***Circus of Plates***

***Mosab Mohamed khames # 39***

***Ahmed ibrahem khattab # 55***

**Description of usage of design pattern in the game :**

***MVC :***

The ***project*** can be divided into ***three*** parts . ***the model*** , ***the view*** and ***the controller*** .   
the model is represented through all the classes of the project except for two classes .

One of them is named view and it contains all the members related to the interface of the game , the ***GUI*** of the game like frames , panels , graphics ,buttons and so on .

The other class is the controller class which is responsible for controlling changes that takes place through model regarding the ***data*** or the ***behavior*** and impacting this change on the view class .

So this design pattern is responsible for the separation of these ***three components*** .

***Observer design pattern :***

We use this design pattern (interface) to make the updates for each player of the two player and the classes that implemented that interface are **player1** and **player2** .  
the observer design pattern should be implemented with another interface (subject) that send the update information to the class that implements the observer.  
the subject interface is implemented by the model class .

***The Memento design pattern*** ***:***

This design pattern is for ***saving*** the state of each player and for saving the state of the balls as free and that belongs to each player and for saving state the balls that in the stack of each player.  
there are two classes that should be implemented with this design pattern the caretaker class and the originator class.  
the caretaker have a list of the previous states of each object and the caretaker for adding new object to save the originator.

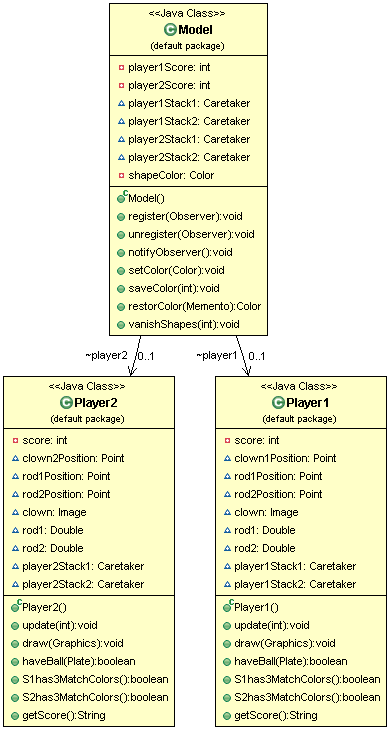
***Object pool design pattern :***

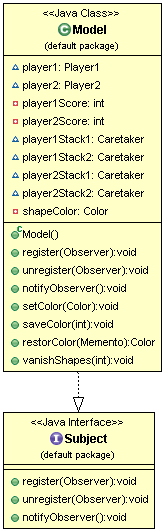
We didn’t implement this design pattern in out game but we use a method that do the same work that the design pattern do.

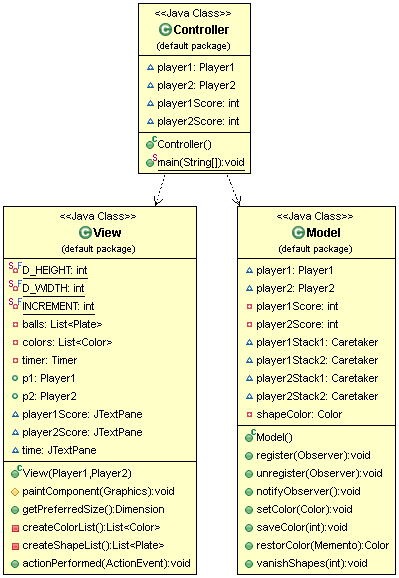
***Iterator design pattern :***

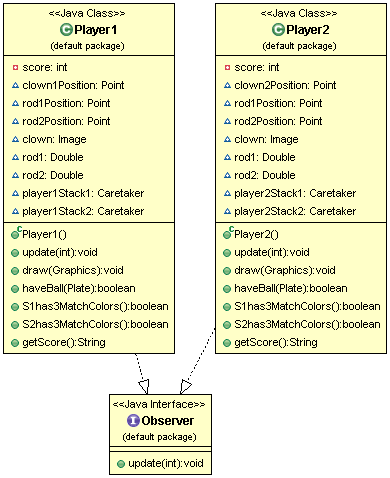
The ***iterator design pattern*** provides a ***common way*** to iterate through the data structure that presents in the program . it is well supported by java to maintain an iterator for any built-in data structure .  
and we use the built-in iterator in java .

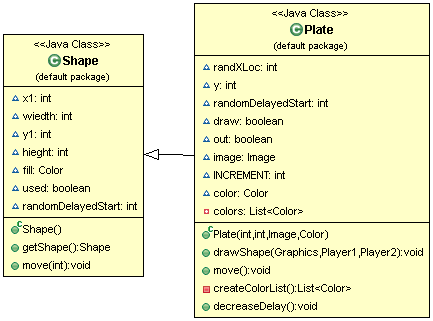
**Other patterns we use their concepts in the code but we wasn’t have the enough time to organize then carefully in the code , so that one class could use more than one design pattern ,and so on.**

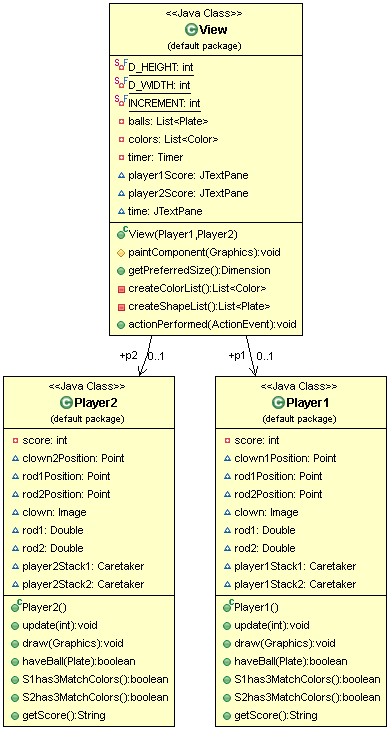
******

******

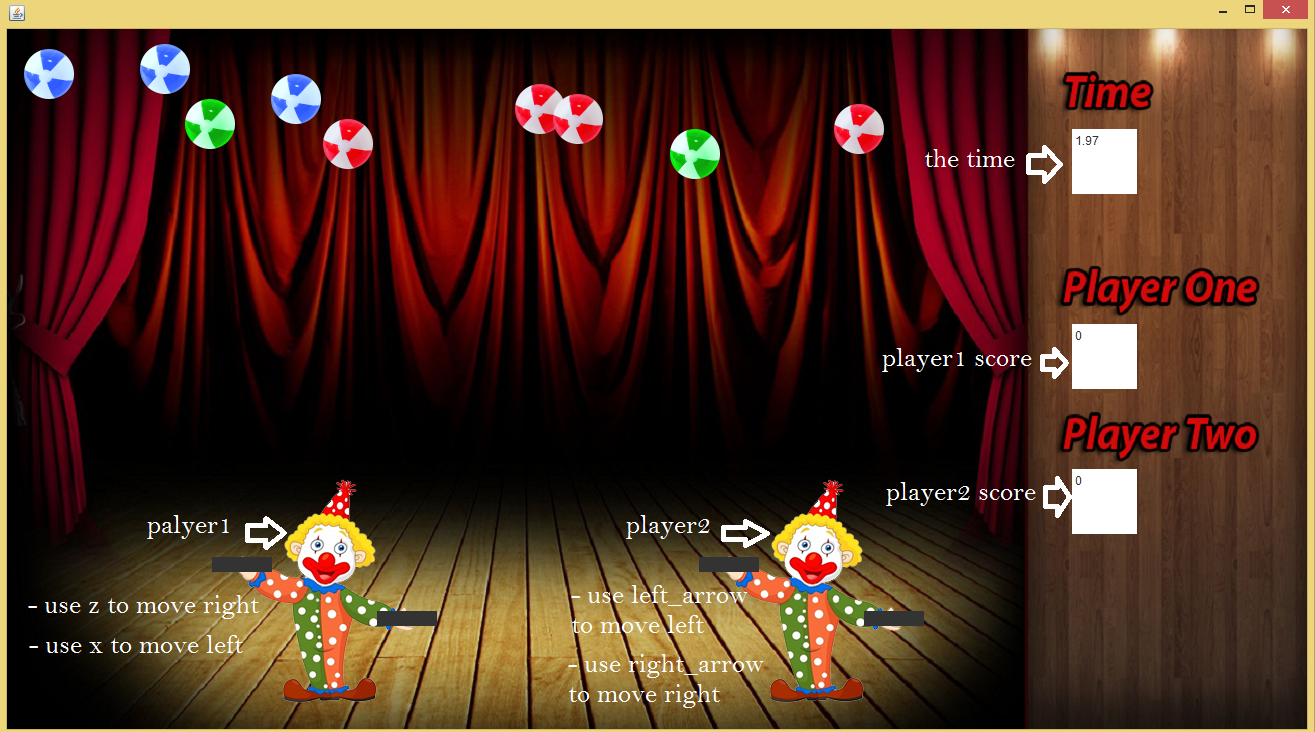
******

******

******

******

***User Guide***

****

1. First run the code from the Collector Class.
2. Player one can use the (x or mouse right-click ) to move right and (z or mouse left-click) to move left to move the Clown.
3. Player two should use the right and left arrows buttons to move the Clown.
4. A continues flow of balls will fall in the game plane.
5. If a player is in suitable range from the plate when it lands then the plate will be held by the Clown in the appropriate hand.
6. If a player gathers Three consecutive balls of the same color , these three balls will be popped and returned to the pool.
7. Press “S” on the keyboard to save and Press “L”on the keyboard.
8. The period of each game is one minute .
9. A player wins when the game ends and he/she has maximum score.