# steps for each phase:

## 1. Planning and Setup

- Write a detailed game design document

- List required features and prioritize them

- Set up Git repository

- Create project in IDE with necessary libraries (Swing)

## 2. Core Game Engine

- Implement GameLoop class with update() and render() methods

- Create abstract Entity class for game objects

- Implement simple collision detection using rectangles

## 3. Graphics and Rendering

- Create GamePanel class extending JPanel

- Implement double buffering for smooth rendering

- Design tilemap system for the game world

- Create Sprite and Animation classes

4. Player Mechanics

- Implement KeyListener for player input

- Create Player class with movement and shooting methods

- Implement projectile system

5. Enemy AI and Spawning

- Create Enemy class with basic AI behaviors

- Implement EnemySpawner class to manage enemy creation

- Design different enemy types (e.g., melee, ranged)

6. UI and Sound

- Create HUD class for rendering game information

- Implement menu screens using Swing components

- Use Java's built-in sound API for audio

7. Game Progression and Polish

- Implement LevelManager or WaveManager class

- Create PowerUp class and system

- Fine-tune game parameters for balance

- Add particle effects for visual polish

8. Final Testing and Packaging

- Write unit tests for core game components

- Perform playtesting and gather feedback

- Use Java's built-in packaging tools to create executable JAR

Remember to use object-oriented principles throughout development, keeping your code modular and maintainable. Also, consider using design patterns like the State pattern for game states or the Factory pattern for entity creation.

Would you like me to elaborate on any specific part of this plan or provide code examples for key components?