COLLISION DETECTION

The Shape.intersect() method found in class java.scene.shape.Shape has been used. It returns a new Shape which is created as an intersection of the specified input shapes. If the new shape in empty, there is no intersection.

OBSTACLE GENERATION

The obstacles are generated in a predefined order at a hardcoded distance from the previous obstacle. Obstacles are removed as they go offscreen, at most 10 obstacles can be added after which we must wait for the first obstacle to be removed to add a new one.

COLOR SWITCH

By
Rachita Saha (2019082)
Karan Abrol (2019366)

MOVEMENT OF BALL

Once a new game has been loaded, all components of the game are stationary until the first input is registered. The ball gets an impulsive upwards velocity everytime the bounce input is given and has a constant downward acceleration throughout the gameplay.

OBSTACLE MOVEMENT

The background moves when the ball has reached the center of the screen. Every obstacle, star and colorswitcher mimics the ball's movement in the opposite direction, and the ball is made stationary. All animations are done using the animation timer class and move functions.

APPLICATION OF OOP CONCEPTS

ABSTRACTION & ENCAPSULATION

The variables inside all classes have been marked private/protected and getter, setter methods have been used to access them.

INHERITANCE

The obstacle class is abstract and the 8 types of obstacles are extending it. This enables code reuse and common variables and methods need not be defined again.

POLYMORPHISM

In the GamePlay class, all obstacles have declared type as Obstacle which allows code reuse and use of the methods like intersect(), move(), which are implemented differently in all obstacles.

DESIGN PATTERNS

The facade design pattern has been used by implementing switch cases in object generation. The abstract factory pattern has been used while designing obstacles.

SERIALIZATION & DESERIALIZATION

The high score and list of all saved games is serialized and stored in a file, and the main menu deserializes the data every time it is created so that the game remains updated.

EXCEPTION HANDLING

All exceptions have been caught and handled properly . User-defined exceptions 'BallNotFoundException' and "ObstacleNotPlacedException' have also been handled.

CLASS RELATIONSHIPS

Ball, obstacles, stars and color switchers have a composition relationship with the GamePlay class as they are all created as the game progrsses. MainMenu has an association relationship with all other pages.

ABSTRACT CLASS

The different obstacles are grouped together as siblings under Obstacle abstract class. It keeps the program organized and understandable. It provides a base for creating all obstacles and ensures they all define the required common methods.

INDIVIDUAL CONTRIBUTIONS

RACHITA SAHA - 2019082

- Digital design of UML diagrams
- Ideation of classes, objects and class relationships.
- Created GUI of Main, MainMenu, SavedGamesPage, PausePage, GameOverPage.
- Placement of obstacles, stars, color switchers in gameplay class.
- Obstacle generation and placement of stars and color switchers.
- Collision detection.
- Added bonus elements - sound effects, game over animation.

KARAN ABROL - 2019366

- Ideated and identified class relationships and multiplicities for the UML diagrams.
- Created GUI for 8 obstacles.
- Coded animations for 8 obstacles.
- Coded the vertical movement of ball , obstacles, stars and color switchers in endless mode.
- Implemented serialization and deserialization of game state.
- Debugging and testing.

BONUS ELEMENTS

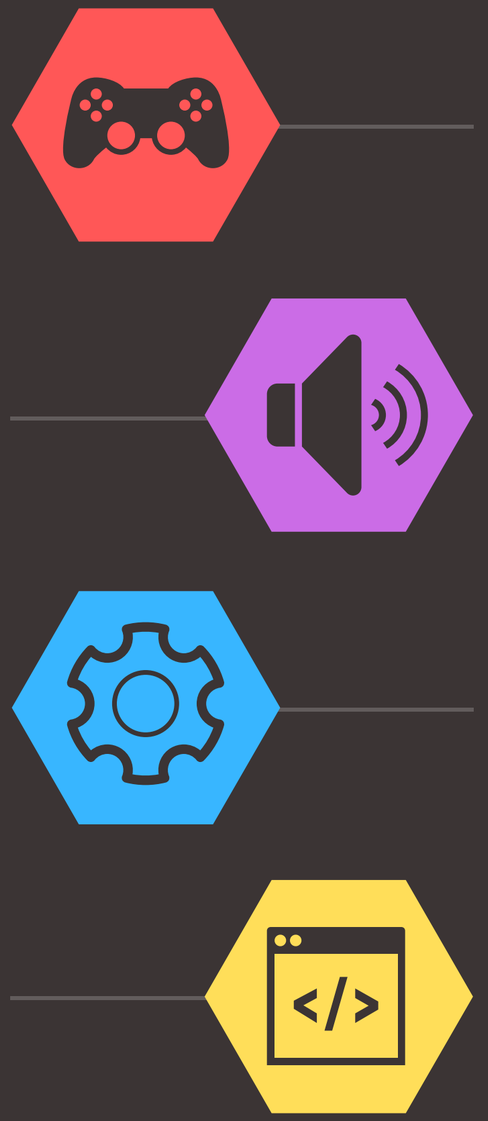
Various features have been added to make our project closely resemble the actual game. In several places, the same sound effects and animations have been added which are present in the actual game.

SOUND EFFECTS

IN-GAME SOUND EFFECT:
Sounds are played when the ball jumps, collects a star, switches color, collides with object, and when a new high score is se
SOUNDS IN PAGES
Sounds are played on button click and when an error dialog box pops up. The theme music plays when the application starts.

ANIMATIONS IN PAGES

When the application is started, an animation similar to the one created in the actual game is displayed. There is a similar animation in the pause page.



GAME OVER ANIMATION

When the ball collides with an obstacle which is not of the same color, it scatters into different small balls which move randomly on the screen and bounce off the edges just like in the actual game.

CHANGE SETTINGS

The player can change the in-game sound settings in the main menu. This disables the sounds produced in game. Dialog boxes have been used to take input for game name and to display information.