

# Snake Game - Complete Documentation

**Version:** 1.0

**Last Updated:** 2023-10-15

---

## 1. Overview

A modern Snake game built in Java with enhanced features:

- ✓ Multiple food types (normal, golden, rotten, bonus)
  - ✓ Power-ups (speed boost, double points, invincibility)
  - ✓ High-score system with file persistence
  - ✓ Particle effects for visual feedback
  - ✓ Sound effects (eating, game over, power-ups)
  - ✓ Difficulty levels (Easy, Medium, Hard)
  - ✓ Obstacles (walls)
- 

## 2. File Structure

Copy

Download

SnakeGame/

```
├── src/
|   ├── main/
|   |   ├── java/
|   |   |   ├── Main.java
|   |   |   ├── GameWindow.java
|   |   |   ├── GameLoop.java
|   |   |   ├── GamePanel.java
|   |   |   ├── Snake.java
|   |   |   ├── Food.java
|   |   |   ├── PowerUp.java
|   |   |   └── ScoreManager.java
```

```
| | | └─ SoundManager.java
| | | └─ ParticleSystem.java
| | └─ resources/
| |   └─ sounds/
| |     └─ normal.wav
| |     └─ gold.wav
| |     └─ rotten.wav
| |     └─ bonus.wav
| |     └─ powerup.wav
| |     └─ gameover.wav
└─ README.md
└─ highscores.dat
```

---

### 3. Key Features

#### 3.1 Game Mechanics

Feature	Description
<b>Snake Movement</b>	Arrow keys (↑, ↓, ←, →)
<b>Food Types</b>	Normal (+10), Golden (+30), Rotten (-5), Bonus (+15)
<b>Power-Ups</b>	Speed Boost ( ⚡ ), Double Points (2X), Invincibility ( ✨ )
<b>Obstacles</b>	Walls (instant death)
<b>Scoring</b>	Persisted high scores (highscores.dat)

---

#### 3.2 Difficulty Levels

Difficulty	Speed	Growth Rate	Score Multiplier
Easy	Slow	+3 segments	1.0x
Medium	Normal	+5 segments	1.5x
Hard	Fast	+10 segments	2.0x

### 3.3 Sound Effects

Sound File	Trigger
normal.wav	Eating regular food
gold.wav	Eating golden food
rotten.wav	Eating rotten food
powerup.wav	Collecting power-up
gameover.wav	Snake dies

## 4. Class Documentation

### 4.1 `Main.java

**Purpose:** Entry point for the game.

**Key Methods:**

- `main()`: Launches the game window.

### 4.2 `GameWindow.java

**Purpose:** Manages the game window and transitions between menu/game states.

**Key Methods:**

Method	Description
startGame(int difficulty)	Initializes game based on selected difficulty
showMenu()	Displays the main menu

#### 4.3 `GameLoop.java

**Purpose:** Core game logic and loop.

**Key Methods:**

Method	Description
update()	Handles collisions, scoring, and power-ups
checkFoodCollision()	Applies food effects (growth/score)
spawnPowerUp()	Randomly generates power-ups

#### 4.4 `Snake.java

**Purpose:** Snake behavior and movement.

**Key Methods:**

Method	Description
move()	Updates snake position
grow()	Increases snake length
collidesWithSelf()	Checks if snake hits itself

#### 4.5 `Food.java

**Purpose:** Manages food types and spawning.

**Key Methods:**

Method	Description
--------	-------------

respawn()	Places food randomly
-----------	----------------------

draw()	Renders food based on type
--------	----------------------------

#### 4.6 `PowerUp.java

**Purpose:** Power-up effects and duration.

**Key Methods:**

Method	Description
--------	-------------

activate()	Starts power-up timer
------------	-----------------------

deactivate()	Clears power-up effects
--------------	-------------------------

#### 4.7 `ScoreManager.java

**Purpose:** Saves/loads high scores.

**Key Methods:**

Method	Description
--------	-------------

saveScore()	Writes scores to highscores.dat
-------------	---------------------------------

getTopScores()	Returns top 10 scores
----------------	-----------------------

#### 4.8 `SoundManager.java

**Purpose:** Plays sound effects.

**Key Fix:**

java

Copy

Download

// Updated to load sounds from JAR/resources

InputStream is = getClass().getResourceAsStream("/sounds/" + filename);

```
AudioInputStream audioStream = AudioSystem.getAudioInputStream(is);
```

---

## 5. How to Run

1. **Compile:**
2. `javac *.java`
3. **Run:**

```
java Main
```

4. **Controls:**
    - **Arrow Keys:** Move snake
    - **Space:** Restart after game over
    - **ESC:** Return to menu
-