A game replica made by

Divyansh Rastogi & Rupanshu Yadav

Source Code

Team 60 – 2019464, 2019475

Design & Implementation / Problems Faced

COLOR SWITCH

- Implemented MVC (Model-View-Controller) architecture for user interaction & data transfer across static screens.
- Used canvas & graphics context for dynamic gameplay and efficient performance, utilizing concepts of linear algebra & physics for movement, rendering and collision checks.
- The project is mainly divided into three components – *gameEngine, gui & data*.
- The role of “Model” is enacted by *gameEngine,* which studies the game’s current state and updates its own state and *data* accordingly.
- Various design patterns such as *template, factory, façade, etc.* are implemented in *gameEngine* with generic & extensive use of OOPS concepts.
- The role of “Controllers” and “View” is given to *gui*, where the screen layout is loaded via FXML files and intuitive user interaction is achieved via corresponding controllers.
- We faced problem in order to how to generate game elements on basis of difficulty in efficient way, we solved it using a obstacle factory comprising of switch statements on basis of score to return a generic abstract obstacle.
- We faced problems regarding exclusivity of various interactable such as star, colour switch & obstacle which we later choose to represent in a template pattern for game elements.
- We also implemented an observer pattern as to change our game’s state based on event handling.

COLOR SWITCH

Individual Contributions

Divyansh (2019464)

- App framework & design
- FXML designs
- Database handling
- Gameplay synchronization
- Player as information keep
- Controllers communication b/w gameplay & pause
- Screen transitions and info. transfer
- Controllers for leader board, game over, input popups & load game
- Revival options, easter egg implementation
- Serialization

Rupanshu (2019475)

- App framework & design
- Rendering mechanisms with canvas & graphics context
- Gameplay loop
- Game mechanics using NLM
- Spawning and refreshing game elements
- Obstacle, star, colour switcher & high score line designs
- Ball physics
- Ball explosion mechanics (Swarm)
- Bubbles mode physics and overlay
- Obstacle Factory

Bonus

COLOR SWITCH

- Implemented a new game mode, **Bubbles**, utilising an immersive and interactive overlay containing **semi-transparent bubbles** offering a challenging gameplay. Achieved via **clipping and restoration** of graphics context.
- Created an obstacle factory responsible for delivering a gameplay of over **25+ different obstacles** utilizing complex combinations of elementary obstacles alongside oscillatory movements with variability in sizes & speeds.
- Implemented a highly dynamic & tooltip loaded **interactive Leader board** comprising of radiant animations. Managed via a **generic Database** utilized for both load game and leader board.
- Created **randomized & complex easter eggs** in the game with unexpected responses yet rewarding results.
- Rendering done via **graphics context for extremely efficient performance** providing enriching and ultra smooth user gameplay, utilizing concepts of **linear algebra** such as **vector cross-product, rotation, etc.** with concepts of physics such as **2D projectile motion, momentum conservation, etc.**
- Implemented an highly interactive user interface with **unique animations** and **audio clips** for **every interactable**.
- Information keep is created for storing minute game details such as **distance travelled, jumps, date, etc.**
- For Bubbles mode, an all-together **separate theme is introduced with swift animations** and smooth transitions.
- Implemented an **adaptable media player with changing music** for every screen with switching transitions.
- Designs patterns such as **template, factory, façade, etc.** implemented with defensive programming, custom exceptions and corner case handling.

In Game Snapshots