Basic 2D Platformer:

Setup:

Start by creating a basic 2D platformer game in Unity. Set up a player character that can move left and right, jump, and one way to interact with the environment. Create a simple level manually to test the player's movement and interaction.

Design the Level Generation System:

Next, design the system that will procedurally generate the levels. Decide on the rules and parameters that will govern the level's layout. For example, you could create rules for platform placement, enemy spawning, and item placement.

Implement Procedural Level Generation:

Begin implementing the procedural level generation using Unity's scripting capabilities. You can use Unity's built-in components such as GameObjects and Prefabs to represent different types of platforms, enemies, and items. Write scripts that will create and position these elements randomly based on the rules you defined.

Add Variety and Challenge:

To keep the levels interesting and challenging, incorporate 3 different level themes and obstacles. You could have different sets of rules for level generation that correspond to different themes. For instance, a jungle-themed level might have more vines and platforms, while a cave-themed level could have more spikes and traps.

Create a Level Manager:

Build a level manager script that keeps track of the player's progress through the game and generates new levels as they complete the previous ones. The level manager should also handle player respawn and checkpoint systems.

Designer Friendly:

The system needs to be highly configurable so that a designer can tweak the PCG settings to meet their requirements.

Test and Iterate:

Test your game extensively to ensure that the procedural generation is creating playable and enjoyable levels. The player must be able to complete the level. Adjust the rules and parameters as needed to improve the gameplay experience.

THE RULES:

- Use the Unity Engine 2023…
- You are able to use any free art assets on the unity store.
- You may not use any Asset Store code libraries etc. or libraries from anywhere else.

YOU ARE NOT REQUIERD TO:

- Have enemies and enemy AI
- Produce you own art assets.
- Have sound.
- To implement any specific design patterns.

WHAT YOU WILL BE ASSESD ON:

- 10% Code Quality (Neatness, Organisation, Simplicity)
- 10% Performance and Optimization Techniques
- 25% Configurability
- 25% Designer Friendly/Proof
- 30% Overall Execution and Features/Functionality

YOU WILL NOT BE ASSESESD ON:

- Art Assets

Don't hesitate to seek help from myself, online tutorials, forums, and the Unity community if you encounter any roadblocks during the development process. Happy coding!