Faculty of Engineering,

Alexandria University

Computer and Systems Engineering Department

Final project
"Circus Of Plates"
Report

Names :

-Yomna Ahmed 72

-Nada Ahmed 67

-Marwan Tarek 62

-Mohamed Elsabaggh 52

Description:

It is single player-game in which each clown carries two stacks of plates, and there are a set of colored plates queues that fall and he tries to catch them, if he manages to collect three consecutive plates of the same color, then they are vanished and his score increases. You are free to put rules ending the game.

Code Design:

Briefly,in this code we use 12 design patterns which are:

1. Singleton
2. Factory or Pool
3. Iterator
4. Dynamic Linkage
5. Snapshot
6. State
7. Strategy
8. Flyweight
9. Observer
10. immutable
11. marker interface

Of course we use the MVC in the whole code.

USER GUIDE:

- In this game we have three levels ,every level has its difficulty and the design differ from level to another as it will be shown in the samples by changing the speed of the objects fall on the clown and the numbers of the objects fall on the clown.

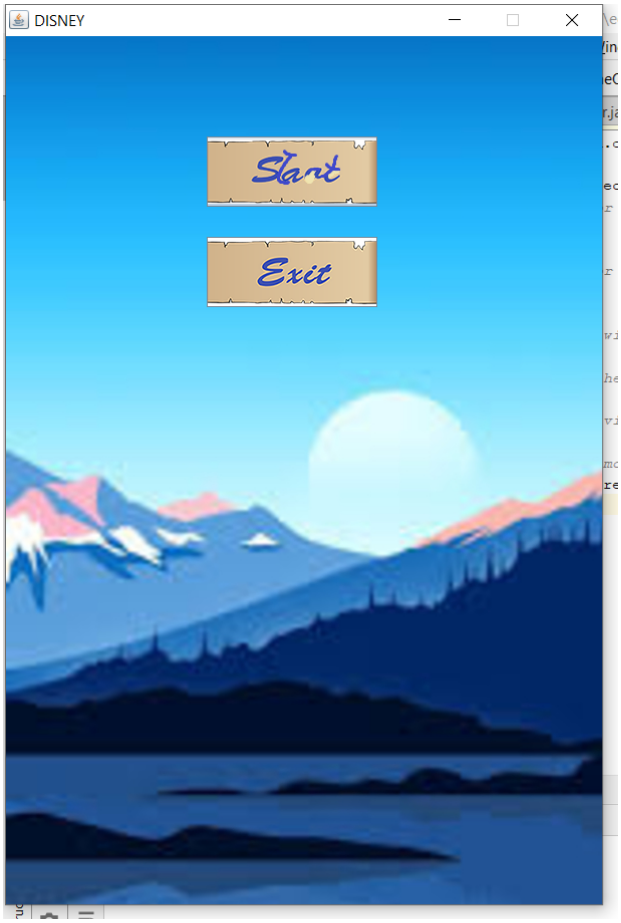
- The player move the left and right arrows on the keyboard to move the clown .

- Levels :

1) level 1 easy , normal speed
2) level 2 medium , speed increase
3) level 3 hard , speed increase , and bombs fall with the plates and the balls .

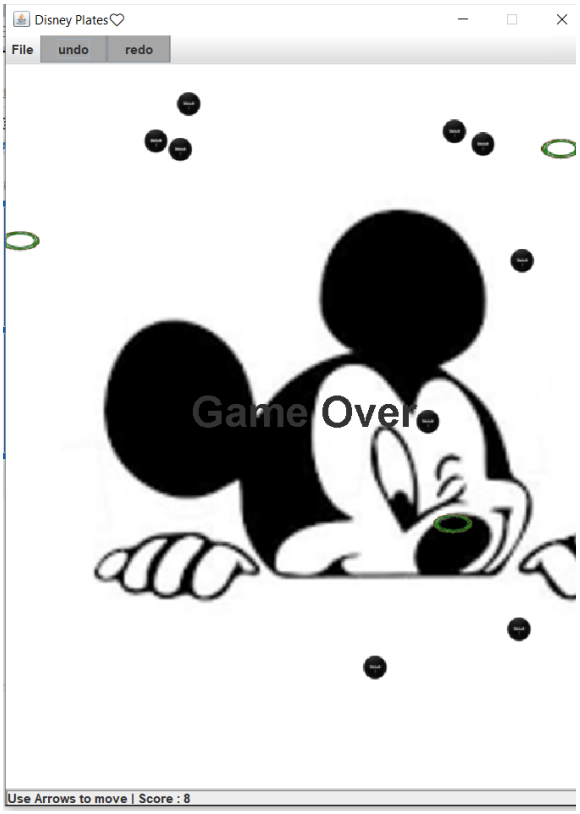
- The player loses by two different ways ,The first way is founded to be the main role of losing as it is the way used in the three levels ,while the second type of losing is only used in the third level
- The first way :by reaching the top of the screen as it will be shown in the samples.
- The second way :by hitting a Bomb

Samples:



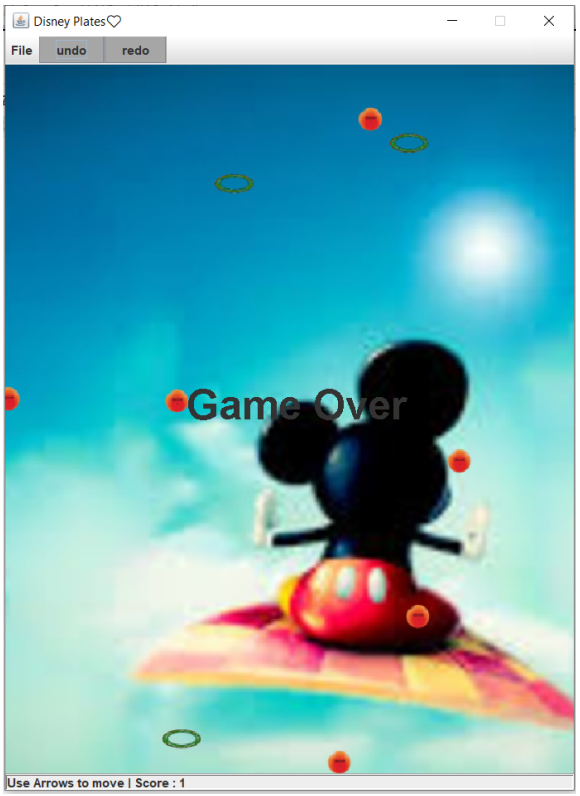
Level 1: "Easy"

The player loses when the plates reach the top of the screen :

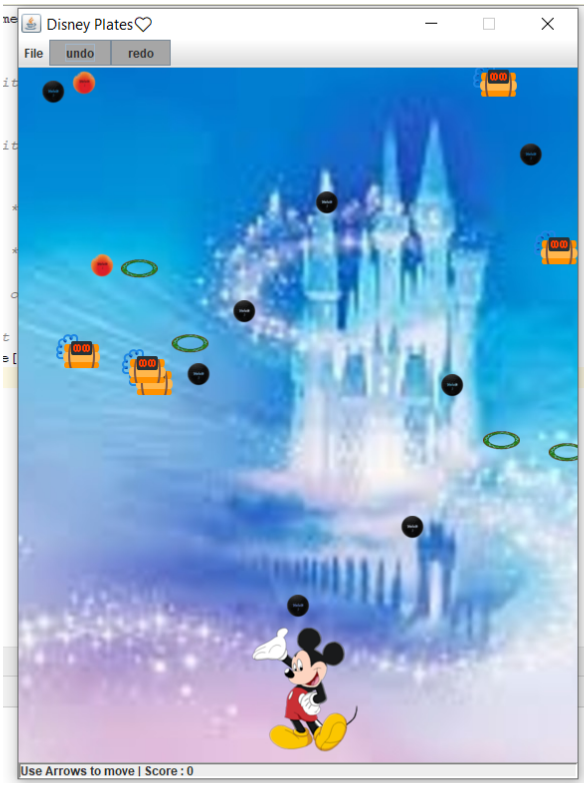


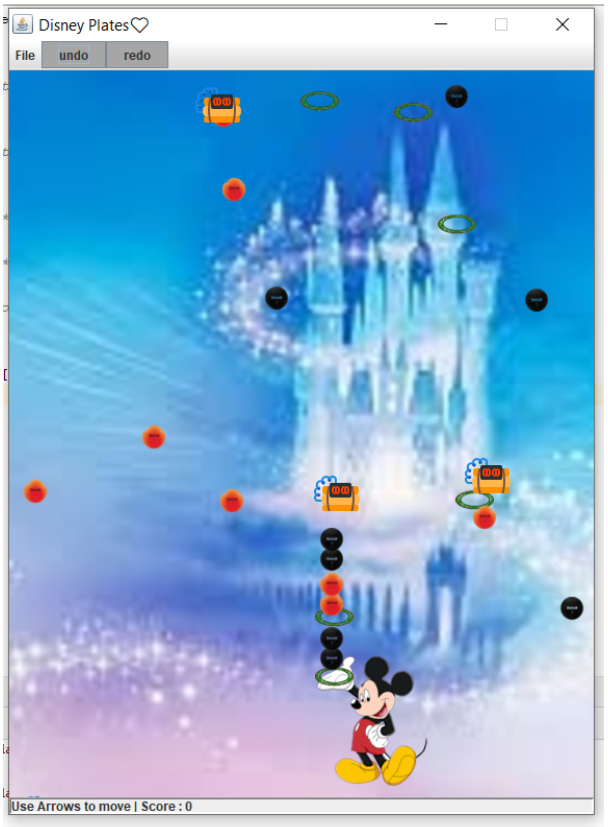
Level2: "Medium"

The player loses when the plates reach the top of the screen :

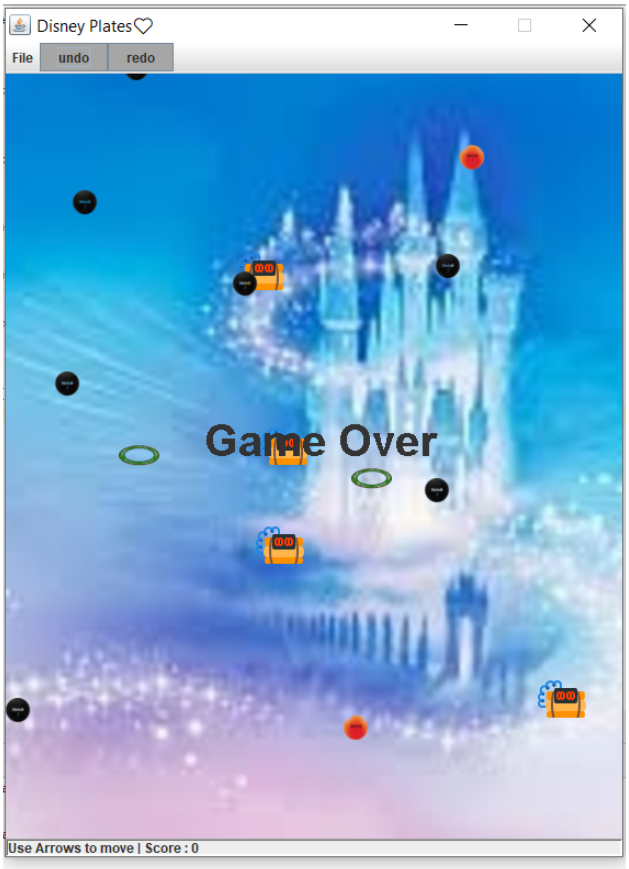


Level3: "Hard"


Disney Plates
File
undo
redo
Use Arrows to move | Score : 0

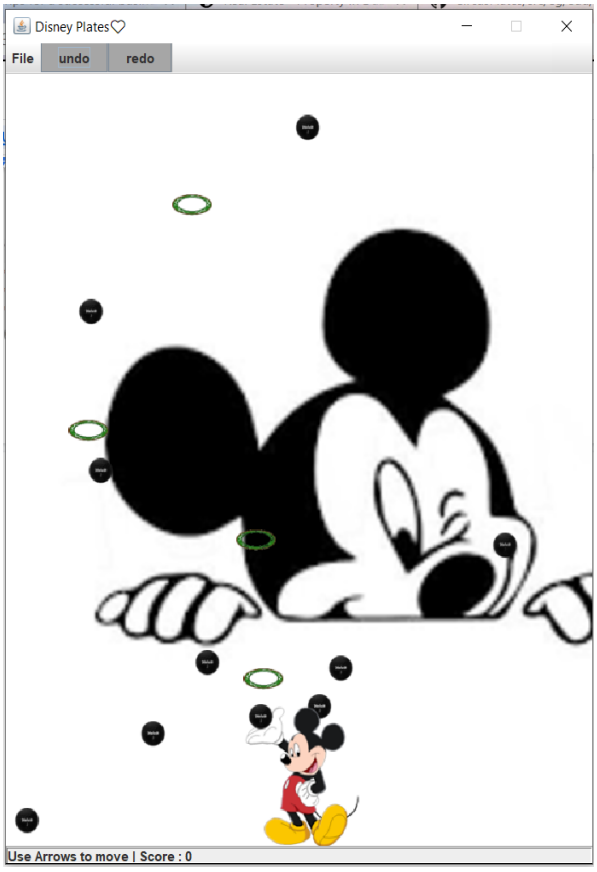

Disney Plates
File
undo
redo
Use Arrows to move | Score : 0

The user loses when the clown hits a bomb:



Undo & Redo samples:

After undo 2 times :

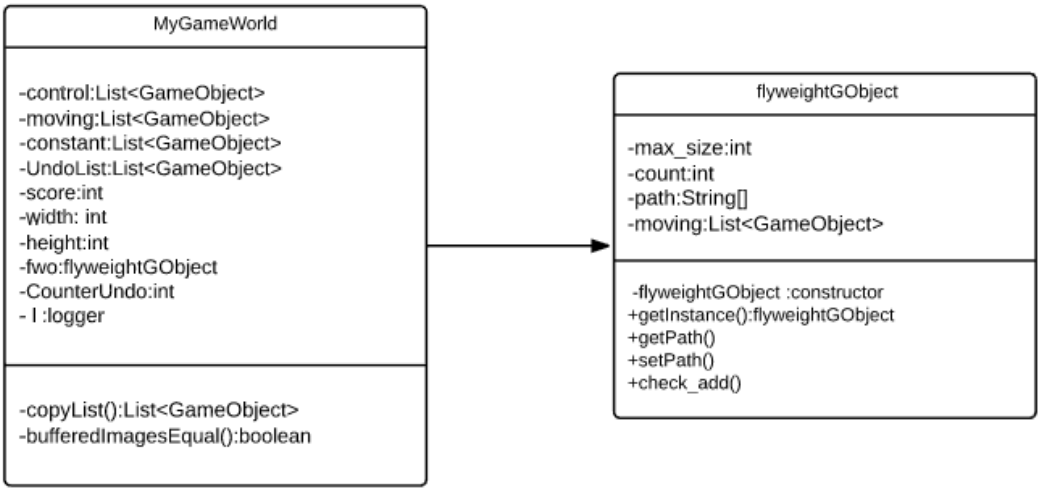


Redo :

Design patterns:

1) flyWeight design pattern:

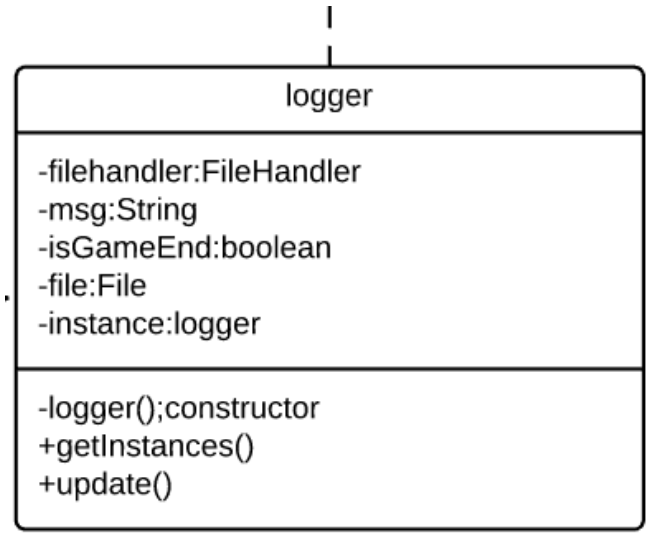
- we use it to reuse the objects fall on the clown ,so that we decrease memory usage

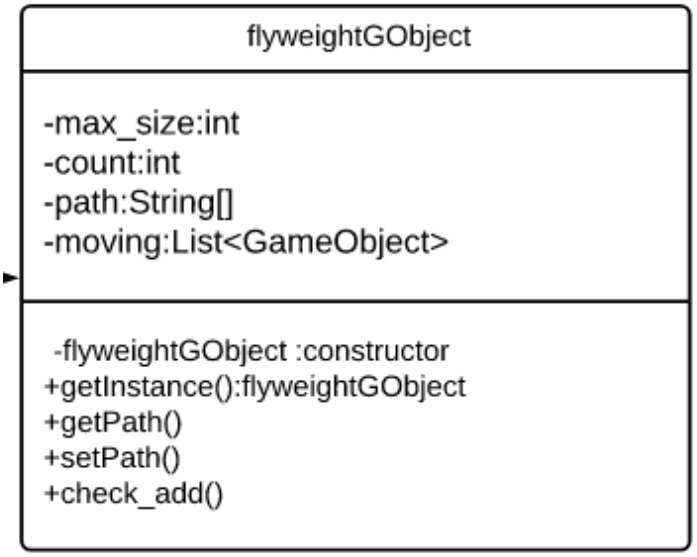


2)Singleton design pattern:

- We use it in two classes :

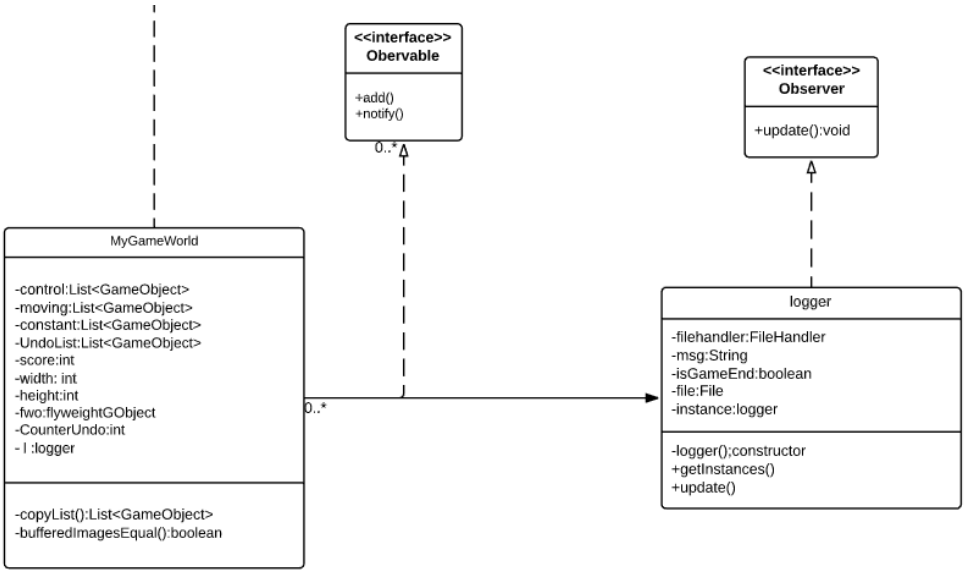
1. In logger class
2. In flyweightGObject class





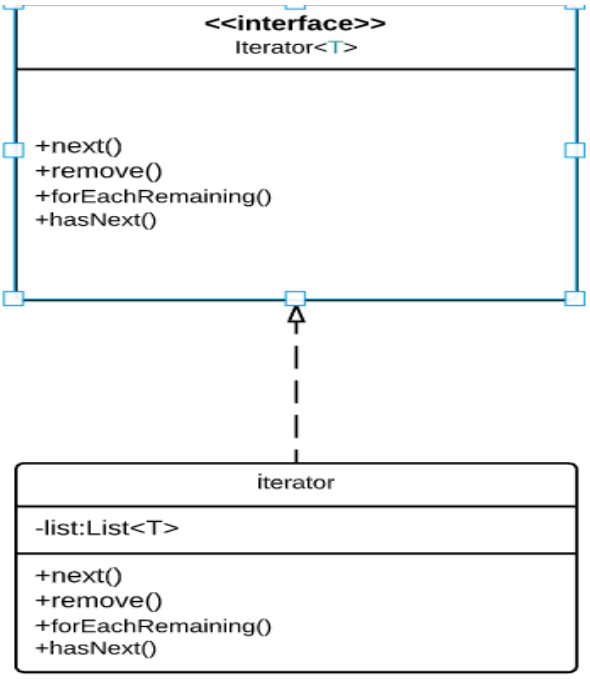
3)Observer design pattern:

- we use it in MyGameWorld class by the help of logger class ,as it used to notify the logger if any important event happen such as:lose ,catch plate ,the score is changed ,and if the bomb is near to the clown



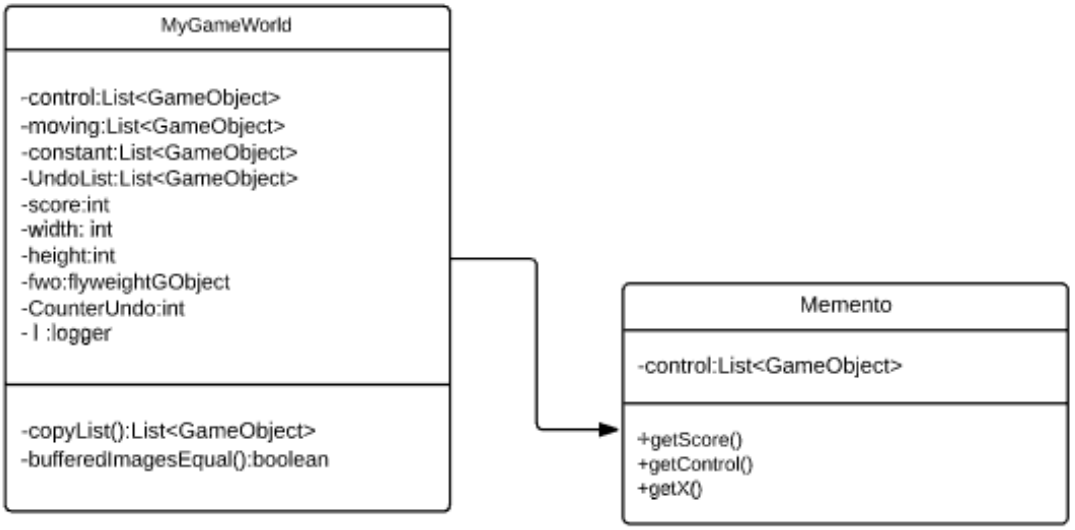
4)Iterator design pattern:

- we use this design pattern in many classes by the help of the iterator class



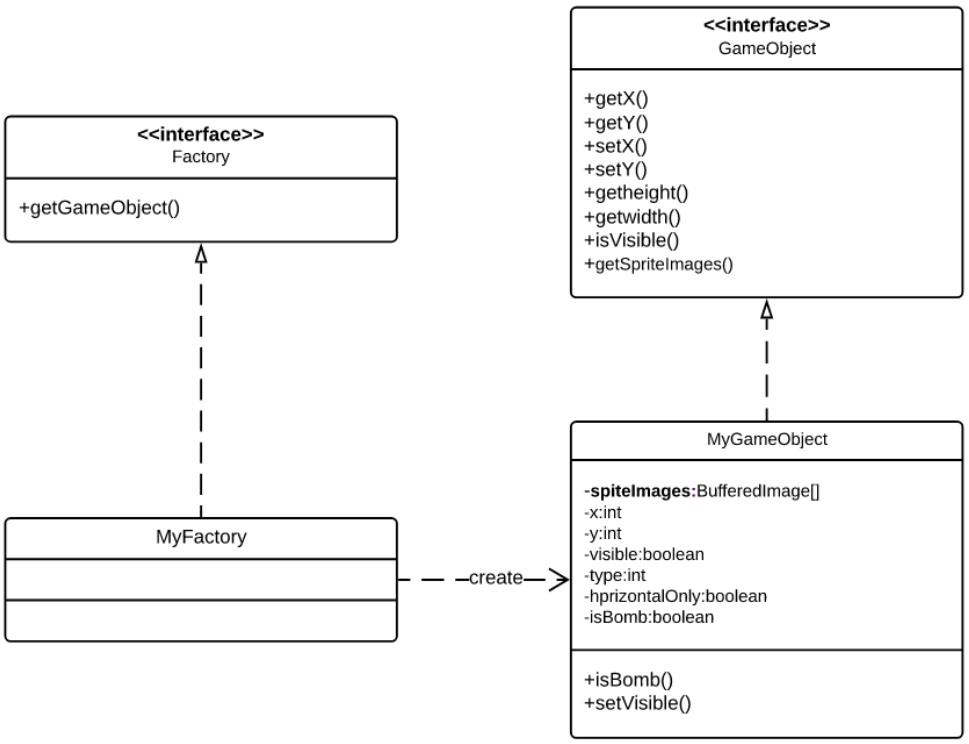
5)Snapshot design pattern:

- we use it to make the redo and undo by the helper class memento which help us save moments in the game



6)Factory method design pattern:

- We use it to create GameObject by the help of MyFactory Class.

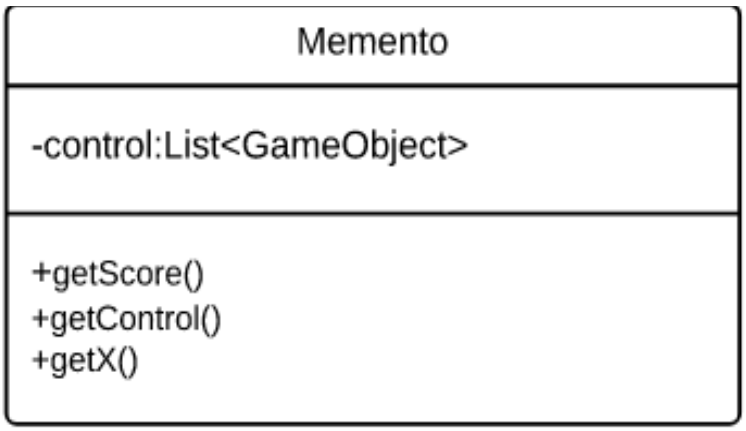


7)Dynamic linkage design pattern:

- We use it so that we dynamically load any photo in the shapes folder to the game in the MyGameWorld class.

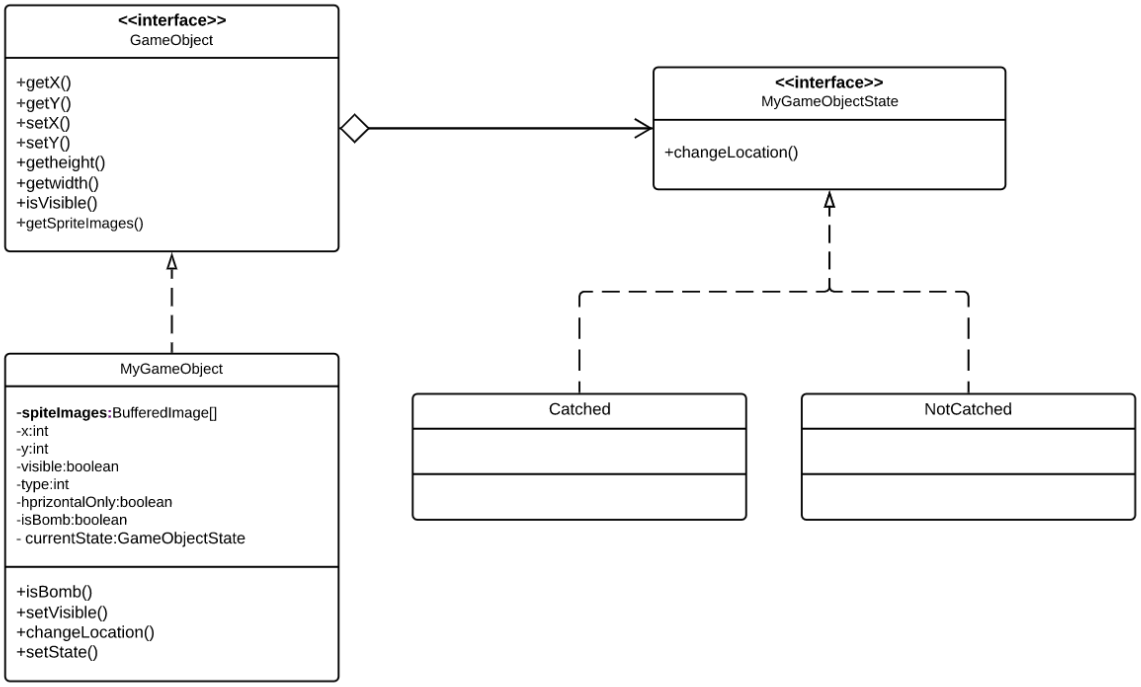
8)Immutable design pattern:

- The memento class has only getters and you can't edit in it.



9)State design pattern:

- we use it as every shape has two states catched and not catched and each of them controls his location change.



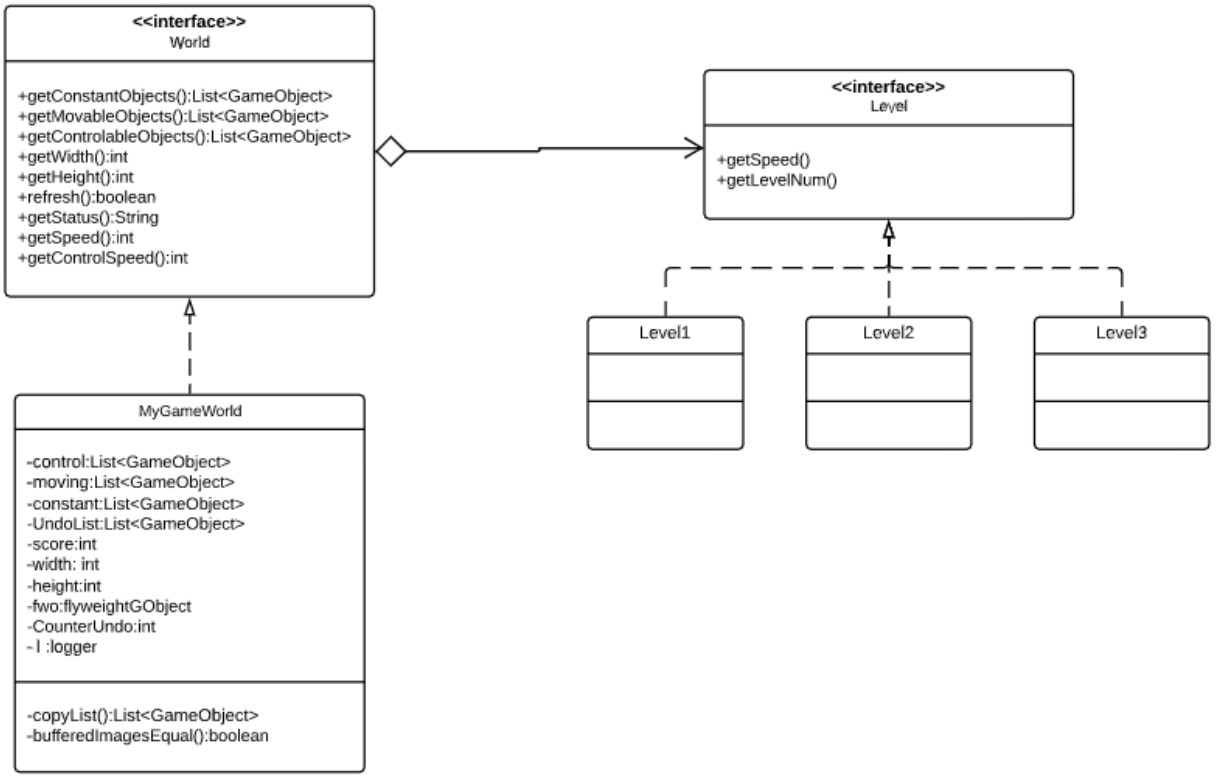
10)Marker interface design pattern:

- we use it in the class of logger to check if it's an instance of String.

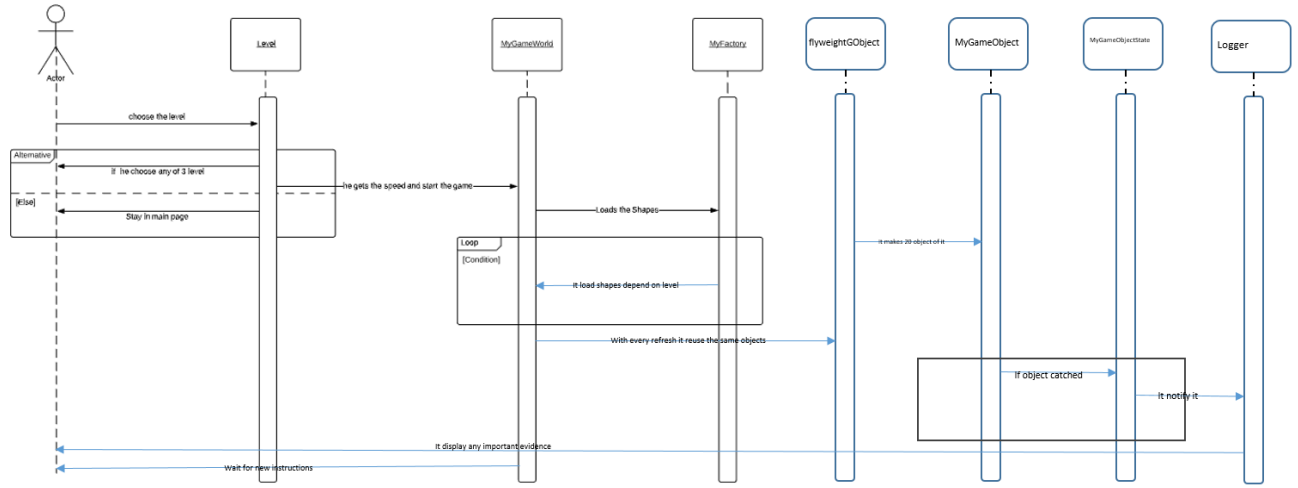
```
} else {
    if (arg instanceof String) {
        msg = "An item is catched!";
```

11)Strategy design pattern:

- We use it to set the 3 levels (easy,medium,hard).



Sequence diagram:



Uml Diagram :