Group 79, sourabh kumar(2022507), Siddhant Singh(2022497)

-----------------------------------------------README-------------------------------------------------------------------

1. Game Overview:

- The game centers around a character, depicted by an ImageView named `playerid`, traversing between two platforms with the goal of covering the maximum distance without falling.

- Player control involves using the SPACE key for stick size and arrow keys for adjusting the character's orientation.

2. Class Structure:

* The main class, `ScreenController`, implements the `Initializable` interface.

* The code leverages JavaFX components, including buttons, images, labels, rectangles, lines, and animations, to create the graphical user interface.

* The Singleton pattern ensures that a class has only one instance and provides a global point of access to that instance. In the given code, the Gamemechanics class has a private static instance of itself (gamemechanics), and it provides a public static method getGamemechanics() to access the instance. The constructor of the class is private, ensuring that no other class can create an instance of Gamemechanics directly.

* Playerinfo class is designed as a flyweight, and it uses a Map<String, Playerinfo> to maintain a pool of unique instances based on the player's name.

Classes:

* ScreenController: This class handles the main gameplay logic, including scene transitions, player movement, collision detection, and scorekeeping.

* Gamemechanics: This class generates random platform positions and distances, ensuring a varied and challenging gameplay experience.

* Pause: This class handles the paused state of the game, displaying a pause menu and resuming the game when unpaused.

* Playerinfo: This class stores player information, such as total cherries collected and highest score.

3. Game Mechanics:

- Dynamic platform generation (`rec1` and `rec2`) with variable widths and distances characterizes the game.

- Cherry collectibles (`cheeryid`) are randomly positioned either at the top or bottom of the screen, contributing additional points upon collection.

4. Audio Integration:

- Sound effects, such as button clicks, character movement, stick falling, and cherry collection, enhance the gaming experience.

5. Transitions and Animation:

- `TranslateTransition` is employed to achieve smooth animations for the character and other in-game elements.

6. Pausing and Exiting:

- Players can pause the game by pressing the 'P' key, triggering a modal pause screen.

- Exiting the game initiates a confirmation dialog to prevent unintended exits.

7. Scoring and Game Over:

- The game keeps a tally of collected cherries, displaying the count on the screen.

- Game over conditions occur if the character falls or fails to reach the next platform, leading to a summary screen where players can choose to exit or continue.

8. Stage Management:

- The code effectively manages multiple stages corresponding to different game states, including the main game screen and exit/summary screens.

9. Repeating and Restarting:

- The game facilitates restarting after successfully reaching a platform, ensuring a continuous and seamless gaming experience.

10. Randomization:

- Randomization is utilized to generate diverse platform characteristics, including widths, distances, and cherry positions, contributing to the game's variability.

HOW TO RUN:

* Locate the ScreenController class, which contains the main method.
* Run the ScreenController class as a Java application.
* A gamestarting window should appears: START THE GAME;
* Controls: SPACE key is used for increasing the stick size, and UP/DOWN arrow keys are used for adjusting the character's orientation.
* Play the game by navigating the character, collecting cherries, and trying to achieve the highest score.
* Press the 'P' key to pause the game. To exit the game, use the in-game options or follow standard window close procedures.
* If you want to play again, follow the provided options to repeat or restart the game.

CREATIVITY:

* The game is in full screen mode giving more immersive experience
* Players can now enter their name
* Player can revive themself if they have a minimum 2 cherries.
* Players can now pause the game in between and resume
* The sound in the home screen makes it more interactive

NOTE:

* If any button is not working , please try with the mouse and vice versa.
* If the game crashes or lags at any point its due to low specification of the computer.
* Player will revive if and only if it will have enough points.

Have fun playing "Stick Hero"!

<https://github.com/Siddhant22497/AP_Project1>