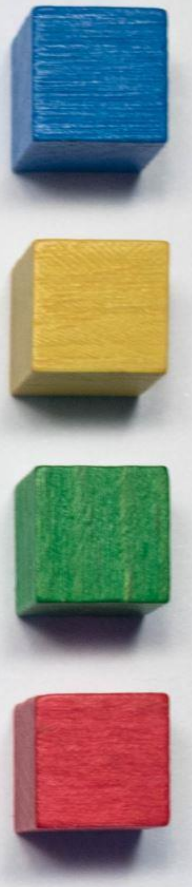
BRICK BREAKER GAME PROJECT

AN INTRODUCTION TO GAME DEVELOPMENT USING JAVA



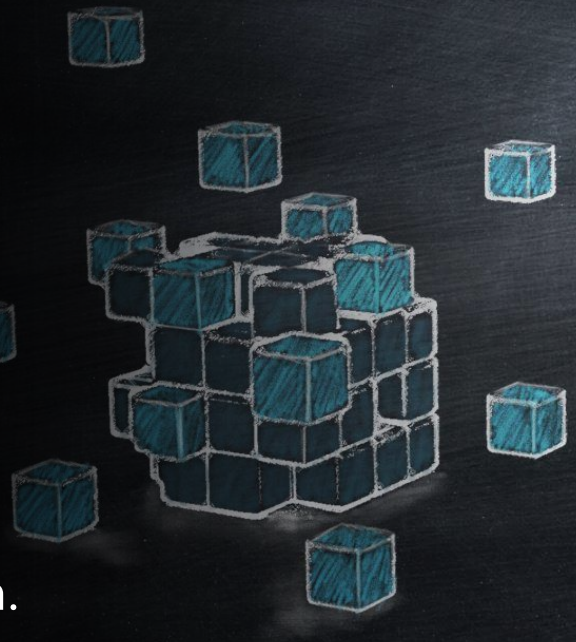
PRESENTED BY: Ishita Dhasmana

INTRODUCTION

Introduction to Brick Breaker

- Classic arcade game popular since the late 1970s.
- Simple yet addictive gameplay.
- Player controls a paddle to hit a ball and break bricks.
- Objective: Destroy all bricks and prevent the ball from falling off the screen.

GAME OBJECTIVE AND MECHANICS



Game objectives

- Break all the bricks using a ball.
- Prevent the ball from falling off the bottom edge.

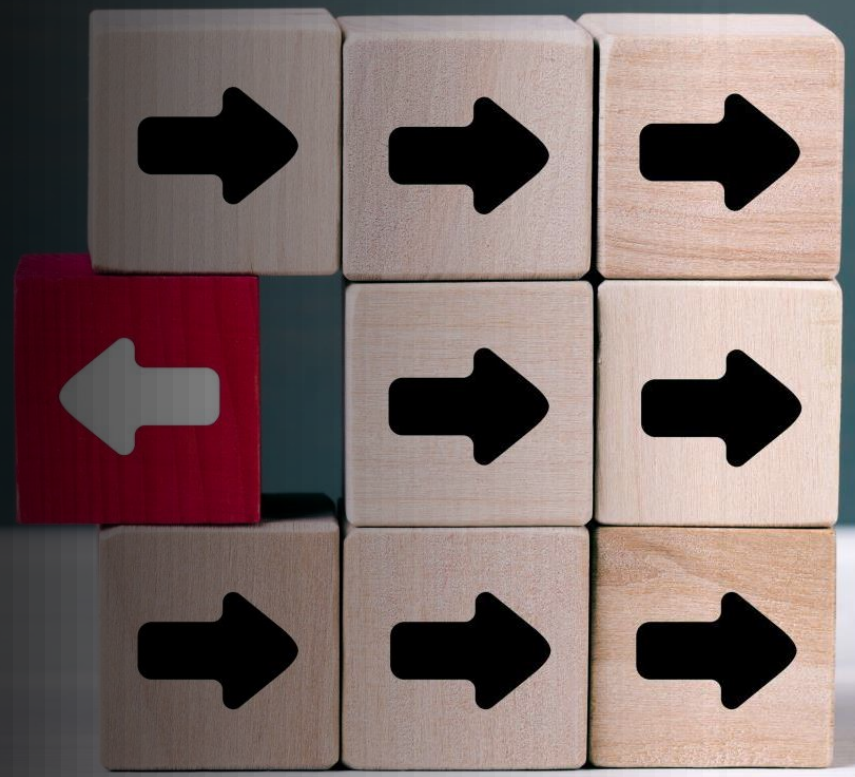
Gameplay Mechanics

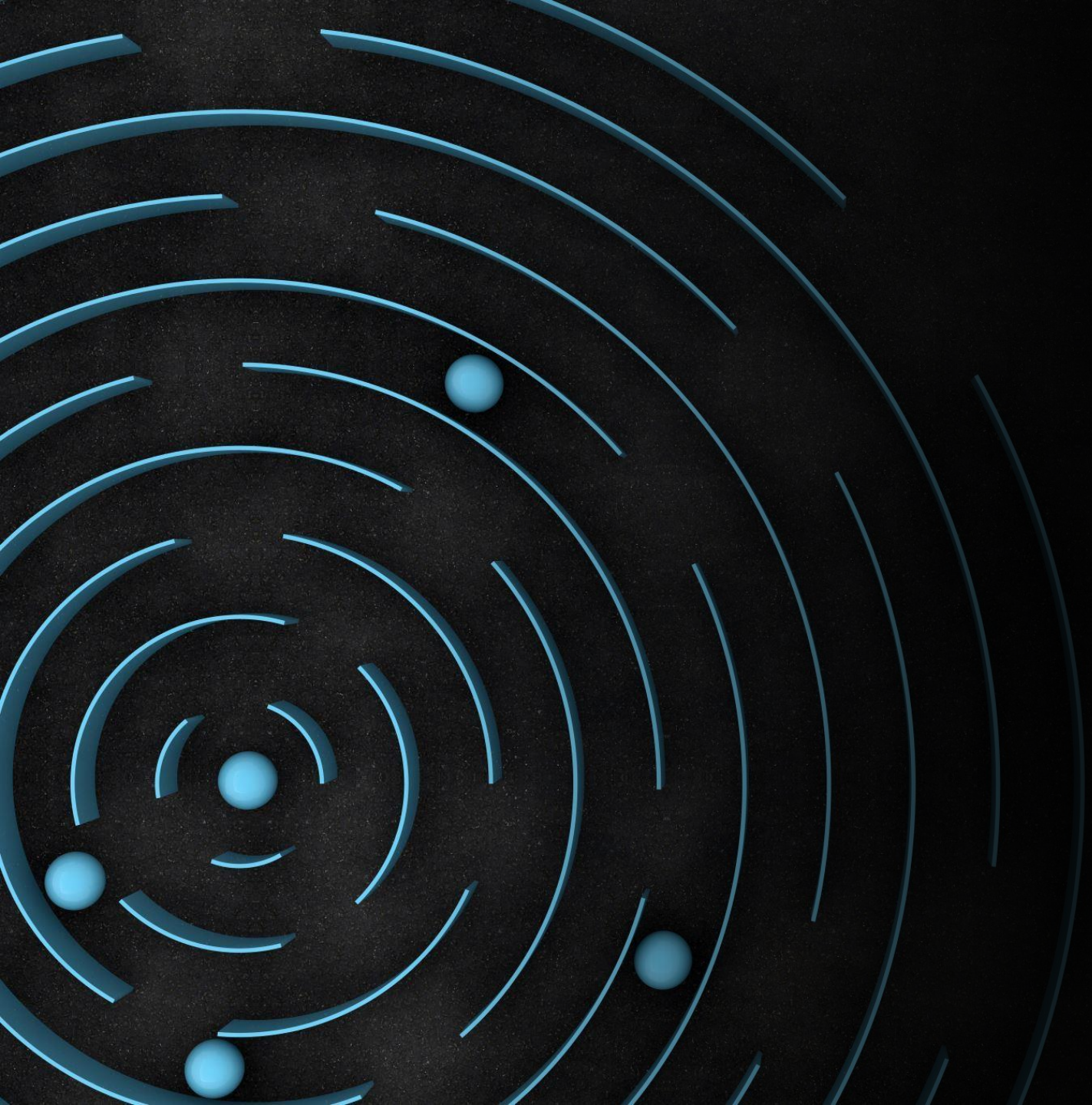
- **Paddle**: Moves left and right to hit the ball.
- **Ball**: Bounces around, breaking bricks on collision.
- **Bricks**: Arranged in a grid, break upon collision with the ball.
- **Score**: Points awarded for each brick broken.

USER CONTROL

User Controls

- **Left Arrow Key**: Move the paddle left.
- **Right Arrow Key**: Move the paddle right.
- **Objective**: Keep the ball in play and direct it towards the bricks.





COLLISION DETECTION

Collision Detection

- **Paddle Collision**: Ball reverses vertical direction upon hitting the paddle.
- **Wall Collision**: Ball bounces off the screen edges.
- **Brick Collision**: Ball breaks bricks and reverses direction.

GRAPHICS AND RENDERING

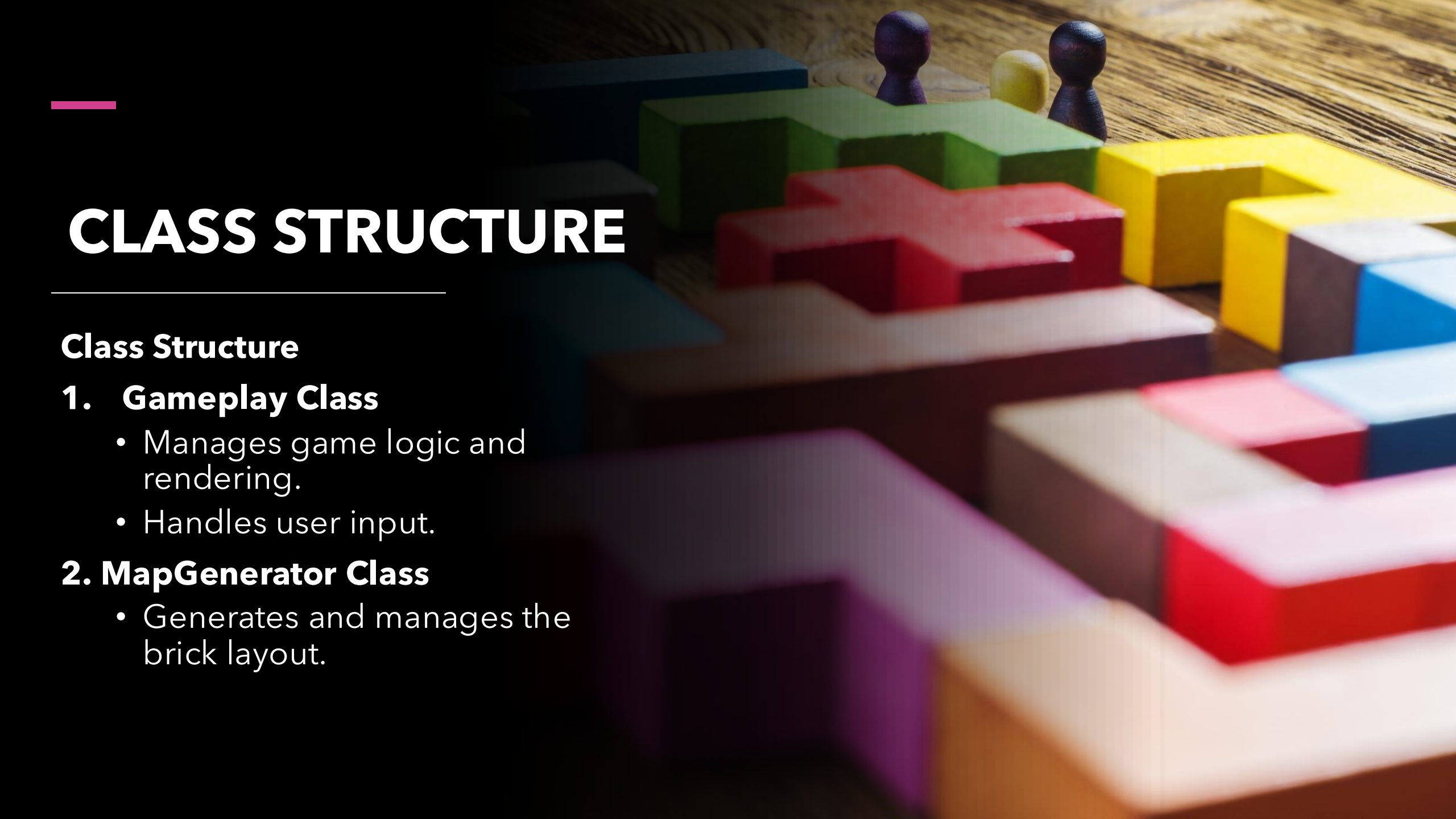
Graphics and Rendering

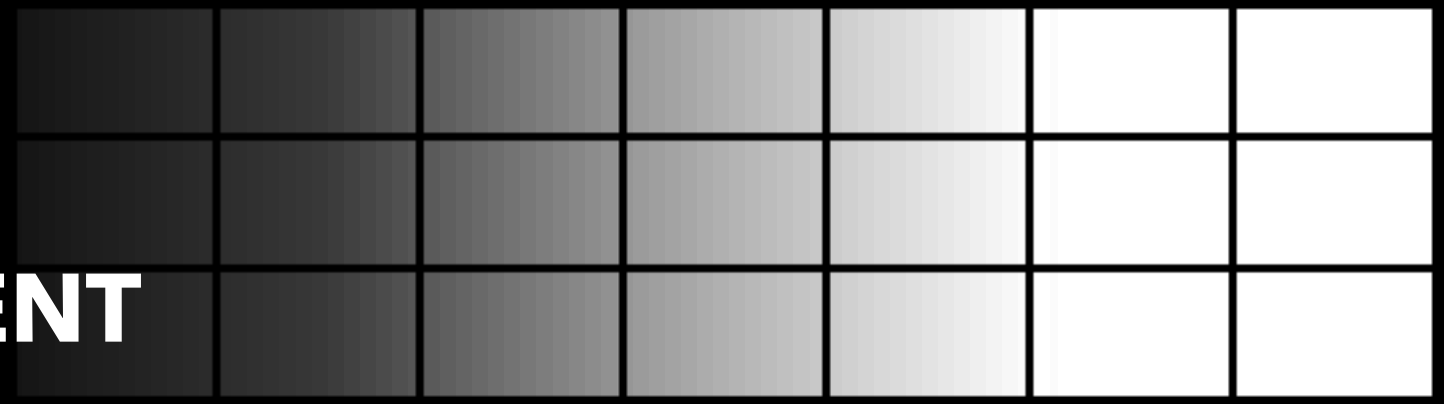
- **Java Graphics**: Uses Graphics and Graphics2D classes.
- **Rendering Elements**:
  - Background
  - Borders
  - Paddle
  - Ball
  - Bricks

CLASS STRUCTURE

Class Structure

1. **Gameplay Class**
   - Manages game logic and rendering.
   - Handles user input.
2. **MapGenerator Class**
   - Generates and manages the brick layout.





GAME STATE MANAGEMENT

Game State Management

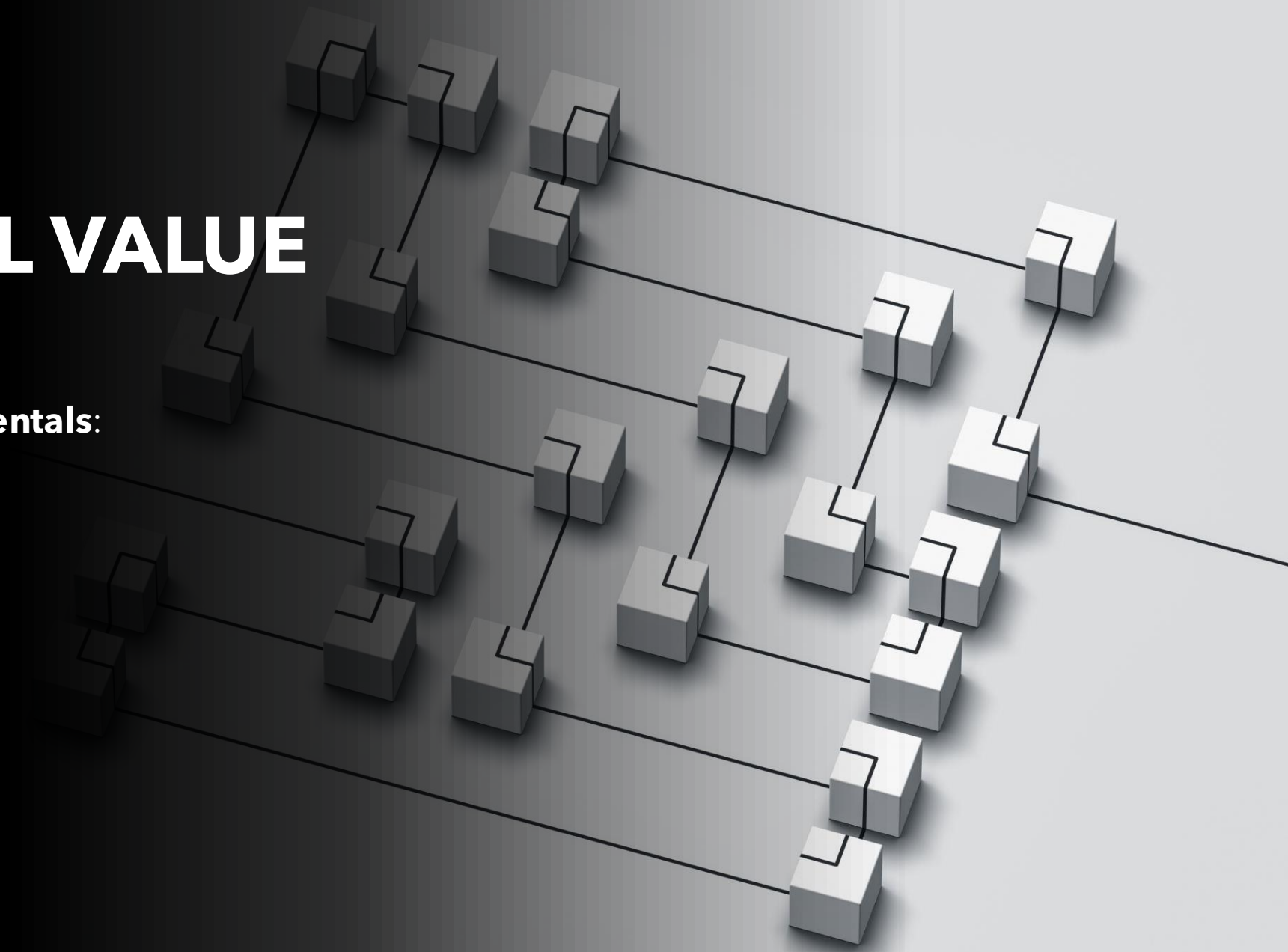
- **Game Loop**: Uses a timer to update and repaint the game.
- **State Handling**:
  - Start
  - Play
  - End

FEATURES AND ENHANCEMENT

Features and Enhancements

- **Power-Ups**: E.g., multi-ball, larger paddle.
- **Multiple Levels**: Increasing difficulty.
- **Sound Effects**: For immersive gameplay.
- **High Scores**: Track and display top scores.

EDUCATIONAL VALUE



Educational Value

- **Game Development Fundamentals**:
  - Collision detection
  - Game loops
  - User input handling
- **Java Programming**:
  - Object-oriented principles
  - Graphics rendering
  - State management
- **Problem Solving**:
  - Implementing mechanics
  - Debugging issues

CONCLUSION

- Comprehensive exercise in game development.
- Solid foundation for beginners.
- Fun and educational project for experienced developers.
- Demonstrates key concepts in Java programming and game mechanics.

THANK YOU