Infinity Dungeon Generator

The walls must be installed to build a dungeon. At least 1 wall element.

public GameObject[] **walls**; - walls prefab
public GameObject[] **ground**; - ground prefab (Optional if you use terrain as a ground, other ways please set ground with prefab)

These elements are optional:

public GameObject[] **environmentToWall**; - objects are located close to walls, like fireplaces and bookcases. *Optional*
public GameObject[] **enemies**; *Optional*
public GameObject[] **lamps**; - these elements are located close to walls, use preferably lamps and torches. *Optional*
public GameObject[] **columns**; - divide the walls. *Optional*
public GameObject[] **furniture**; - different objects, environments, loot and so on. *Optional*.
public GameObject[] **roofs**; - perfect for ego shooter video games. Set roofs prefab. *Optional*

public GameObject **player**; - set the player.

public int transformY = 0; - dungeon starting position
public int groundTransZ = 0; - dungeon starting position
public int groundTransX = 0; - dungeon starting position

each room will be randomly generated between the minimum and maximum size.

public int roomSizeIlow = 4; - min room size on Z achse
public int roomSizeImax = 10; - max room size on Z achse
public int roomSizeJlow = 4; - min room size on Z achse
public int roomSizeJmax = 10; - max room size on Z achse
public int maxDungeonSize = 15; - the maximum size of the dungeon means the number of rooms in the dungeon. The minimum value is always 5.

public int enemyRange = 10; - 1/10 chance to call the enemy to an accessible floor position. For example, if the enemy radius of action is 2, then the probability that there is an enemy on the floor is ½.

public int furnitureRange = 2; - ½ chance to create furniture on each floor, except the floors close to the walls.

public int lampsRange = 2; - ½ chance to create lamp.
public int wallEnvRange = 2; - ½ chance to create environment close to the wall.
public int columnRange = 2; - ½ chance to create columns between the walls. select 1 if you want to create columns between each wall. This makes sense for many scenarios.

public bool hasRoof = **false**; - optional if you need a roof in your dungeon.

public int playerSpawnPosY = **1**; - player spawn coordinate achse y
public int playerSpawnPosX = **1**; - player spawn coordinate achse x
public int playerSpawnPosZ = **1**; - player spawn coordinate achse z

public float sizeOfFloat = 4; - width and height of wall, floor and roof.
public int sizeOfInt = 4; - width and height of wall, floor and roof.
Note that the value of these 2 arguments must be equal.

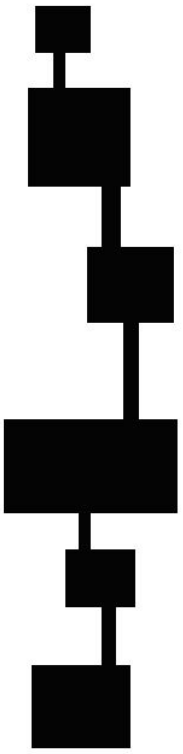
Dungeons:

Infinity Dungeon - if a player passes a corridor, the infinity dungeon creates a new room and the next corridor, the previous room and corridor disappear.

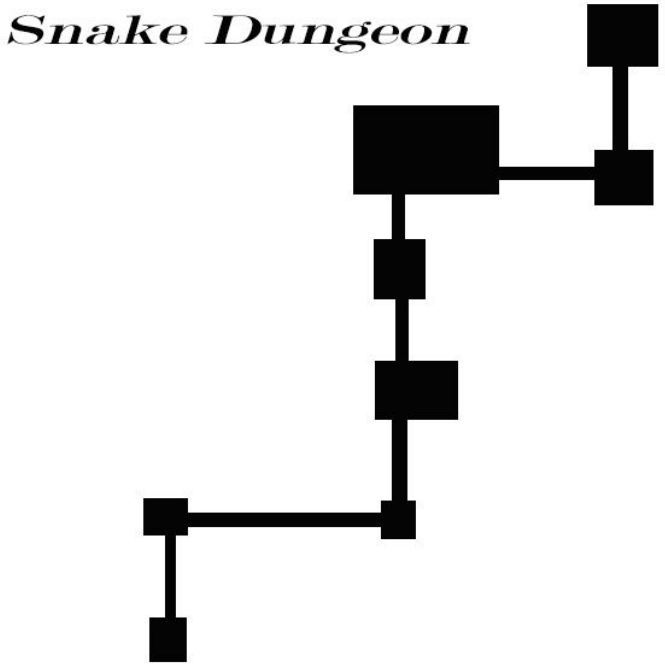
Clean Dungeon - clean all objects from generated dungeon.

Generate Line Dungeon - Generate simple one way dungeon.

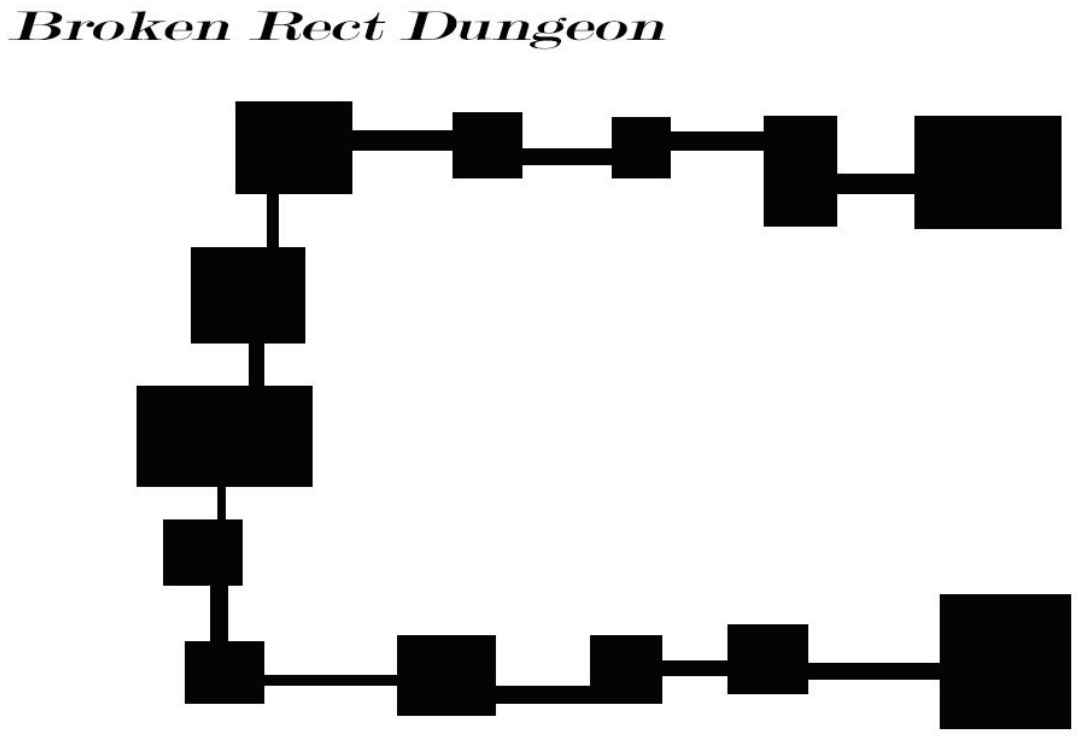
Line Dungeon



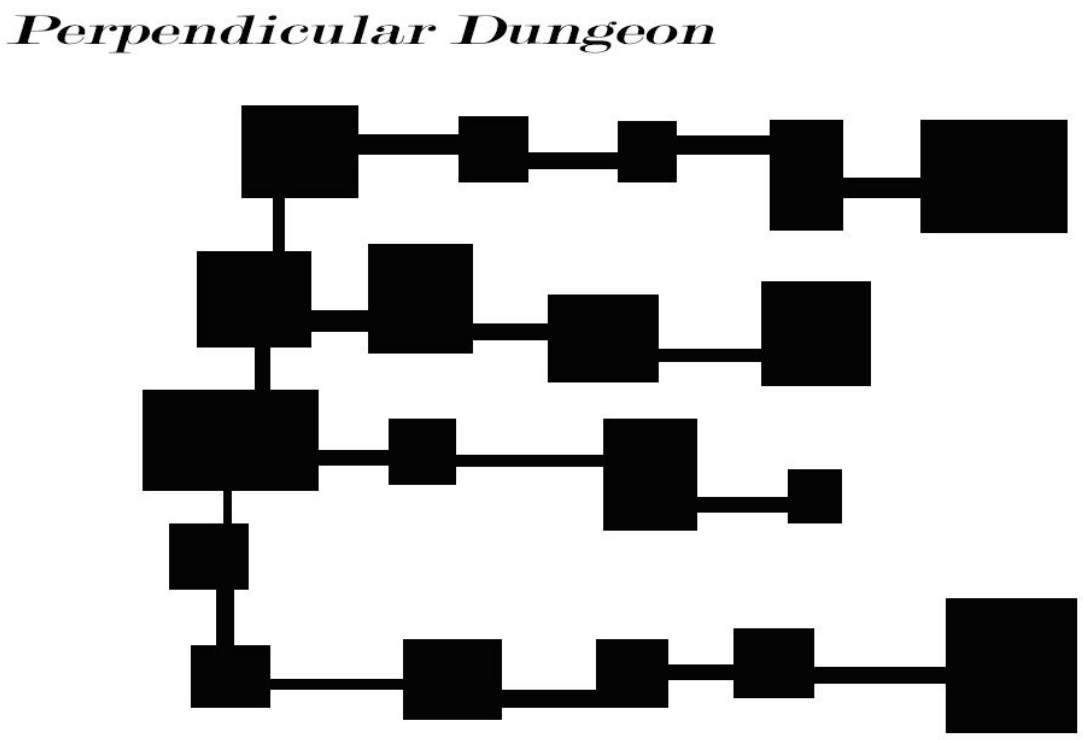
Generate Snake Dungeon - Generate simple one way dungeon with changing exit directions.



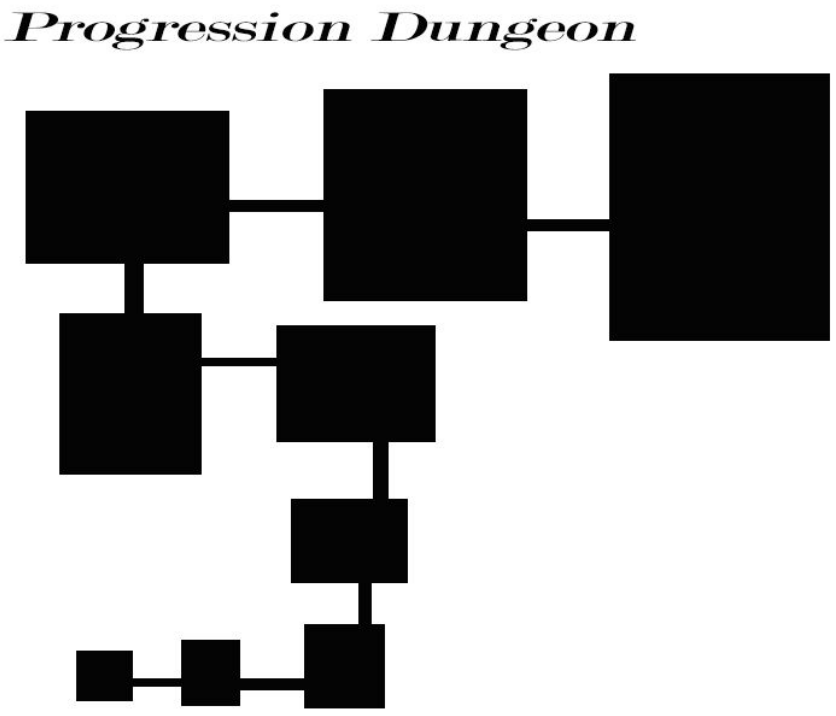
Generate Broken Rect Dungeon - Like a perpendicular dungeon, but only the first and last room of the dungeon has another way.



Generate Perpendicular Dungeon - Generate dungeon with many/random perpendicular lines.



Generate Progression Dungeon - this dungeon is a mix between Snake and Normal Dungeon. Rooms have a progression size, it begins by roomSizeIlow and roomSizeJlow value and progress +1 with next one room, but cannot be bigger than the max size of the room.



Generate Normal Dungeon - generate normal dungeon with long corridors.