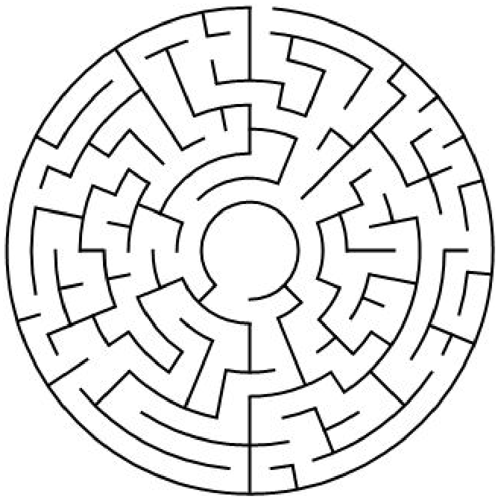


LABYRINTH FIGHTS

DESIGN PATTERNS – 4A – IBO2 – MANASA PRAKASH – MAXIME LE CORRE

22 December 2017

INDEX

“

At some point we all look up and realize that we are lost in a maze.

–John Green

”

INTRODUCTION

The game is a labyrinth fight game with a maze consisting of objects that arm fighters who fight each other on their quest to find the exit of the maze. A set of objects stay with a user only for 10 seconds. The user is characterized with 100 points for health and 1-10 damage points depending on the object in hand. The game ends when one of the players finds an exit. Moreover, each fighter could be in an offensive/ defensive mode.

Fig. 1. Example of a maze fight

The programming challenges in this exercise are
-displaying the maze
-using a factory structure for object and fighter generation
-usage of threads for management of multiple players

Object Oriented Approach

The program is structured into 9 classes, each handling their unique purpose. The Main class launches the Menu where the user can choose between launching the game or ending it. If starting the game is chosen, the Menu instantiates the Game Class and starts the Game Loop. The Game class takes care of the Labyrinth generation. The class Labyrinth calls the factory method for producing the appropriate object of Class type: attack object/ fighter/wall of which it maintains lists. It also consists of a Class Exit which represents the exit point of the maze. The Fighter class (Combattant) consists of player’s details- score, life, list of objects. The Item class represents the attack objects and holds damage power.

The class diagram is as follows:

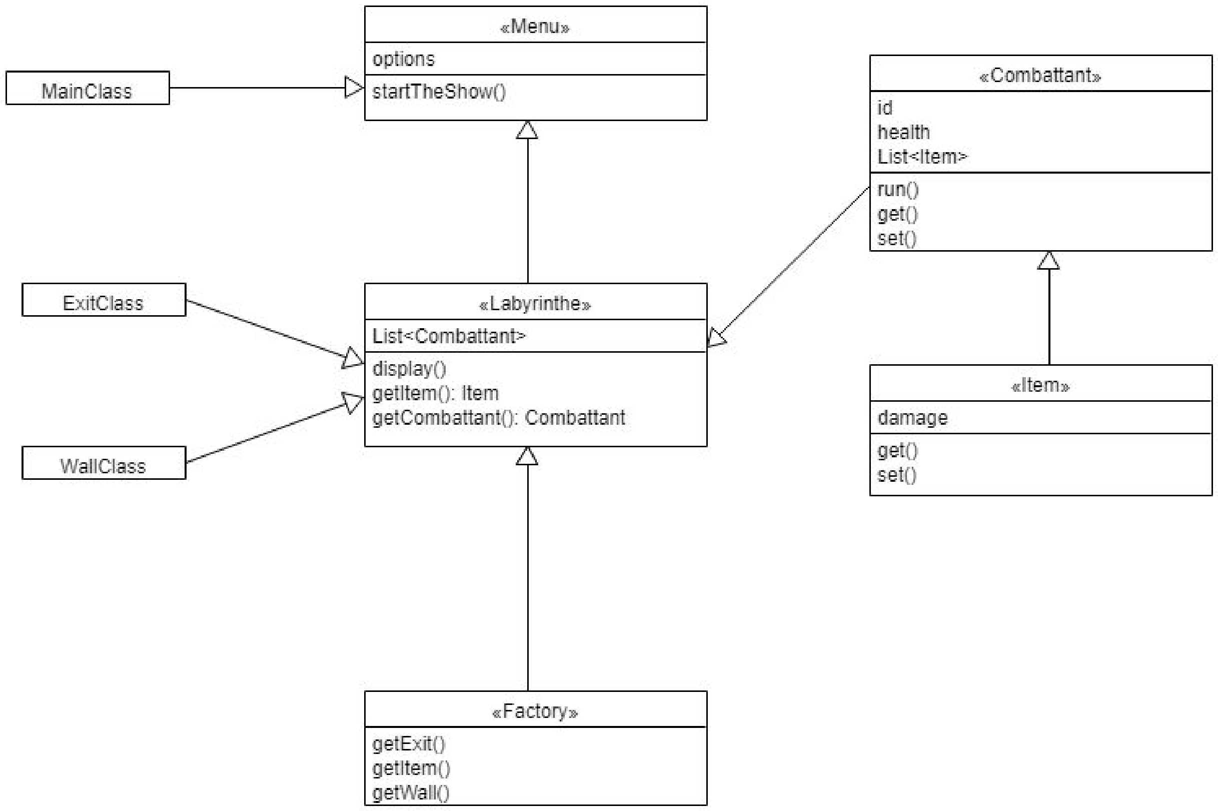


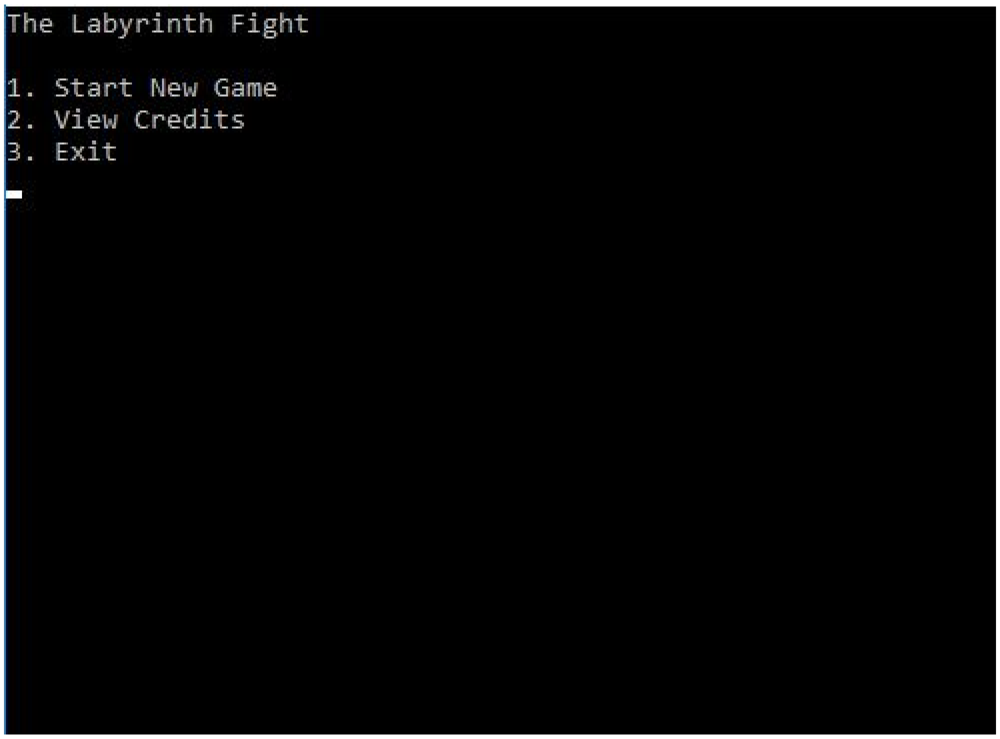
Fig. 2. Class Diagram

GAME WALKTHROUGH

Game screens

Screen 1:

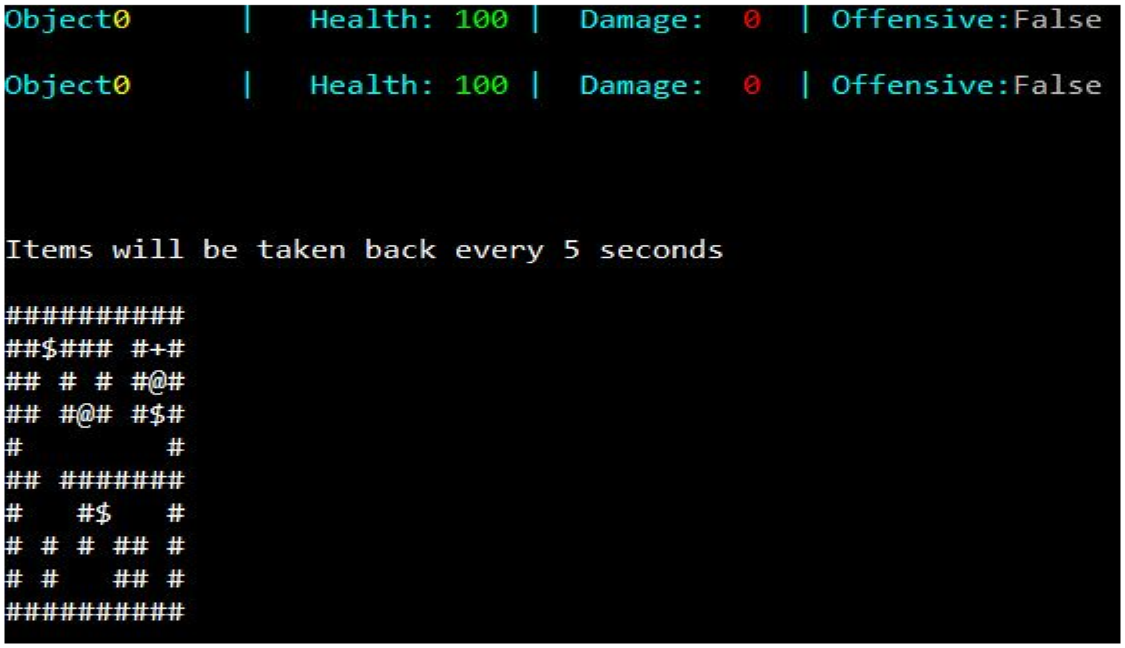
Menu.



```
The Labyrinth Fight

1. Start New Game
2. View Credits
3. Exit
```

Screen 2:

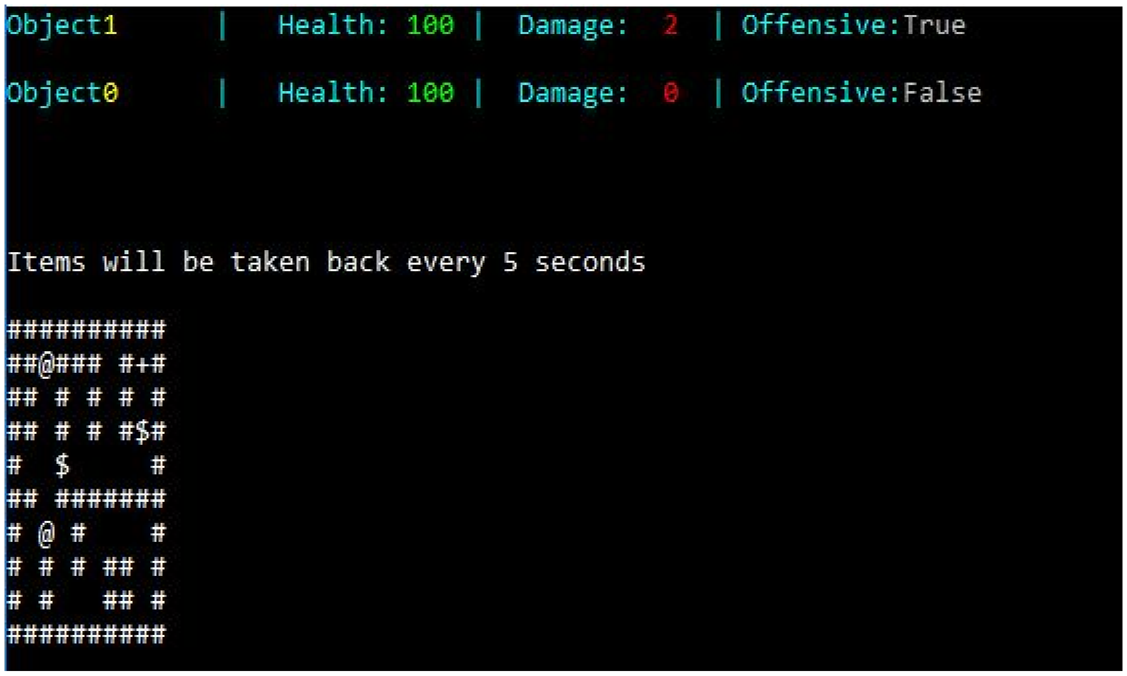
Maze generated.



```
Object0       |   Health: 100 |  Damage:  0  | Offensive:False

Object0       |   Health: 100 |  Damage:  0  | Offensive:False



Items will be taken back every 5 seconds

##########
##$### #+#
## # # #@#
## #@# #$#
#        #
## #######
#   #$   #
# # # ## #
# #   ## #
##########
```

Screen 3 - (n-1):

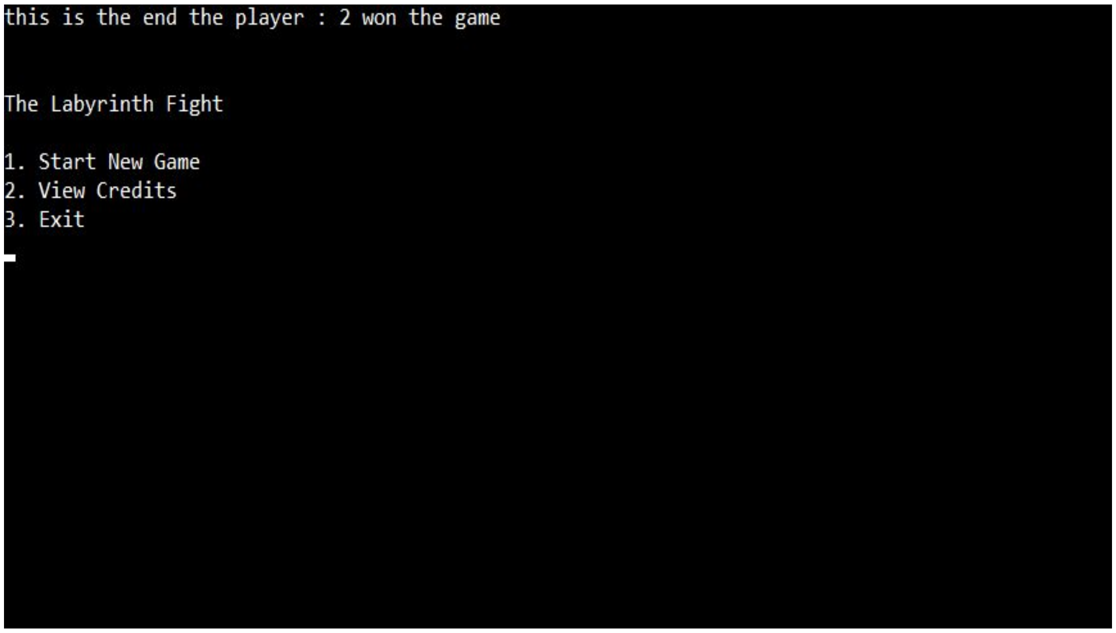
Players moving and objects being fired.



```
Object1       |   Health: 100 |  Damage:  2  | Offensive:True

Object0       |   Health: 100 |  Damage:  0  | Offensive:False



Items will be taken back every 5 seconds

##########
##@### #+#
## # # # #
## # # #$#
#  $     #
## #######
# @ #    #
# # # ## #
# #   ## #
##########
```

Screen n:

Game over with one winner.



```
this is the end the player : 2 won the game


The Labyrinth Fight

1. Start New Game
2. View Credits
3. Exit
```

Game loop Algorithm

```
while (no one at exit):
    displayMaze
    movePlayers
    updateState
```

Factory Logic

10% of the board is filled with attack objects and 1% of the board is filled with fighters

Fighter Movement Algorithm

```
Initialize stack of visited nodes = {0}
Initialize list of visited nodes = {0}
```

```
While (no winner):
     If (curr_pos not in visited_nodes_list):
          append_to_list(curr_pos)
          push_to_stack(curr_pos)
     If (no possible moves):
          Curr_pos = pop_from_stack(pos)
```

Fighter Attack Algorithm

```
If (offensive = true and pos_occupied_by_fighter = true):
     attack_fighter()
```

Display Logic

Display number of objects, health, damage of best item and if offensive or not followed by the maze.

Object Timer

Thread timer running every 5 seconds to empty item list from fighter and place it back in the maze.

IMPROVEMENTS:

On the graphics front, the game could have a more advanced user interface and smoother transitions.

On the programming front, the interactions between the fighters could be ameliorated.