

Circus of Plates – Game

Participants: -

1 - Omar Ahmed Fouad Hassan Wasfy.

2 - Sherif Mohamed Abd El-Rahman.

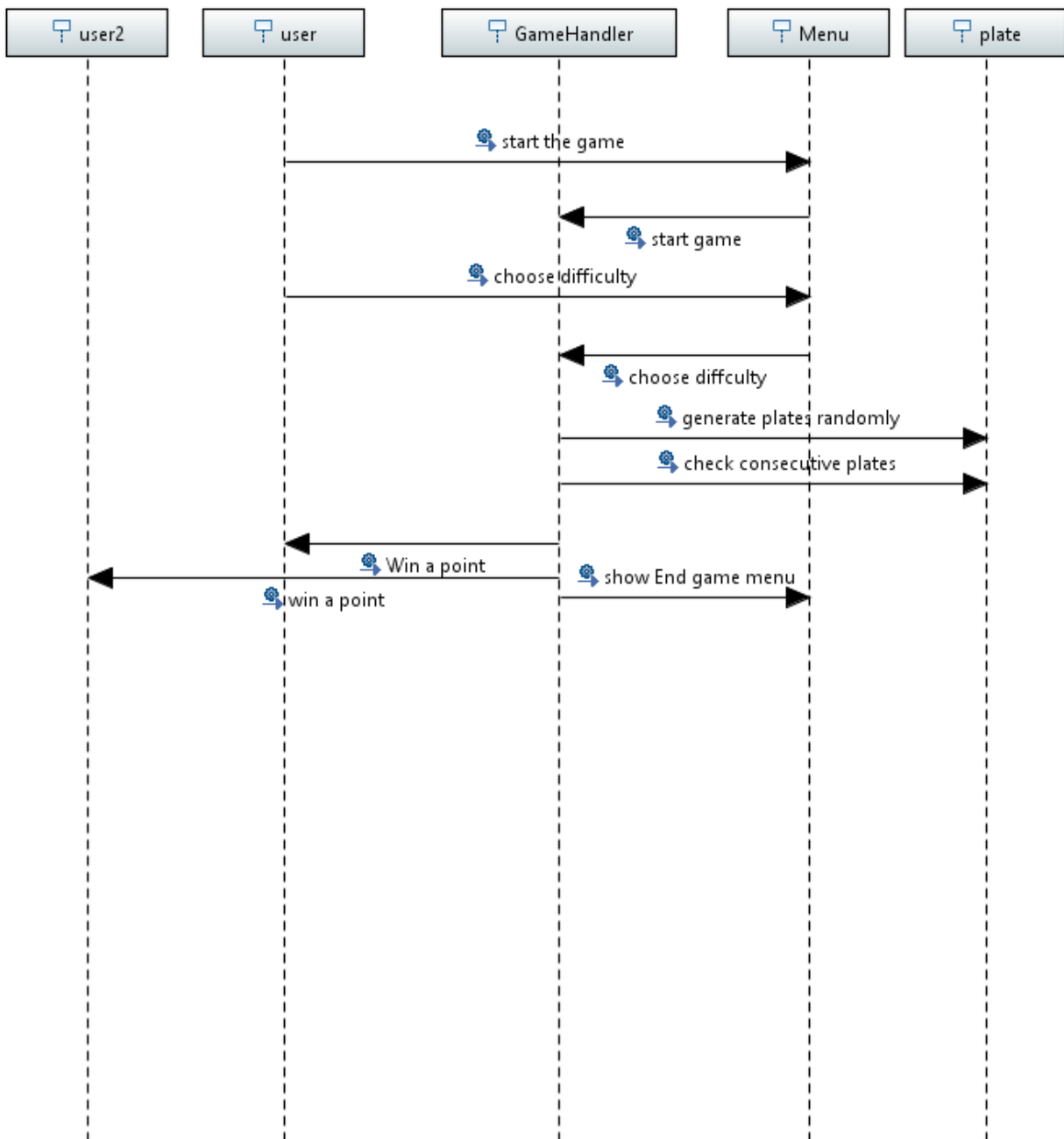
3 – Mostafa Tarek Ibrahim Mahmoud.

The description of design:-

We used most of the design patterns that are recommended in the assignment , at all our design is based on how to make more than one object make an action at the same time so that we have to use threads in our design and we used in our project an interfaces, classes, abstract classes and enums .

Class diagram:-

Sequence diagram



Design pattern:-

1 - At first we design an abstract factory which include two different object which represent the stone corner of this project, two different objects are clown and plates, At all we have two clown which use the same object clown and we have a lot of plates which have the same object but different in some its attributes like color,... etc.

2 – After that we have to make singleton design pattern on both clown that we create it only one but it is not available for the plates that we created it more than ones at the same time.

3 – We have to use state design pattern such that at the beginning of running game we will be at the menu then we will move to another state until reach to play.

4 – We have to use an iterator design pattern at looping for the game's object by rapid velocity in order to achieve our goal that for playing with high quality and high performance and you will find this at two method (Tick():void) and (Render():void).

5 – We have to use an observer design pattern at giving each clown its score because it will change dynamically that when

three object of the same color falls one over the other so that the score will increase by one so we need to notify observers that the score changes and at once all observers will be notified and update the right value of score for both of them.

6 – (Model – View – Controller) design pattern is considered as the big image of our work we have three package one for model , the other for view and the last for controller ,one main usage for that design pattern is to separate the logic of the game from the view

7 – We have to use Dynamic Linkage design pattern in case of loading an image to be an object of falling objects.

8 – We have to use factory design pattern when creating the clown.

9 – We have to use object pool design pattern to reduce the creation of object that you initialize the minimum and maximum size of their creations.

10 – we used strategy design pattern to differentiate between plates and clowns behavior (tick()method and render())

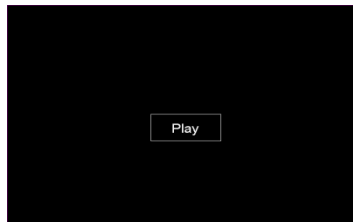
11 – we used snap shot design pattern to save two states menu state and game state and traverse between them

When needed.

User guide:-

When you run the play you should follow the instruction:

1 – Click on play button.



2 – Select the difficulty of the game.



3 – Now you have two players on the game.

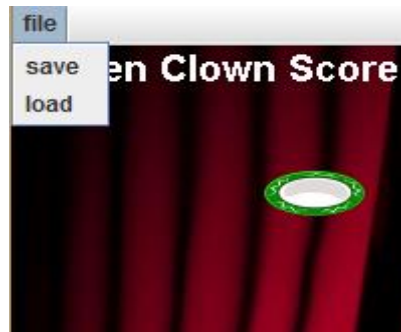


4 – First player uses mouse to move the clown right and left.

5 – Second player uses keyboard to move the clown right and left.

6 – When a player can carry three shapes with the color but sequential then they will removed from that hand and the score increase by one and so on.

7 – When you want to save the game at any location in your on computer, you have a menu at left corner of the screen then you can select from it save.

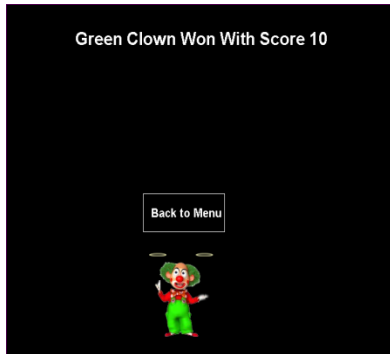


8 – When you want to load the game you saved before, you have a menu at left corner of the screen then you can select from it load.



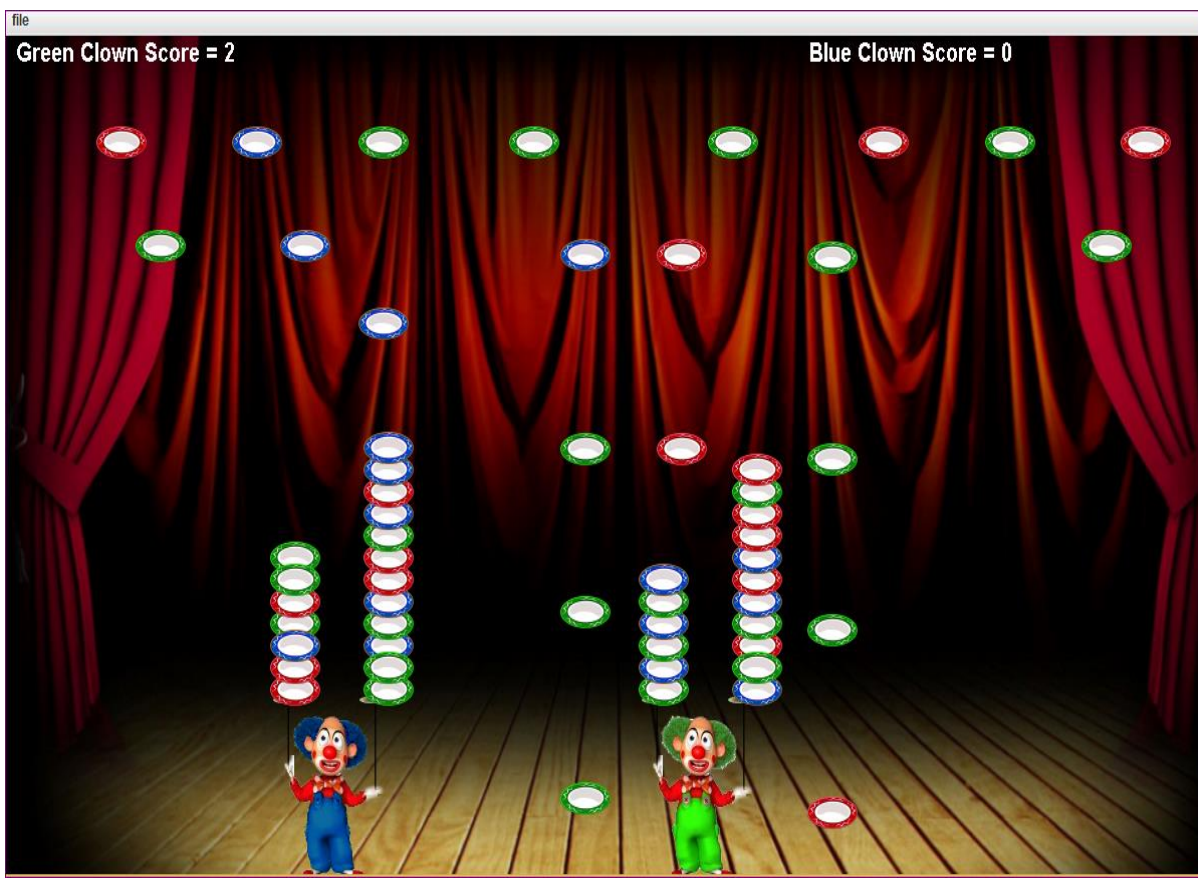
9 – When you want to pause the game click on space.

7 – When the number of the plates reaches its maximum then the maximum player's score will win.



8 – The game will end at once.

Snapshots of GUI



Design decisions:-

1 - We decided to use singleton design pattern

In clowns classes only

2 - We decided to use snapshot design pattern to save two states menu state and select difficulty