Room-Based Procedural Level Builder v1.0

Documentation PDF – by domdgn

Table of Contents

---

1. Introduction

Welcome, and thank you for purchasing **Room-Based Procedural Level Builder v1.0**! This package allows developers to generate modular dungeon-like levels at runtime using customizable prefab sets and generation rules.

---

2. Requirements

- Unity 2022.3.47f1 or higher
- Basic understanding of prefabs, colliders, and ScriptableObjects

---

3. Getting Started

3.1 Importing the Asset

1. Import the package via Unity Package Manager or drag it into your project.

2. Let Unity compile all scripts before interacting with the system.

3.2 Demo Scene Overview

- Navigate to the Scenes folder and open Demo Scene.unity.
- Press **Play**, then press the **E** key to generate a random level using the default settings.

3.3 Generating a Level

- In Play mode, pressing **E** triggers the level generation system.
- A new layout is constructed using the default Level ScriptableObject configuration.

---

4. Customizing Levels

4.1 Creating a New Level

1. In the Project window, right-click and go to Create > DungeonGenerator > Level Data.
2. Assign your own room and hallway prefab sets.
3. Adjust generation settings as needed (max rooms, hallway chance, etc.).

4.2 Setting Up Room and Hallway Prefabs

- Each room must be a prefab with:
  - A **parent empty GameObject**
  - **Trigger Collider(s)** that cover the entire room
  - The GameObject must be on the **"Rooms"** layer **(BUT NOT CHILDREN)**
- All **entry point empties** (used to connect rooms) must have their **local Z axis *(blue arrow)* facing outward**.

---

5. Setup Rules

- Ensure **all prefabs** used in a level are tagged and layered correctly.

---

6. Script Reference

DungeonGenerator.cs

Description: Main controller for level generation.
Key Methods:

- ResetDungeon() – Removes current dungeon and begins generating a dungeon layout using a random level data.
- ClearLevel() – Destroys all generated rooms and resets the Start entries.

LevelSO.cs

Description: ScriptableObject that stores generation data and prefab sets.
Fields:

- List<GameObject> roomPrefabs
- List<GameObject> hallwayPrefabs
- int totalRooms
- float hallwayChance
- float subsequentialHallwayChanceMultiplier

EntryPoint.cs

Description: Attached to entry point empties. Stores data about room connectivity and used status.

---

7. Troubleshooting

Issue: Rooms not connecting

- Ensure entry points face outward (Z+ axis)
- Confirm prefab colliders are properly set to trigger and on "Rooms" layer

Issue: Nothing spawns

- Check LevelSO has valid prefab lists
- Ensure you are calling ResetDungeon() properly

---

8. Contact

For support, feedback, or collaboration:
Discord: budgied
I'm happy to help with any setup issues or to hear your suggestions!