Kevin Shilla

Pattern Memory Game | JavaFX

Project Description: Developed an interactive Pattern Memory Game using JavaFX, where players are challenged to remember and replicate an increasingly complex sequence of illuminated squares on a grid. The game includes the following features:

- **Dynamic Grid Selection**: Players can choose from three grid sizes (3x3, 6x6, 12x12), each offering a different level of difficulty.
- **Interactive Gameplay**: The game highlights a sequence of squares that the player must memorize and click in the correct order. Each round, the sequence grows longer, increasing the challenge.
- **Visual Feedback**: Implemented color transitions using `Fill Transition` to visually indicate which squares are part of the sequence. Squares transition from a default gray color to yellow when highlighted, and back to gray.
- **User Interface**: Designed a user-friendly interface with styled buttons for grid size selection and a game over screen that displays the player's score and offers a restart option.
- **Event Handling**: Managed user interactions through mouse click events on the squares, checking for correctness and providing feedback on success or failure.
- **Real-Time Updates**: Utilized threading to handle sequence highlighting and delays, ensuring smooth and responsive gameplay.

Code Highlights:

- **Grid Initialization**: The grid is dynamically created based on the selected size, with each square being interactive and capable of color transitions.
- **Sequence Management**: The game generates a random sequence of squares that the player must follow. The sequence is highlighted at the start of each round.
- **Game Logic**: Handles user input to verify the correct sequence, advances the game round upon success, and shows a game over message upon failure.

Impressiveness: This project demonstrates advanced proficiency in JavaFX, including:

- **GUI Design**: Creating a visually appealing and functional user interface.
- **Animation**: Using transitions to enhance user experience.
- **Event-Driven Programming**: Handling user interactions in real time.
- **Game Development**: Implementing game logic and managing game state.