**Circus of Plates – Game**

Team members:

* Mohamed Mohamed AbdElHakem Mohamed.
* Karim Atef Ahmed.
* Abdallah Mohamed Fawzy ElSaman.
* Yousef Mohamed Fathy.

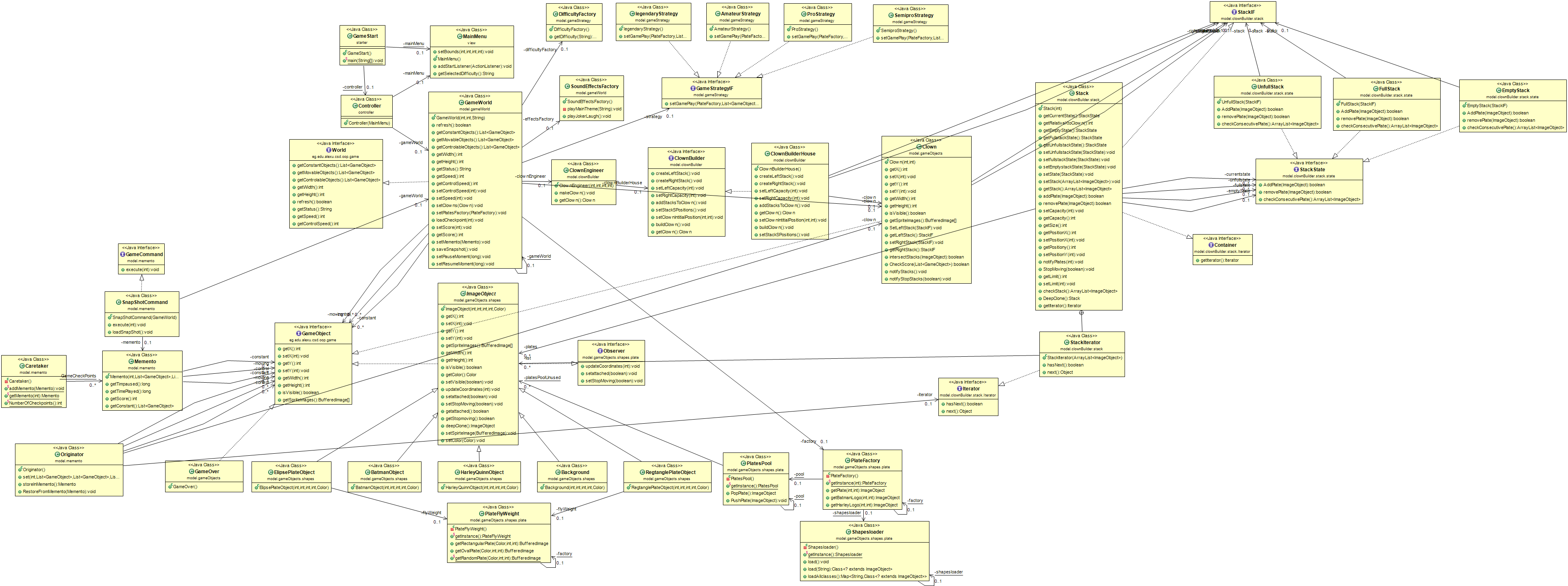
**Description**

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.

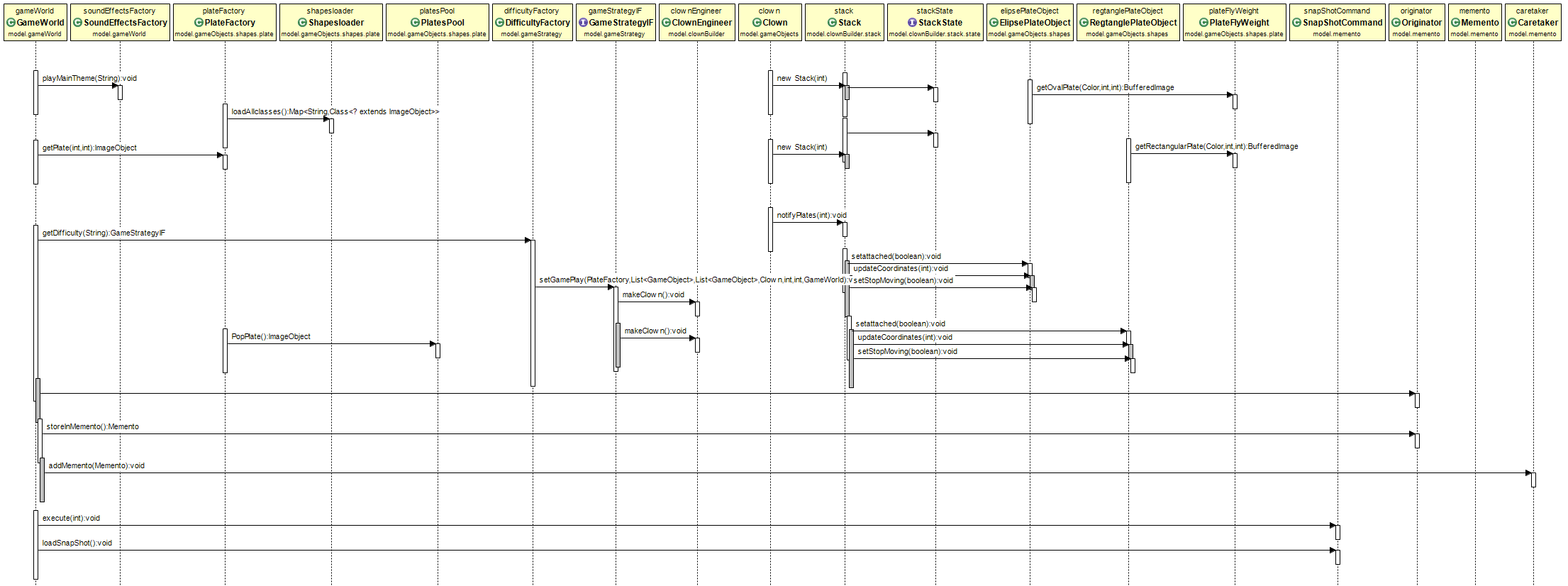
**User guide:**

* In the beginning of the game, the user should choose the difficulty of the game then click start.
* Once the user collects 3 consecutive plates of same color he gets one on his score.
* The batman object is a fighter object, it affects your score down, try to avoid it.
* Harley Quinn logo helps you, it saves a check point and increases the user’s score, it should be collected it.
* The user can save and load checkpoints in the game whether you saved it manually or if the user collected Harley Quinn gift.

**Class Diagram:**

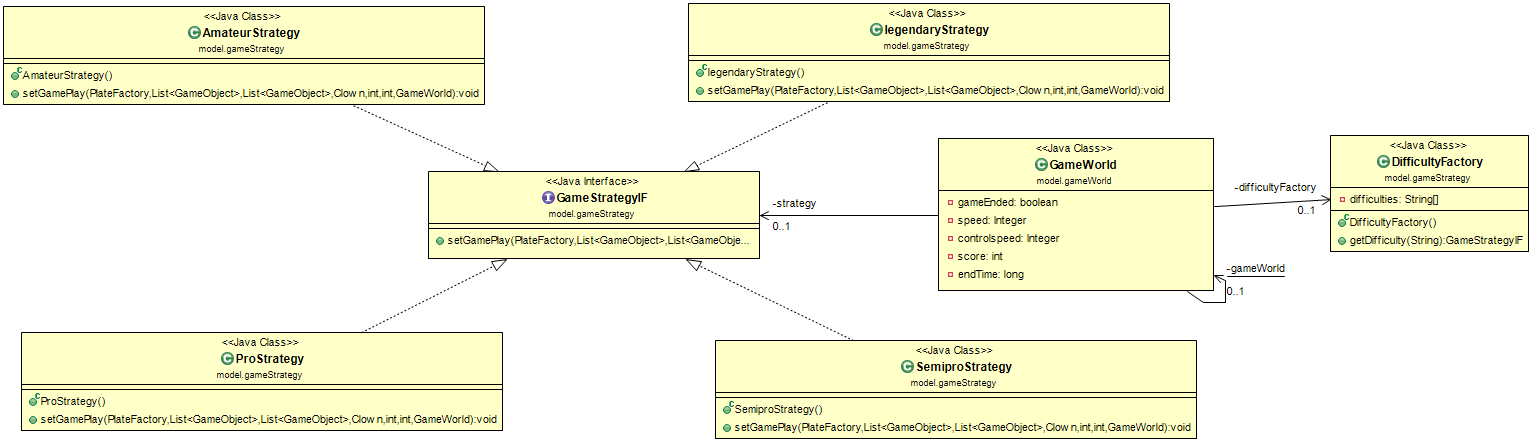


**Sequence Diagram:**



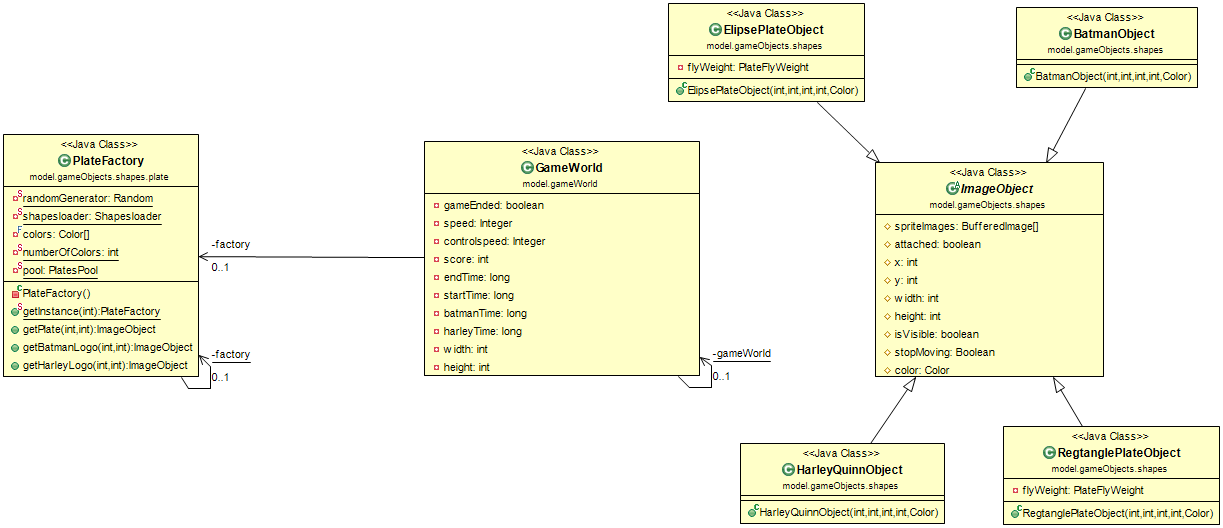
**Design Patterns**

* **Strategy**: the strategy design pattern is used in handling the difficulty of the game that the user chooses at the start of the game. The difficulty factory return the corresponding difficulty class as the user choose in the start menu of the game. The game strategy varies in lots of criteria like the game speed, the number of plates color falling and the stack capacity of the clown. All of them contributes together to make the gameplay unbeatable.

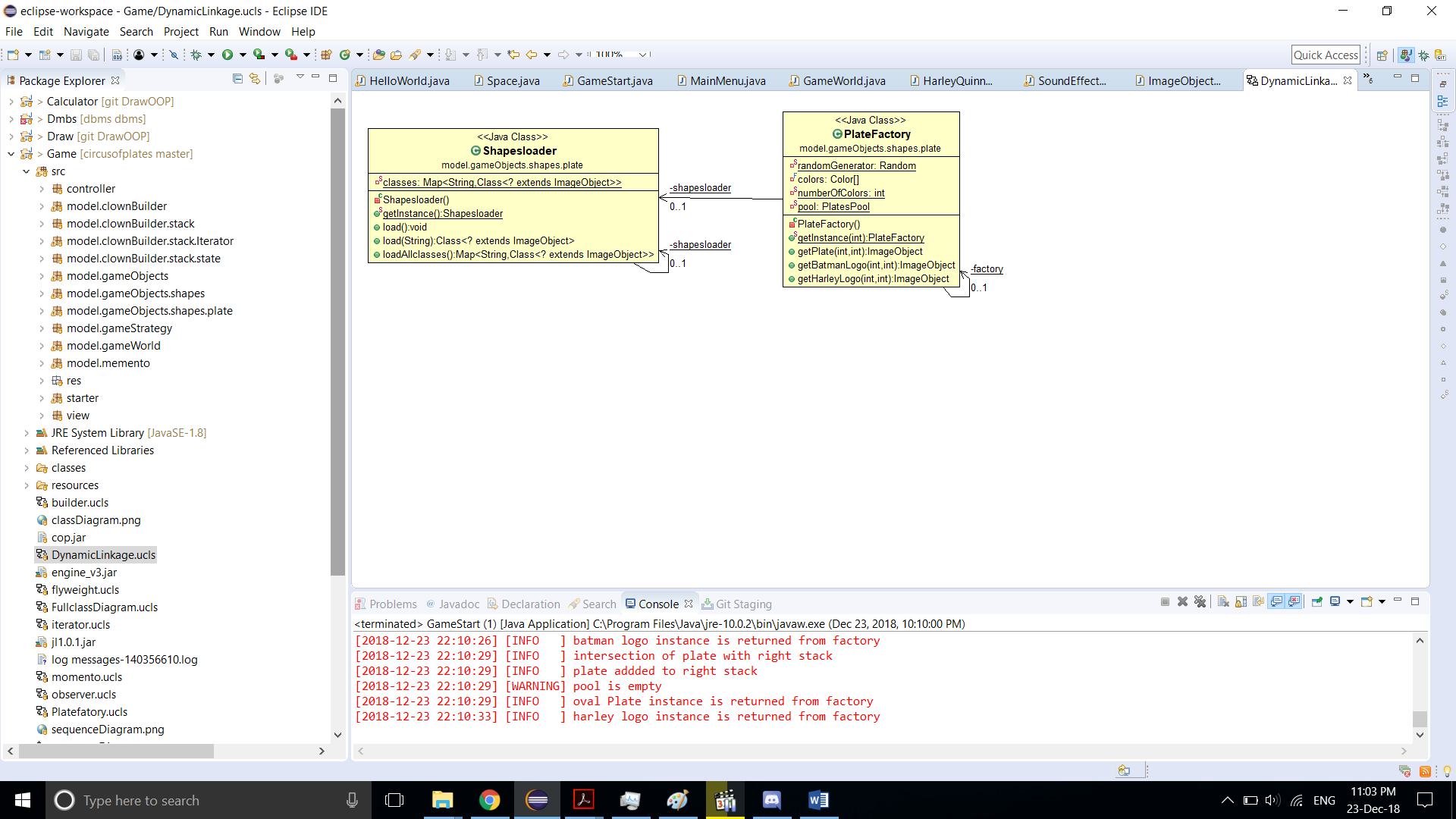


* **Factory:** the factory handles most of objects’ creation operations in the game as:

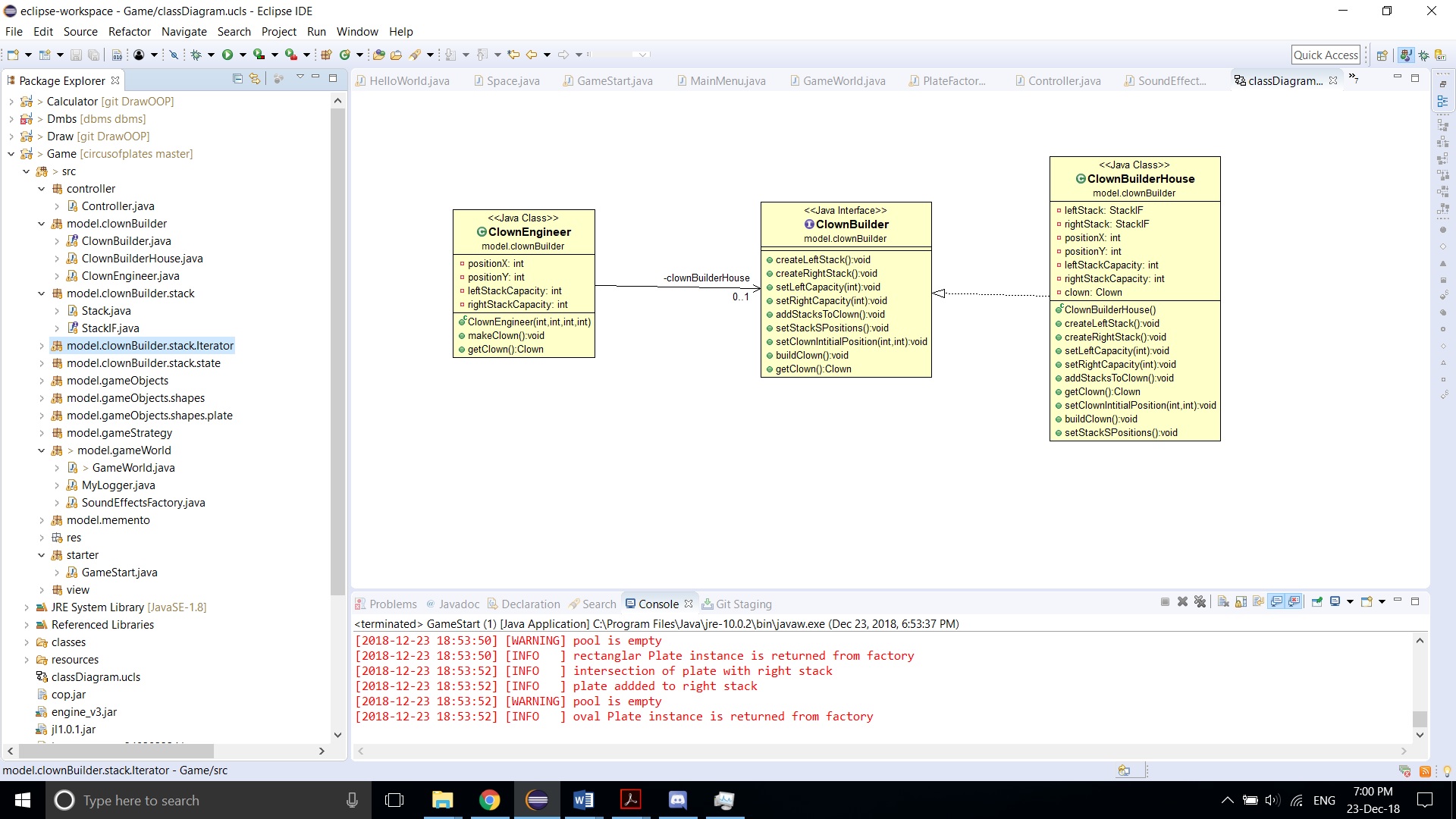
1. The plate factory class which creates the movable objects in the game like the oval plate, the rectangle, batman logo and Harley queen logo.
2. Difficulty factory as mentioned in the strategy pattern.



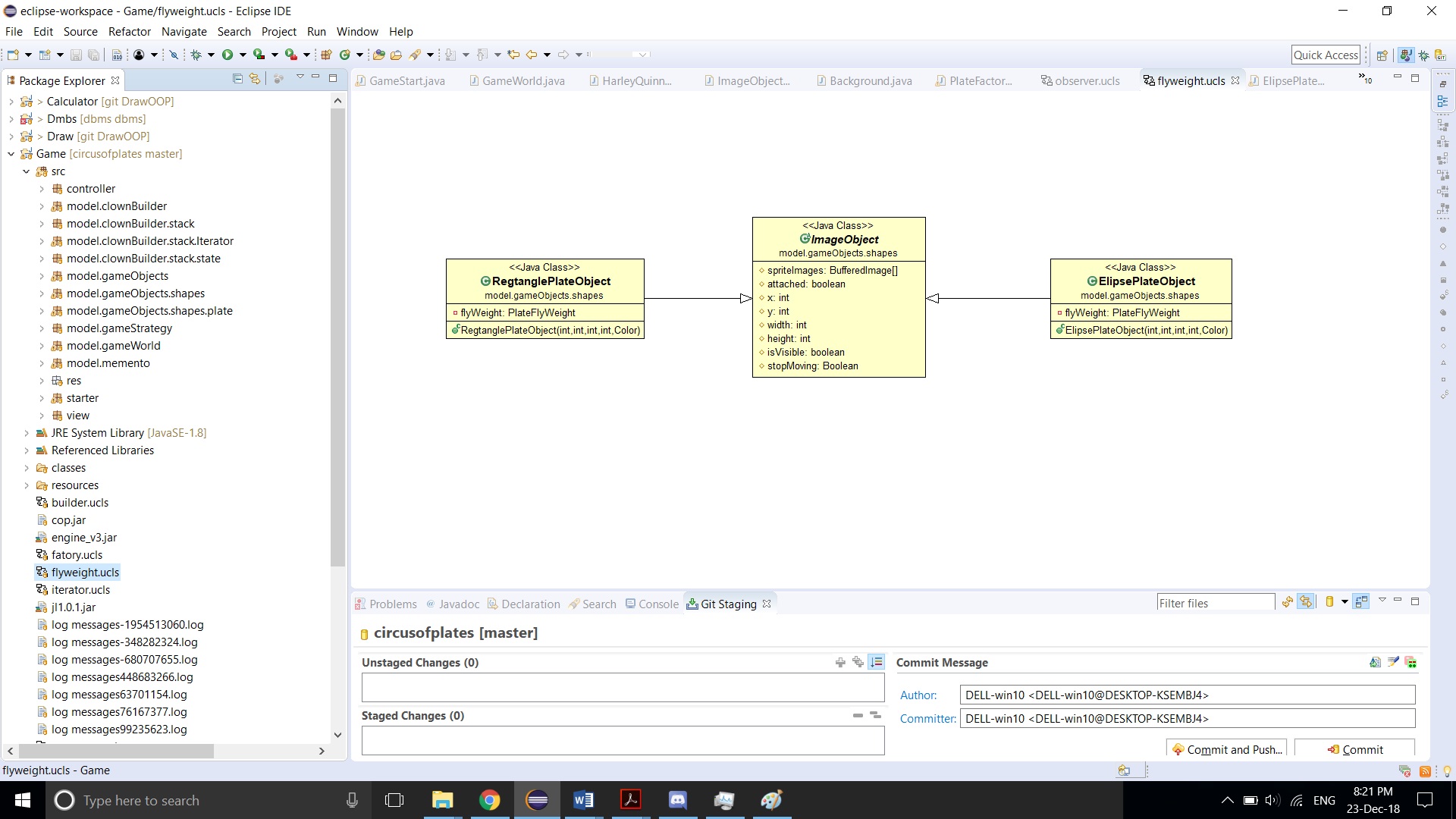
* **Dynamic Linkage:** the class shapes loader loads the shapes jar file at the start of the program for use of these 4 classes that extends image object images.



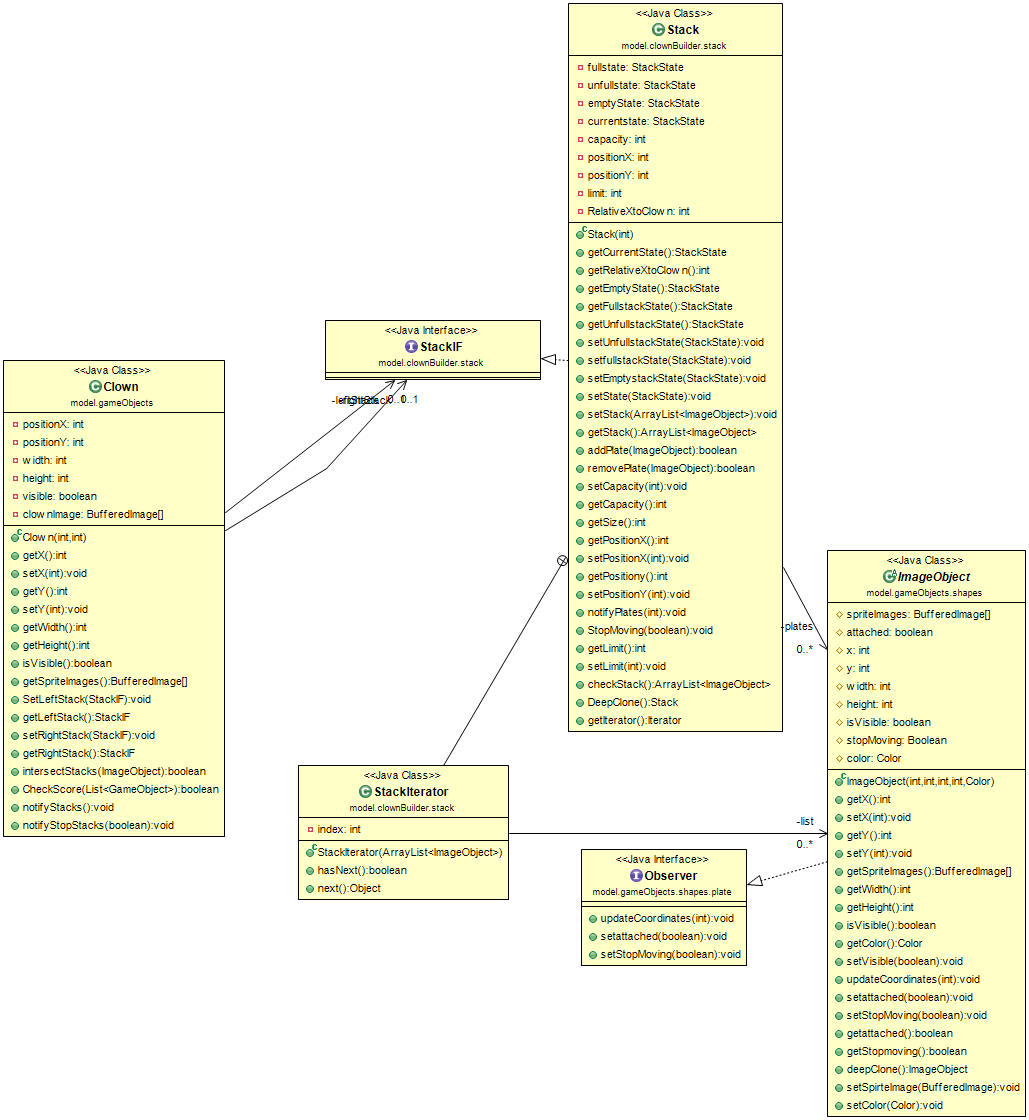
* **Builder:** builder design pattern is used in building the clown object which consists of left and right stacks used for collecting plates, and setting the clown position.



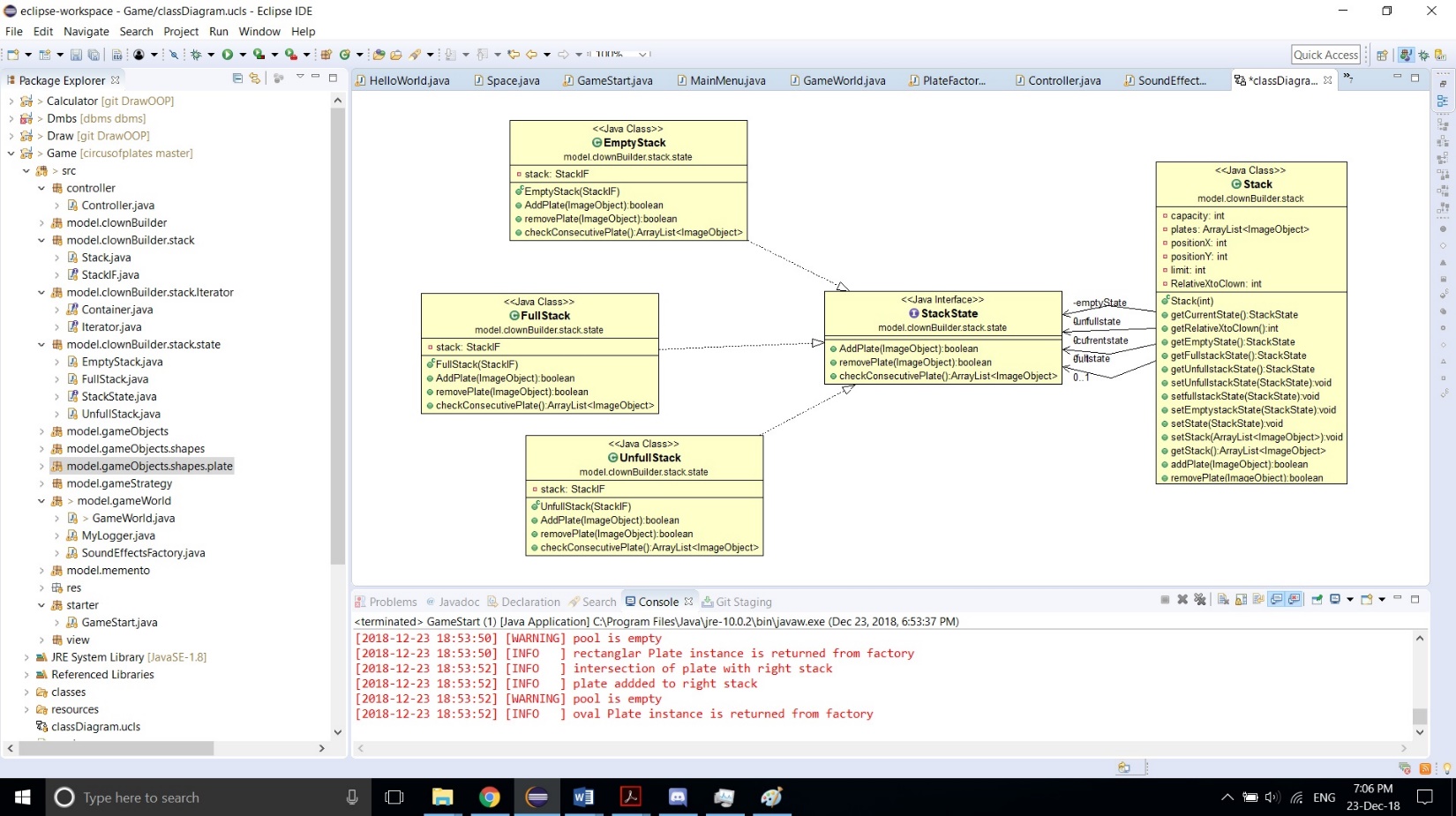
* **Flyweight:** used in sharing the graphic shapes between plates of the same shape, be it rectangle or ellipse plate.

****

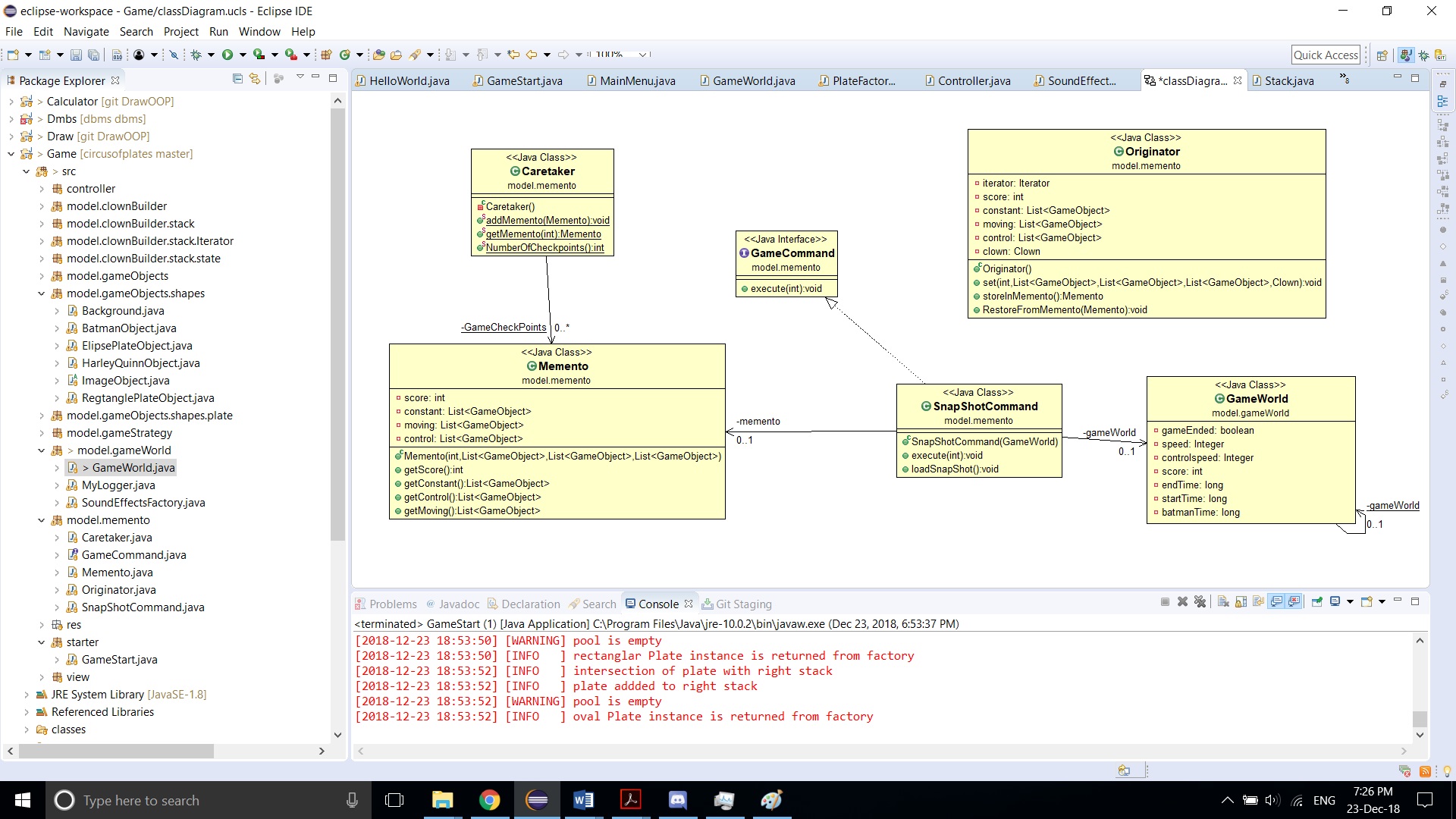
* **Observer:** used in updating the coordinates of the left and right stacks according to the clown, and to notify the stacks when to start moving and when to stop.



* **State:** used in defining the state of the stack whether it is empty of full or not full which controls the functions of the stack.

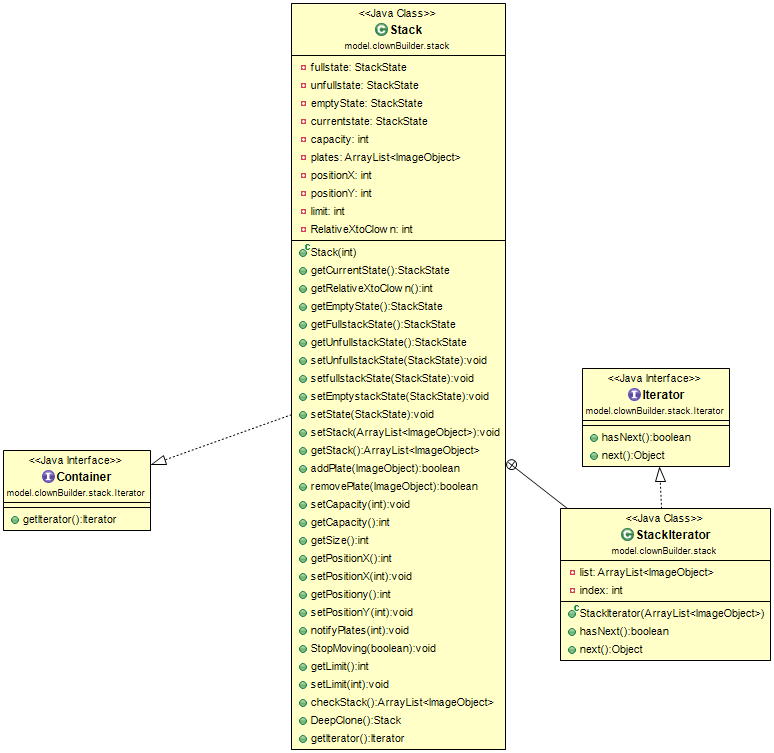
****

* **Memento:** used in taking a snapshot of the game by performing a deep copy of every property in the game, so we are able to create checkpoints and can navigate through them. Memento contains state of an object be restored. Originator creates and stores the states in Memento objects and Caretaker object is responsible for restoring object state from Memento.

****

* **Iterator:**

It iterates the stack elements in the stack and encapsulates the user from list of the plates.



* **Singleton:**

This pattern is used in the plate’s factory and plates pool and plate flyweight and the logger. All these classes are initialized once and only one instance is used during the program execution to get access to the data changed by all classes uses the singleton classes.

