WIGGLE ESCAPE - LEVEL GENERATOR INSTRUCTIONS

1. OVERVIEW

The Level Generator is a powerful tool for creating procedurally generated, GUARANTEED SOLVABLE levels for Wiggle Escape. It uses a simulation-based algorithm that verifies each snake placement won't create an unsolvable puzzle.

Key Features:
- Procedural generation with configurable parameters
- 100% solvable level guarantee (verified through simulation)
- Visual preview in the Unity Editor
- Quick presets for common grid sizes
- Export to ScriptableObject assets for use in-game

2. GETTING STARTED

Access LevelGenerator GameObject from Hierarchy.

You'll see a custom inspector with all the configuration options and generation controls.

3. CONFIGURATION PARAMETERS

GRID CONFIGURATION

- Grid Width: Horizontal size of the level grid
- Grid Height: Vertical size of the level grid
- Quick Presets:
  * 8x8 - Small levels (10 snakes) - Easy
  * 12x12 - Medium levels (20 snakes) - Normal
  * 20x20 - Large levels (40 snakes) - Hard
  * 25x25 - Extra large (50 snakes) - Expert

SNAKE CONFIGURATION

- Target Snakes: Number of snakes to generate
  * The generator will try to place this many snakes
  * May place fewer if grid space runs out

- Min Length is 2: Minimum snake body length
  * Shorter snakes = easier puzzles
  * Recommended: 3 for normal difficulty

- Max Length : Maximum snake body length
  * Longer snakes = more challenging
  * Recommended: 8 for normal difficulty

- Curve Chance (0.0-1.0): Probability of snake body turning
  * 0.0 = Straight snakes only
  * 0.4 = Moderate curves (recommended)
  * 1.0 = Maximum curves (can be very twisty)

SNAKE COLORS

- A list of colors that will be cycled through for snake visualization
- Click "Reset to Default" to restore the original color palette
- Add custom colors for your own visual style

4. USING THE EDITOR INTERFACE

GENERATING A LEVEL
---
1. Configure your desired parameters (or use a preset)
2. Click the green "GENERATE LEVEL" button
3. Wait for generation to complete (progress bar shown)
4. A success dialog will show the results

VIEWING THE PREVIEW

After generation, the inspector displays:

- Statistics:
  * Grid dimensions (e.g., "25 × 25")
  * Number of snakes placed
  * Grid density (percentage of cells occupied)

- Grid Preview (for grids ≤30x30):
  * Visual representation of the generated level
  * Colored squares represent snake segments
  * Arrows (↑↓←→) on snake heads show exit direction
  * Dark squares are empty cells

UNDERSTANDING THE PREVIEW

- Each colored block is a snake segment
- The arrow on a snake's head shows which direction it needs to exit
- All snakes can escape in some order (guaranteed by simulation)

5. SAVING GENERATED LEVELS

Once you've generated a level you like:

1. Click the blue "SAVE AS ASSET" button

2. Choose a save location (default: Assets/ScriptableObjects/Levels/)
3. Name your level file
4. Click "Save"

The level is saved as a LevelData ScriptableObject containing:
- Grid width and height
- All snake data (positions, colors, exit directions)

USING SAVED LEVELS IN YOUR GAME
---
1. Reference the LevelData asset in your game manager from GameManager GameObject from hierarchy.
2. Access snake data via:
   - levelData.gridWidth / levelData.gridHeight
   - levelData.snakes (List<SnakeData>)
   - Each snake has: color, segments, exitDirection

6. UNDERSTANDING THE ALGORITHM

GENERATION PROCESS
---
1. RANDOM PLACEMENT: Selects random empty cells as potential snake heads
2. RANDOM WALK: Builds snake bodies using random walk algorithm
3. EARLY ELIMINATION: Rejects snakes that would face existing snakes
   (two snakes facing each other = guaranteed deadlock)
4. SIMULATION VERIFICATION: After placing each snake, simulates the entire
   level to ensure it's still solvable

SOLVABILITY GUARANTEE

The simulation works by:
1. Finding a snake with a clear path to exit
2. "Removing" that snake from the simulation
3. Repeating until all snakes can escape
4. If any snake gets stuck, the placement is rejected

This ensures EVERY generated level is 100% solvable.

7. TROUBLESHOOTING

"Generation Failed" Message

- Check Console for detailed error messages
- Try reducing snake count or increasing grid size
- Reduce max snake length if grid is too crowded

Fewer Snakes Than Requested

- Normal behavior when grid space is limited
- The generator stops when grid is full
- Increase grid size or reduce target count

Generation Is Slow

- Large grids (25x25+) with many snakes take longer
- Simulation verification adds processing time
- Normal: 1-5 seconds for most configurations

Console Shows "Level is NOT solvable"

- This should rarely happen
- Try regenerating with different parameters
- Report if this occurs consistently

Preview Not Showing

- Only displays for grids ≤30x30
- Generate a smaller level to see preview
- Check that generation succeeded