Samyak Jain (2019098)

Utkrisht Sikka (2019215)

Design and Implementation

1.**Singleton** : There is only one instance of Database class needed. Every api of Database class requires invocation of getInstance() api of Database.
2. **Iterator**: GameAchievements, OwnedTrails, List of Stars, Obstacles are traversed using specific iterator for each of them respectively.
3. **Observer**: Ball , obstacles, stars, colorswitcher in the scene listen for up key press. When up key is pressed, the ball moves up and stops at half mark. From there the obstacles ,stars and colorswitcher move down ,giving the effect that ball is moving up in space.
4.**Factory** : Whenever an obstacle gets out of screen, a new obstacle is created at top of screen by Obstaclefactory object.
5.**Facade** : The handling of keystrokes in GameMain has been done in a facade DP implementation type of fashion using a switch case.

Issue Faced:

1.Movement of ball was a very tricky and time consuming task. Not just the ball moved up but also all the obstacles in the scene had to move down. This together had to give an effect that only ball is moving up.Also the ball had to move down to give gravity effect.
First problem was that we wrote the algo assuming that Y axis extends towards top. But later realised that Y axis extends towards bottom. We resolved this by going through every function(one by one) to adjust the code to modified axis.
Second problem was to stop the ball at half way mark.
Solution: We computed the remaining distance from half way mark and if ball was too close, we also moved the obstacles and star, down after completion of ball's motion.
Third problem : the amount of distance the ball should move up at every up key.
Solution: We made distance to move dependent on the distance of ball from half mark.
Fourth problem: handling of multiple up keystrokes.
Solution: We stopped the original motion of the ball and defined new trajectory on every key press.
2. JavaFX components are not Serializable.
Solution: We serialized the necessary info that we would have required to recreate the scene. All Java FX components were reinstantiated and set.

Implementation

Ball, star, ColorSwitcher,Obstacle extend from GameObject. We have used a separate thread for collision checking and detecting when obstacles go out of bounds. Color switcher is implementing SpecialObject interface . It overrides method specialChange(Ball b).

The various types of individual obstacles are constructed using Path type objects and their animation is implemented using Timeline, Rotate and TranslateTransition.They all extend from Obstacle.

MainPageMenu , InGameMenu, ObstacleHitMenu extend abstract class Menu which further extends Application. TranslateTransition has been used extensively in the animation overall.

Efforts by Samyak

Deadline 1: Designing and constructing UML Class Diagram.Partially designing use case diagram.

Deadline 2: Construction of obstacles and implementing their rotating animations. Implementing main menu page, ingame menu, game over menu along with buttons.

Deadline 3: Implemented transitions between different menus, partially implementing endless game algorithm, load and save page and saving-loading, serialization and deserialization of games, pause and save game,all the handlers, adding GUI elements.

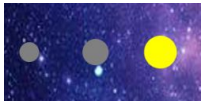
Efforts by Utkrisht

Deadline 1: Designing and Constructing UML Use Case diagram. Partially designing class diagram.

Deadline 2: Implementing animations of ball, stars , hand and color switcher. Implemented downward motion of obstacles, display score.

Deadline 3: Implementing the main algorithm for endless game and collision checking, pause , resume functionalities, splitting of ball effect,partially handling menu functionalities loading and saving attributes of game objects, adding GUI elements.

Bonus Components



1. Trail effect
2. 7 different types of Obstacles
3. Achievements page based on stars collected
4. Shop to buy 3 different trails: fire, neon laser,mist
5. Spin wheel allowed to be spinned once in a while.
6. Super colorswitcher: gives the ball power to pass through obstacles
7. Cool sound and music for every interaction in the game, Background Music
8. Splitting of Ball on game over
9. Press X to take screenshot of game screen(not of laptop screen).
10. Online Feedback provision functionality
11. Splash screen
12. Fluid animations
13. Hand to hold the ball on new Game
14. Colorful and interactive menu buttons,gifs
15. HowToPlay page