CGS Brief – Procedural Dungeon Generator

Procedural Dungeon Generator

Create a 2D or 3D Procedural Dungeon Generator where rooms and hallways can be specified by the developer, using editor windows and inspector tools to assign them, these rooms can then be overwritten from a grey box to final assets by the developer. The rooms they create can all be assigned to spawn pick ups, or entities.

Proof-of-Concept Application

What to think about when creating the application:

- There needs to be at least 3 different rooms sizes created.
- There needs to be the ability to create hallways based on an algorithm to determine the next room it connects to.
- Rooms can connect directly to other rooms.
- Rooms do not overlap, making parts inaccessible.
- The rooms must be generated in a grid system.
- Dungeon layouts can be saved in editor.
- Dungeon layouts can be create in run time.
- Assets for walls floors and ceilings should be interchangeable from specific managers.
- Pickups (Coins, Keys, Bombs, Power-ups, etc) can be assigned to spawn in a room in specific or random locations.
- NPCs (Allies or Enemies) can be assigned to spawn in a room in specific or random locations.

Extension

How can we take this further to showcase a skillset in a portfolio piece:

- For a 3D dungeon, dungeons can be created on all axis.
- Use multithreading to improve the speed when generating dungeons.
- Create a boss room and encounter that is at the end of the dungeons.
- Create a combat system that is engaging and allows the player to interact with pickups and enemies in a meaningful way.
- Create an endless mode where the dungeon will continue to generate new rooms and tougher monsters, tracking the players progress by rooms cleared.