Color Switch

CSE201: Advanced Programming

Utkarsh Dubey 2019213
Robin Garg 2019092

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY **DELHI**

Implementation

1. FXML has been heavily used, for all obstacles, pages and navigation has been done using FXML only.
2. Saved game menu is used to load the instances of available saved games.
3. Mainpage menu is the home of the game and controller of mainpage acts as a manager for the new game started.
4. Pause menu and game over menu are also present to do the job as there names suggest.
5. Game can be saved using save game button in pause menu.
6. Option to continue using all collected stars is available in gameover menu.
7. All the gameplay is based on how the real color switch works.
8. Difficulty of the game increases as the game proceeds.

Design

Design patterns:

1. Singleton: We are making only one instance of the Database, as this single instance can be used where ever needed.
2. Factory: While generating random obstacles we have made use of the logic that all obstacles are inherited by a single class, so we can operate on that too generate random obstacles.
3. Iterator: For synchronised iterations of all the obstacles and color switchers in the list we have used this design pattern.

FXML made via - SceneBuilder

Image sources - various sites

Individual Contributions

Utkarsh	Robin
● FXML - cross2, ninjastar, doublecross, NewMainPage, PauseMenu, pauseSymbol, savedgames, gameOver ● UML - 50% of both Use case diagram and Class diagram ● Working of all buttons. ● Checked collision, star collection and working of color switcher ● Added sound effects ● Made classes and attributes	● FXML - ball,circle, concircle, cross, colorswitcher, dottedcircle, ● UML - 50% of both Use case diagram and Class diagram ● Serialisation Deserialisation ● Transitions for balljump and many obstacles. ● Working of pause and resume ● Working of continue option ● Made classes and attributes

All individual contributions can be seen on the github repo of the project. Link

Bonus Features

1. Added sound effects.
2. More stars can be earned by visiting a website.
3. Added extra obstacles than needed.
4. Reverse mode.
5. Alert windows.

Difficulty we faced and how we solved then.

- Linking all the FXMLs and working around them was a big problem, we solved this by reading a lot about FXML and it’s translation into code.
- Most of the classes of Javafx are not serialisable, hence in order to serialise the data of all those objects, extra classes were made and all the important data was dealt with separately.
- Handling exceptions was a big task and seperate care was needed.
- Pausing, resuming and continuing were some challenges which we had to face.