trap or treasure.

At beginning, the knowledge of the map is zero, this is the being doesn't know nothing of it



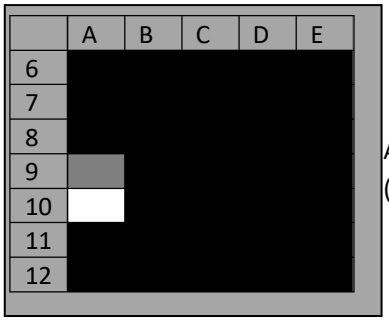
Initial

But when it sense the (10,A) position it knows that the terrain type is Road.



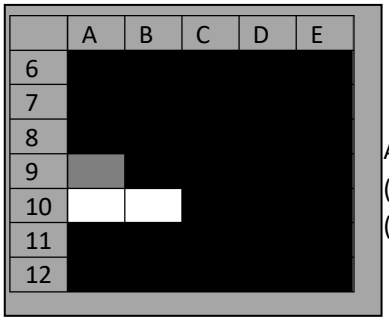
After sense (10,A)

After it can also sense another cell, for example (9,A)



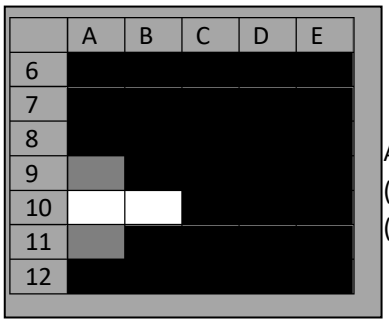
After sense (10,A), (9,A)

After it can also sense another cell, for example (10, B)



After sense (10,A), (9,A), (10,B)

After it can also sense another cell, for example (10, B)



After sense (10,A), (9,A), (10,B), (11,A)

Options for a being will be defined accordingly of how it discovers its world, for example we can have two beings:

Being 1: only one sensor, in this case direction of sensor is needed and would behave like showed before, consuming one unit time, for each cell

Being 2, with four sensors, being able to sense Up, Down, Left and Right, sensing all of them at once using only one unit time.

3.2 Agent Actions

What agent can make at a given moment is defined by actions that agent is able to do. For example, consider the following agents definition:

Agent 1:

1. Turn left
2. Advance forward

Agent 2:

1. Turn Left
2. Turn Right

3. Advance forward

Agent 3

1. Move left
2. Move right
3. Move up
4. Move down

Each one of the previous definitions is valid but is mainly defined by the problem nature and it is necessary to consider advantages and disadvantages.

3.4 Additional information based on terrain types

Sometimes additional information in necessary and depends on one or more environment values, or other processes. An example of this is besides a cell has a terrain type the cost to occupy that cell is different for different agent types

Color	Means	Cost movement			
		Human	Monkey	Octopus	Sasquatch
	Mountain	N/A	N/A	N/A	15
	Earth	1	2	2	4
	Water	2	4	1	N/A
	Sand	3	3	N/A	N/A
	Forest	4	1	3	4
	Swamp	5	5	2	5
	Snow	5	N/A	N/A	3

For example, for a human, moving from any cell to a sand cell will cost

4 CREATE A BEING CONTROLLED BY USER

Allow in the system the possibility to:

1. Choose a being (human, Monkey, etc),
2. Stablish a start point
3. Stablish a final point
4. put the chosen being at Start point
5. Allow the user to control this being using keyboard or mouse until it arrives the final point, moving through map.
6. Mark the positions visited with ‘V’
7. Mark the position where the user must choose one way between 2 or more with an ‘o’
8. Calculate number of movements (Each presed key)
9. Calculate total cost ,initial 0, and with each movement the cost to move to the terrain type of the cell arrived is added