TEAM NO - 148

COLOR SWITCH

Ramit - 2019086

Satwik - 2019100

Design and Implementation

We have used

1. Factory Pattern Design Pattern because Object Creation was cumbersome and assigning it a helper function was very useful.
2. Facade Design Pattern which simplified the relationships between entities used by giving separate functions to specific classes.(Most Significant is Navigation)
3. Iterator Design Pattern wherever possible because it is so elegant and we often need to traverse over similar type of objects and do similar operation.
4. Decorator Design Pattern because we needed to wrap around the ObjectInputStream and ObjectOutputStream around FileOutputStream & FileInputStream else we would had to make an unnecessary Object.
5. State Design Pattern (Game_State) which is very elegant for storing the Game_Screen state as it can have different states and each state has different behaviour.

Problems Faced

1. Classes present in JavaFX are not serializable, and hence we cannot serialised any of the objects of those classes ,
   So we made a serializable class which stores the Game state information and we recreated the game screen from the game state whenever needed.
2. It was very difficult to create a Multi-User environment where any number of player can enter and load or save his games. The main part was saving a player and all its games in a single file as the header of the file was being inserted everytime we insert a new player in the file.
3. Adding random objects and moving all the on-screen obstacle was also a tough job but both gave large amount of time understanding how JavaFx works and implemented it correctly.
4. We did a lot of research before using anything in JavaFx , because it was very important to know how it was implemented in the backend so that we can add our own features to it.
5. Making the animations smooth for every entity used was tough and needed a good knowledge of how the timer was working and in which frame of the second what we have to do.

Contribution

Equal amount of contribution is made by both of us. Which can be easily seen by seeing the contributions make by both.

Total Contributions - 156 commits
Ramit's Contribution - 53 commits
Satwik's Contribution - 75 commits

Ramit-2019086

1. Implemented Collisions of all the Objects present in the game.
2. Added Hover Effect over Buttons.
3. Serializable and Deserializable of the game state.
4. Implemented Revive for the Player
5. Implemented Multi-User environment
6. Implemented Navigation which helped in navigation of screens

Satwik Tiwari -2019100

1. Created FXML Pages and GUI designs for all Screens such as Login menu and Player menu.
2. Saved game state for serialization
3. Implemented game physics like falling and gravity feature of the ball
4. Added sound effects
5. Implemented components like color switch, stars etc.
6. Added random obstacles on GameScreen

BONUS

01 We have added sounds of Color Switch which make the game to be more fun.

02 We had added lots of animations which are so fluid, there are no jerks whatsoever at any point of the game.

03 We have also added Multi-User Interface. Which means many user can play the game and save the state of infinite number of games that they want to.
I.e We multiple player can login with unique username and then store any number of games which they want.

04 Added Hover Effects and Sounds on Buttons.

05 Added Twinkling Effects on the Stars