

Projet 3 : Save MacGyver

Diagramme de classe and organisation of project	P.02
Update Diagram	P.04
Update Python code	P.07
Pygame test	P.09
Interface Pygame	P.10
MVC	P.11

Diagramme de classe_Projet 3 : Aidez MacGyver à s'échapper

```
Class Game get an matrix attribut to create a maze  
Matrix of 15 Sprites ( 15 columns, 15 lines)  
Maze =  
[ [D,--,--,--,--,--,--,++,++,++,++,++,++,++,++,++],  
[++,++,++,++,++,++,--,--,++,--,--,--,--,--,++,++],  
[--,--,++,--,--,++,--,--,--,--,--,--,++,++,++,++ ],  
[--,--,++,--,--,++,--,--,++,--,--,++,--,++,++,++,++ ],  
[--,--,++,--,--,++,--,--,++,--,--,++,--,--,++,--,++ ],  
[--,--,++,--,--,--,--,--,--,--,--,--,--,--,++,++,++ ],  
[++,--,++,--,++,++,++,++,++,++,--,--,++,++,++,++,++ ],  
[--,--,++,++,--,--,--,--,--,--,--,--,--,--,--,++,++ ],  
[--,--,--,--,--,--,--,+,--,--,--,--,--,--,--,--,++,++ ],  
[--,++,--,--,++,++,++,++,++,++,++,++,++,++,++,++,++,++ ],  
[--,--,--,--,--,--,--,--,++,++,++,++,++,++,++,++,++,-- ],  
[--,++,--,++,++,++,--,--,++,++,++,++,++,++,++,++,++,--,-- ],  
[--,--,--,--,--,--,--,--,--,--,--,--,--,--,++,--,--,-- ],  
[++,++,++,++,++,++,++,++,--,--,++,++,++,++,++,++,++,--,-- ],  
[--,--,--,--,--,--,--,--,--,--,--,--,--,--,--,--,++,--,-- ],  
[--,--,--,--,--,--,--,--,--,--,--,--,--,--,--,--,--,--,--,A ]]
```

Game : Départure, Wall, Floor, Arrived

Départure = D

Wall = +

Floor = -

Arrived = A

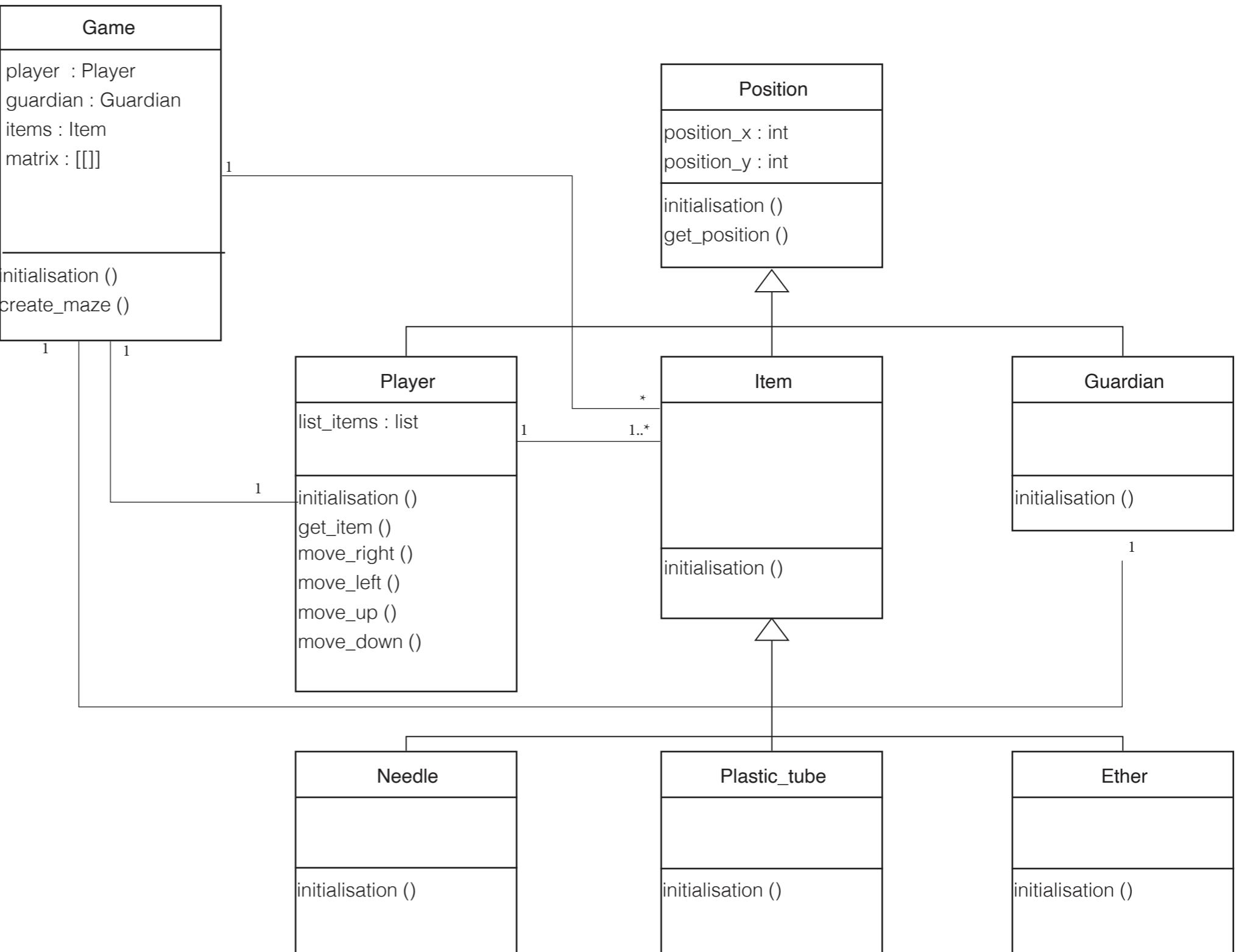


Diagramme de classe_Projet 3 : Save MacGyver - work in progress -

Game's structure : **Départure, Wall, Floor, Arrived**

Départure = D -> P (player)



Arrived = A-> G (guardian)

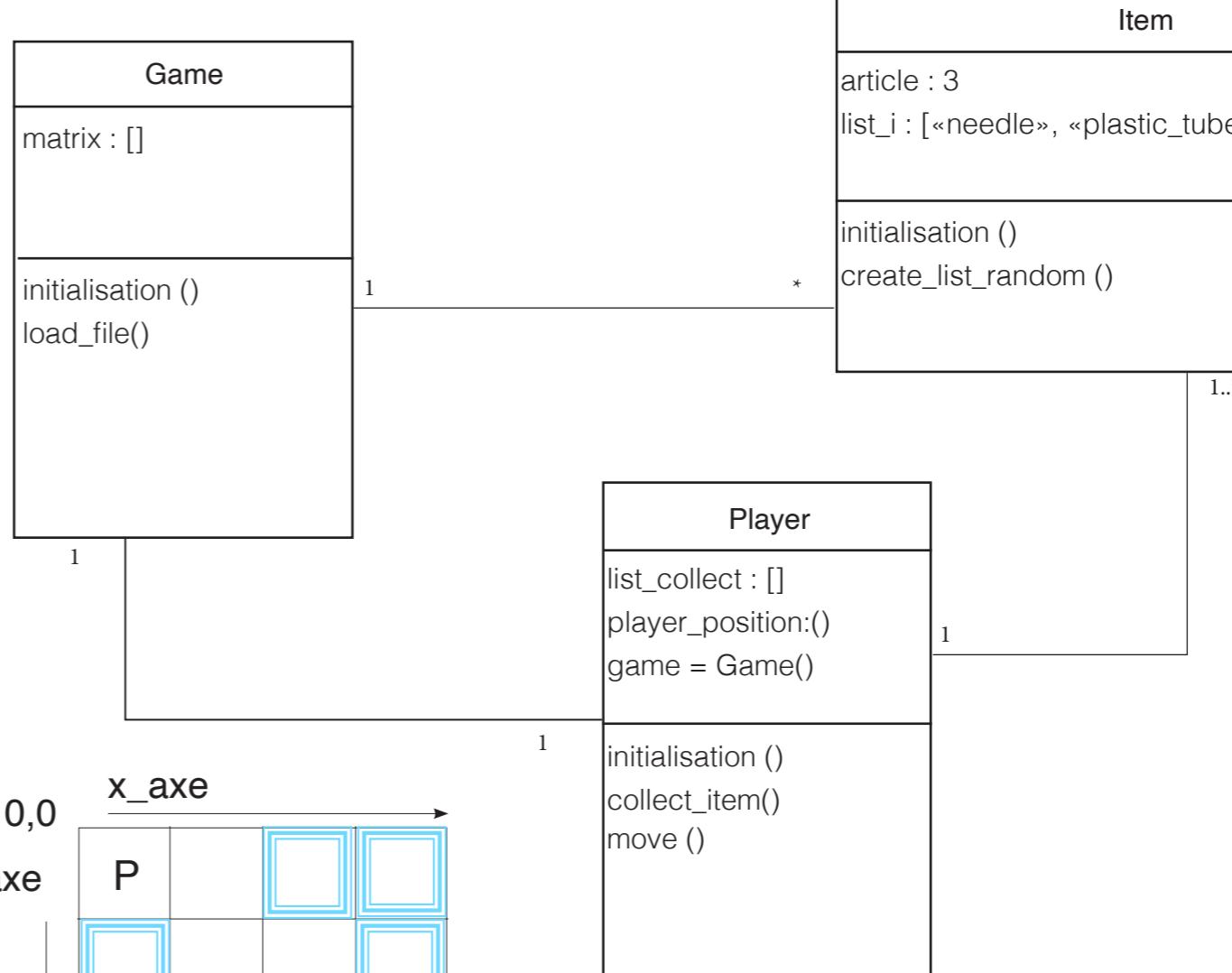
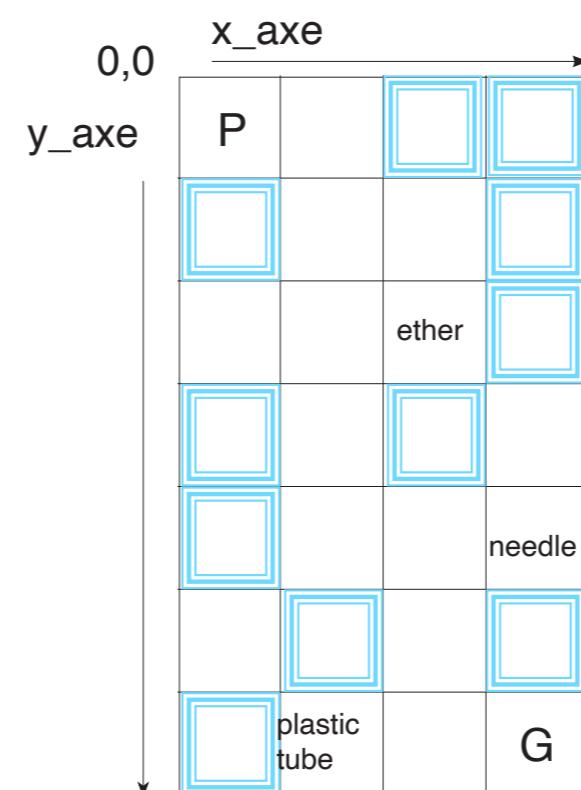
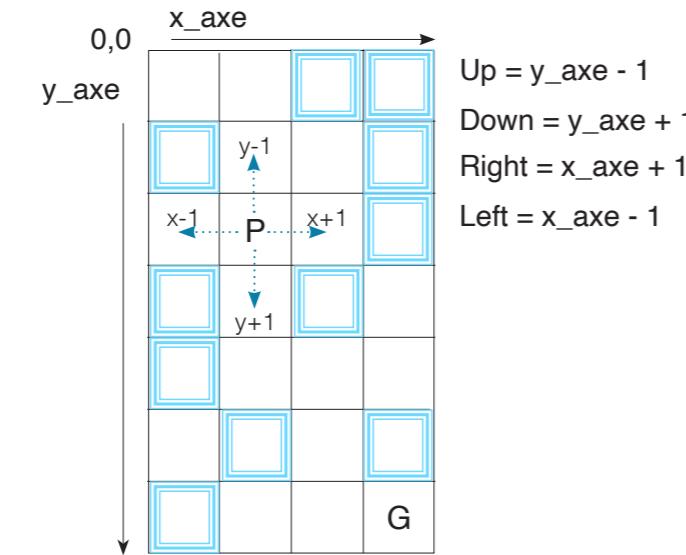
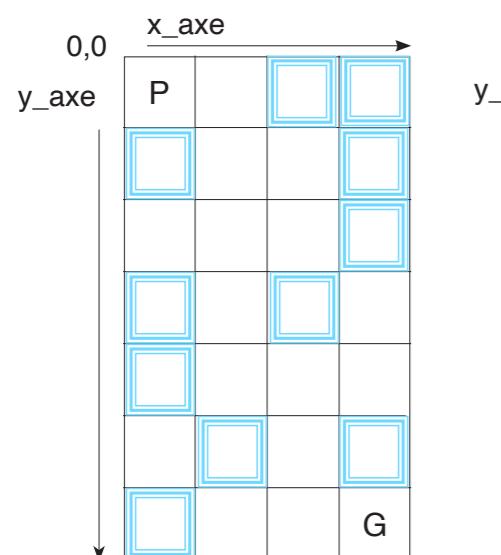
Floor = -

Save in a separate file maze.txt



We don't need a Guardian and Position class because Game has x_axe, y_axe and matrix. Game's attributes can be used by Item and Player.

The `Guardian` class is not useful because he has a static position.



Result on Terminal

```
(base) iMac-de-XavGab:PROJET_3 xavgab$ python P00_P3.py
[['P', '-', '+', '+'], ['+', '-', '-', '-'], [y, position(), '+'], ['+', '-', '+', '-'], ['+', 'needle', 'plastic_tube', 'ether', '+'], ['+', '-', '-', 'G']]
[['P', '-', '+', '+'], ['+', '-', '-', '-'], [-, '+', '+'], ['+', '-', '+', '-'], ['+', 'needle', 'plastic_tube', 'ether', '+'], ['+', '-', '-', 'G']]
game = Game()
4 2
6 1
[2 2
initialisation()
[['P', '-', '+', '+'], ['+', '-', '-', '-'], [initialisation(), 'ether', '+'], ['+', '-', '+', 'create'], [st, random, 'needle', '-'], ['-', '+', '-', '+'], ['+', 'plastic_tube', '-', 'G']]
[['P', '-', '+', '+'], ['+', '-', '-', '-'], [leds_itoh, ('-', '+'), ['+', '-', '+', '-'], ['+', '-', '+', '-'], ['-', '+', '-', '+'], ['+', '-', '+', '-'], ['+', 'plastic_tube', '-', 'G']]]
press a key u->up, d->down, r->right, l->left u
move()
```

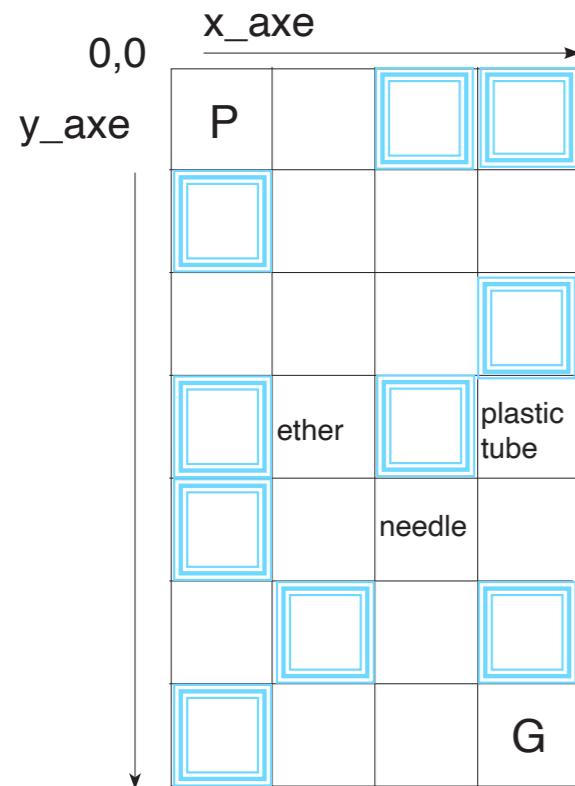
```
POO_P3.py x
1 from random import randint
2
3
4 class Game:
5
6     def __init__(self):
7
8         self.matrix = []
9         self.x_axe = None
10        self.y_axe = None
11        self.load_file()
12        self.get_position_player()
13
14    def load_file(self):
15        with open("maze.txt", 'r') as maze:
16            for x_axe, line in enumerate(maze.readlines()):
17                self.x_axe = x_axe
18                self.matrix.insert(x_axe, [])
19                for y_axe, char in enumerate(line.strip()):
20                    self.y_axe = y_axe
21                    if char == "D":
22                        self.matrix[x_axe].insert(y_axe, "P")
23                        self.player = (x_axe, y_axe)
24                        print(self.x_axe, y_axe)
25                    elif char == "-":
26                        self.matrix[x_axe].insert(y_axe, "-")
27                    elif char == "+":
28                        self.matrix[x_axe].insert(y_axe, "+")
29                    elif char == "A":
30                        self.matrix[x_axe].insert(y_axe, "G")
31        # print(self.matrix)
32
33    def get_position_player(self):
34        return self.player
35        print(self.x_axe, y_axe)
36
37
38 class Player():
39
40     def __init__(self):
41         self.list_collect = []
42         self.player_position = ()
43         self.new_pos = ()
44         self.game = Game()
45         self.move()
46
47     def move(self):
48         running = True
49         #self.player_position = self.game.x_axe, self.game.y_axe
50         while running:
51             option = str(
52                 input("press a key u->up, d->down, r->right, l->left "))
53             if option == "u":
54                 self.game.player = self.game.y_axe + 1
55                 self.new_pos = self.game.matrix[self.game.y_axe].append(
56                     self.game.player)
57                 print(self.new_pos)
58             elif option == "d":
59                 self.player_position == self.game.y_axe
60                 new_pos = player_position
61             elif option == "r":
62                 self.player_position += self.game.x_axe
63                 new_pos = player_position
64             elif option == "l":
65                 self.player_position == self.game.y_axe
66                 new_pos = player_position
67             elif option == "q":
68                 return exit()
69             # else: raise exception (other letters or wall)
70
71     def collect_item(self):
72         pass
73
74
```

I have some trouble with the player position ...
Python_code create a new matrix, i want a item matrix to move the player and collect items (needle, ether, plastic_tube)

```
POO_P3.py x

74
75 class Item():
76
77     def __init__(self):
78         self.article = 3
79         self.list_i = ["needle", "plastic_tube", "ether"]
80         self.game = Game()
81         self.create_list_random()
82
83     def create_list_random(self):
84         art = 0
85         random_x = 0
86         random_y = 0
87         for art in range(0, self.article):
88             while self.game.matrix[random_x][random_y] != "-":
89                 random_x = randint(0, self.game.x_axe)
90                 random_y = randint(0, self.game.y_axe)
91                 print(random_x, random_y)
92                 self.list_i.append(self.article)
93                 # if self.matrix[random_x][random_y] == "-":
94                 self.game.matrix[random_x][random_y] = self.list_i[art]
95                 print(self.game.matrix)
96
97
98     game = Game()
99     obj = Item()
100    user = Player()
101
```

Diagramme de classe_Projet 3 : Save MacGyver - work in progress - Update diagram



Game	Player
matrix : [] player_position : None initialisation () load_file() load_items() update_maze()	game = Game() list_collect = int numbers = int new_position = () initialisation () move () collect_item()
1	1

We don't need a Guardian and Position class because Game has x_axe, y_axe and matrix. Game's attributes can be used by Player.
The player collect the items placed on the game (matrix)
The Guardian class is not useful because he has a static position.

Result on Terminal with pprint

```
(base) iMac-de-XavGab:PROJET_3 xavgab$ python P00_P3.py
(0, 0)
[['D', '-', '+', '+'],
 ['+', '-', '-', '-'],
 ['-', '-', '-', '+'],
 ['+', 'ether', '+', 'plastic_tube'],
 ['+', '-', 'needle', '-'],
 ['-', '+', '-', '+'],
 ['+', '-', '-', 'A']]
```

```

POO_P3.py      x
1  from random import randint
2  from pprint import pprint
3
4
5  class Game():
6
7      X_SIZE = 4
8      Y_SIZE = 7
9
10     def __init__(self):
11
12         self.matrix = [
13             [None for x in range(0, Game.X_SIZE)]
14             for y in range(0, Game.Y_SIZE)
15         ]
16         self.player_position = None
17         self.list_article = 0
18         self.article = 3
19         self.load_file()
20         self.load_items()
21
22         # self.collect_item()
23
24     def load_file(self):
25         with open("maze.txt", 'r') as maze:
26             for y_axe, line in enumerate(maze.readlines()):
27                 for x_axe, char in enumerate(line.strip()):
28                     self.matrix[y_axe][x_axe] = char
29                     if char == "D":
30                         self.player_position = (x_axe, y_axe)
31         pprint(self.player_position)
32
33     # def can_i_move_to(self, x, y):
34     #     if x < 0 or y < 0 or x >= Game.X_SIZE or y >= Game.Y_SIZE:
35     #         return False
36     #     # verif s'il y a un sol a la pos x, y
37     #     return True
38
39     def load_items(self):
40         list_i = ["needle", "plastic_tube", "ether"]
41         random_x = 0
42         random_y = 0
43         for art in list_i:
44             while self.matrix[random_y][random_x] != "-":
45                 random_x = randint(0, Game.X_SIZE - 1)
46                 random_y = randint(0, Game.Y_SIZE - 1)
47             self.matrix[random_y][random_x] = art
48         pprint(self.matrix)
49
50     def update_maze(self, new_position):
51         old_x, old_y = self.player_position
52         x_axe, y_axe = new_position
53         if x_axe < 0 or y_axe < 0 or x_axe >= Game.X_SIZE or y_axe >= Game.Y_SIZE:
54             return False
55             pprint("can not move, try again")
56         else:
57             self.matrix[old_x][old_y] = "-"
58             self.matrix[x_axe][y_axe] = "D"
59             self.player_position = new_position
60
61

```

Load_file to read and create a matrix (4 columns, 7 lines)
 initialize D like a Player (player_position)

Ruler to 80 characters

Fonction load_items to put items on the Game.
 (0, Game.X_SIZE -1) we decided to put -1 on code because i had error
 message «List out of range»

fonction to update the matrix with the D (player) moved.
 Error message if the player is out of matrix pprint(**«can not move, try again»**)

```
POO_P3.py x
61
62 class Player():
63
64     def __init__(self, game, *args, **kwargs):
65         self.game = game
66         self.list_collect = 0
67         self.numbers = 3
68         self.player_position = game.player_position
69         self.new_position = (0, 0)
70         self.move()
71
72     def move(self):
73         running = True
74         x_axe, y_axe = self.new_position
75         # self.player_position = self.game.x_axe, self.game.y_axe
76         # 0,0
77         while running:
78             option = str(
79                 input("press a key u->up, d->down, r->right, l->left "))
80             if option == "d":
81                 x_axe += 1
82                 new_position = (x_axe, y_axe)
83                 self.collect_item(new_position)
84                 self.game.update_maze(new_position)
85                 pprint(self.game.matrix)
86
87             # if self.game.can_i_move_to(x_axe, y_axe):
88
89             elif option == "u":
90                 # if self.game.can_i_move_to(x_axe, y_axe):
91                 x_axe -= 1
92                 new_position = (x_axe, y_axe)
93                 self.collect_item(new_position)
94                 self.game.update_maze(new_position)
95                 pprint(self.game.matrix)
96
97             elif option == "r":
98                 # if self.game.can_i_move_to(x_axe, y_axe):
99                 y_axe += 1
100                new_position = (x_axe, y_axe)
101                self.collect_item(new_position)
102                self.game.update_maze(new_position)
103                pprint(self.game.matrix)
104
105            elif option == "l":
106                # if self.game.can_i_move_to(x_axe, y_axe):
107                y_axe -= 1
108                new_position = (x_axe, y_axe)
109                self.collect_item(new_position)
110                self.game.update_maze(new_position)
111                pprint(self.game.matrix)
112
113            elif option == "q":
114                return exit()
115
116    def collect_item(self, position_items):
117        x_axe, y_axe = position_items
118        if self.game.matrix[x_axe][y_axe] in ("needle", "plastic_tube", "ether"):
119            pprint(self.game.matrix)
120            self.list_collect += 1
121            print("vous en avez :", self.list_collect)
122            self.game.matrix[x_axe][y_axe] == "-"
123            if self.game.matrix[x_axe][y_axe] == "A" and self.list_collect < self.numbers:
124                pprint("you loose")
125                return exit()
126
127
128 game = Game()
129 user = Player(game)
```

Result on Terminal :

the player move until 4 columns, then i have a error message :
List index out of range

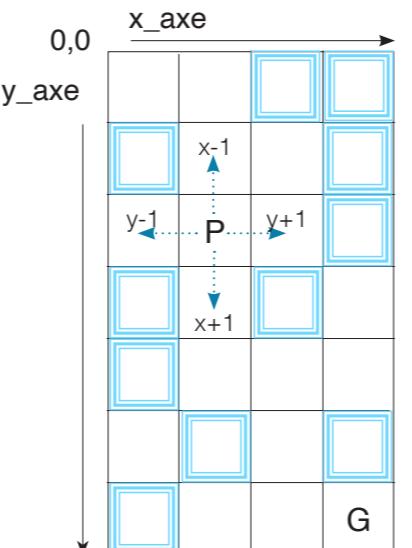
solution : i have made a mistake with X and Y axes.
switch the x_axe and y_axe, the line and columns

Function to move the player.

x_axe += 1 (move the player to the down direction)

`new_position = (x_axe, y_axe)` (*the player has a new position*)
`self.collect_item(new_position)` (*call function to collect items if the player is in items cell*)

self.game.update_maze(new_position) (update the maze with the new player position and if the player get a items)



`Up = y_axe - 1`

Down = y_axe + 1

Right = x_axe + 1

Left = x_axe - 1

Function to collect items.

if the player position is in items cell, he put the item on list_collect.

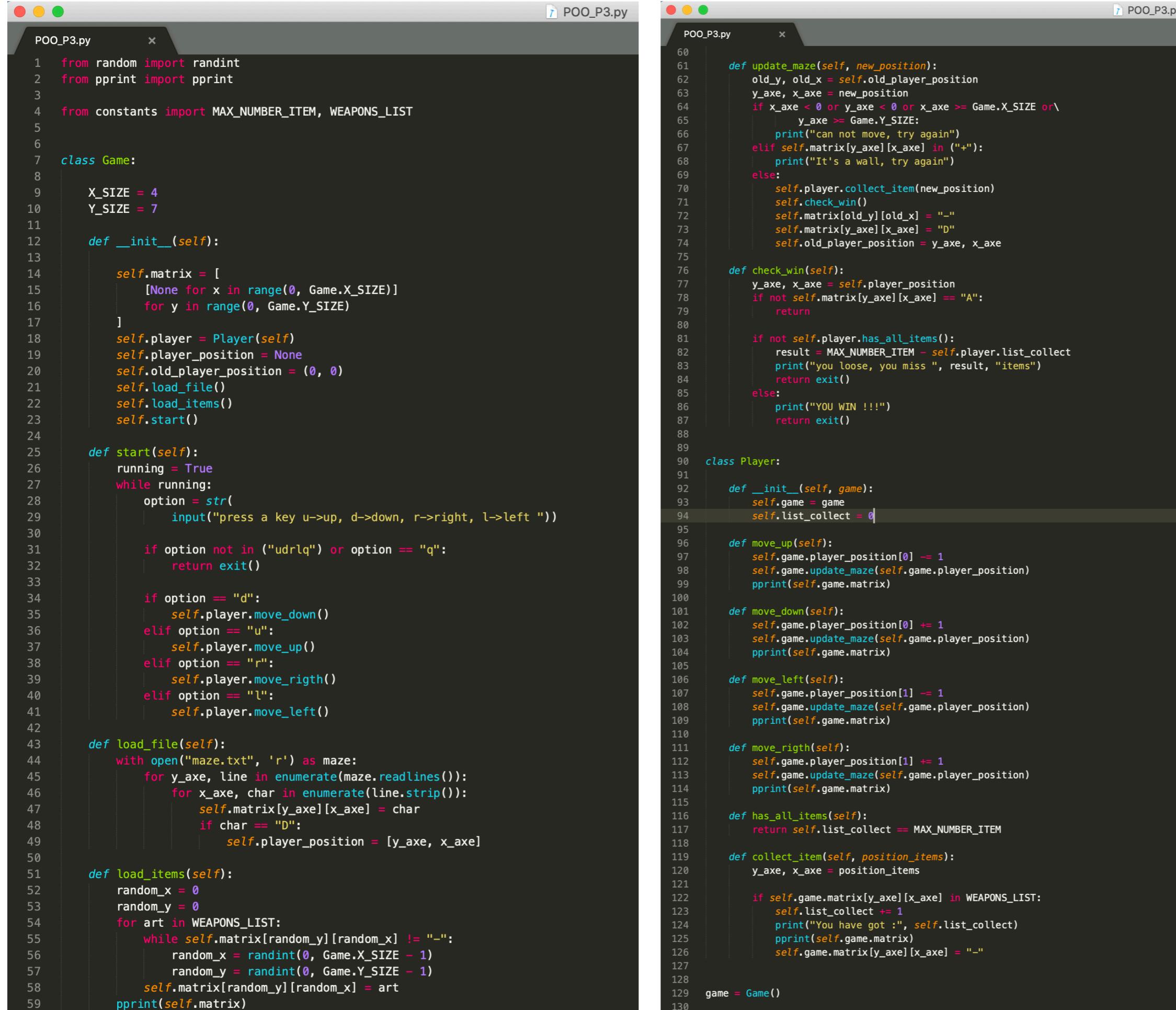
```
if self.game.matrix[x]
```

self.list_collect += 1

self.game.matrix[x_axe][y_axe] == '-' (*change item on Floor '-'*)

```
If the player go on to the guardian with items missing, he loose.  
if self.game.matrix[x_axe][y_axe] == 'A' and self.list_collect < self.numbers:  
    print('you loose')
```

Terminal_Projet 3 : Save MacGyver - work in progress - Update python code



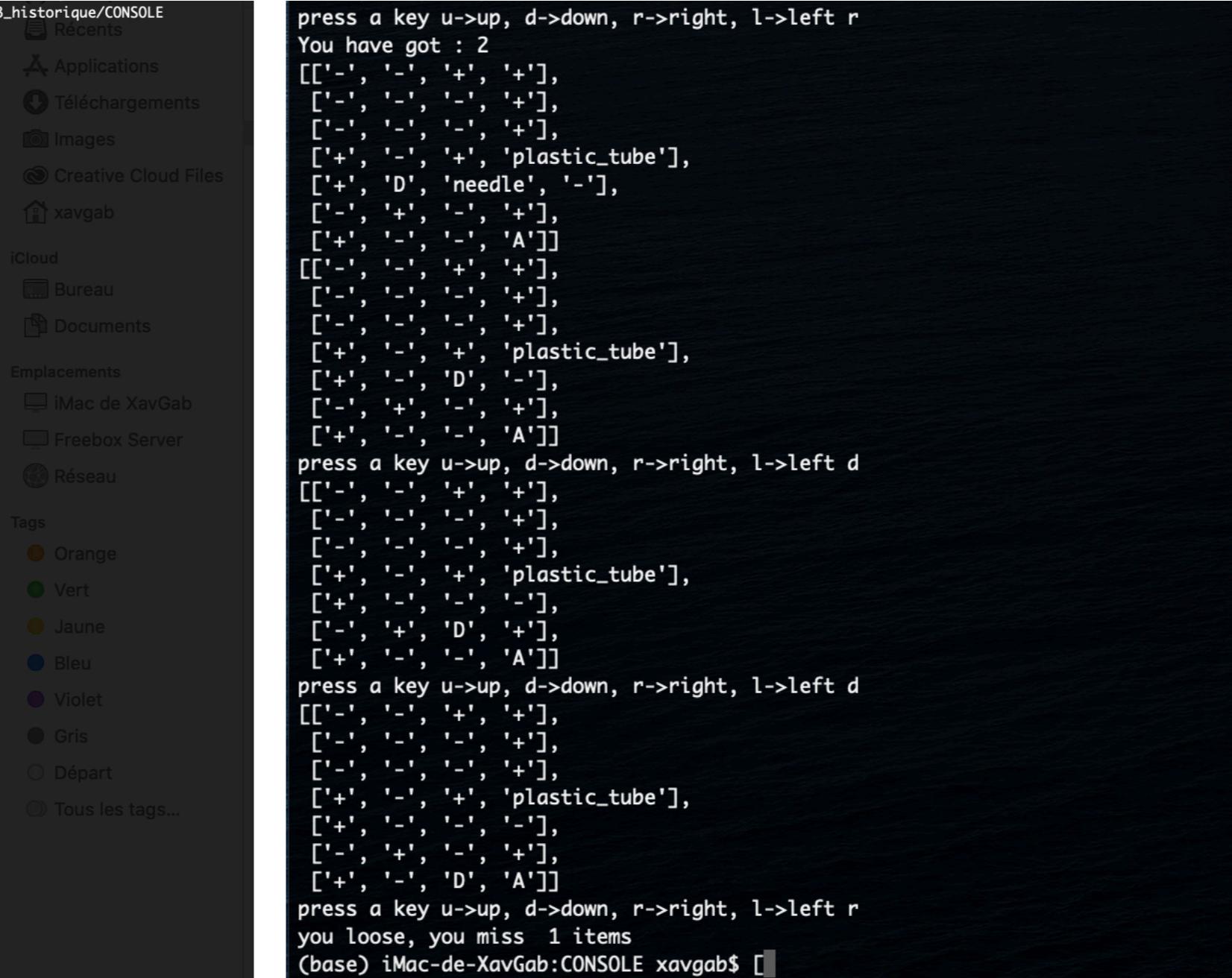
```

POO_P3.py
1  from random import randint
2  from pprint import pprint
3
4  from constants import MAX_NUMBER_ITEM, WEAPONS_LIST
5
6
7  class Game:
8
9      X_SIZE = 4
10     Y_SIZE = 7
11
12     def __init__(self):
13
14         self.matrix = [
15             [None for x in range(0, Game.X_SIZE)]
16             for y in range(0, Game.Y_SIZE)
17         ]
18         self.player = Player(self)
19         self.player_position = None
20         self.old_player_position = (0, 0)
21         self.load_file()
22         self.load_items()
23         self.start()
24
25     def start(self):
26         running = True
27         while running:
28             option = str(
29                 input("press a key u->up, d->down, r->right, l->left "))
30
31             if option not in ("udrlq") or option == "q":
32                 return exit()
33
34             if option == "d":
35                 self.player.move_down()
36             elif option == "u":
37                 self.player.move_up()
38             elif option == "r":
39                 self.player.move_righth()
40             elif option == "l":
41                 self.player.move_left()
42
43     def load_file(self):
44         with open("maze.txt", 'r') as maze:
45             for y_axe, line in enumerate(maze.readlines()):
46                 for x_axe, char in enumerate(line.strip()):
47                     self.matrix[y_axe][x_axe] = char
48                     if char == "D":
49                         self.player_position = [y_axe, x_axe]
50
51     def load_items(self):
52         random_x = 0
53         random_y = 0
54         for art in WEAPONS_LIST:
55             while self.matrix[random_y][random_x] != "-":
56                 random_x = randint(0, Game.X_SIZE - 1)
57                 random_y = randint(0, Game.Y_SIZE - 1)
58             self.matrix[random_y][random_x] = art
59             pprint(self.matrix)
60
61     def update_maze(self, new_position):
62         old_y, old_x = self.old_player_position
63         y_axe, x_axe = new_position
64         if x_axe < 0 or y_axe < 0 or x_axe >= Game.X_SIZE or \
65             y_axe >= Game.Y_SIZE:
66             print("can not move, try again")
67         elif self.matrix[y_axe][x_axe] in ("+"):
68             print("It's a wall, try again")
69         else:
70             self.player.collect_item(new_position)
71             self.check_win()
72             self.matrix[old_y][old_x] = "-"
73             self.matrix[y_axe][x_axe] = "D"
74             self.old_player_position = y_axe, x_axe
75
76     def check_win(self):
77         y_axe, x_axe = self.player_position
78         if not self.matrix[y_axe][x_axe] == "A":
79             return
80
81         if not self.player.has_all_items():
82             result = MAX_NUMBER_ITEM - self.player.list_collect
83             print("you loose, you miss ", result, "items")
84             return exit()
85         else:
86             print("YOU WIN !!!")
87             return exit()
88
89     class Player:
90
91         def __init__(self, game):
92             self.game = game
93             self.list_collect = []
94
95         def move_up(self):
96             self.game.player_position[0] -= 1
97             self.game.update_maze(self.game.player_position)
98             pprint(self.game.matrix)
99
100        def move_down(self):
101            self.game.player_position[0] += 1
102            self.game.update_maze(self.game.player_position)
103            pprint(self.game.matrix)
104
105        def move_left(self):
106            self.game.player_position[1] -= 1
107            self.game.update_maze(self.game.player_position)
108            pprint(self.game.matrix)
109
110        def move_righth(self):
111            self.game.player_position[1] += 1
112            self.game.update_maze(self.game.player_position)
113            pprint(self.game.matrix)
114
115        def has_all_items(self):
116            return self.list_collect == MAX_NUMBER_ITEM
117
118        def collect_item(self, position_items):
119            y_axe, x_axe = position_items
120
121            if self.game.matrix[y_axe][x_axe] in WEAPONS_LIST:
122                self.list_collect += 1
123                print("You have got :", self.list_collect)
124                pprint(self.game.matrix)
125                self.game.matrix[y_axe][x_axe] = "-"
126
127
128    game = Game()
129

```

Terminal_Projet 3 : Save MacGyver - work in progress - Terminal results

```
(base) iMac-de-XavGab:P3_historique xavgab$ cd /Users/xavgab/Documents/Python/P3_historique/CONSOLE
(base) iMac-de-XavGab:CONSOLE xavgab$ python P00_P3.py
[['D', '-', '+', '+'],
 ['-', '--', 'ether', '+'],
 ['-', '--', '+', '+'],
 ['+', '--', '+', 'plastic_tube'],
 ['+', '--', 'needle', '-'],
 ['-', '+', '--', '+'],
 ['+', '--', '--', 'A']]
press a key u->up, d->down, r->right, l->left d
[['D', '-', '+', '+'],
 ['D', '--', 'ether', '+'],
 ['--', '--', '+', '+'],
 ['+', '--', '+', 'plastic_tube'],
 ['+', '--', 'needle', '-'],
 ['-', '+', '--', '+'],
 ['+', '--', '--', 'A']]
press a key u->up, d->down, r->right, l->left r
[['D', '-', '+', '+'],
 ['D', '--', 'ether', '+'],
 ['--', '--', '+', '+'],
 ['+', '--', '+', 'plastic_tube'],
 ['+', '--', 'needle', '-'],
 ['-', '+', '--', '+'],
 ['+', '--', '--', 'A']]
press a key u->up, d->down, r->right, l->left r
You have got : 1
[['D', '--', '+', '+'],
 ['D', '--', 'ether', '+'],
 ['--', '--', '+', '+'],
 ['+', '--', '+', 'plastic_tube'],
 ['+', '--', 'needle', '-'],
 ['-', '+', '--', '+'],
 ['+', '--', '--', 'A']]
[['D', '--', '+', '+'],
 ['D', '--', 'ether', '+'],
 ['--', '--', '+', '+'],
 ['+', '--', '+', 'plastic_tube'],
 ['+', '--', 'needle', '-'],
 ['-', '+', '--', '+'],
 ['+', '--', '--', 'A']]
press a key u->up, d->down, r->right, l->left d
[['D', '--', '+', '+'],
 ['D', '--', 'ether', '+'],
 ['--', '--', '+', '+'],
 ['+', '--', '+', 'plastic_tube'],
 ['+', '--', 'needle', '-'],
 ['-', '+', '--', '+'],
 ['+', '--', '--', 'A']]
press a key u->up, d->down, r->right, l->left l
[['D', '--', '+', '+'],
 ['D', '--', 'ether', '+'],
 ['--', '--', '+', '+'],
 ['+', '--', '+', 'plastic_tube'],
 ['+', '--', 'needle', '-'],
 ['-', '+', '--', '+'],
 ['+', '--', '--', 'A']]
press a key u->up, d->down, r->right, l->left d
[['D', '--', '+', '+'],
 ['D', '--', 'ether', '+'],
 ['--', '--', '+', '+'],
 ['+', '--', '+', 'plastic_tube'],
 ['+', '--', 'needle', '-'],
 ['-', '+', '--', '+'],
 ['+', '--', '--', 'A']]
press a key u->up, d->down, r->right, l->left d
[['D', '--', '+', '+'],
 ['D', '--', 'ether', '+'],
 ['--', '--', '+', '+'],
 ['+', '--', '+', 'plastic_tube'],
 ['+', '--', 'needle', '-'],
 ['-', '+', '--', '+'],
 ['+', '--', '--', 'A']]
press a key u->up, d->down, r->right, l->left d
[['D', '--', '+', '+'],
 ['D', '--', 'ether', '+'],
 ['--', '--', '+', '+'],
 ['+', '--', '+', 'plastic_tube'],
 ['+', '--', 'needle', '-'],
 ['-', '+', '--', '+'],
 ['+', '--', '--', 'A']]
```



```
press a key u->up, d->down, r->right, l->left r
You have got : 2
[['D', '--', '+', '+'],
 ['D', '--', 'ether', '+'],
 ['--', '--', '+', '+'],
 ['+', '--', '+', 'plastic_tube'],
 ['+', '--', 'needle', '-'],
 ['-', '+', '--', '+'],
 ['+', '--', '--', 'A']]
[['D', '--', '+', '+'],
 ['D', '--', 'ether', '+'],
 ['--', '--', '+', '+'],
 ['+', '--', '+', 'plastic_tube'],
 ['+', '--', 'needle', '-'],
 ['-', '+', '--', '+'],
 ['+', '--', '--', 'A']]
press a key u->up, d->down, r->right, l->left d
[['D', '--', '+', '+'],
 ['D', '--', 'ether', '+'],
 ['--', '--', '+', '+'],
 ['+', '--', '+', 'plastic_tube'],
 ['+', '--', 'needle', '-'],
 ['-', '+', '--', '+'],
 ['+', '--', '--', 'A']]
press a key u->up, d->down, r->right, l->left d
[['D', '--', '+', '+'],
 ['D', '--', 'ether', '+'],
 ['--', '--', '+', '+'],
 ['+', '--', '+', 'plastic_tube'],
 ['+', '--', 'needle', '-'],
 ['-', '+', '--', '+'],
 ['+', '--', '--', 'A']]
press a key u->up, d->down, r->right, l->left d
[['D', '--', '+', '+'],
 ['D', '--', 'ether', '+'],
 ['--', '--', '+', '+'],
 ['+', '--', '+', 'plastic_tube'],
 ['+', '--', 'needle', '-'],
 ['-', '+', '--', '+'],
 ['+', '--', '--', 'A']]
press a key u->up, d->down, r->right, l->left r
you loose, you miss 1 items
(base) iMac-de-XavGab:CONSOLE xavgab$
```

Terminal_Projet 3 : Save MacGyver - work in progress - Pygame test

Flake8

```
POO_P3_pygame.py  x

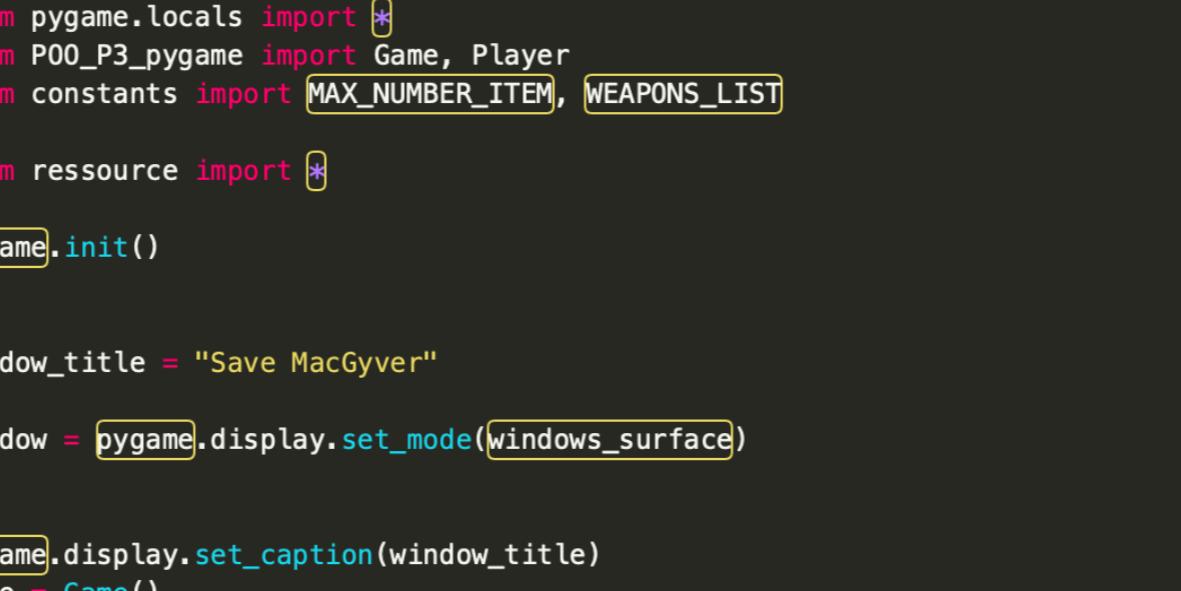
73         elif self.matrix[i][j] == 'plastic_tube':
74             window.blit(pipe, (j * 44, i * 44))
75         elif self.matrix[i][j] == 'ether':
76             window.blit(ether, (j * 44, i * 44))
77     pygame.display.flip()
78
79 def load_file(self):
80     with open("maze.txt", 'r') as maze:
81         for y_axe, line in enumerate(maze.readlines()):
82             for x_axe, char in enumerate(line.strip()):
83                 self.matrix[y_axe][x_axe] = char
84             if char == "D":
85                 self.player_position = [y_axe, x_axe]
86
87 def load_items(self):
88     random_x = 0
89     random_y = 0
90     for art in WEAPONS_LIST:
91         while self.matrix[random_y][random_x] != "-":
92             random_x = randint(0, Game.X_SIZE - 1)
93             random_y = randint(0, Game.Y_SIZE - 1)
94         self.matrix[random_y][random_x] = art
95     # pprint(self.matrix)
96
97 def check_maze(self, new_position):
98     old_y, old_x = self.old_player_position
99     y_axe, x_axe = new_position
100
101    if x_axe < 0 or y_axe < 0 or x_axe >= Game.X_SIZE or \
102        y_axe >= Game.Y_SIZE:
103        print("can not move, try again")
104        self.player_position = [old_y, old_x]
105        print(self.player_position)
106    elif self.matrix[y_axe][x_axe] == "+":
107        print("It's a wall, try again")
108        self.player_position = [old_y, old_x]
109
110 def update_maze(self, new_position):
111     old_y, old_x = self.old_player_position
112     y_axe, x_axe = new_position
113     self.player.collect_item(new_position)
114     self.check_win()
115     self.matrix[old_y][old_x] = "-"
116     self.matrix[y_axe][x_axe] = "D"
117     self.old_player_position = y_axe, x_axe
118
119 def check_win(self):
120     y_axe, x_axe = self.player_position
121     if not self.matrix[y_axe][x_axe] == "A":
122         return
123
124     if not self.player.has_all_items():
125         result = MAX_NUMBER_ITEM - self.player.list_collect
126         print("you loose, you miss ", result, "items")
127         font = pygame.font.SysFont("helvetica.ttc", 15)
128         message = font.render(
129             "YOU LOOSE !!!", False, pygame.Color("#ff0000"))
130         message_rect = message.get_rect()
131         message_rect.center = [rect_window.center
132             window.blit(message, message_rect)
133             pygame.display.flip()
134             pygame.display.update()
135             return exit()
136     else:
137         font = pygame.font.SysFont("helvetica bold.ttc", 30)
138         message = font.render(
139             "YOU WIN !!!", False, pygame.Color("#ff0000"))
140         message_rect = message.get_rect()
141         message_rect.center = [rect_window.center
142             window.blit(message, message_rect)
143             pygame.display.flip()
144             # pygame.display.update()
145             print("YOU WIN !!!")
146             # return exit()
```

```
POO_P3_pygame.py  x  flake8(25|0), Line 25, Column 26  master [36]  UTF-8
```

```
136     else:
137         font = pygame.font.SysFont("helvetica bold.ttc", 30)
138         message = font.render(
139             "YOU WIN !!!", False, pygame.Color("#ff0000"))
140         message_rect = message.get_rect()
141         message_rect.center = rect_window.center
142         window.blit(message, message_rect)
143         pygame.display.flip()
144         # pygame.display.update()
145         print("YOU WIN !!!")
146         # return exit()
147
148
149 class Player(pygame.sprite.Sprite):
150
151     def __init__(self, game):
152         pygame.init()
153         self.game = game
154         self.list_collect = 0
155
156     def move_up(self):
157         self.game.player_position[0] -= 1
158         self.game.check_maze(self.game.player_position)
159         self.game.update_maze(self.game.player_position)
160         pprint(self.game.matrix)
161
162     def move_down(self):
163         self.game.player_position[0] += 1
164         self.game.check_maze(self.game.player_position)
165         self.game.update_maze(self.game.player_position)
166         pprint(self.game.matrix)
167
168     def move_left(self):
169         self.game.player_position[1] -= 1
170         self.game.check_maze(self.game.player_position)
171         self.game.update_maze(self.game.player_position)
172         pprint(self.game.matrix)
173
174     def move_right(self):
175         self.game.player_position[1] += 1
176         self.game.check_maze(self.game.player_position)
177         self.game.update_maze(self.game.player_position)
178         pprint(self.game.matrix)
179
180     def has_all_items(self):
181         return self.list_collect == MAX_NUMBER_ITEM
182
183     def collect_item(self, position_items):
184         y_axe, x_axe = position_items
185         if self.game.matrix[y_axe][x_axe] in WEAPONS_LIST:
186             self.list_collect += 1
187             # print("You have got :", self.list_collect)
188             # pprint(self.game.matrix)
189             self.game.matrix[y_axe][x_axe] = "-"
190
191
192 game = Game()
```

Terminal_Projet 3 : Save MacGyver - work in progress - Terminal results + Pygame interface

PYGAME — python start_pygame.py — 118



```
start_pygame.py      x
• 1 from pygame.locals import *
• 2 from P00_P3_pygame import Game, Player
• 3 from constants import MAX_NUMBER_ITEM, WEAPONS_LIST
• 4
• 5 from ressource import *
• 6
• 7 pygame.init()
• 8
• 9
10 window_title = "Save MacGyver"
11
• 12 window = pygame.display.set_mode(windows_surface)
13
14
• 15 pygame.display.set_caption(window_title)
16 game = Game()
17 user = Player(game)
18 |
```



Projet 3 : Save MacGyver

Final proposition_Projet 3 : Save MacGyver - Pygame Game - MVC Pattern

