ProD Gold 3.1 Manual

What is ProD?

ProD supplies you with several scripts that help you create procedurally generated environments using a two dimensional grid system. The package is designed with roguelike development in mind and has the means to support both retro-style dungeon crawlers and contemporary roguelikes together.

How to use this Manual?

Use 1: If you want to see the **online code documentation** go to this link or to "" in your folder. We used **Doxygen**, a free documentation tool to make proper documentation possible for you.
Use 2: Please, continue reading for your other documentation needs. You will find a quickstart, an in depth explanation of how ProD works, a FAQ, Contact Info and an Update Log.

QuickStart

ProD contains three pre-made test scenes in the folder "**ProD/Scenes**". These scenes serve as examples on how to quickly generate several types of worlds, maps and dungeons.

We highly encourage you to test all three scenes, they will take you a minute each**.**

Scene 1 - Generator Dungeon Parameters: This scene encourages you to generate Dungeons with various parameters. The GUI will allow you tweak multiple parameters and helps you see what ProD is capable of. After you play with this scene feel free to open up ProDManager.cs and Generator_Dungeon.cs to see what's inside.

Scene 2 - Other Map Generator Samples: This scene will show you a two by two example of various map types. The GUI calls on different map types for each button. You may find the scripts for these map types in the "**ProD/Scripts/Map Generators**" folder.

Scene 3 - Player Movement & Camera Tracking: This scene will appear to do the same thing as Scene 2 but will additionally give you a Player to control through numpad/cursor keys. This is an example of how the player, player tracking and minimap functionalities work.

How does ProD work?

1. **Cells, Maps, WorldMaps**
    a. **Worldmaps are made out of maps.** For example a worldmap can be, "the Ancient Dungeon Ruins", and its maps can be "Dungeon Ruins - Entrance",

"Dungeon Ruins - Maze" and "Dungeon Ruins - Lich's Lair".

b. **Maps are made out of cells**. For example "Dungeon Ruins - Maze" will have Wall, Path, Trap, Entrance and Exit as some of its cells.

2. **Generators and the MethodLibrary.cs**
    a. **WorldMap Generators use Map Generators.** For example "the Ancient Dungeon Ruins" can use Generator_DungeonRuins.cs and Generator_Maze to make its individual maps.
    b. **Map Generators always use MethodLibrary.cs methods**: The MethodLibrary.cs consists all fundamental methods in making maps. For example Generator_DungeonRuins.cs uses the following and more:
        i. MethodLibrary.CreateRooms(...);
        ii. MethodLibrary.CreateMaze(...);
        iii. MethodLibrary.SetCellsOfTypeAToB(...);
        iv. MethodLibrary.CloseDeadEndCells(...);
        v. MethodLibrary.ReduceUCorridors(...);
        vi. MethodLibrary.ConvertUnreachableCells(...);

3. **Materializer.cs**
    a. **Materializer.cs uses the theme of a map and the resources folder to instantiate game objects.** If a map has its theme as "Terminal Theme", then the Materializer.cs will look into **"ProD\Visual Assets\Resources\Terminal Theme\Cells"** and search for respective Prefabs such as Wall, Path, Trap, Entrance and Exit. Please look into the necessary naming conventin by navigating to aforementioned folder. You will find a wall prefab in the terminal theme at **"ProD\Visual Assets\Resources\Terminal Theme\Cells\Wall\PRE_Wall_Default"**

4. **Further Clarification:**
    a. Every test scene uses the **ProDManager.cs** and the **Materializer.cs**. This shows a clear **distinction between generating the dungeon and actually showing it.** The GUI in the test scenes work in two steps. First calling a world map generation via: Generator_Generic_World.Generate(). Then calling a prefab placement via: Materializer.Instance.MaterializeWorldMap()
    b. Inside a World Generator, for example Generator_Generic_World.cs, a world is created which can consist of **multiple maps.** Each map is created using a Map Generator, for example Generator_Dungeon.cs.
    c. These generators are static, so they don't need to be instantiated. They also serve as examples for you, on how to make your own generators. Each of them use methods from the static **MethodLibrary.cs**.
    d. MethodLibrary.cs contains all methods for creating rooms, caverns, doors etc. You should see these methods as building blocks and we encourage you to experiment with them and write your own map generator scripts. All the methods

of the method library are carefully commented and documented in our online documentation here.

F.A.Q.

1. **Can I use 3D models with ProD?**
   Yes you can. Check out how our Materializer.cs works.

2. **Does ProD generate maps in Play Mode or before I press play, in the Editor Mode?**
   a. **ProD can do both**, **however**, the test scene in the package will only let you generate maps in the play mode, that is after you press Play icon in Unity.

   b. ProD works in two steps. First step is generation and second step is materialization. You are capable of generating maps without viewing them.

   c. If you want to generate maps on the Editor Mode **you have to write your own Editor script** to do so and it is possible since we've done it on earlier versions.

3. **I can't get pixel-perfect textures in my game. How do I resolve this issue? (Pre 4.3)**
   a. Your texture size, your object's scale and your resolution need to be in sync for getting pixel-perfect images in Unity3D. Follow these steps:
      i. Switch your camera to Orthographic. This will display everything in 2-D.
      ii. Go to import settings of your textures and raise their compression to a higher resolution. Your texture's Filter Type should be Point and its Format should be highest possible.
      iii. Power of 2 is your friend. You get warped pixels when they need to split into half. For example I use a 32x32 pixel PNG file that is attached to a prefab with scale of X:32 Y:1 Z:32. I display this in a resolution of 512x512 using 512 as my camera size.
      iv. Go to Unity>Edit>Project Settings>Quality and Disable Anti Aliasing option.
      v. If these don't help you then google and unity forums are your friend.

4. **What's your e-mail?**
   Unity3D@graylakestudios.com

5. **Is there a community forum where I can make posts and communicate with the other users?**
   Yes! We finally have a forum. We are using reddit.com and our community subreddit is located at r/ProD.

6. **I e-mailed you but you didn’t reply. What's up with that?**
    a. If for whatever reason you’re not getting answers, we highly recommend you try the community forum subreddit at r/ProD.
    b. I usually reply immediately. I might be very busy due to the fact that I’m working at my own start-up. Feel free to e-mail me again in case I missed it.
    c. Also we don’t receive notifications for reviews you’ve made in the asset store. This means if you complain about something we may not see it until later. I kindly advise you to e-mail us your issues since we want to help you asap.

7. **Do you guys have a video tutorial or other media teaching Prod?**
   We will be making videos very soon. Check for video links on our blog.

I want to contact you!

You may contact us via Unity3D@graylakestudios.com for **support**. Please send us your **feedback**, **complaint**, **praise** or **questions** at aforementioned address.. We are an independent development studio and we hope to reply to you as soon as we can, however, given the circumstances we may be very busy and reply late depending on our schedule. ProD is being developed slowly and you may follow the latest development news on our blog.

Update Log for ProD Gold 3.1

1. Changed the name of the package from ProD to ProD Gold.
2. Added the community information to the manual.