



# Circus of Plates

OOP – Lab #5

Names :

Amr Fathy Eldfrawy (48)

Mostafa Gabriel (66)

# Table of Contents

- 1. Class Diagram**
- 2. description of design**
- 3. Pattern Used**
- 4. Snapshots**

CIRCUS OF PLATES - [SELECT DATE]



# description of design and patterns Used

---

- The Game is based on a high level of Object Oriented Programming and use the following
  - Interfaces
  - Inheritance technique
  - Singleton Pattern
  - MVC Pattern
  - Observer Pattern
  - Object Pool Pattern
  - Dynamic linkage
  - Factory Pattern
  - snapshot Pattern
  
- Interfaces and Observer Pattern
  - Using the Observer Pattern help The Clown Object to notify the Shapes on his hand about his moves , so when a the Clown moves by +dx or – dx he used the Observer .Update to notify every Shape in his hand to change their positions

## ● Inheritance technique

- The Shape Class is the main geometric shape in the design and we used Rectangle Class as a subclass from Shape Class

## ● Singleton Pattern

- This is useful when exactly one object is Created from the One class
- The Controller Class (used in the MVC pattern) is good example for The Singleton Pattern

## ● Object Pool Pattern

- When there existed an object which is expensive to create and the initialization operation will be done multiple Times and expensive we use The Object Pool pattern.
- When you need a new object you request it from the pool. If a previously prepared object is available in it, the object is returned immediately to avoid the instantiation cost. If no objects are present in the pool, a new item is created and returned. When you have used the object and no longer need it, you return it to the pool, allowing it to be used again in the future without going through the slow instantiation process.
- In Our Game the Shape Class was the expensive Object .

## ● Dynamic linkage

- Amazing technique in the OOP help us to add new Objects and features to Our program , like adding new shape like (Circle ) Or (Oval ).
- In Game of Plates we write A FIXED PATH in the Controller Class this Path hold the class. Class files which inherited from the Shape class , by this way we can add new Shapes and object to the Game , so we can add new objects to the game by there binary Files Only

## ● Factory Pattern

- we used the Factory design pattern when we wanted to define the class of an object at runtime. It also allows us to encapsulate object creation so that we can keep all object creation code in one place

# User Guide

---

1. Player1 Control keys: left and right arrows
2. Player2 Control keys: 'S' & 'D'
3. Save: Ctrl + S
4. Load: Ctrl + L
5. Player x wins if his score reached 20
6. If stack has more than 20 plates the game ends in fair

# Snapshots

---







