**PCG Level Master**

**Hybrid PCG Solution for Game Developers**

*Version 1.0.0*

*Copyright © 2025 Dineshkumar, Mahmud Hasan, Kevin A. Moberly, & Kamalanathan*  
*All rights reserved.*

**Table of Contents**

* [Introduction](https://claude.ai/chat/47d6dea5-507a-475b-bece-525221ba00fc#introduction)
* [Installation](https://claude.ai/chat/47d6dea5-507a-475b-bece-525221ba00fc#installation)
* [Getting Started](https://claude.ai/chat/47d6dea5-507a-475b-bece-525221ba00fc#getting-started)
* [Generation Methods](https://claude.ai/chat/47d6dea5-507a-475b-bece-525221ba00fc#generation-methods)
  + [Fully Procedural](https://claude.ai/chat/47d6dea5-507a-475b-bece-525221ba00fc#fully-procedural)
  + [Hybrid Procedural](https://claude.ai/chat/47d6dea5-507a-475b-bece-525221ba00fc#hybrid-procedural)
  + [User Defined Layout](https://claude.ai/chat/47d6dea5-507a-475b-bece-525221ba00fc#user-defined-layout)
* [Tilemap Integration](https://claude.ai/chat/47d6dea5-507a-475b-bece-525221ba00fc#tilemap-integration)
* [Entity Placement](https://claude.ai/chat/47d6dea5-507a-475b-bece-525221ba00fc#entity-placement)
* [Parameter Reference](https://claude.ai/chat/47d6dea5-507a-475b-bece-525221ba00fc#parameter-reference)
* [Troubleshooting](https://claude.ai/chat/47d6dea5-507a-475b-bece-525221ba00fc#troubleshooting)
* [Support](https://claude.ai/chat/47d6dea5-507a-475b-bece-525221ba00fc#support)

**Introduction**

PCG Level Master is a professional procedural level generation system for Unity that combines algorithmic generation with template-based design. This hybrid approach provides unlimited level variety while maintaining complete creative control over the results.

The system uses Binary Space Partitioning (BSP) as its foundation, enhanced with custom room templates, L-shaped corridor connections, and advanced tile placement algorithms. All generated levels are output directly to Unity's built-in Tilemap system, making integration with your existing game seamless.

*Fig 1: Level generated in the Unity scene view using PCG Level Master*

**Key Features**

* **Three Generation Methods:** Choose from Fully Procedural, Hybrid Procedural, or User Defined Layout to match your game's needs.
* **BSP Algorithm:** Industry-standard Binary Space Partitioning creates balanced, naturally flowing level layouts.
* **Custom Room Templates:** Design your own room templates and integrate them with procedural generation.
* **Visual Designer:** Drag and drop interface for manually designing level layouts with precise control.
* **Entity Placement:** Automatic player, enemy, and decoration placement based on customizable rules.
* **Full Source Code:** Complete C# source code included for maximum customization and extension.

**Installation**

**Option A: Import from Unity Asset Store (Recommended)**

1. Purchase PCG Level Master from the Unity Asset Store.
2. In Unity, go to Window → Package Manager.
3. Select "My Assets" from the dropdown menu.
4. Find PCG Level Master and click Download.
5. Once downloaded, click Import.
6. In the import window, ensure all components are selected and click Import.

**Option B: Manual Installation**

1. Download the PCG Level Master package file (.unitypackage).
2. In your Unity project, go to Assets → Import Package → Custom Package...
3. Navigate to the downloaded package file and select it.
4. Ensure all components are selected and click Import.

**Note:** After importing, you can find example scenes in the Assets > PCGLevelGenerator > Examples folder.

**Getting Started**

Get up and running with PCG Level Master in minutes:

**Quick Start**

1. **Open an Example Scene or Set Up Your Own**  
   Navigate to Assets > PCGLevelGenerator > Examples and open one of the example scenes, or follow the "Setting Up a New Scene" section below.
2. **Select the HybridLevelGenerator GameObject**  
   Find and select the HybridLevelGenerator GameObject in the Hierarchy.
3. **Initialize Assets**  
   In the Inspector, click the **Initialize Assets** button. This will set up the necessary references for generation.
4. **Generate Level**  
   Click the **Generate Level** button to create a game level. You'll see a complete procedural level appear in the scene view!
5. **Explore and Experiment**  
   You can click the Generate Level button again to create new variations, or adjust parameters to customize the generation.
6. **Play the Scene**  
   Click the Play button in Unity to experience the generated level.

*Fig 2: PCG Level Master inspector showing the Initialize Assets and Generate Level buttons*

**Setting Up a New Scene**

1. Create a new Unity scene or use an existing one.
2. Create a Grid GameObject: GameObject → 2D Object → Grid
3. Add two Tilemap children to the Grid:
   * Right-click on Grid → 2D Object → Tilemap (name it "GroundTilemap")
   * Right-click on Grid → 2D Object → Tilemap (name it "WallTilemap")
   * Add a Tilemap Collider 2D component to the WallTilemap
4. Create a new empty GameObject and name it "LevelGenerator"
5. Add the HybridLevelGenerator component to the LevelGenerator GameObject: Add Component → PCG Level Master → Hybrid Level Generator
6. Click the Initialize Assets button, then Generate Level

**Generation Methods**

PCG Level Master offers three distinct generation methods, each providing different levels of control and procedural variety. You can select the generation method from the dropdown in the Inspector.

**Fully Procedural**

Fully Procedural generation uses Binary Space Partitioning (BSP) to create classic dungeon layouts with rectangular rooms and corridors. This method offers the highest level of procedural variety but less control over specific room layouts.

*Fig 3: Fully Procedural settings in the inspector of HybridLevelGenerator GameObject*

**Key Features**

* Pure BSP-based generation algorithm
* Rectangular rooms with configurable size parameters
* Automatic L-shaped corridor connections using Minimum Spanning Tree (MST)
* Complete randomization (with seed control)
* Fastest generation speed

**Best For:** Roguelike games, infinite dungeons, or any application where variety and replayability are more important than specific room designs.

**Parameters**

The Fully Procedural generation method includes the following parameter groups:

**Level Dimensions & Seed**

* **Level Width:** Width of the level grid in tiles (default: 100)
* **Level Height:** Height of the level grid in tiles (default: 100)
* **Use Random Seed:** When enabled, generates a new random seed each time
* **Seed Value:** The specific seed value to use when "Use Random Seed" is disabled
* **New** button: Generates a new random seed value

*Fig 4: Tilemap integration settings for fully procedural generation*

**BSP Algorithm Settings**

* **Min Room Size:** Minimum width/height for rooms in tiles (default: 8)
* **BSP Iterations:** Number of BSP splits (controls room count) (default: 5)
* **Room Padding:** Minimum space between room edge and partition (default: 2)

**Corridor Settings**

* **Corridor Width:** Width of corridors in tiles (default: 2)

*Fig 5: Entity placement settings for fully procedural generation*

**Algorithm Details**

The Fully Procedural method implements the following algorithm:

1. Start with the entire level space as a single partition
2. Recursively split partitions either horizontally or vertically
3. Split direction is determined by partition aspect ratio
4. Stop splitting when minimum size is reached or max iterations is hit
5. Create rooms within the resulting leaf partitions
6. Connect rooms with L-shaped corridors using Minimum Spanning Tree (MST) algorithm
7. Apply tiles to Unity Tilemaps
8. Spawn entities (player, enemies, decorations) if specified

**Examples & Tips**

**Creating More Rooms**

To increase the number of rooms in your level:

* Increase the **BSP Iterations** value (try 6-7)
* Decrease the **Min Room Size** value (try 6-7)
* Increase level dimensions for more space

**Creating Larger Rooms**

For larger, more spacious rooms:

* Increase the **Min Room Size** value (try 10-12)
* Decrease **BSP Iterations** for fewer, larger spaces
* Reduce **Room Padding** to allow rooms to fill more of their partition

**Creating Consistent Layouts**

To create reproducible level designs:

* Disable **Use Random Seed**
* Set a specific **Seed Value**
* Keep all other parameters the same
* The same seed will always produce the same layout with the same parameters

**Hybrid Procedural**

Hybrid Procedural generation combines BSP-based level structure with custom room templates. This method offers a balance between procedural variety and designer control, allowing for specialized room designs within a procedurally generated framework.

*Fig 6: Hybrid Procedural generation method in the inspector*

**Key Features**

* Combination of BSP algorithm and pre-designed templates
* Support for both rectangular and L-shaped rooms
* Template probability control
* MST corridor connections for optimal pathways
* Mix of procedural and hand-crafted elements

**Best For:** Games that need some specialized rooms (boss rooms, treasure rooms, puzzles) within an otherwise procedural environment.

*Fig 7: Hybrid Procedural settings showing room templates and probabilities*

**Parameters**

The Hybrid Procedural generation method includes all the parameters from Fully Procedural plus:

**Hybrid Room Settings**

* **L-Shape Chance:** Probability of generating L-shaped rooms instead of rectangles (0-1)
* **Template Chance:** Probability of using a room template instead of procedural generation (0-1)

**L-Shape Leg Ratios**

* **Min Leg Ratio:** Minimum ratio of the smaller leg to the larger leg in L-shaped rooms
* **Max Leg Ratio:** Maximum ratio of the smaller leg to the larger leg in L-shaped rooms

**Room Templates List**

The list of room template prefabs that can be used in generation. Each template has a green indicator showing its validity.

* Click the "+" button to add a new template
* Click the "x" button to remove a template
* Drag and drop template prefabs from your project

*Fig 8: Room Templates from Assets > PCGLevelGenerator > Prefabs > Room Templates*

**Creating Custom Room Templates**

You can create your own room templates to be used in the Hybrid Procedural generation:

*Fig 9: Creating custom room prefabs using the Tile Palette*

1. Create a new Unity scene for template design.
2. Add a Grid GameObject with a Tilemap child.
3. Design your room layout using the Tile Palette. Make sure your room has clear entrance/exit points for corridors.
4. Drag the Tilemap to your Project window to create a prefab, or use the Create Prefab option.
5. Add your prefab to the Room Templates List in the HybridLevelGenerator component.

**Note:** Room templates should follow a consistent size convention for best results. The system will attempt to scale templates to fit the available space, but designs work best when they're relatively similar in size.

**Examples & Tips**

**Creating a Boss Room**

To create a special boss room that appears in your levels:

* Design a larger room template with distinctive features
* Add it to the Room Templates List
* Set **Template Chance** high enough to ensure templates appear (0.5+)
* The boss room will be integrated into the level along with other rooms

**Balancing Template Usage**

To control how often templates appear versus procedural rooms:

* Adjust the **Template Chance** value:
  + 0.0 = No templates used (pure procedural)
  + 0.5 = About half the rooms will be templates
  + 1.0 = All rooms will attempt to use templates

**Creating L-Shaped Rooms**

To generate interesting L-shaped rooms alongside rectangles:

* Increase **L-Shape Chance** to 0.3-0.5
* Adjust **Min Leg Ratio** and **Max Leg Ratio** to control the shape proportions
* L-shaped rooms add visual variety and interesting spaces

**User Defined Layout**

User Defined Layout provides complete control over level design while still leveraging the powerful tile placement and entity spawning systems. This method allows you to manually place and connect rooms for maximum design control.

*Fig 10: User Defined Layout settings in the inspector*

**Key Features**

* Visual node-based layout designer
* Manual room placement and connection
* Room property configuration
* Support for both custom templates and procedural room generation
* Dynamic corridor generation between manually placed rooms

**Best For:** Specific level designs for story missions, tutorials, or any scenario where precise control over the layout is required.

**Using the Visual Level Designer**

The User Defined Layout method revolves around the Visual Level Designer window:

*Fig 11: Visual Level Designer window with room creation options*

1. Click the "Open Visual Level Designer" button in the Inspector.
2. In the designer window, right-click on the canvas to create a room. Choose from "Create Rect Room", "Create L-Shape Room", or "Create Template Room".
3. Configure room properties using the panel on the right. You can set size, type, and other parameters.
4. Create additional rooms as needed for your level. You can move rooms by dragging them.

**Connecting Rooms**

Once you've created rooms, you need to connect them to form a cohesive level:

*Fig 12: Connecting rooms in the Visual Level Designer*

1. Click the "Connect" button on one room node.
2. Click on a target room to create a connection between the two rooms.
3. Repeat to connect all rooms in your layout. Each room should connect to at least one other room.
4. Right-click on connections to configure corridor properties if needed.

**Warning:** All rooms must be connected to at least one other room for the level to generate properly. Isolated rooms will be ignored.

1. Click the "Create Scene Objects & Generate" button at the bottom of the Visual Level Designer.
2. Confirm the operation in the dialog that appears, then close the Visual Level Designer window.
3. The level will be generated according to your design and visible in the scene view.

*Fig 13: The level created based on the User Defined Layout*

**Examples & Tips**

**Creating a Linear Level**

To create a level with a specific progression path:

* Place rooms in a sequence (start room, middle rooms, end room)
* Connect them in order to create a main path
* Add optional side rooms connecting to the main path
* Designate the starting room using the properties panel

**Using Keyboard Shortcuts**

Speed up your workflow with these shortcuts:

* **F key:** Center view on all nodes
* **O key:** Focus on origin (0,0)
* **Middle-click drag:** Pan the view
* **Mouse wheel:** Zoom in/out

**Saving and Loading Layouts**

Save your work for later editing:

* Use the "Save" button to store your current layout
* Use "Open" to load a previously saved layout
* Layouts are saved as assets in your project

**Tilemap Integration**

PCG Level Master integrates seamlessly with Unity's built-in Tilemap system:

*Fig 14: Floor and Wall Tile assets from the Project window*

*Fig 15: Directional Wall Tiles from the Project window*

**Required Tilemaps**

* **Ground Tilemap:** For floor tiles
* **Wall Tilemap:** For wall tiles

**Tile Types**

The system works with all Unity tile types:

* **Regular Tiles:** Basic sprites
* **Rule Tiles:** Auto-tiling sprites that change based on neighbors
* **Animated Tiles:** Tiles with animation frames
* **Random Tiles:** Randomly selected variants

**Directional Wall System**

When "Use Directional Walls" is enabled, you can assign specific wall tiles for different orientations:

* **Basic Wall Directions:** Top, Bottom, Left, and Right wall tiles
* **Inner Corner Tiles:** Inner Top-Left, Inner Top-Right, Inner Bottom-Left, Inner Bottom-Right
* **Outer Corner Tiles:** Outer Top-Left, Outer Top-Right, Outer Bottom-Left, Outer Bottom-Right

**Entity Placement**

PCG Level Master includes a system for automatically placing entities in generated levels:

*Fig 16: Entity placement settings in the inspector*

**Player Placement**

* Player is placed in one of the rooms (typically the first room)
* Positioning ensures the player starts on a valid floor tile
* Can be customized to start in specific room types in Hybrid or User Defined modes

**Enemy Placement**

* Enemies are distributed throughout rooms based on the Max Enemies/Room setting
* Placement algorithm ensures enemies don't overlap or block pathways
* Enemy density can be customized per room in User Defined Layout

**Decoration Placement**

* Decorations are placed along walls or in open floor areas depending on the prefab
* Controlled by the Max Decors/Room setting
* Can be customized per room in User Defined Layout

**Parameter Reference**

This section provides a complete reference of all parameters available in PCG Level Master.

**Level Dimensions & Seed**

| **Parameter** | **Description** | **Default** | **Range** |
| --- | --- | --- | --- |
| Level Width | Width of the level grid in tiles | 100 | 20-500 |
| Level Height | Height of the level grid in tiles | 100 | 20-500 |
| Use Random Seed | Generate a new random seed each time | Checked | Boolean |
| Seed Value | The specific seed value to use | Random | Integer |

**BSP Algorithm Settings**

| **Parameter** | **Description** | **Default** | **Range** |
| --- | --- | --- | --- |
| Min Room Size | Minimum width/height for rooms | 8 | 4-20 |
| BSP Iterations | Number of BSP splits (controls room count) | 5 | 1-10 |
| Room Padding | Minimum space between room edge and partition | 2 | 0-10 |

**Hybrid Room Settings**

| **Parameter** | **Description** | **Default** | **Range** |
| --- | --- | --- | --- |
| L-Shape Chance | Probability of generating L-shaped rooms | 0.048 | 0-1 |
| Template Chance | Probability of using a room template | 0.844 | 0-1 |
| Min Leg Ratio | Minimum ratio for L-shaped room legs | 0.4 | 0.1-0.9 |
| Max Leg Ratio | Maximum ratio for L-shaped room legs | 0.7 | 0.1-0.9 |

**Corridor Settings**

| **Parameter** | **Description** | **Default** | **Range** |
| --- | --- | --- | --- |
| Corridor Width | Width of corridors in tiles | 2 | 1-5 |

**Entities & Decorations**

| **Parameter** | **Description** | **Default** | **Range** |
| --- | --- | --- | --- |
| Max Enemies/Room | Maximum number of enemies per room | 2 | 0-10 |
| Max Decors/Room | Maximum number of decorations per room | 3 | 0-20 |

**Troubleshooting**

**Common Issues**

**No Level Generated After Clicking "Generate Level"**

**Issue:** Clicking "Generate Level" doesn't produce any visible results.

**Solution:**

* Check that you clicked "Initialize Assets" first
* Verify that Ground Tilemap and Wall Tilemap are properly assigned
* Ensure Floor Tile and Wall Tile references are set
* Check the Console for any error messages

**Rooms Are Too Small or Too Large**

**Issue:** Generated rooms don't match desired size.

**Solution:**

* Adjust Min Room Size parameter (higher for larger rooms)
* Modify BSP Iterations (lower for larger rooms)
* Change Room Padding (lower for larger rooms)

**Corridors Are Missing or Disconnected**

**Issue:** Some rooms appear disconnected without corridors.

**Solution:**

* Ensure Corridor Width is at least 1
* In User Defined Layout, verify all rooms are connected in the Visual Designer
* Try regenerating with a different seed

**Room Templates Not Appearing in Hybrid Mode**

**Issue:** Custom room templates aren't being used in generation.

**Solution:**

* Check that templates are properly added to the Room Templates List
* Verify Template Chance is set high enough (0.5 or higher)
* Ensure templates have valid entrance/exit points

**Performance Considerations**

* Very large levels (>200x200) may take longer to generate
* High BSP Iteration values (>7) will create more rooms and may impact performance
* Consider using Rule Tiles for walls instead of individual directional tiles for better performance
* For runtime generation, consider generating levels in a background thread

**Support**

If you encounter any issues or have questions not covered in this documentation, please reach out:

**Contact Information**

* **Email:** [support@pcglevelmaster.com](mailto:support@pcglevelmaster.com)
* **Discord:** Join our community at [PCG Level Master Discord](https://discord.gg/pcglevelmaster)
* **Asset Store Comments:** Leave a comment on our Unity Asset Store page

**Note:** This documentation can be found in your project at:  
Assets/PCGLevelGenerator/Documentation/PCGLevelMaster-Documentation.docx

**Updating PCG Level Master**

We regularly release updates with new features, improvements, and bug fixes. To ensure you have the latest version:

* Check the Unity Asset Store for updates
* Import any new versions using the Package Manager
* Review the changelog for details on what's changed

**Feature Requests**

Have an idea for improving PCG Level Master? We'd love to hear it! Send your feature requests to:

* Email: [features@pcglevelmaster.com](mailto:features@pcglevelmaster.com)
* Discord: #feature-requests channel

*PCG Level Master*  
*Hybrid PCG Solution for Game Developers*  
*Version 1.0.0*

*Copyright © 2025 Dineshkumar, Mahmud Hasan, Kevin A. Moberly, & Kamalanathan*  
*All rights reserved.*