Overview

- **Title**: SpeedMaze
- **Genre**: First-Person Maze Puzzle
- **Engine**: Unity

Concept

- The game is a time-based first-person maze challenge. Players must navigate through a series of maze blocks, each containing a green door that acts as a checkpoint. The objective is to solve the maze as quickly as possible while managing a 10-second timer for each block.

Gameplay

- **Starting Point**: The player spawns at the entrance of the maze in Block 0.
- **Objective**: Reach the green door within 10 seconds to proceed to the next block.
- **Failure Condition**: If the timer runs out before reaching the green door, the player respawns at the last checkpoint.
- **Checkpoints**: Each green door serves as a checkpoint for subsequent blocks.
- **Completion**: Escape the maze by completing all blocks.

Difficulty Levels

- **Easy**: 5 blocks
- **Normal**: 10 blocks
- **Hard**: 15 blocks

Timer Mechanics

- Starts at 10 seconds upon entering a new block.
- Resets when the player reaches a checkpoint or respawns.

Maze Structure

- **Blocks**: Pre-designed sections of the maze with distinct layouts and challenges.
- **Generation**: A procedural algorithm assembles the maze using pre-designed blocks. The number of blocks corresponds to the selected difficulty level.

Visual and Audio Design

- **Art Style**: Minimalist with atmospheric lighting.

- **Key Elements**:
    - Green Door: Glows or is otherwise visually distinct to attract attention.
    - Maze Walls: Different textures/themes for variety.
    - Timer: A visible countdown displayed on the HUD.
- **Sound**:
    - Background: Ambient maze sounds.
    - Feedback: Success (door crossing), failure (respawn), and timer warnings.

Core Features

- Procedurally generated maze from pre-designed blocks (not fully sure yet)
- Checkpoint system with respawn mechanics.
- Adjustable difficulty levels.
- First-person view with immersive controls.

Technical Requirements

- **Engine**: Unity
- **Programming Language**: C#
- **Assets**:
    - 3D Models for maze components.
    - UI elements (timer, prompts).
    - Sound effects and music.

Notes Focus on creating a polished, engaging experience within the given timeline. Prioritize core mechanics.