

# END TERM PROJECT

Course: CSE201- Advanced Programming

Project Details: End Term Project

Create a clone of the famous arcade game 'Color-Switch' using java, javafx and essential principles of object oriented programming.

Team:

Abhishek Saini 2019007

Raghav Nakra 2019083

# COLOR SWITCH

# IMPLEMENTATION & FEATURES

**Navigation:** navigation of on-screen components done using FXML.

**Saving/Loading a game:** every time a user saves a game, the game state is added in the database along with the count of stars earned, which is then serialized using I/O streams.

**Delete:** game state is automatically removed in which the user loses the game and is unable to revive.

**Difficulty:** levels become harder as the user progresses through the game. Rotation rates of obstacles start increasing along with consistent change in their sizes.

**Revive:** The user can revive if he/she has enough stars (10 stars per revival) and feels like continuing or start afresh otherwise.

Difficulty starts increasing after 10 stars are earned in a game.

Colors available are red, yellow, blue, purple.

Obstacles available are square, circle, triangle and cross.

User can unlock and choose between different shapes of the ball.

User can save multiple game states and reload or restart them at any time.

Cool background music.

# INDIVIDUAL CONTRIBUTIONS

## Abhishek Saini

Game Play  
Pause Menu  
Load Menu  
Interaction between ball and other elements  
The game loop  
Increase of difficulty  
Enabling and stopping animation  
Serialization/Deserialization  
Help Menu

## Raghav Nakra

Main Page & Leaderboard  
Load game Options  
Transition of elements  
Buttons and Background  
Initializing classes and their attributes  
Randomness of obstacles  
Background Music  
Saved game state  
Different shapes of ball

# DESIGN PATTERNS & BONUS FEATURES

## Design Patterns

Iterator - Ordered accessing of lists when required.

Template - Used in interface and in-game interactions.

Facade - Distribution of tasks between entities.

Decorator - Scanner and File streams used to save the game.

Composite - Current game holds object of its own type.

Observer - Event handler and Key listeners as observers.

Chain of responsibility - Event handlers and key listeners use chain of responsibility internally to function.

State - Game is implemented using various states such as collision, state saving, revival.

Threading - All the in-game animations are done using AnimationTimers each being an individual thread in JVM

## Bonus Features

Item Shop - The user can unlock exciting shapes of the ball by redeeming stars collected during the game.

Scoreboard - The user can see top 5 highest scores ever reached in the game.

Help Menu - Users that are new to the game are provided with a guide to start off good.

Changing Size - Obstacles will be changing sizes every second to make the game more interesting and difficult.