PSUEDOCODE – Dayton Heywood

RoomActorBase

SetRoomMesh(StaticMeshActor\*)

Set Actor as the Mesh and Attach to this

Set Overlapping box component to cover the mesh size.

PopulateEmptySockets()

Get all Static Mesh Components

For each component

Get all sockets

For each socket

Spawn Actor based on socket name parsing

If Actor is a Door add to door array

If Actor is an AssetTemplate add to Asset array and call SpawnLoot on the Asset

GenerateDoorConnections()

Get the room array from GameMode

Filter viable rooms to connect to using the following rules:

* Room is not ourself (we don’t connect our doors together)
* Room cannot have 2 doors connected to the same room
* Rooms with all connected doors are ignored

For each unconnected door

Connect that door to a random viable room door

Remove that room from the list of viable rooms

AssetTemplate

PopulateLootSockets()

Get all Static Mesh Components

For each component

Get all sockets

For each socket

If socket contains “Loot” keyword

Get a random LootActor template from GameMode and spawn it on that socket

DoorActorBase

PostInitializeComponents()

If unconnected make door invisible and unresponsive

ApplyConnection(DoorActor\*)

Connect to door

Set this door visible and responsive

TeleportPawnToOtherDoor(…) //is an event from collision sphere

Set triggering actor position and rotation to connecting door

If triggering actor is a player

Rotate the controller to the arrow component

LootingLootersGameModeBase

Constructor()

Load all room blueprints into memory

Load all asset blueprints into memory

Load all loot blueprints into memory

GenerateRandomRoomLayout()

For every room we are creating

Calculate grid offset using index

Spawn a RoomActor

Spawn a random Room asset

Attach them together

GenerateRandomRoomConnections()

For every room

Make that room connect their doors to other rooms doors

PopulateRoomSockets()

For every room

Make that room spawn Assets on its sockets

GenerateLoot()

For every room

Make that room’s Assets spawn Loot on its sockets

GetARandomAssetOfTypes(types[])

For every Asset blueprint

If that Asset has all types provided

Add as viable mesh

Return a random viable mesh