# Circus of plates

Team Names:

- Abdelrahman Mahmoud Kamal Mahmoud Nour (39)
- Yousef Raouf Wadie Tadros (73)
- Ayman Emad Hussien Darwish (19)
- Abdelrahman Ibrahim Aly Hagrass (35)

# Game Description:

This is completely GUI game. The game consists of one player who can move using controls of (keyboard) and the player has two sticks and can collect the shapes which are detected.

When the player collects three shapes from the same color they will disappear and increases the score of the player.

The game finishes when the player shapes reaches the end of the frame of the green bar then the game ends.

# Design Assumptions:

We assumed that the game finishes if a player's plates reach the level of the green bars.

# Design patterns used:

## 1. Singleton:

Classes: “LevelFactory” , “logger” ,” platefactory” , “score” , “manageMoving”.

## 2. Factory:

Classes: “Platefactory” ,” level factory” .

## 3. Dynamic Linkage:

We use it to load any shape which have the same implementation of the Plate.

## 4 Snapshot:

Originator , memento , Gui: Caretaker

5. State: In class level factory “context class “where the state changes “ current level” from level 1, 2,3

According to the observed score

We use this design pattern there are three levels in our game. We consider the three classes of the difficulty as states which change during the game. the three classes (LevelOne - LevelTwo – LevelThree) all implement the same abstact Level. it represents various states and context object whose behavior varies as its state object changes.

### 6. Strategy:

"ManageMoving", "CheckCollisions", and "ManageCollisions" implement the interface "Strategy" , "Context" class organize the flow of operations,

And changes the Strategy to the class to perform the current operation according to the sequence.

### 7. Flyweight:

In "PlateFactory" Class: In generating plates of same type and color.

### 8. Observer:

Observers : "Managemoving" , "PlateFactory" , "LevelFactory"

Observable : "Score"

## Another patterns:

### 9-Command:

Classes: "Managemoving" ," Managecollision" and "CheckCollision"

### 10- Fascade: "Context"

## *MVC:*

The whole project depends on this design pattern. the player need to observe the shapes and detect their position to get them. and the frame detect the movement of the player and the shapes and detect the score of each play.

Model: Plates Objects and the player.

Controller: is the class context which manages controllong

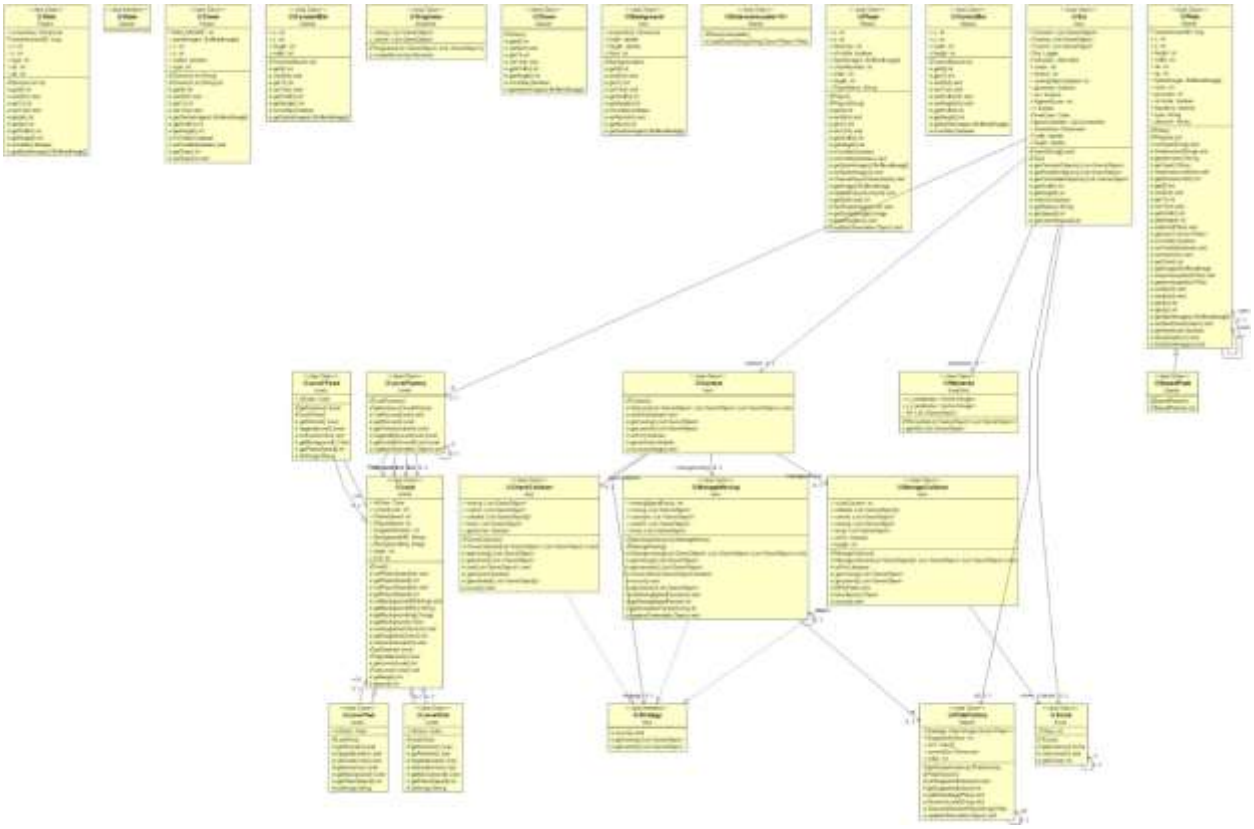
View: class GUI which displays the game.

# UML Diagrams:

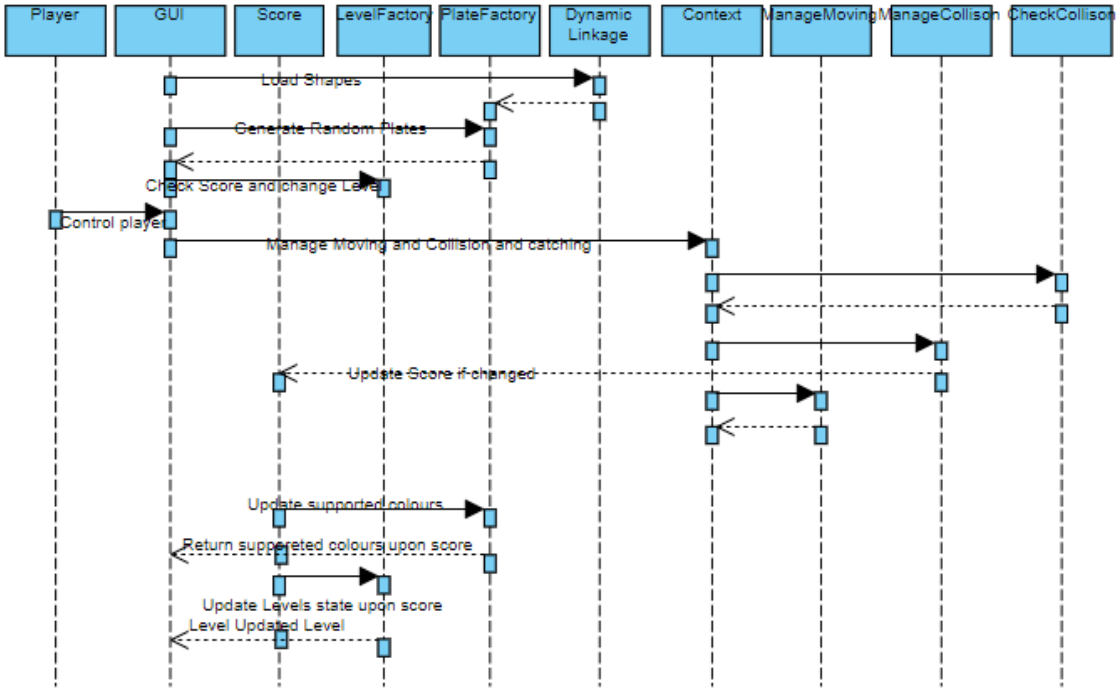
## Class Diagram

For better viewing and zoom check:

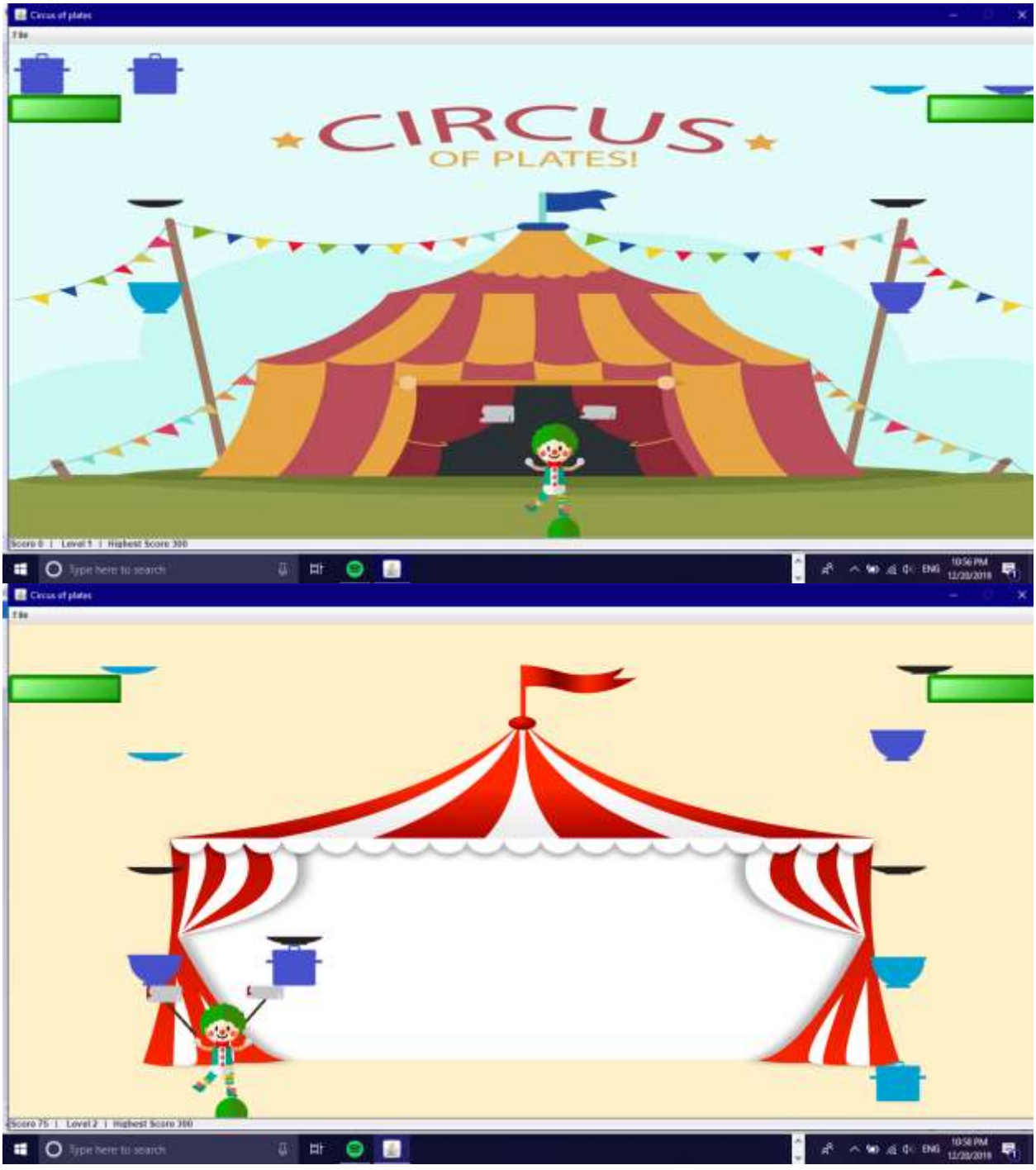
https://easyzoom.com/image/165829/album/0/4?mode=manage



# Sequence Diagram



## Snapshots:


Circus of plates
File
CIRCUS
OF PLATES!
Score 0 | Level 1 | Highest Score 300
Type here to search
ENG
10:56 PM
12/28/2019
Circus of plates
File
Score 75 | Level 2 | Highest Score 300
Type here to search
ENG
10:58 PM
12/28/2019


Circus of plates
File
Score 175 | Level 3 | Highest Score 300
Type here to search
ENG
11:56 PM
12/20/2019
Circus of plates
File
CIRCUS
OF PLATES!
Score 25 | Level 1 | Highest Score 300
Type here to search
11:51 PM
12/20/2019