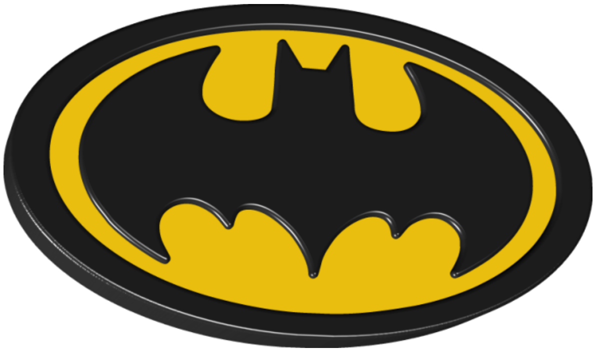
CCE 2020 – Programming II

Circus-Of-Plates (Batman version)



Prepared by:

- Ahmed Ibrahim Ali

Design Description:

Circus of plates is a GUI game based on JavaFX library with an intensive usage of different design patters to ease the development and extension anytime later.

The main design is based on MVC pattern to separate the data of the game from the controller and the user interface.

Other design patters are used as follows:

- Singleton is used to represent any system component and controllers that doesn't need more than one object during the runtime.
- Factory pattern is used with the generating the buttons of the main UI and creating new shapes from whatever loaded shapes during the runtime.
- Object Pool is used to limit the usage of resources represented at the falling shapes by setting a maximum limit for the used shapes and reusing the shapes after they vanish instead of creating new shapes.
- Iterator is used to iterate through all the GUI Components to build it at the start of the game.
- Dynamic Linkage is used for dynamically loading the shapes used in the game at run-time.

- Snapshot Design Pattern is used to save the game status
- State is used to differ the behavior of the character's stacks depending
on the current state of the stack.
- Strategy is used to distinguish between the two modes of the game,
will be clarified in the user guide section.
- Observer is used to notify the system with any change would lead to
changing the current game state (e.g. scoring a point or ending the game).

Log4j is used to provide a complete system logging.

User guide:

If you one of Superman fans, don't play this game

Unless you want to see the dark Knight collecting his logo in a crazy way ☺.

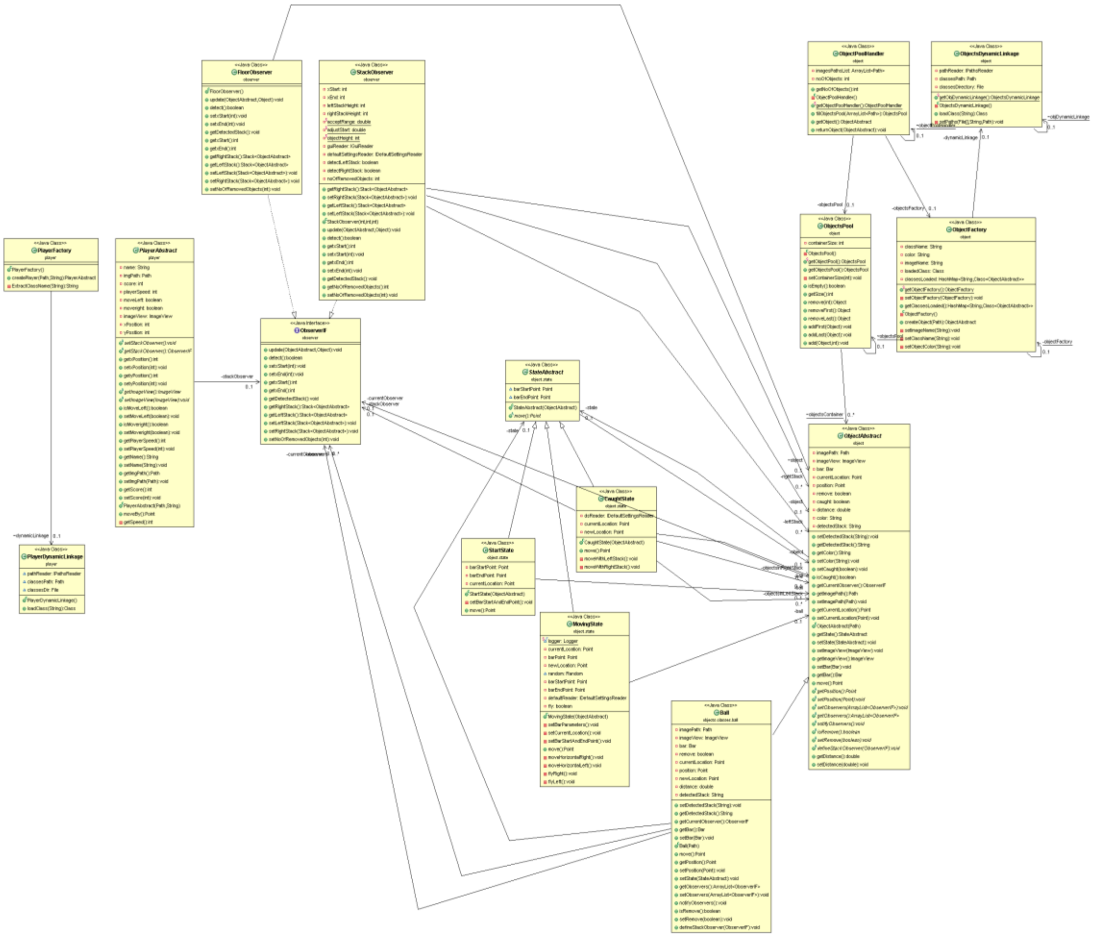
*** How to play:**

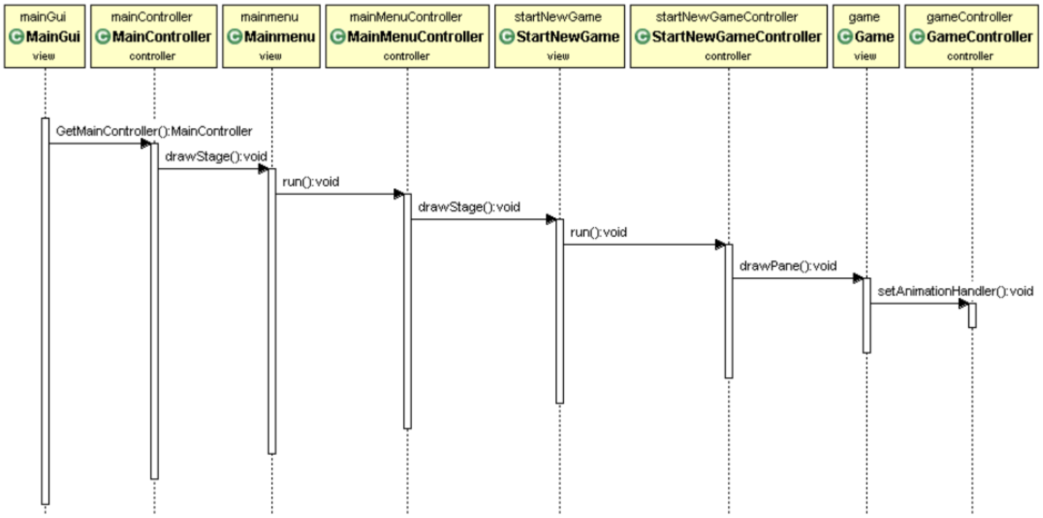
first, you need to choose your level in the game .. then you can control your Batman to collect his logo in his both hands ..
every once you collect three logos from the

same color, your score increasing one by one .. don't pause the game to any reason .. but you can close it when " Justice League " become in the cinema :D .

the more you collect, the more you love Batman ☺.

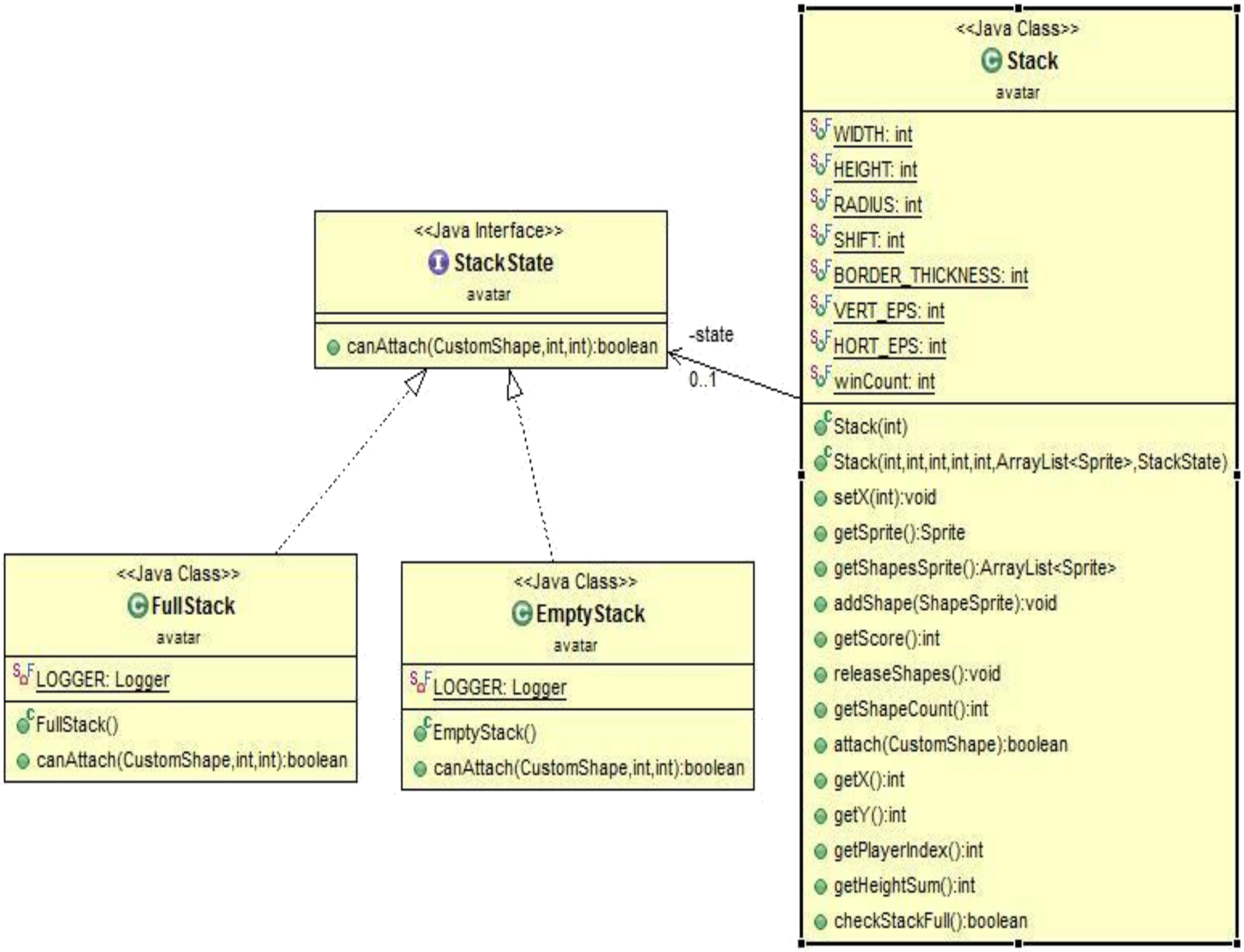
Class Diagram & Sequence Diagram:





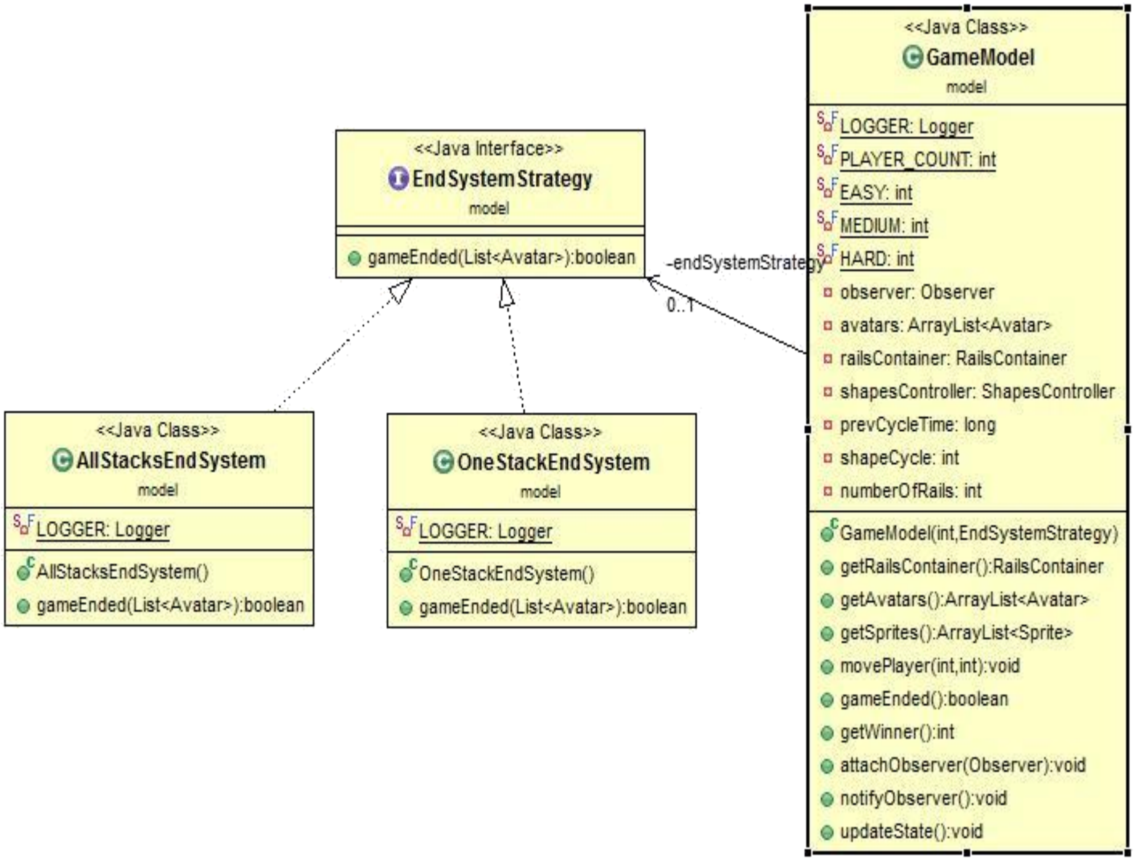
- **State Pattern**

State pattern is used to vary the behavior of the character’s stacks depending on the current state of it.



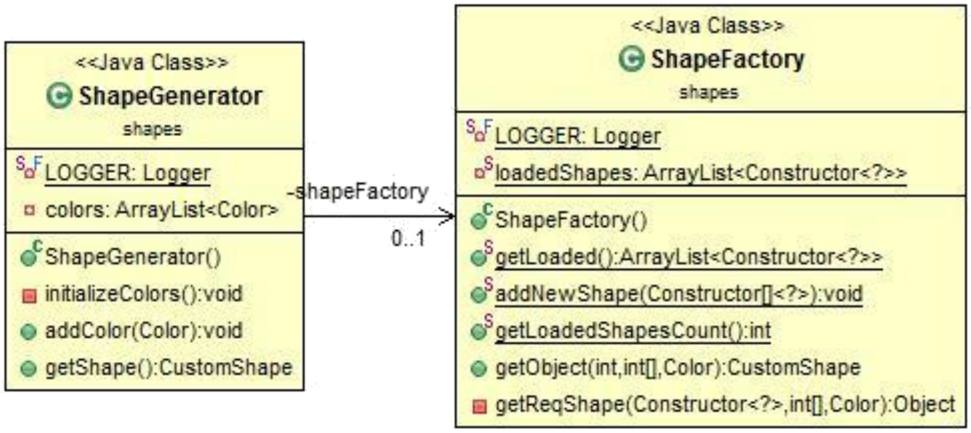
- **Strategy Pattern**

Strategy pattern is used to vary the behavior of the game in the two game modes, One-Stack-Full and All-Stack-Full.



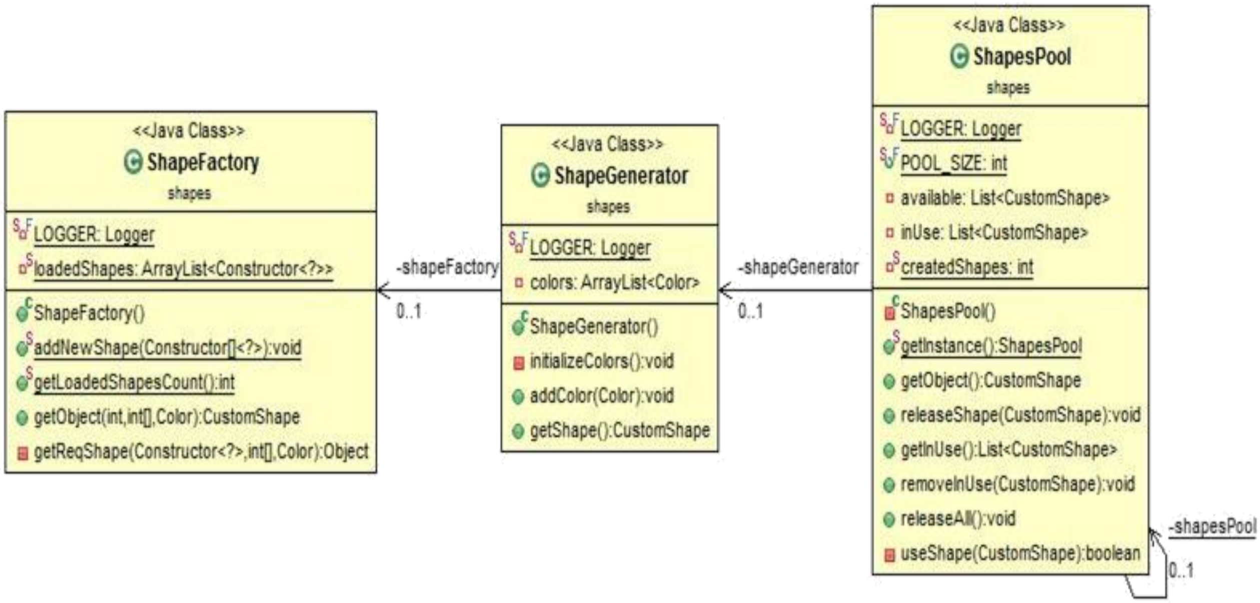
- **Factory Pattern**

Factory is used to generate the dynamically loaded falling shapes during the run-time.

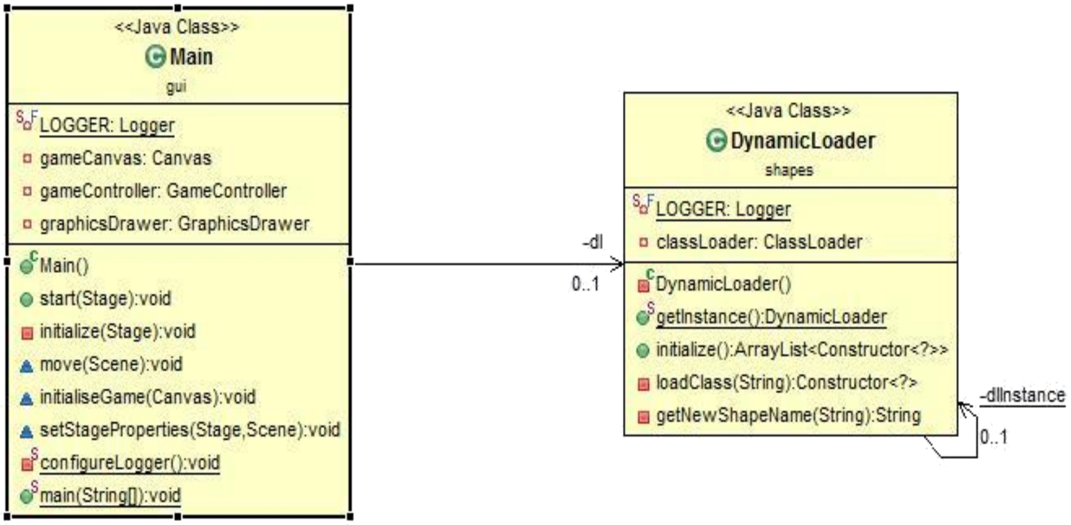


- **Object Pool Pattern**

Object pool is used to manage the creation and usage of the shapes to limit the usage of JavaFX objects since they're heavy objects.

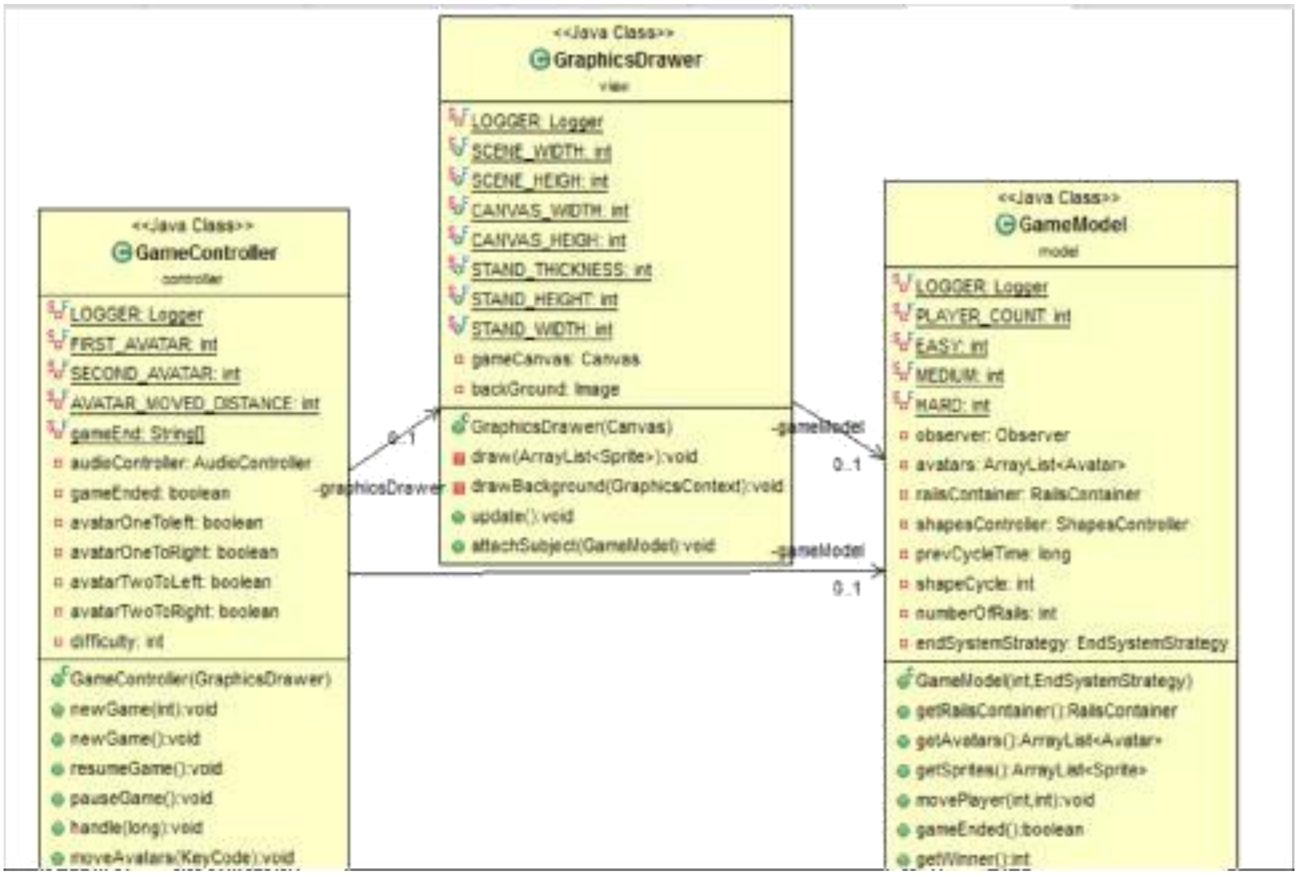


- **Dynamic Linkage Pattern**
- Dynamic linkage is used to dynamically load the falling shapes to be used in the game as long as they’re compatible with the defined interface.



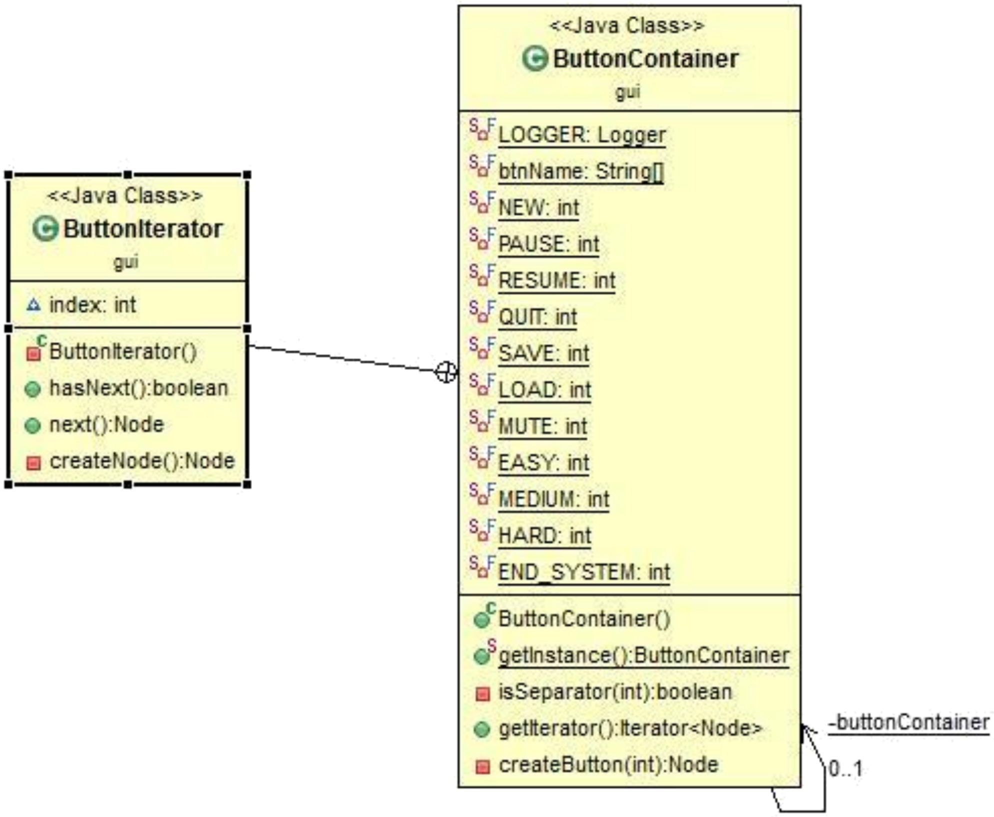
- **Observer Pattern**

Observer pattern is used to keep the GUI updated with the last results and game stats by notifying the GUI components whenever a change happens in the data to be displayed.



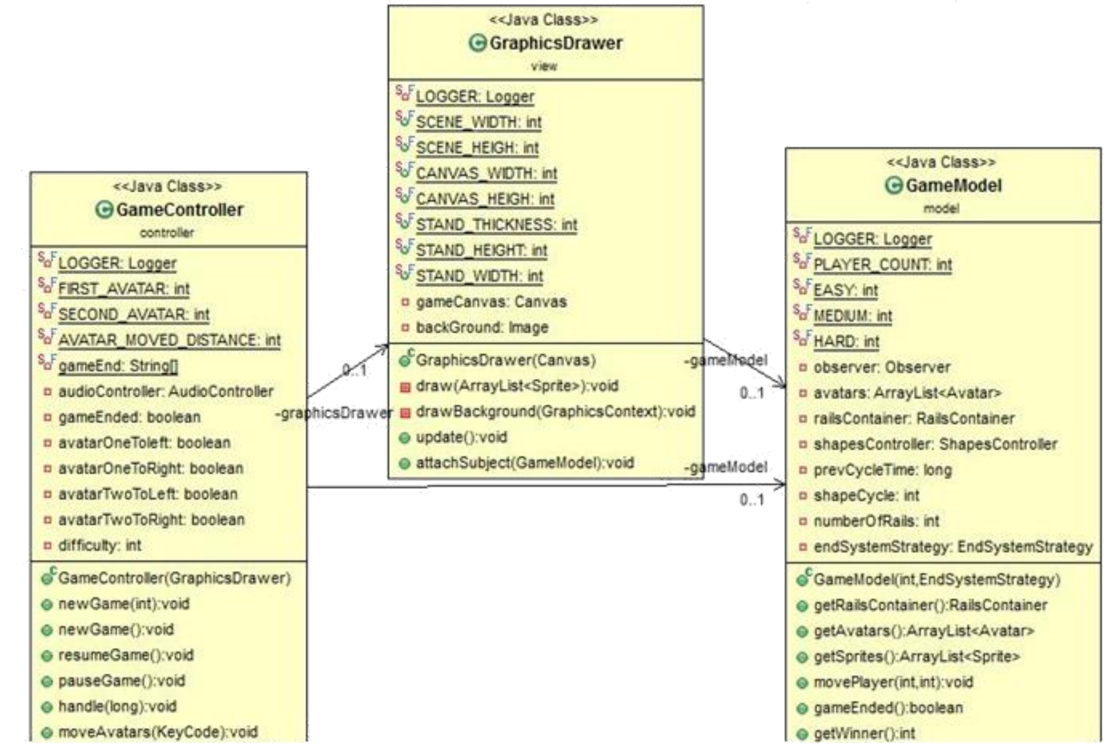
- **Iterator Design Pattern**

Iterator is used to iterate through all the buttons used in the GUI to set the user interface.



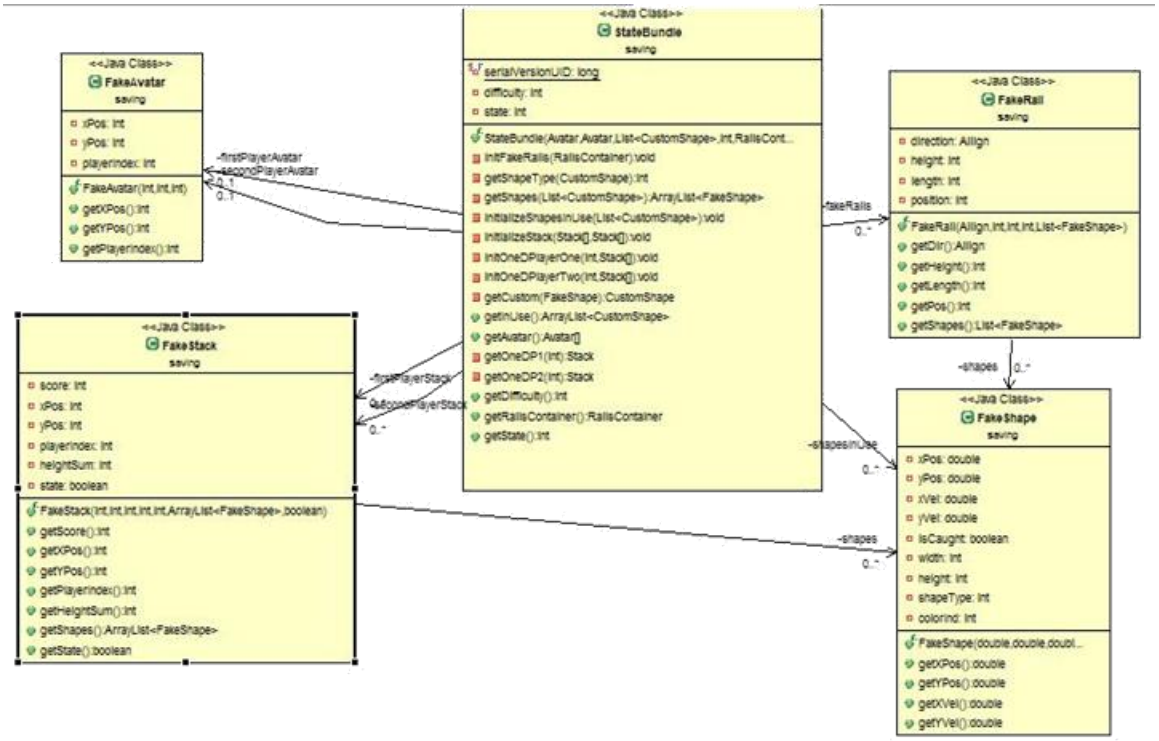
- **MVC design pattern**

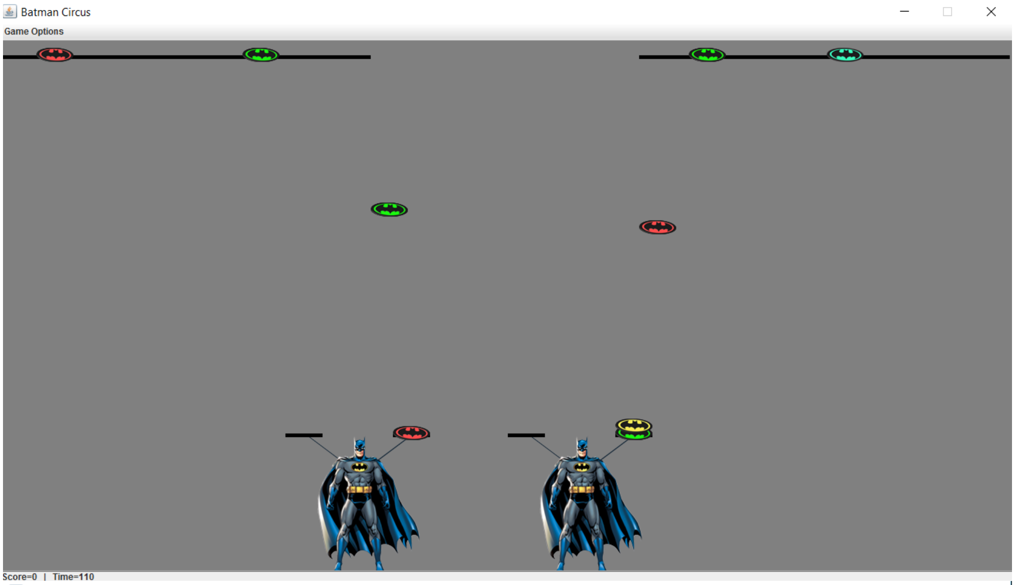
MVC is used to separate the three main components, model which holds the data of the game, view which holds the UI and controller which controls the game logic.

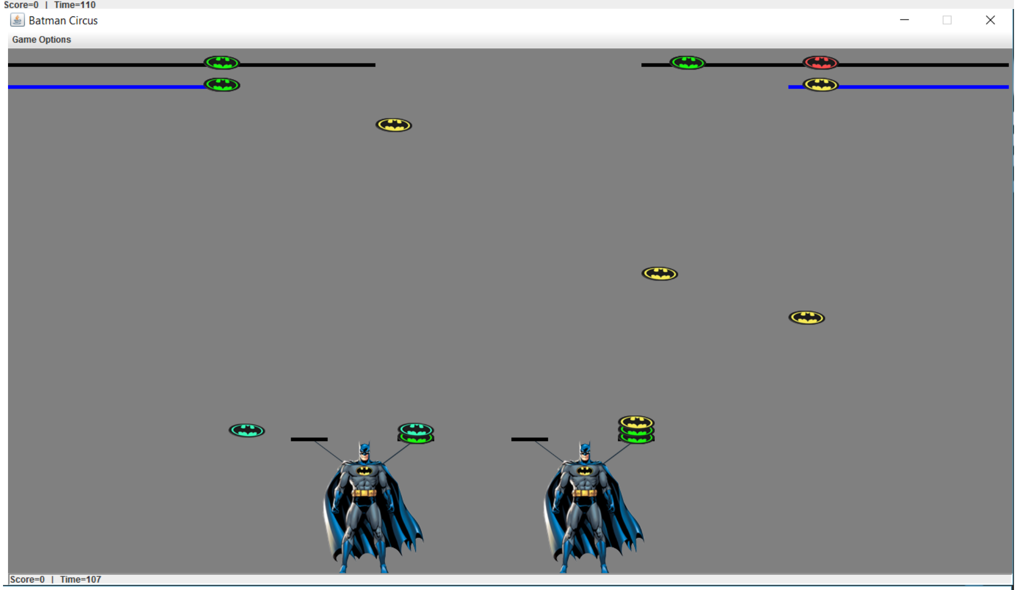


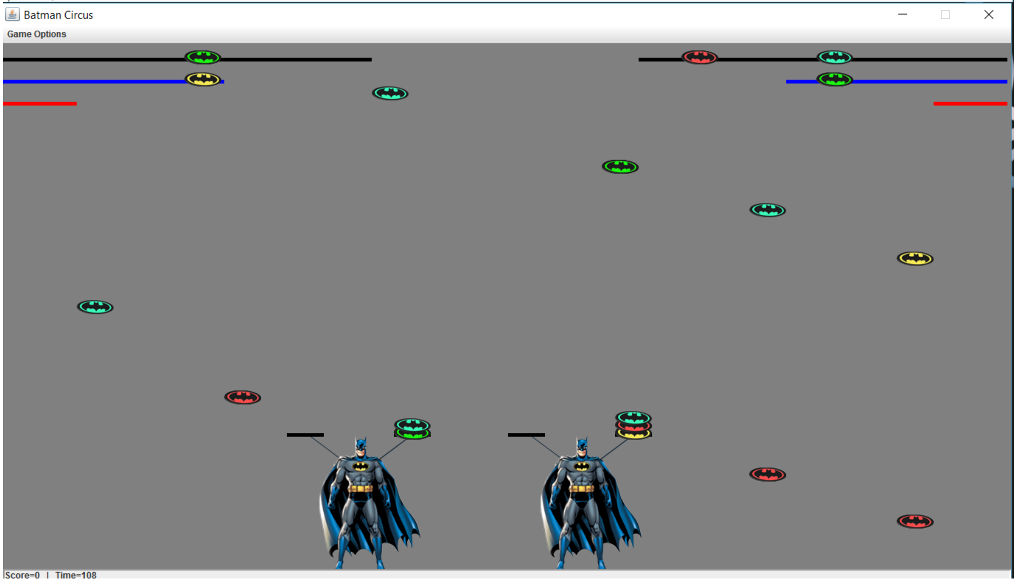
- **Snapshot Design Pattern**

Snapshot takes a moment (the moment of clicking the save button) and returns an object to be saved using JSON.




Batman Circus
Game Options
Score=0 | Time=110


Batman Circus
Game Options
Score=0 | Time=107


Batman Circus
Game Options
Score=0 | Time=108