Project 3: Circus of Plates Game

Introduction:

The circus of plates game 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 down and he tries to catch them, if he manage to collect three consecutive plates of the same color, then they are vanished and his score increases.

Design:

We have a package called res contains the images we uses and another called source packages that contains one for each design pattern and also objects , world and the main.

We import the game engine that supports three types of objects: movable plates, balls….etc. Constant the stand which the plates stand on and user-controlled object Clown.

We have two interfaces:

1. World: which is the Circus.

2. Game Objects: shapes, triangle, ball, rectangle, plates, plates, and clown.

## Class diagram of your design

## Sequence diagram showing the typical scenarios of the game

## Design patterns

**In this project we used..**

**Some** [**Creational design patterns**](https://sourcemaking.com/design_patterns/creational_patterns) **such as:**

**-**[Factory](https://sourcemaking.com/design_patterns/factory_method)

**-**[Singleton](https://sourcemaking.com/design_patterns/singleton)A class of which only a single instance can exist

**a** [**Structural design patterns**](https://sourcemaking.com/design_patterns/structural_patterns) **which is:**

**-**[Flyweight](https://sourcemaking.com/design_patterns/flyweight) a fine-grained instance used for efficient sharing

**-**Dynamic Linkage Allow a program, upon request, to load and use arbitrary classes that implement a known interface.

**Some** [**Behavioral design patterns**](https://sourcemaking.com/design_patterns/behavioral_patterns) **such as:**

**-**SnapshotCapture a snapshot of an object's state so that the object's state can be restored later. The object that initiates the capture or restoration of the state does not need to know anything about the state information. It only needs to know that the object whose state it is restoring or capturing implements a particular interface

**-**[Strategy](https://sourcemaking.com/design_patterns/strategy)   
encapsulates an algorithm inside a class

**-**[State](https://sourcemaking.com/design_patterns/state)  
Alter an object's behavior when its state changes

**-**[Iterator](https://sourcemaking.com/design_patterns/iterator)  
sequentially access the elements of a collection

**-** [Observer](https://sourcemaking.com/design_patterns/observer)   
A way of notifying change to a number of classes

.

## Sample Run

Score increase when collecting 3 similar shapes of same color



Score increase will collecting 3 consecutive non similar shapes but similar Color .