Game Design Document (GDD)

Project: Spike Runner

(No scoring system — matches the current implemented game)

---

1. Game Overview

Title: Spike Runner
Genre: Endless Runner / 2D Platformer
Target Platform: PC (Windows)
Engine: Unity 6

Short Description:
Spike Runner is a minimalistic endless runner where the player must avoid spikes and saw blades by timing their jumps. The goal is simply to survive as long as possible. There is no scoring system; the challenge comes from the increasing speed and density of obstacles.

---

2. Core Concept

The player controls a small creature running endlessly across a pixel-art platform world. The game increases in speed over time, making survival progressively harder. The simplicity of mechanics makes the game accessible while still challenging.

---

3. Game Mechanics

3.1 Player Movement

- The player runs automatically at a constant speed (which increases over time).
- Press Space to jump.
- There is no double jump, but the game uses a variable jump height mechanic:
    - A short tap triggers a short jump.
    - Holding Space slightly longer results in a higher jump.
- This makes jumping more responsive and helps the player avoid different obstacle types.

3.2 Obstacles

Two hazardous obstacle types appear:

1. **Spikes**
    - Spawn on the ground.
    - Must be jumped over.
2. **Saw Blades**
    - Spawn in the air.
    - Float toward the player.
    - Can require precise timing to avoid.

3.3 Game Difficulty

- Player speed gradually increases over time.
- Obstacles spawn more frequently as difficulty rises.

3.4 Failure Condition

- Any collision with a spike or saw immediately ends the run.
- Player returns to the start state.

There is no scoring system, and survival time is not displayed.

---

4. Visual & Art Direction

4.1 Art Style

- Pixel-art visual style
- Simple blocky environment graphics
- Bright, clean background tiles
- Classic platformer aesthetic

4.2 Characters

- A small green creature (frog/monster-like)
- Animated idle, run, and jump states

4.3 Environment

- Ground tiles
- Parallax scrolling backgrounds
- Obstacle sprites (spikes, saws)

---

5. Audio

5.1 Sound Effects

The game includes only one sound effect:

- Collision / death sound — played when the player hits a spike or a saw.

No jump sound or additional environmental sounds are implemented.

5.2 Music

- Looping background music plays continuously during gameplay.
- Music is handled through a persistent audio object (DontDestroyOnLoad) to avoid restarting between resets.

---

6. Controls

Action	Key
Start game	Space
Jump	Space
Restart after death	Space

---

7. User Interface

Start Screen

- Text: **"Press Space to start!"**

In-game UI

- None (no score, no HUD)

Game Over Behavior

- Game resets automatically to start state
- "Press Space to start!" message reappears

---

8. Game Flow

Start State

- Game paused
- Background is visible
- Player idle
- UI message instructs user to press Space

Gameplay

- Player runs
- Obstacles spawn
- Difficulty increases
- Collision ends gameplay

Game Over

- Player dies instantly
- Game stops
- All obstacles are removed
- Player respawns at start position

---

9. Technical Specification

Main Systems

---

1. GameManager

The central controller of the game. Responsible for:

- Starting and stopping the game.
- Managing the game state (active or inactive).
- Increasing game speed over time.
- Spawning obstacles (spikes and saw blades).
- Tracking and clearing all active obstacles.
- Resetting gameplay after collision.

This script contains:

- Speed scaling logic

- Obstacle spawn timing logic
  All inside a single manager class.

---

2. PlayerController

Handles:

- Automatic forward movement
- Jumping, including variable jump height (hold-to-jump-higher)
- Collision detection with spikes/saws
- Triggering game over state via GameManager

---

3. AudioManager

A simple audio controller responsible for:

- Playing the background music loop
- Playing the collision sound when the player hits an obstacle
- Ensuring audio persists or resets correctly

This is the only audio system in the game.
There are no jump sounds, UI sounds, saw sounds, etc.

---

4. Obstacle Prefabs

Two prefab types:

- Spikes (ground hazard)
- Saw (air hazard)

Spawned at random intervals and positions according to GameManager logic.

---

10. Scope & Constraints

- Approx. 6 hours of work per team member
- Minimal feature set
- No scoring
- Simple endless mechanics

---

11. Optional Future Improvements

(Not required for this project, but listed for completeness)

- Scoring system
- Main menu
- Game Over screen
- Collectible items
- Character skins
- Additional environments
