



Color Matching Ping Pong Game

2310030028

Srikartik Mateti

2310030224

Pranay Varma

2310030223

Sujan Cherukuri

2310030324

Karthik V.S





Introduction

Project Overview

- This project is a **ping pong-style game** with an added element of **color matching**. The game consists of a bat and balls appearing from the right side of the screen. The bat changes color randomly at regular intervals, and the player must ensure that the ball and bat colors match when they collide. If the colors match, the player continues playing.

Project Objectives

- Develop an **interactive Java-based game** with a simple yet engaging mechanism.
- Improve **reaction time and decision-making skills** through fast-paced gameplay.
- Integrate **audio feedback** to enhance the gaming experience.
- Implement **Object-Oriented Programming (OOP) principles** for better structure and modularity.
- Use **JavaFX and Java Sound API** for graphics and sound output.

Literature Review

+ **Color Recognition and Human Perception**

- Studies in **cognitive psychology** suggest that reaction-based games improve reflexes and decision-making skills.
- Color differentiation plays a crucial role in **human visual perception and attention. Importance of Auditory Feedback in Gaming**
- Studies indicate that **auditory feedback enhances memory retention and response accuracy.**
- Examples include rhythm-based games like **Guitar Hero** and **Color Switch**, which use sound cues to aid player decisions.

+ **Evolution of Ping Pong Games**

- Modern versions of ping pong games have evolved by integrating **physics-based mechanics and interactive challenges.**
- The addition of **color-matching elements** in this project makes it distinct from traditional ping pong games.



Survey of Existing Websites/WebApps

- + **Classic Pong Game (Atari, 1972)**
 - + Simple mechanics, two paddles and a bouncing ball.
 - + No color-matching concept, only movement-based gameplay.
- + **Color Switch (Mobile Game)**
 - + Players navigate a colored ball through matching color segments.
 - + Focuses on reaction time and pattern recognition.
- + **Color Matching Memory Games**
 - + Involves matching colors for points but lacks the real-time movement aspect.
 - + Often used for cognitive skill training.
- + Key Takeaway: Our game is unique because it combines reaction-based gameplay from Pong with color-matching mechanics and audio feedback.



Technologies Used

+ 1. Java Programming Language

- Java provides **object-oriented principles**, making the game easy to scale and maintain.
- Java is **cross-platform**, meaning the game can run on multiple operating systems.

+ 2. JavaFX for Graphical User Interface (GUI)

- JavaFX is used for rendering **game graphics, animations, and user interface elements**.
- Supports **real-time rendering** and smooth **frame transitions**.
- Provides built-in support for handling **keyboard and mouse events**.

+ 3. Java Sound API for Audio Output

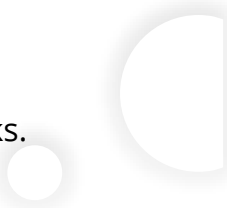
- Used for **announcing the color names** of the balls.
- Converts **text-based color names into speech** using pre-recorded sound files or text-to-speech libraries.

+ 4. Random Class for Color Generation

- The bat and ball colors are randomly selected using Java's **Random class**.
- Ensures unpredictability and keeps the game engaging.

+ 5. Multithreading for Game Loop and Performance Optimization

- Ensures **seamless execution of animations and background tasks** without freezing the game interface.
- Separates the game loop from rendering logic to avoid performance bottlenecks.



Challenges

+ Color Visibility and Differentiation

Problem: Some colors may be too similar, making them hard to distinguish.

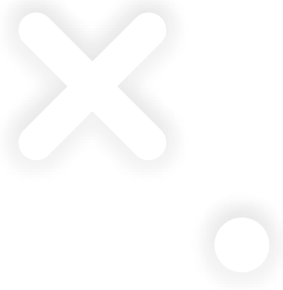
Audio and Graphics Synchronization

Problem: Delays in announcing the color name might disrupt gameplay.

Performance Optimization

Problem: Smooth ball movement may be affected by lag.

Solutions

- + **Solution 1:** Use **high-contrast color palettes** and avoid colors that are difficult to differentiate for color-blind users.
 - + **Solution 2:** Optimize **Java Sound API buffering** and use **asynchronous processing** to minimize delay.
 - + **Solution 3:** Implement **separate threads for game logic and rendering** to improve performance.
- 

Game Features and Explanation

+ 1. Randomized Bat Color Change

- The bat's color changes at **regular intervals** to increase unpredictability.
- This prevents players from anticipating the bat's color in advance.

+ 2. Ball Movement and Collision Detection

- Balls spawn **from the right side of the screen** and move towards the bat.
- Collision detection ensures that when the ball reaches the bat, its color is compared to the bat's color.

+ 3. Color Matching Mechanism

- If the bat and ball **colors match**, the player earns points and continues playing.
- If the **colors do not match**, the game either ends or applies a penalty (such as reducing the score or lives).

+ 4. Audio Feedback for Color Announcement

- The game **announces the color name** of the ball upon collision, regardless of whether it matches the bat's color.
- This feature enhances **player engagement and reaction speed**.

+ 5. Increasing Difficulty Over Time

- The ball's speed increases as the game progresses, making it more challenging.
- The bat's color-changing interval may also become shorter over time to test the player's reaction skills.





Development Workflow



A decorative graphic consisting of three 'X' marks and three circles. One 'X' and one circle are in the top left, a larger 'X' and one circle are in the top right, and one 'X' and one circle are in the bottom right. The 'X' marks are larger and more prominent than the circles.

Thank You