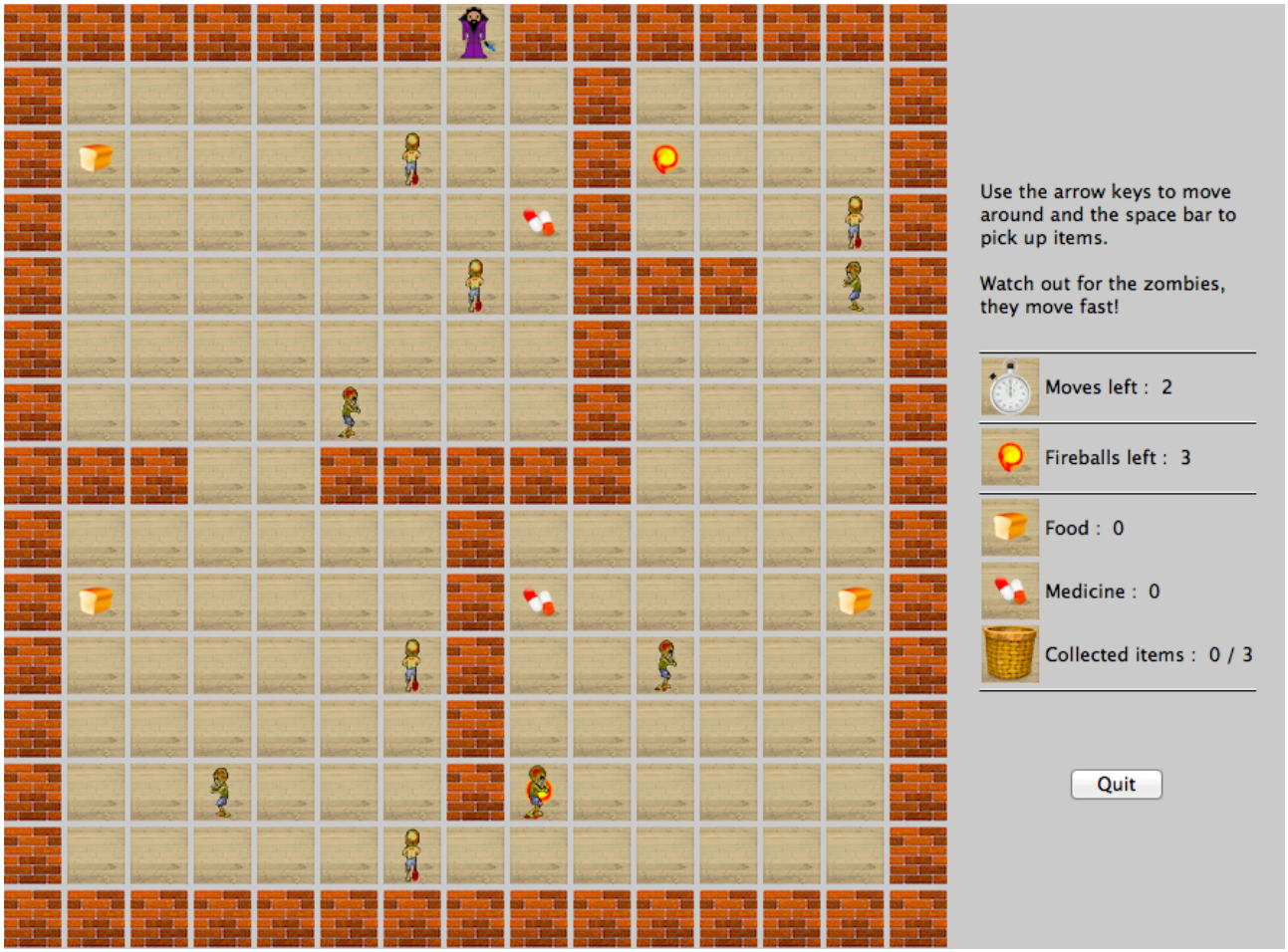
Université Catholique de Louvain

INGI1131 - Computer Language Concepts

Zombieland

Course Project



Abstract *Following an explosion of secret U.S. government laboratories, fast and fearless living beings with a thirst for human blood have invaded the planet. Few survivors have been hiding in a secret place, but they are starting to run out of victuals. A "brave" has been designated to collect some. To assist him in this task, we have implemented a simulator that will help him to take all the possible unexpected events into account. Zombies have indeed been studied for a while, so that we are now able to precisely tell you how they move and behave...*

Group 43 : Lena Peschke 58261100
Mélanie Sedda 22461100

Professor : Peter Van Roy
TAs : Zhongmiao Li
Manuel Bravo

May 10, 2014

1 The game

The room the brave has to enter contains food, medicine and packs of three bullets. He can kill a zombie with one bullet and can only leave the room if he has collected a certain number of objects (i.e. food + medicine). Our simulator takes four arguments : the map of the room, the number of bullets the brave initially has with him, the percentage of the objects in the room he has the collect and the number of zombies present (default values are used if the user does give any input).

Percentage of objects If the percentage is < 0 (resp. > 100), we transform it into 0 (resp. 100).

Number of zombies We have decided to limit the number of zombies to the number of empty cells on the map, because it would be suicidal to enter a room with more zombies. The zombies can nonetheless still stand on a cell containing an item.

Moves The brave and the zombies are moving one after another. When the brave has made 2 moves, all of the zombies can make 3 moves at the same time and, when they are done, the brave can move again. Displacements and pickups are considered as moves while killing is not. Zombies and brave cannot overlap.

Pickups If the the number of objects needed to leave the room is out of reach for the brave (because the zombies have destroyed some), the brave automatically loses.

Kills A brave automatically kills a zombie if he has at least one bullet and if the zombie his in the cell in front of him. However, if he is running out of bullets and if the zombie is facing him, the zombie will automatically kill him. Furthermore, the zombie automatically kills the brave if the brave is in the cell in front of him and isn't facing him.

2 Architecture and design

We have identified some port objects that we would need. The main ones are the brave and the zombies, with one port object per zombie. To interact with the map, we also decided to create one port object per cell because it was a quite effective approach. To manage and synchronize the turns of the brave and the zombies, we also created a controller. The functions relative the each entities are located in separated files. Furthermore, we have a file for the GUI and a file for the launch of the game. To implement the interactions between all these entities, we made a component and several state diagrams. Since the codes of the port objects are quite self speaking and always following the same pattern we will only briefly describe what each port object does (NB : we also sometimes used unbound variables to enforce atomicity in the interaction).
Here is the pattern :

```
case Mode
of Mode_1 then ...
        case Msg
        of Msg_1 then ....
        ...
        [] Msg_n then ...
        else we ignore the message, but it never happens
        end
...
[] Mode_m then ...
else we ignore the message, but it never happens
end
```

We have decided to make the ports of the controller, the cells and the brave known by everybody. All of the zombie's ports are known by the controller, and a cell knows the port of a zombie that is one it.

Controller The controller has to tell the brave and to the zombies when it is their turn. It knows when a zombie is dead, so it doesn't warn him in this case. It also has to decide when the brave couldn't win anymore because there are to few objects left in the room.

Cell The room is a grid of cells. Each cell knows which item and person is on it. If it is the brave, it knows his facing direction and the number of bullets he has (this is important for the management of the kills). If it hosts a zombie, it knows his facing direction and his port (also important for the kills).

All the characters All the characters warn a cell when they enter or leave it. They also notice the controller when they are done and when they have picked up/destroyed an object. After each move, they call a function that makes all the needed kills. In order to do that, we check if there is an enemy in front or at their left or their right. If the dead conditions are verified (see 1), we send a kill message to the right character.

Brave The brave knows his position, the number of each items he possesses, the item on the cell he stands on and the actions he can still perform before the zombies move. He receives the messages from the keyboard and, if he can, moves and updates what has to be updated. The pickups aren't mandatory; the player has a say if he really wants to pickup an object.

Zombies A zombie know his position, the item on the cell he stands on and the actions he can still perform before the brave moves. He automatically tries to move : he takes the same direction as previously, unless he encounters an obstacle. In this case, he randomly changes his direction. If after 15 time he hasn't found a valid direction, he passes his turn. We decided that the zombies destroy an item 20% of the time when they can. He warns the controller if he dies.

Conclusion

We think that our simulator fulfills the requirements and we hope that it will provide you some help to survive the zombie apocalypse out there.