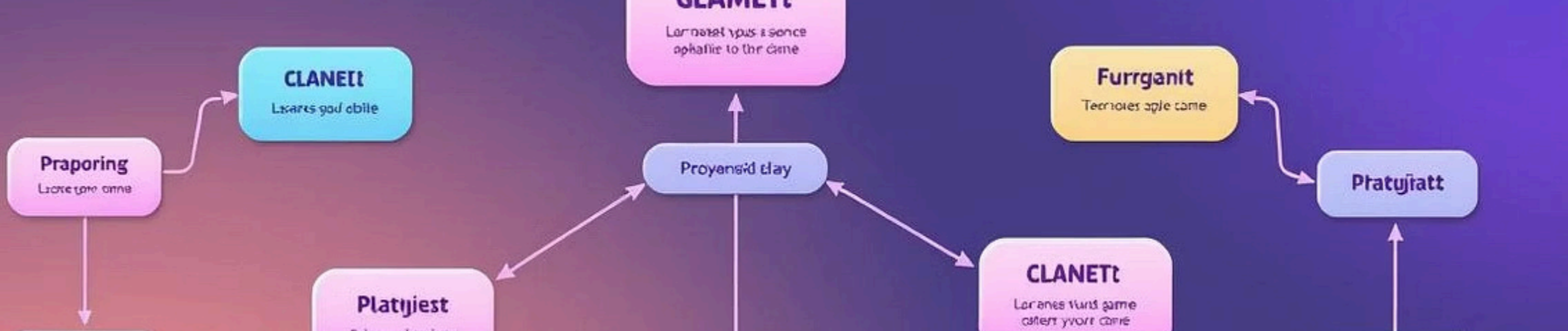# Rock Paper Scissors: A Java Implementation

This presentation covers how to create a Rock Paper Scissors game using Java.

We'll explore design, coding, testing, and future improvements.

S. **by S. Pragadheesh**

Project Overview: Objectives and Scope

Objective

Build a console-based Rock Paper Scissors game in Java.

Scope

Include user input, game rules, round logic, and score tracking.

Outcome

Familiarity with basic Java programming and control structures.

Core Logic: Implementing Game Rules in Java

1 **Define Choices**

Represent Rock, Paper, Scissors as constants or enums.

2 **Compare Moves**

Use conditional statements to decide the winner per round.

3 **Handle Ties**

Recognize and replay when both players choose identically.

Made with **GAMMA**

Code Structure: Classes and Methods Explained

Main Class

Controls game flow, user input, and output display.

GameLogic Class

Contains methods to evaluate moves and track scores.

Utilities

Assist with input validation and random computer choices.

User Interface: Designing for Interaction

⌨ **User Input**

Text prompts for player choices (Rock, Paper, Scissors).

🖥 **Output Messages**

Show round results and cumulative scores clearly.

🔁 **Replay Option**

Allow users to restart or quit after each game session.

Welcome to Java Console!

```
yoine welbome to jaav a conrack:

the feedlback to .scule jelcursof tolp tb)
the java faldebock:
ree feedleaschbogkar -- up)
```

Testing and Debugging: Ensuring Game Accuracy

1 **Unit Tests**

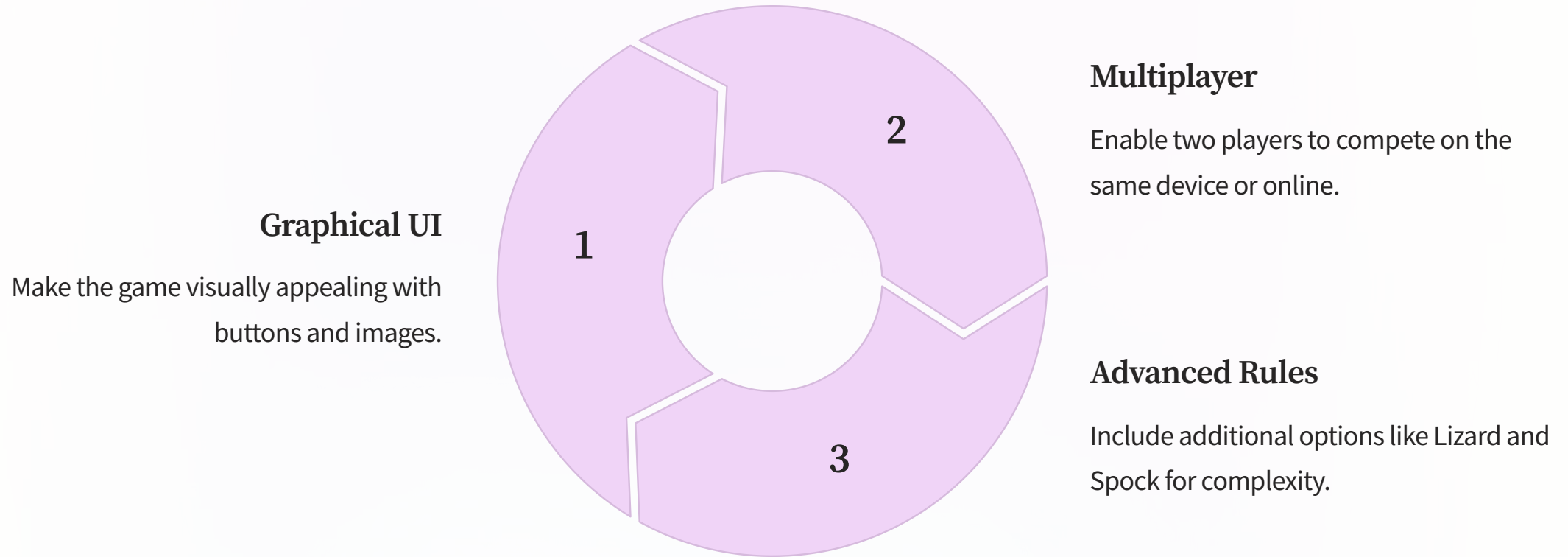Test methods for correct winner determination and input handling.

2 **Playtesting**

Simulate multiple rounds for logical consistency and edge cases.

3 **Debugging**

Use IDE tools to find and fix errors or unexpected behaviors.

Made with GAMMA

Enhancements: Adding Features and Complexity

Graphical UI

Make the game visually appealing with buttons and images.

Multiplayer

Enable two players to compete on the same device or online.

Advanced Rules

Include additional options like Lizard and Spock for complexity.

Made with GAMMA

Conclusion: Key Takeaways and Future Development

Core Skills

Java basics with loops, conditionals, and input handling mastered.

Next Steps

Add GUI and multiplayer features for better user experience.

Keep Learning

Explore advanced Java topics and game development frameworks.