Circus of Plates Game

Hassan Tamer 7405
Mostafa AbdelAtty Sultan 7900
Omar Essam Mohamed 7869
Yassin Medhat Helmy 7574

Fall 2022

—

Programming 2

—

1.Design and how to play

It is a 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 vanish and his score increases.

The game starts by asking the user to choose the difficulty to start the game as shown in fig 1.1

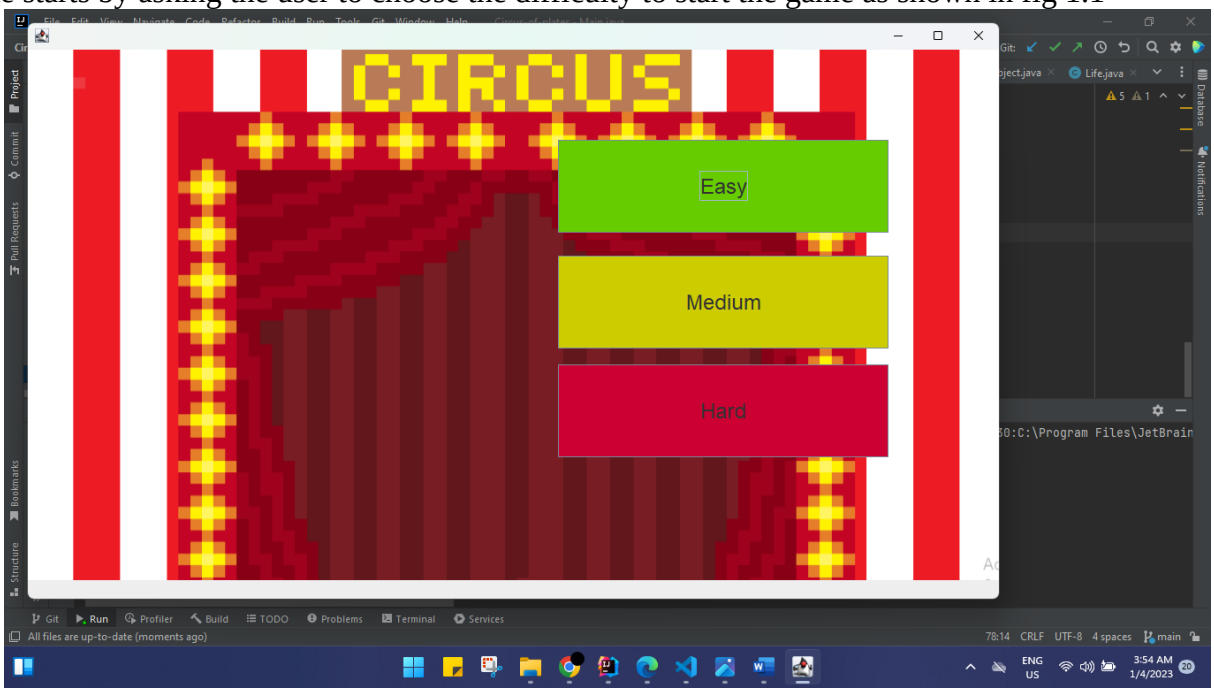


Fig 1.1 the start menu of the difficulties

The difficulties correspond to the speed of the shapes and the number of the bombs falling

Then you start the game with showing the clown and the 2 sticks to collect the cavity to fall into the stick as shown in fig 1.2

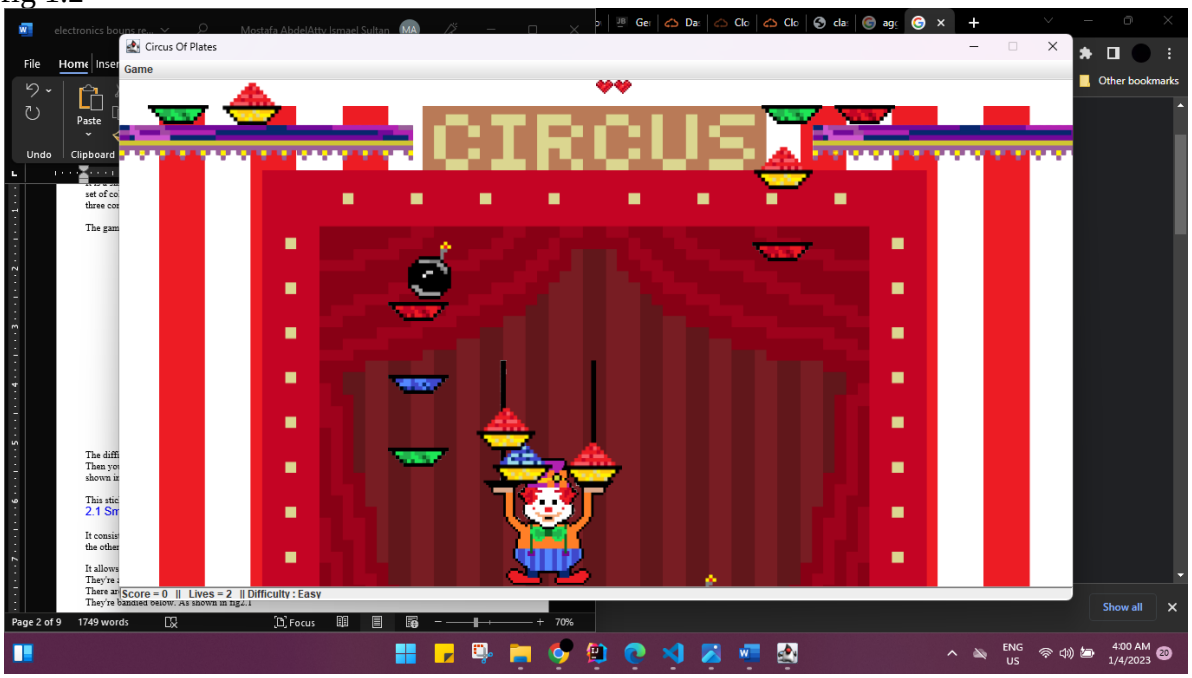
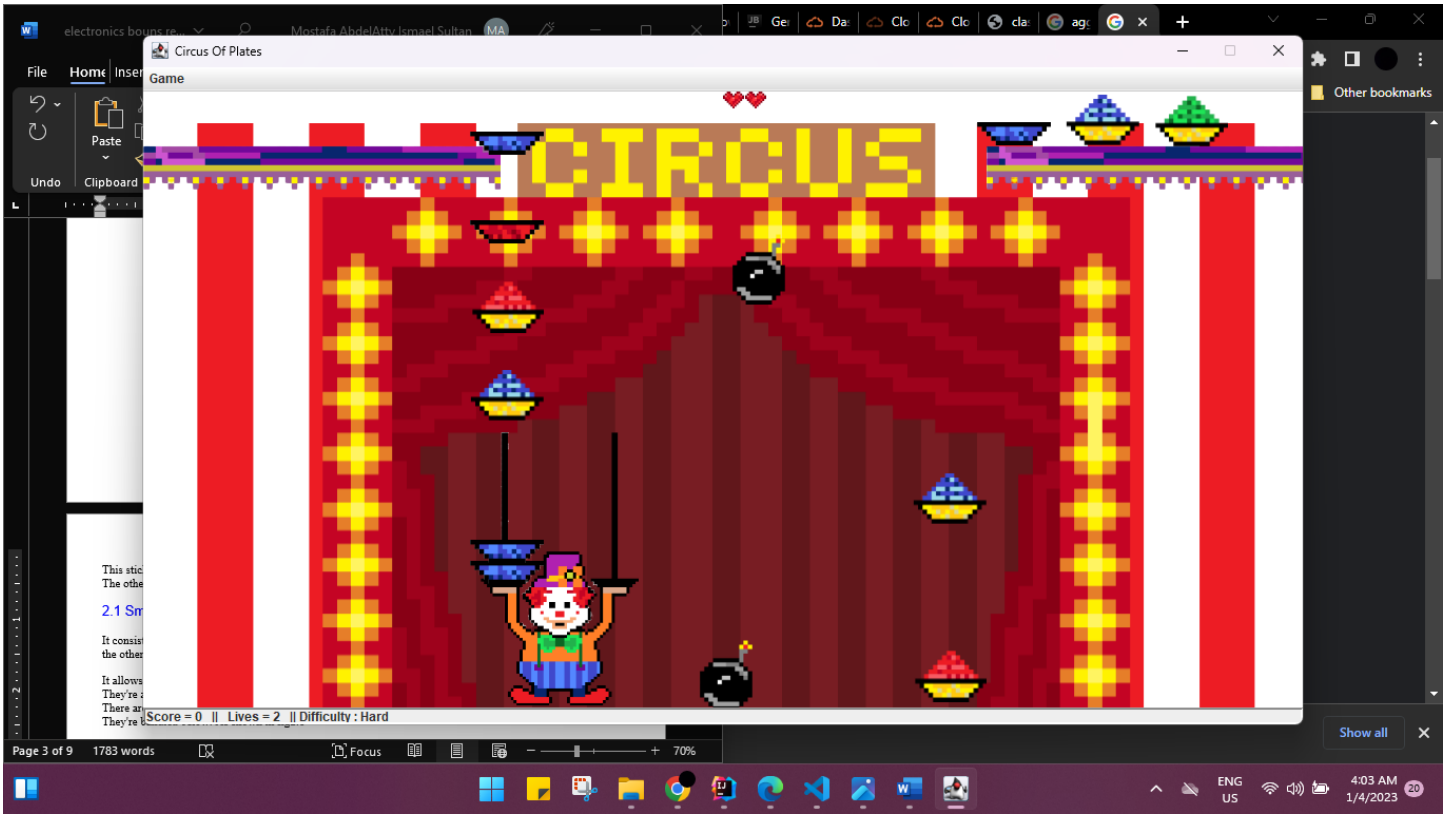


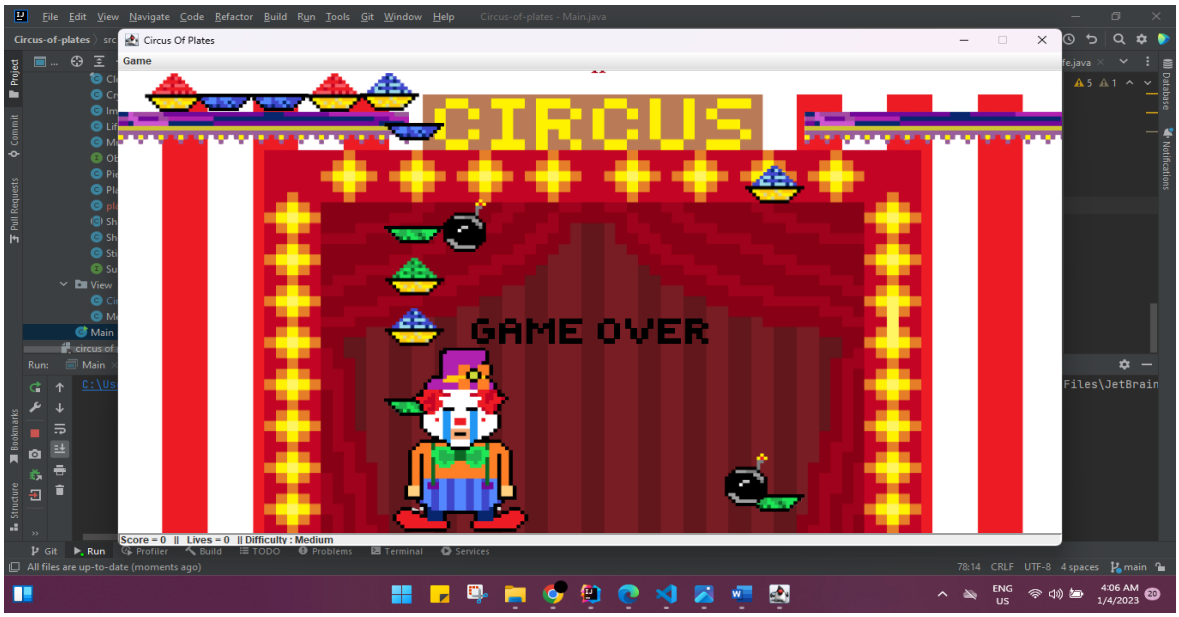
Fig 1.2 how the clown collects shapes

This stick has a maximum number of plates and they all fall once they reach the maximum height of the stick
The other way is to collect 3 consecutive shapes with the same color



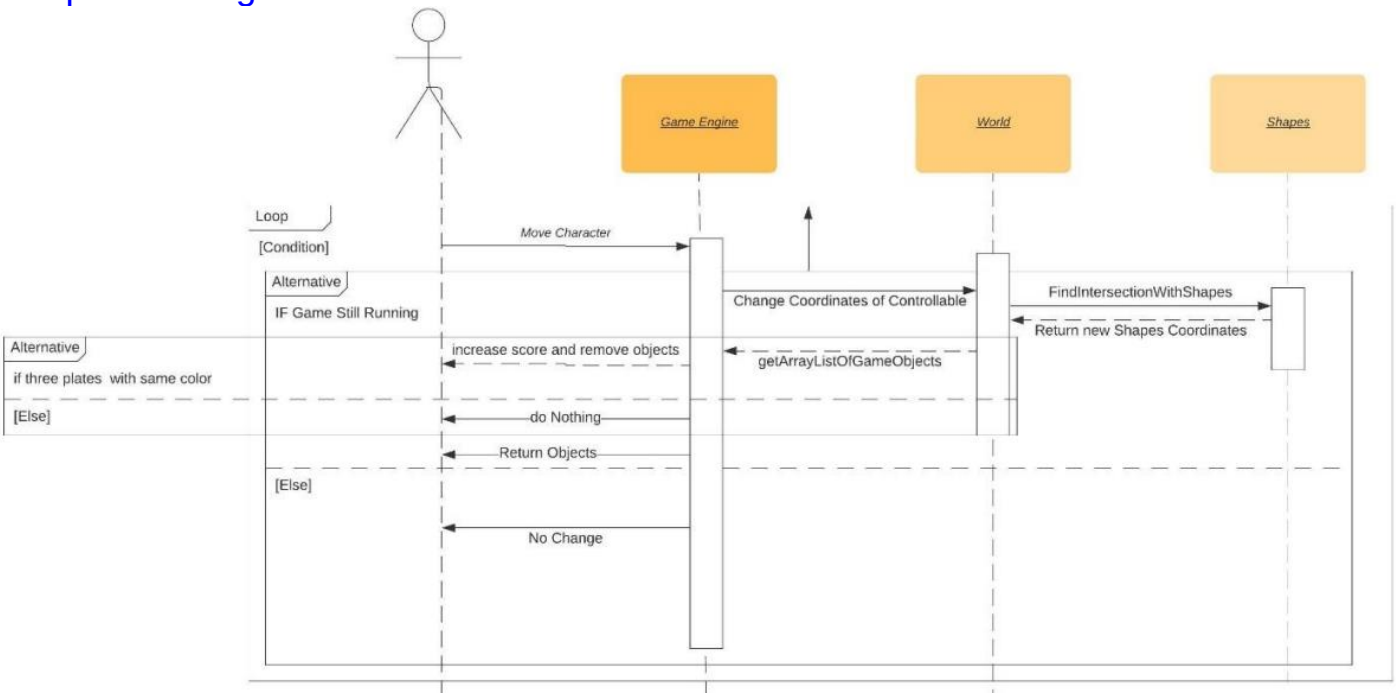
Scores and lives

Once you complete 3 plates you get one point and you start with 2 lives you lose lives in 2 ways getting the sticks to the full or receiving a bomb. Which makes the clown cry and the game is over



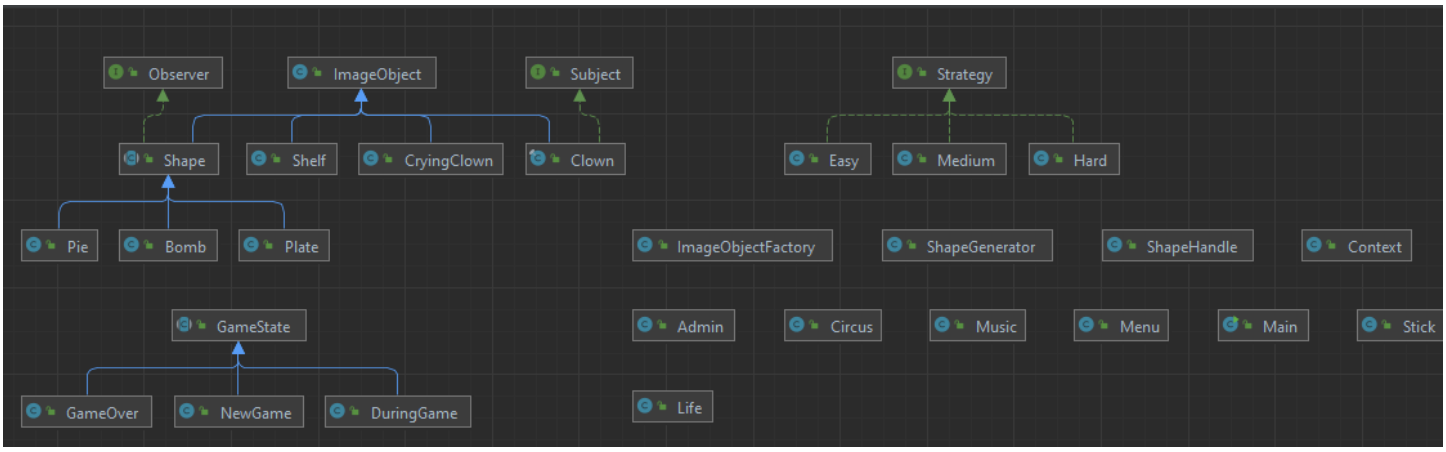
It allows the inflow of current in the forward direction and blocks it in rear direction.

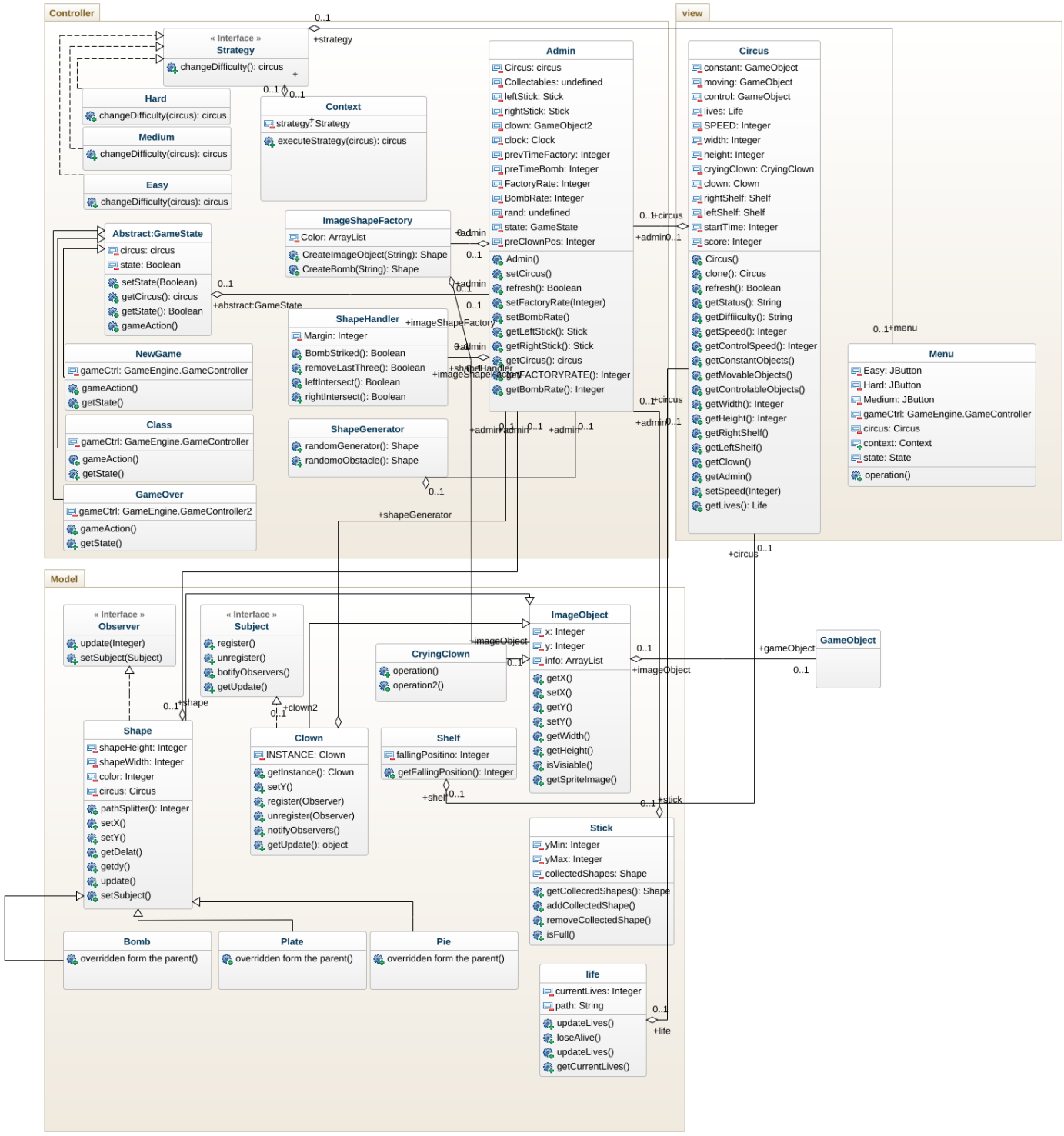
2.Sequence diagram



The Class diagram

The relations between classes and their attributes and different uses and instances





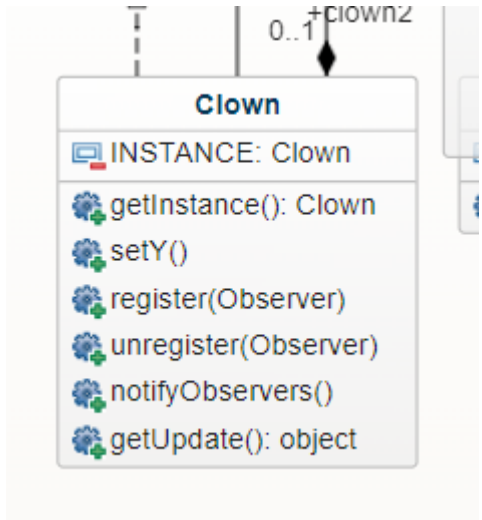
This follows the MVC model to control how the game works removing some dependency from the game engine

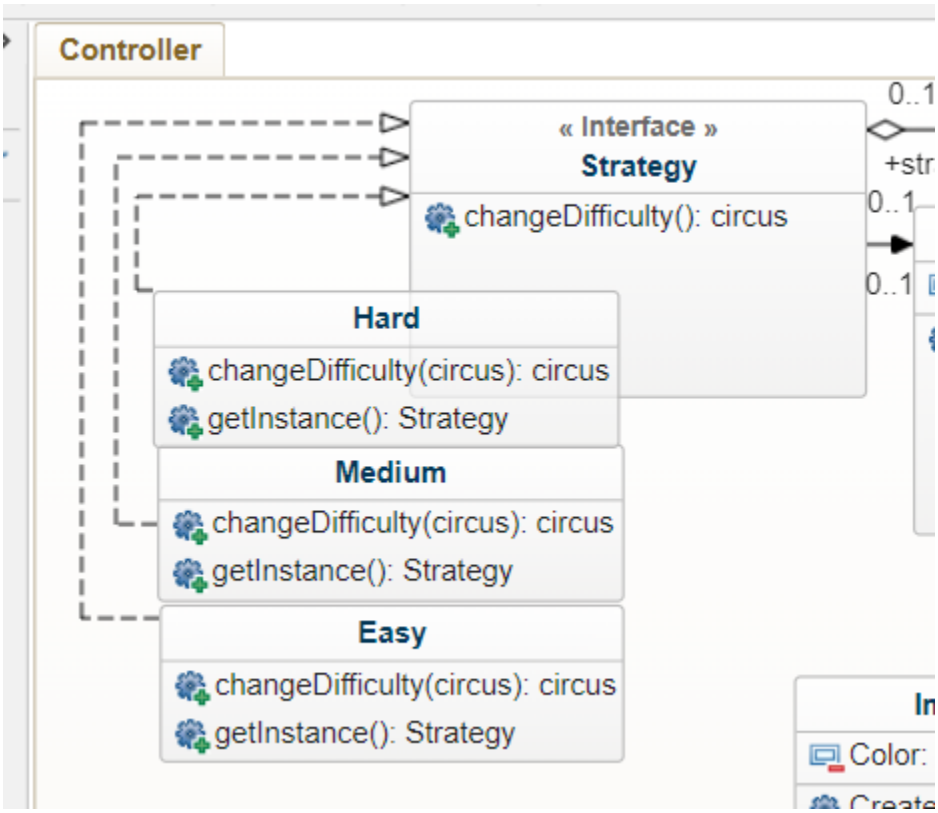
Design patterns

1.Singleton

We used it in the clown and the strategies part not to load it once and use the same reference every time using the method get Instance

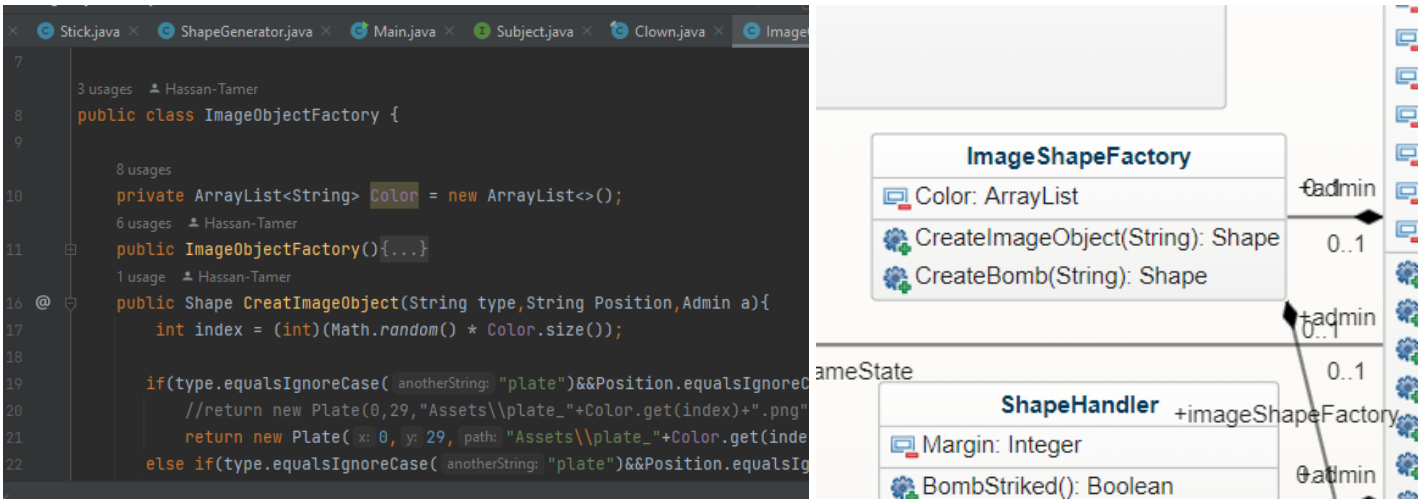
```java
public final class Clown extends ImageObject implements Subject{
    3 usages
    private static Clown INSTANCE;
    3 usages
    private List<Observer> observers = new ArrayList<>();
    1 usage  Hassan-Tamer
    private Clown(int x, int y, String path) {
        super(x, y, path);
    }

    1 usage  Hassan-Tamer
    public static Clown getInstance(int x, int y, String path) {
        if(INSTANCE == null) {
            INSTANCE = new Clown(x,y,path);
        }
        return INSTANCE;
```





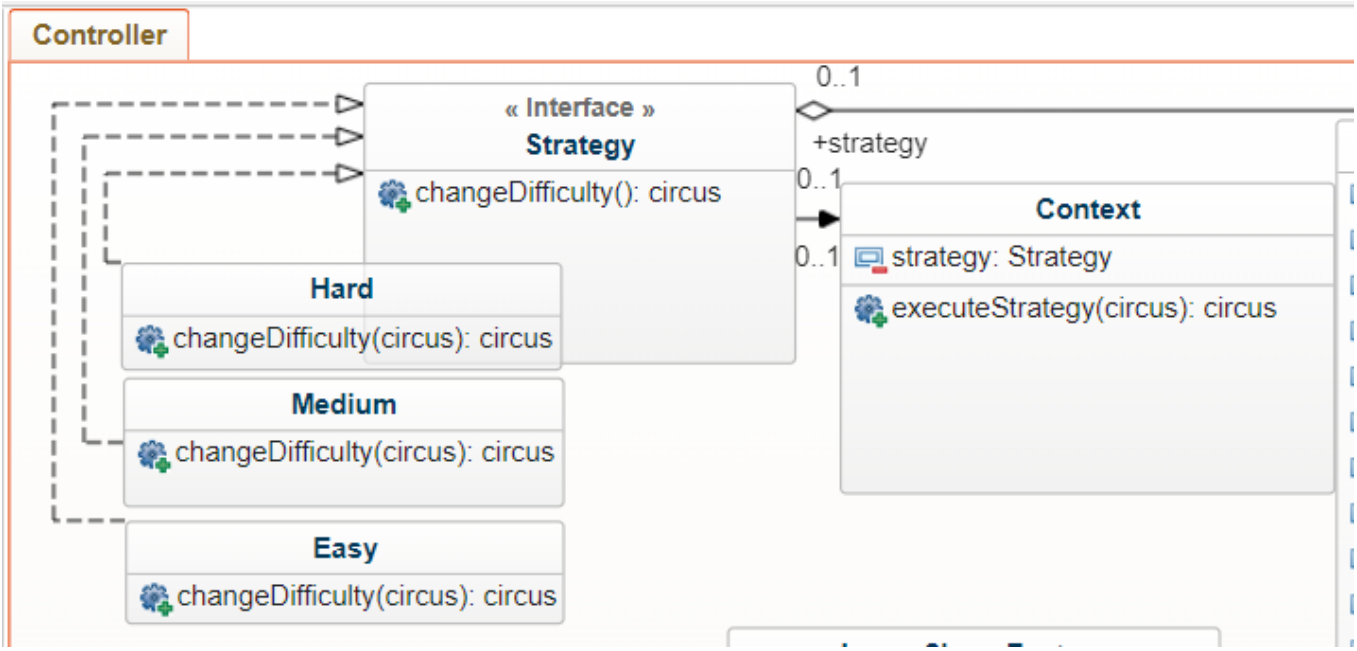
2.factory

Creates the images of the plates and pies randomly without making separate classes and constructors for each and we can all it once



3.Stratgy

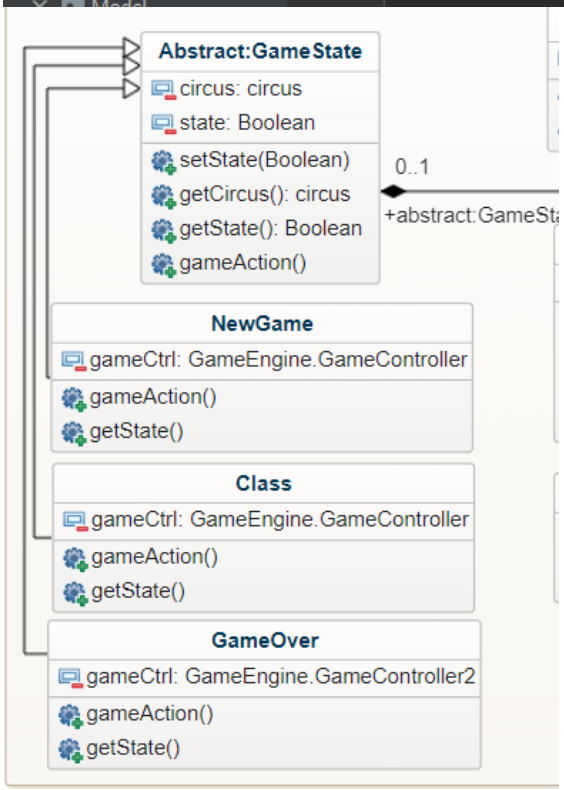
We used it to determine the difficulty of the game the user set



4.State

Used in determining the state of the game while it was working or when we are choosing a new game or game over part

```java
package Controller;

import View.Circus;

5 usages  Hassan-Tamer
public class Context {

    2 usages
    private Strategy strategy;

    4 usages  Hassan-Tamer
    public Context(Strategy strategy) { this.strategy = strategy; }

    3 usages  Hassan-Tamer
    public Circus executeStrategy(Circus c) { return strategy.changeDifficulty(c); }
}
```



5.Observer

Used in the clown moving part when each refresh updates the shapes with the new state of the clown so they can more consequently.

And we register them and unregister them when they reach the intersection part in the code