LAB – 3

Test the Multiplayer System with a Project Prototype

2. Creating a Multiplayer Game Prototype

2.2 Setting Up the project from a template

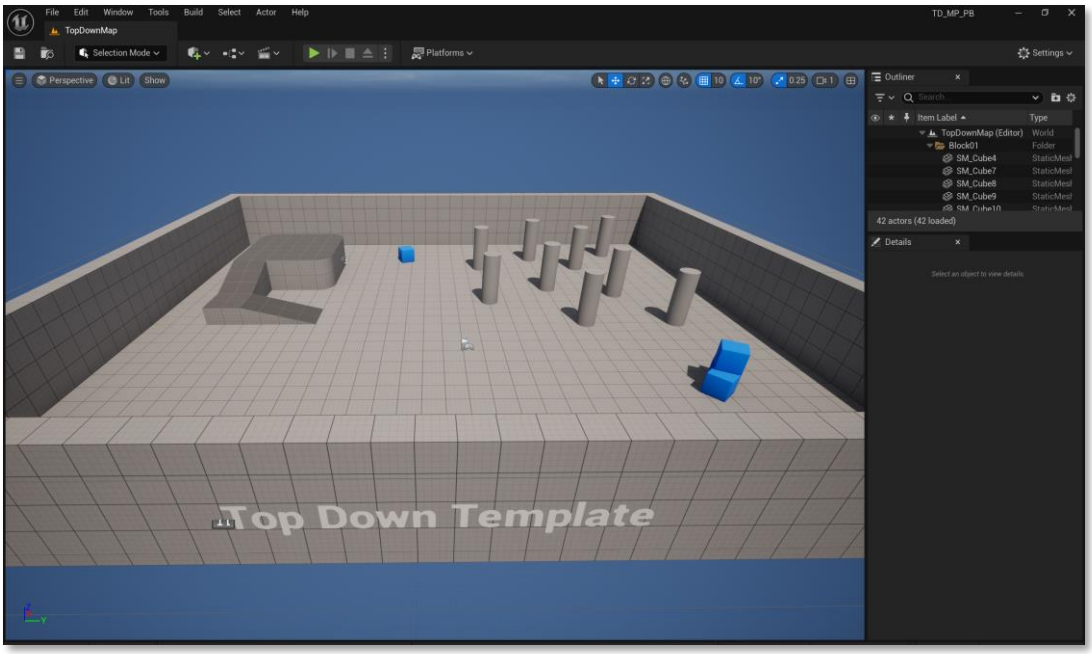

Fig 2.2.1: Screenshot showing Top-Down Template

2.3 Obtain some assets for the props

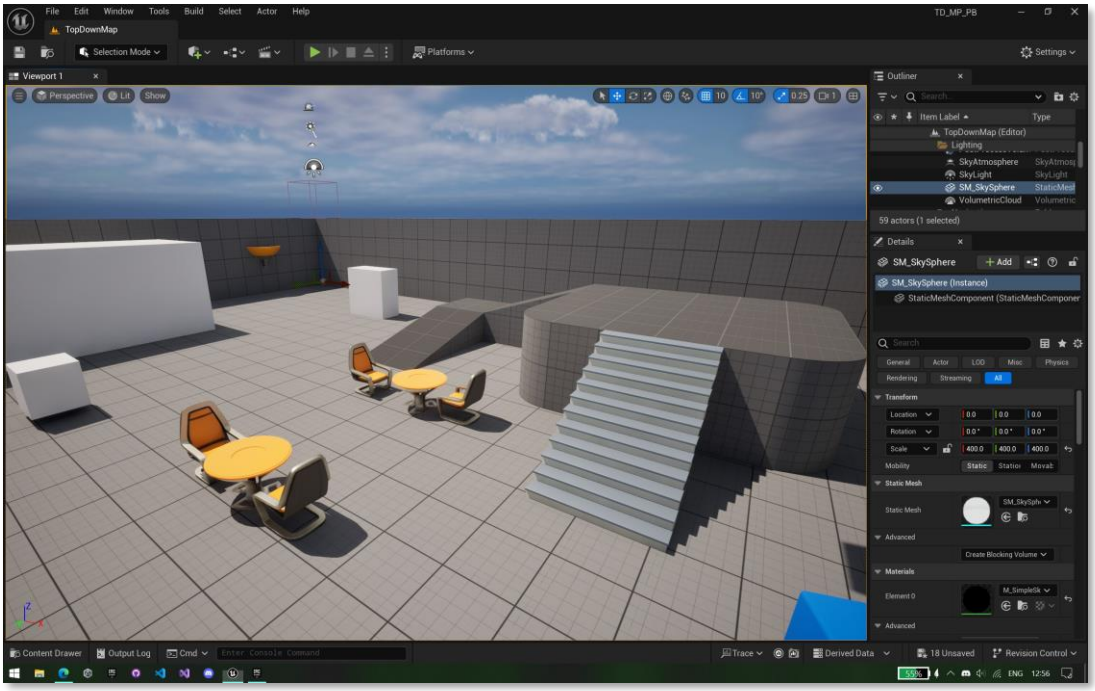

Fig 2.3.1: Screenshot showing addition of props

2.4 Resizing Assets

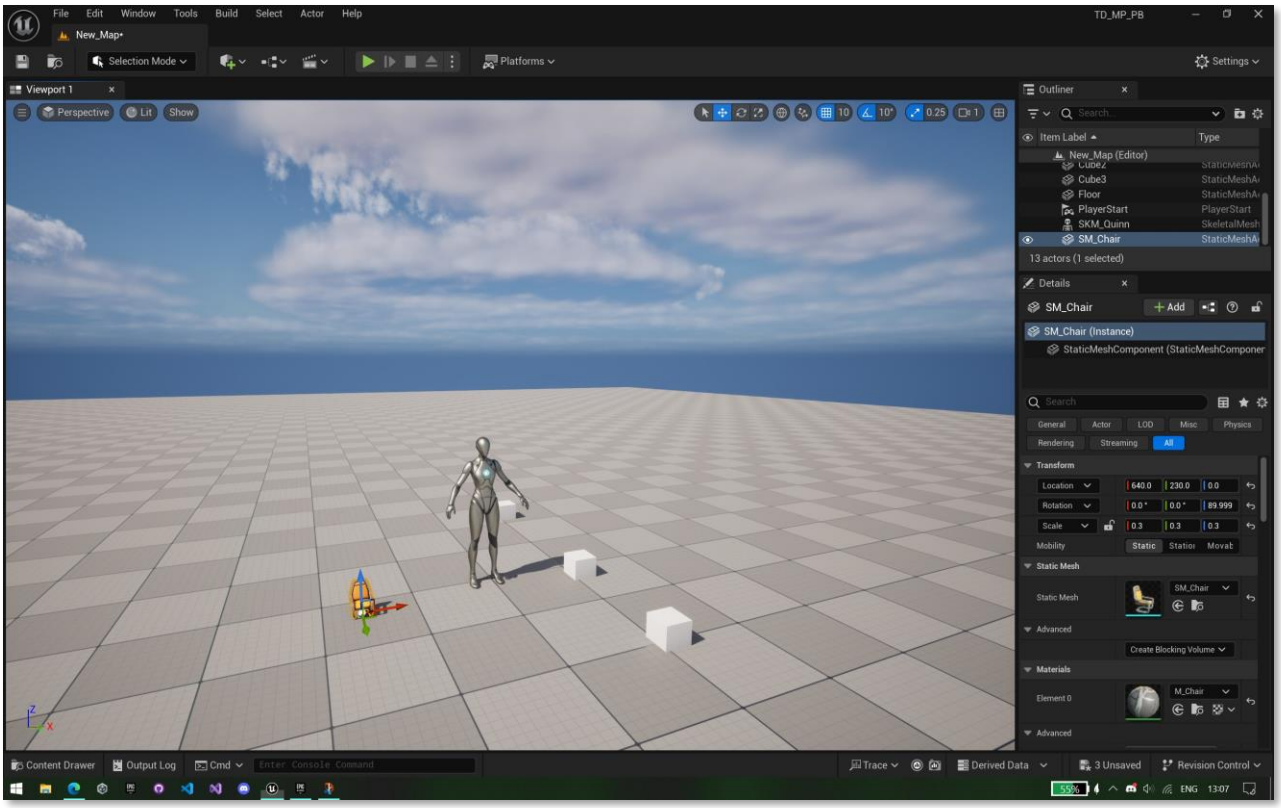

Fig 2.4.1: Screenshot showing new map and scaling of props to 1/3 of Mannequin

2.5 Baking (in Unity was Creating Prefabs)

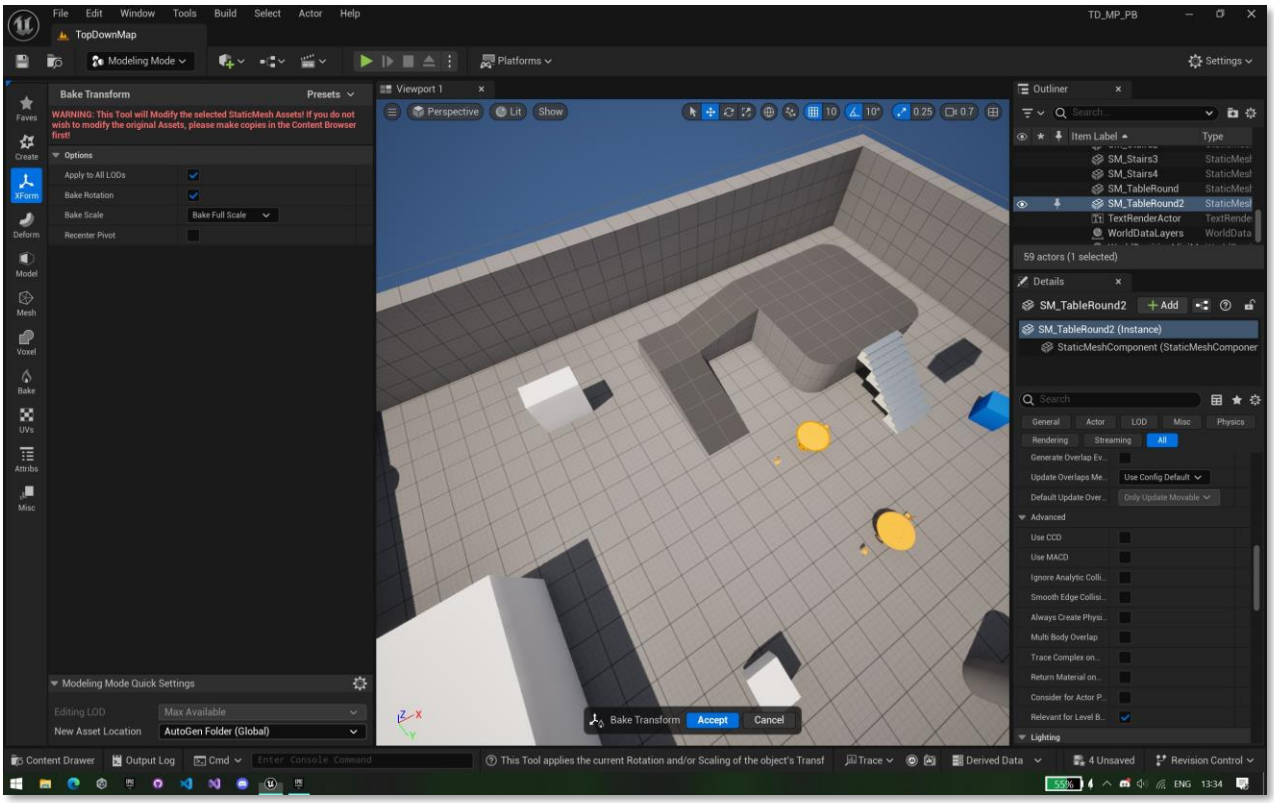

Fig 2.5.1: Screenshot showing baking

2.6 Generate Collisions for the Models (in Unity were called "Colliders")

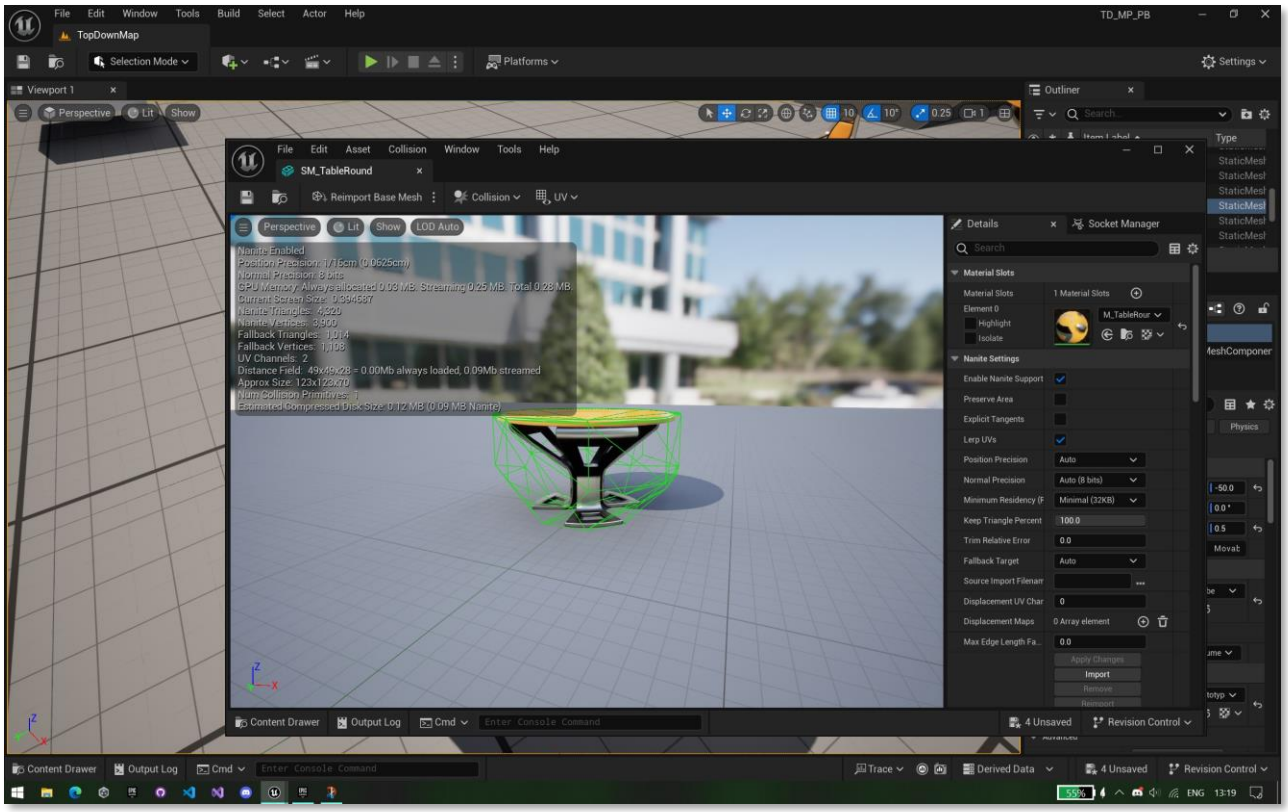

Fig 2.6.1: Screenshot showing 26DOP Changes to Collision to asset

2.7 Modifying the Player Controller

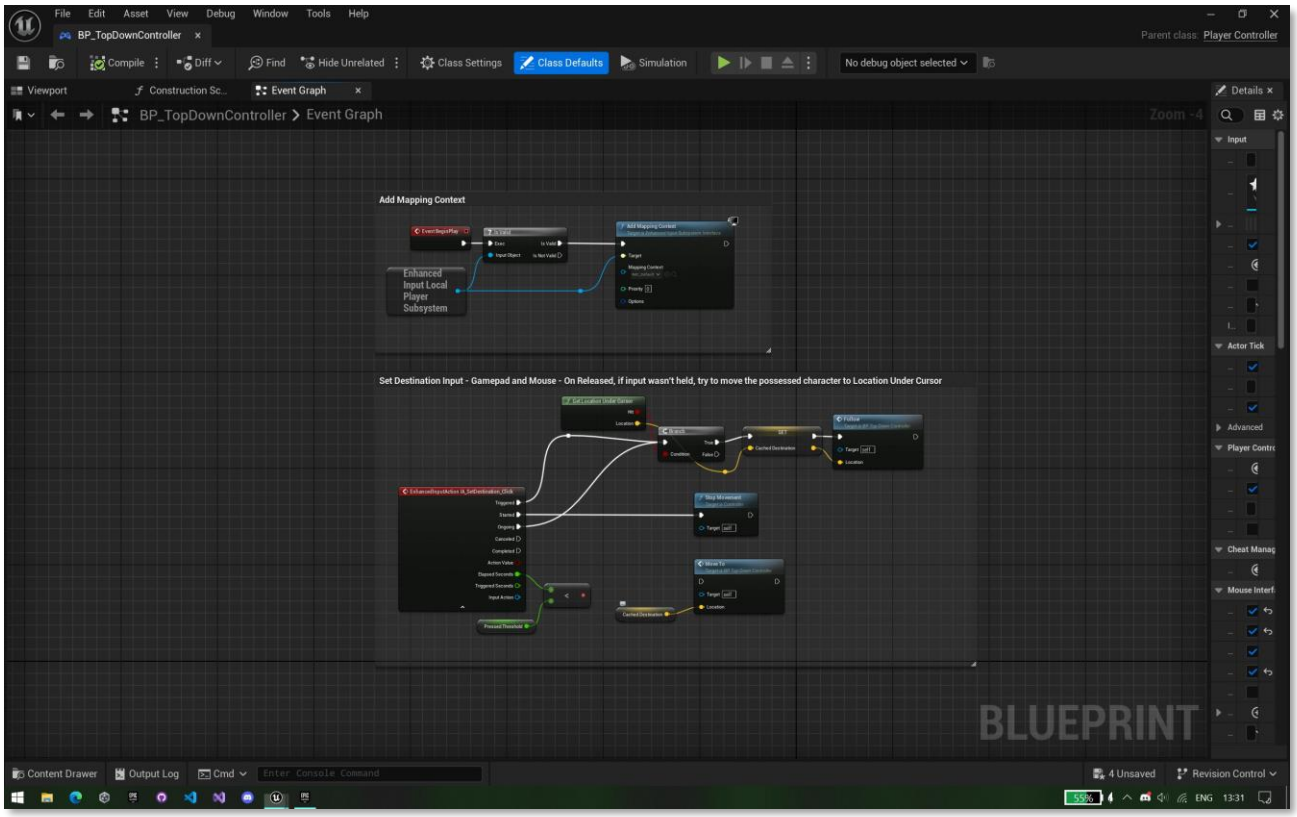

Fig 2.7.1: Screenshot showing nodes readjusted to in the controller

3. Testing a multiplayer game locally

3.1 Playing as a Listen Server

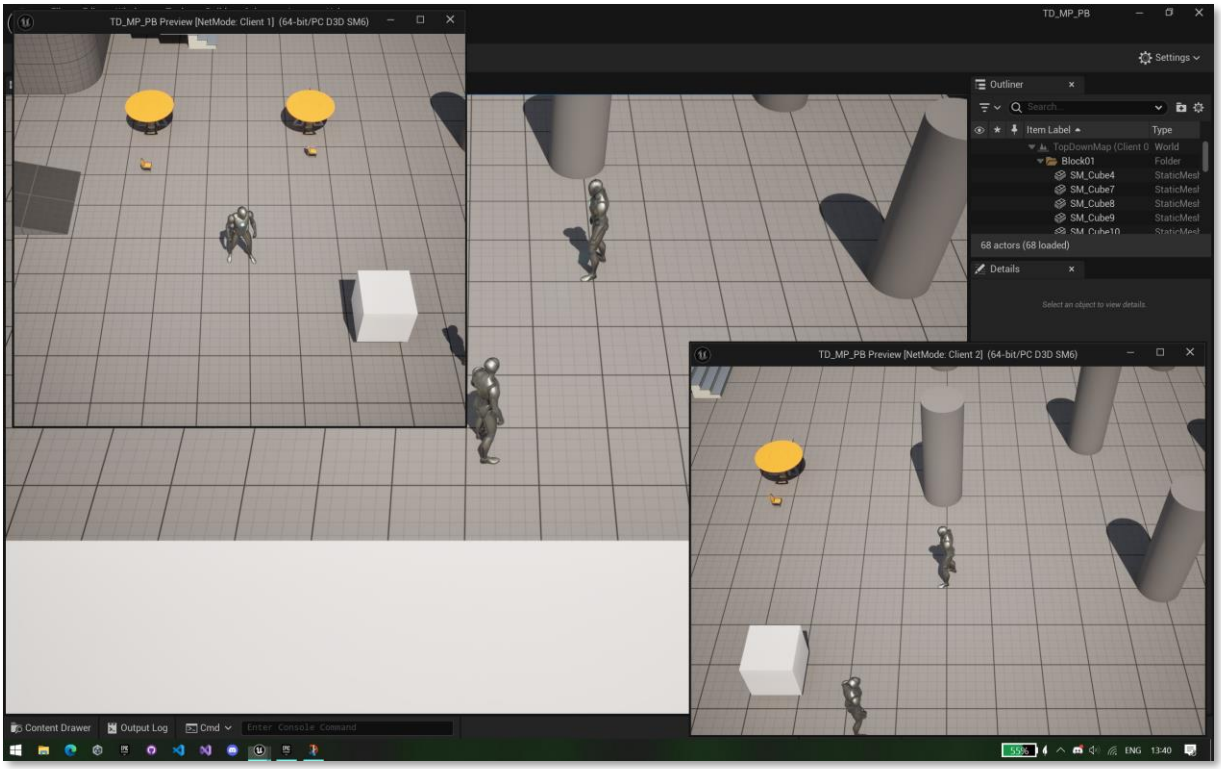

Fig 3.1.1: Screenshot showing Net Mode feature playtest with 2 clients and 1 server

3.2 Updating over the network

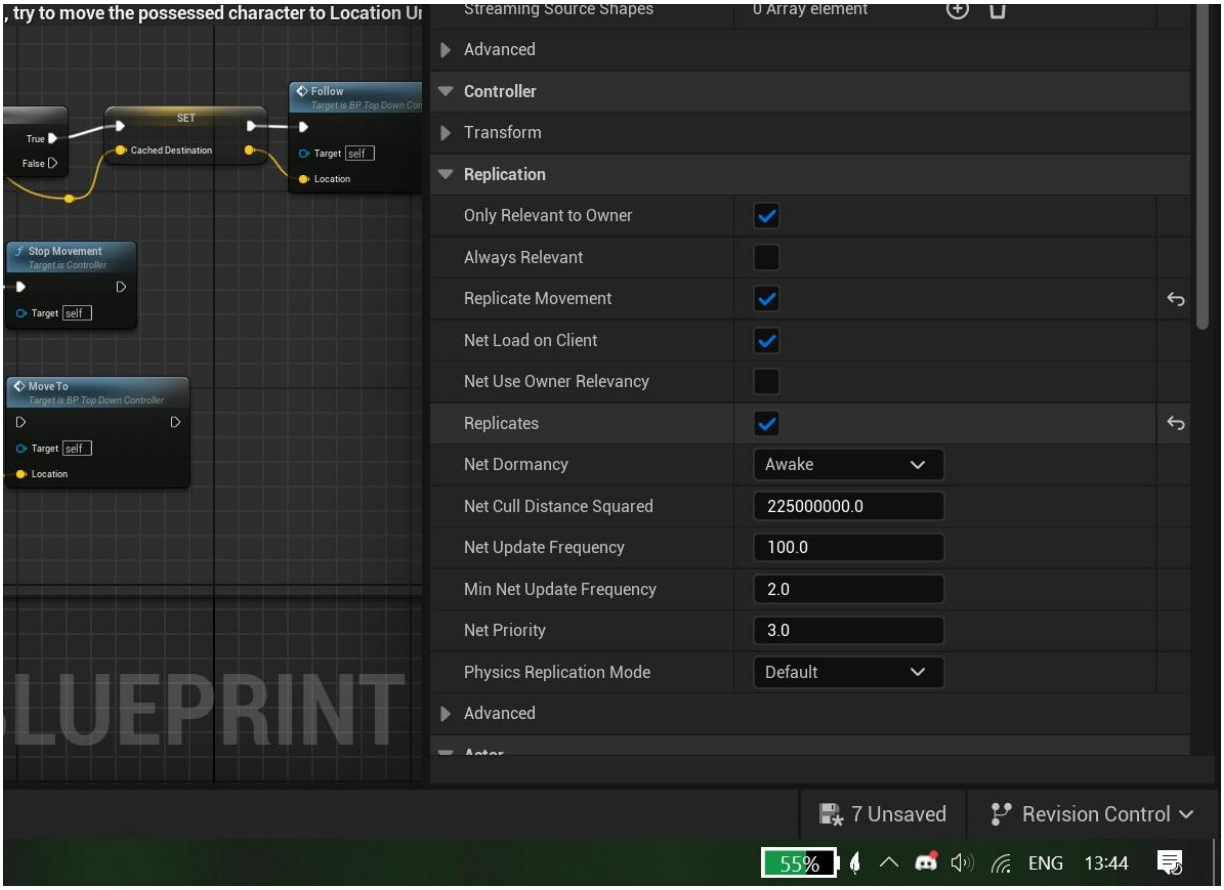

Fig 3.2.1: Screenshot showing Replication settings and attributes

3.3 Adding additional character spawn points

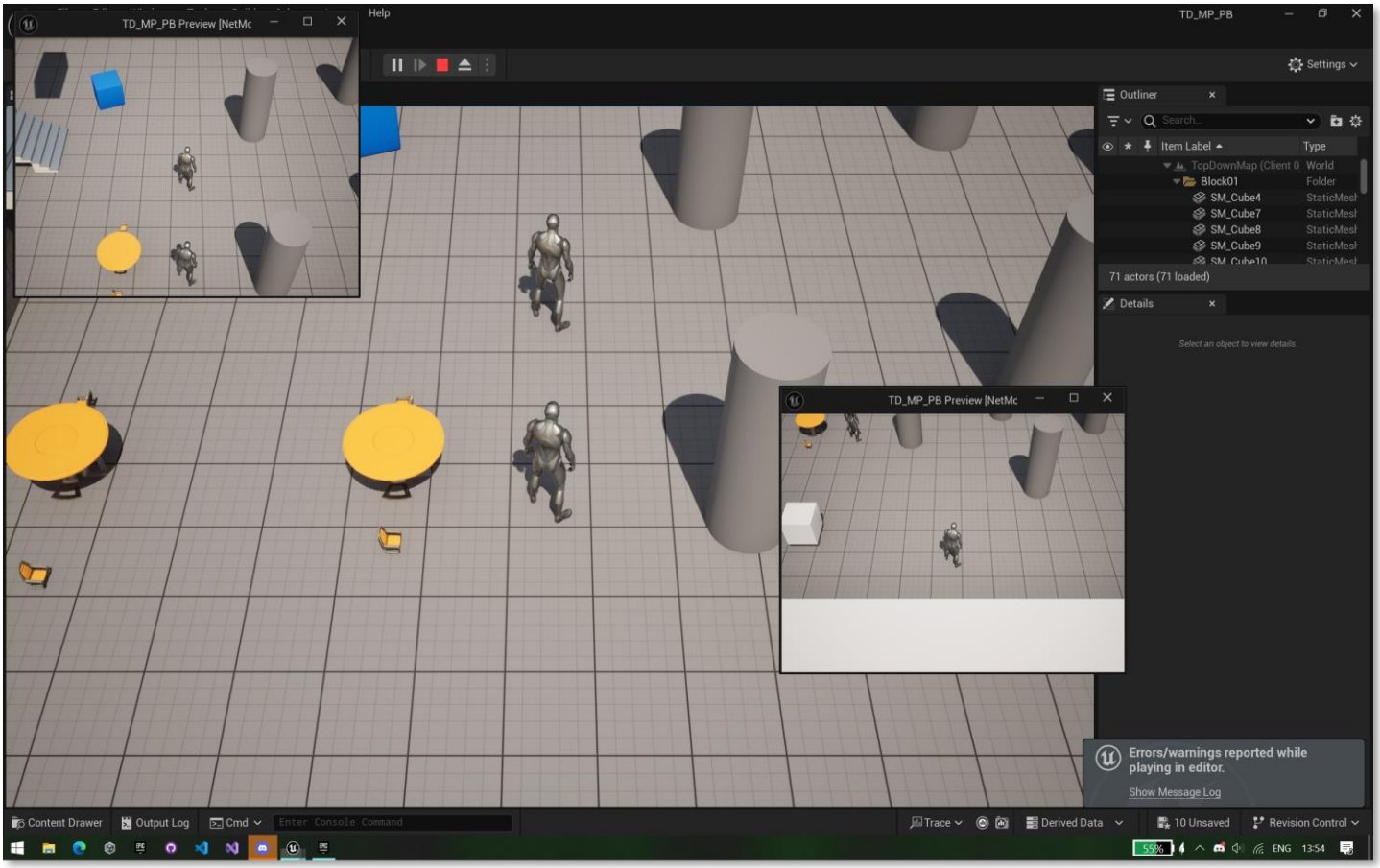

Fig 3.3.1: Screenshot showing Player Start

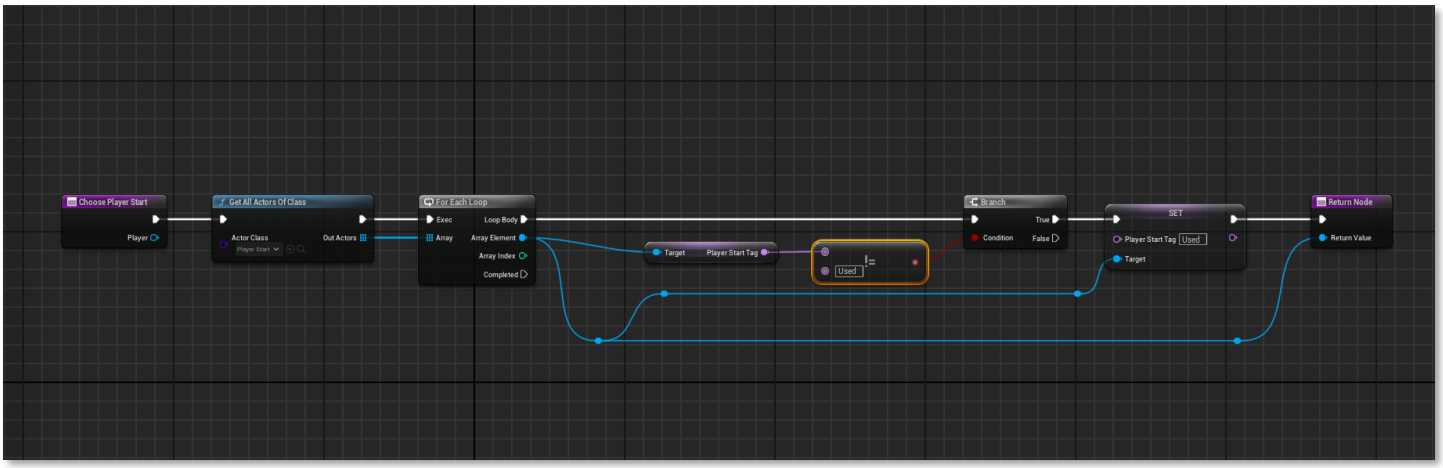

Fig 3.3.2: Screenshot showing Player Start

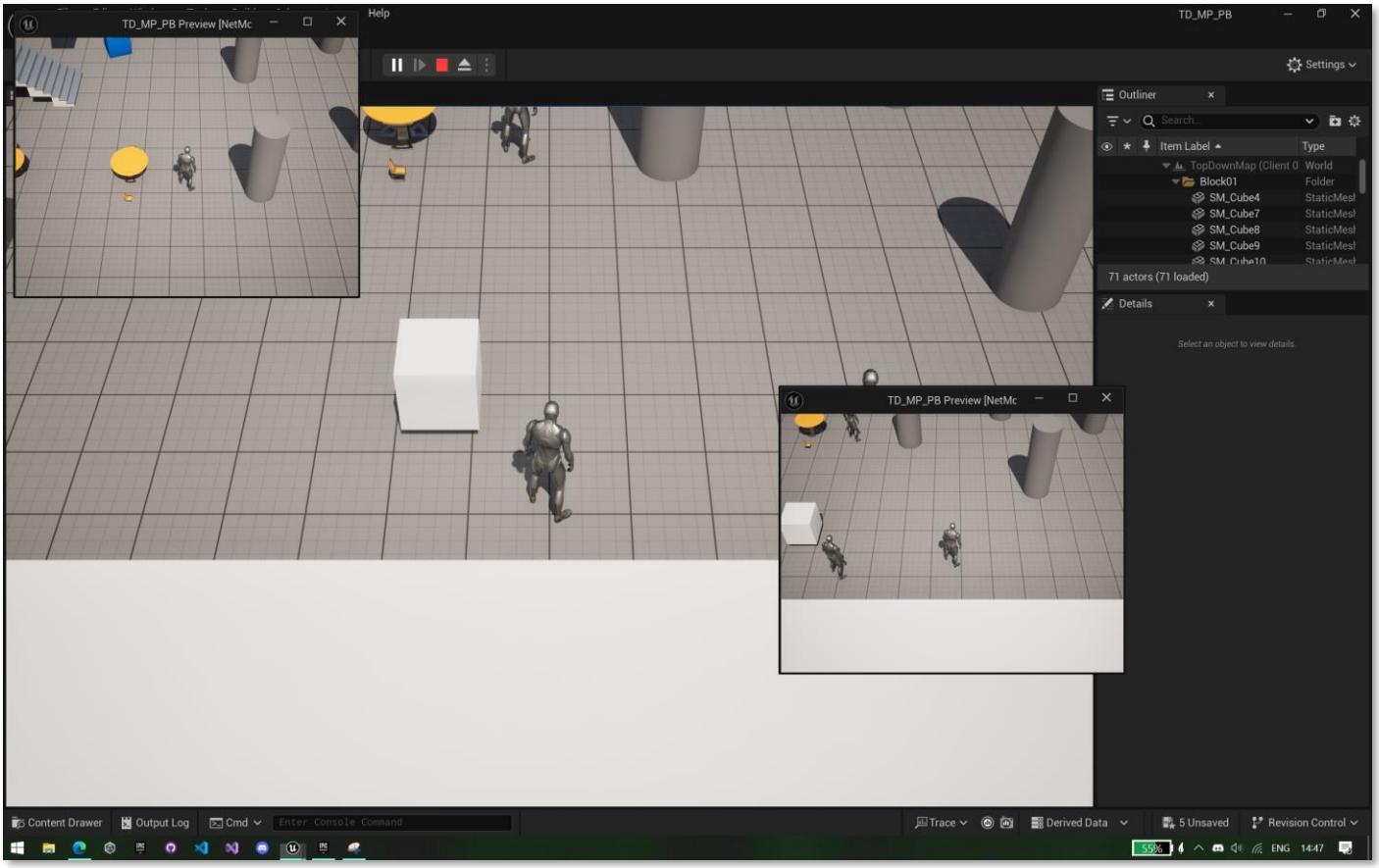

Fig 3.3.3: Screenshot showing Player now not spawning together

4. Updating properties over the network

4.1 Creating the pickup Blueprint

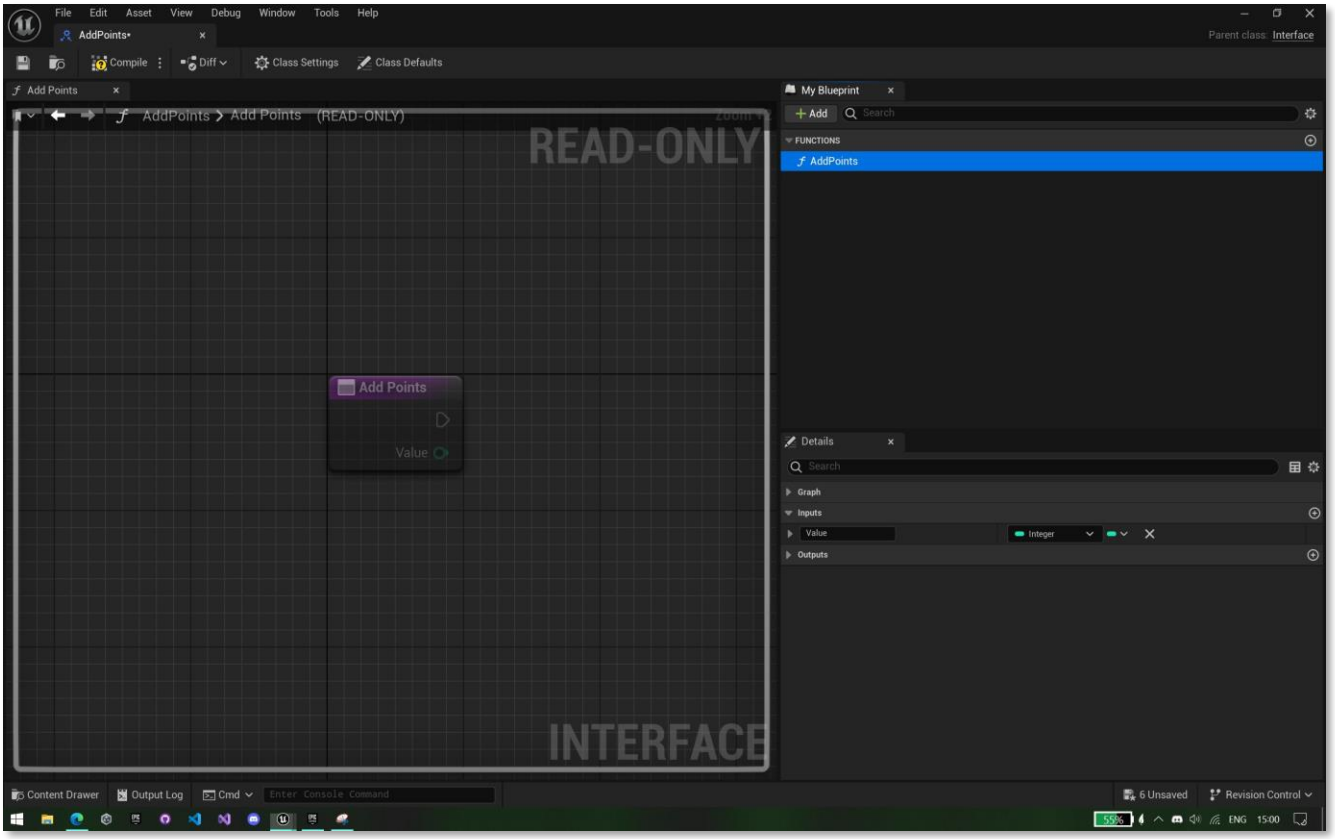

Fig 4.1.1: Screenshot showing pickup blueprint interface with function AddPoints with integer input value

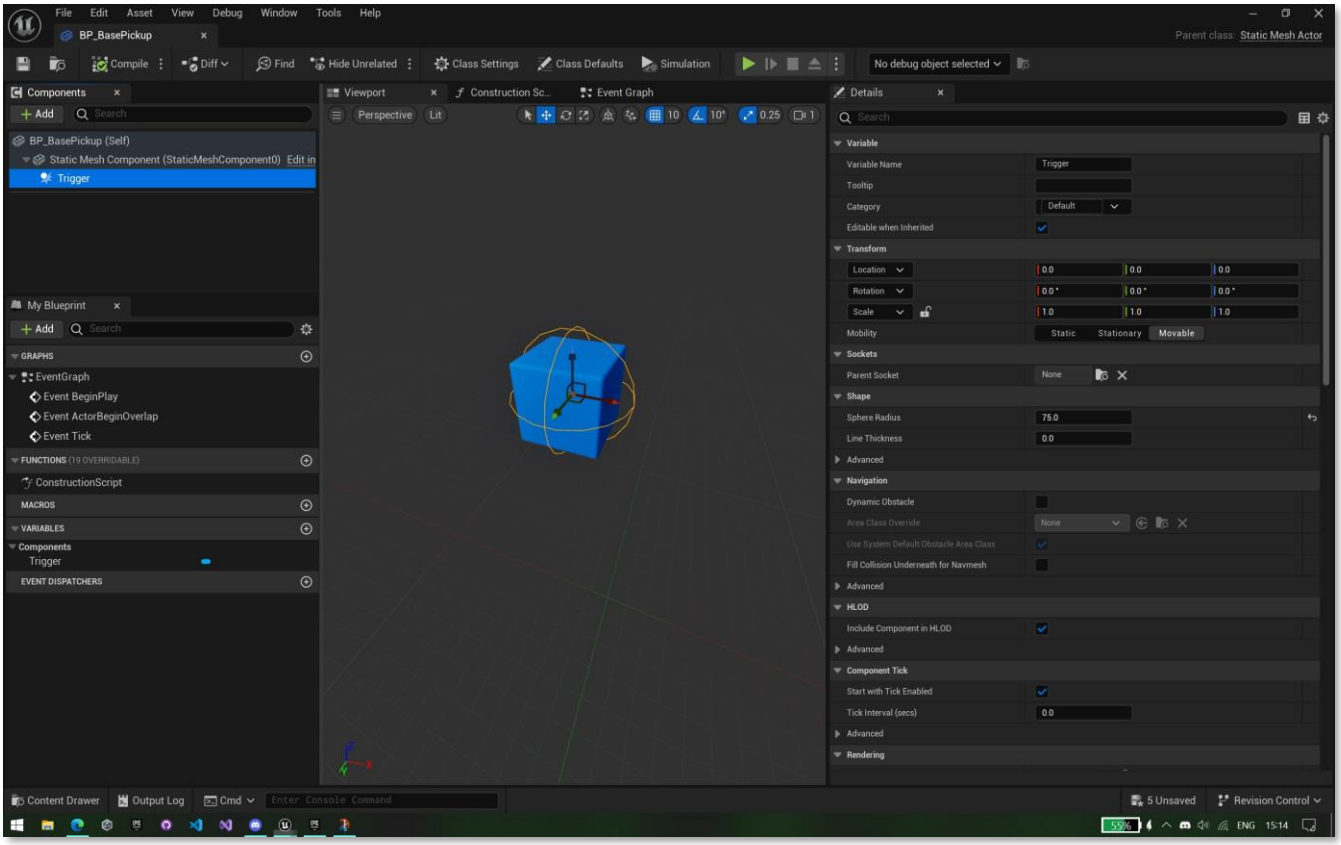

Fig 4.1.2: Screenshot showing pickup static mesh with sphere collider

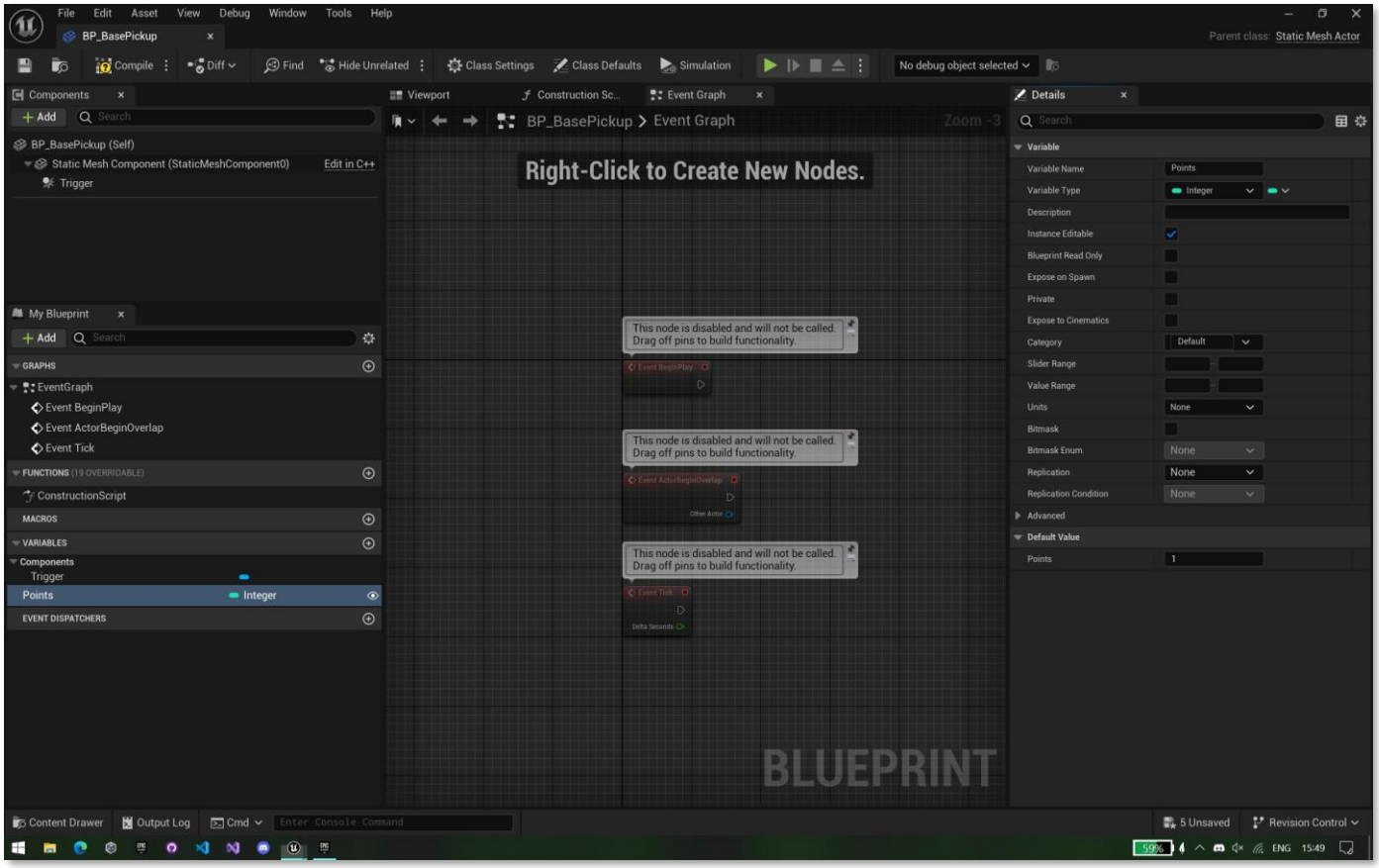

Fig 4.1.3: Screenshot showing creation of Points variable and setting it to 1 as default.

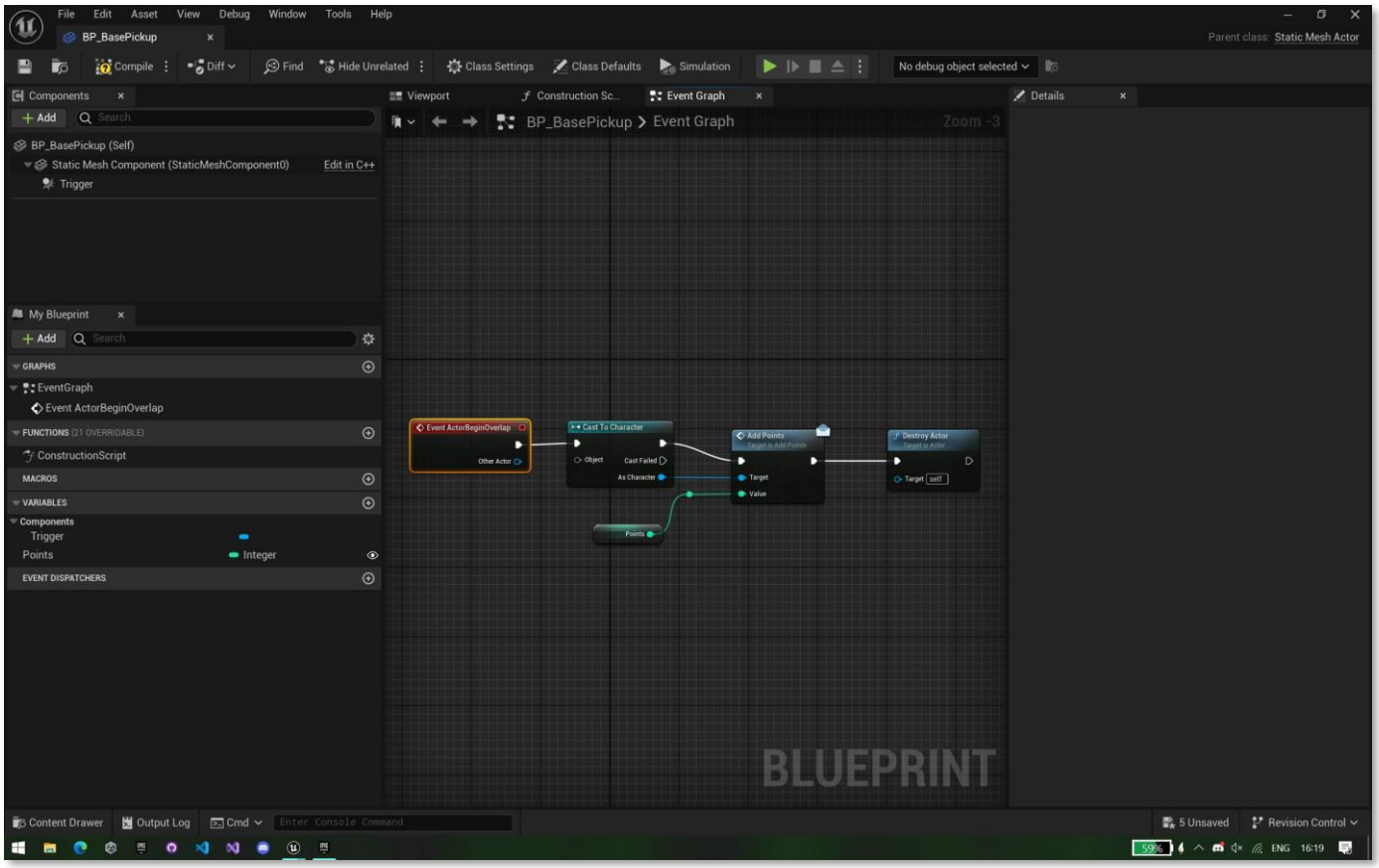

Fig 4.1.4: Screenshot showing overlap of event behaviour for blueprint.

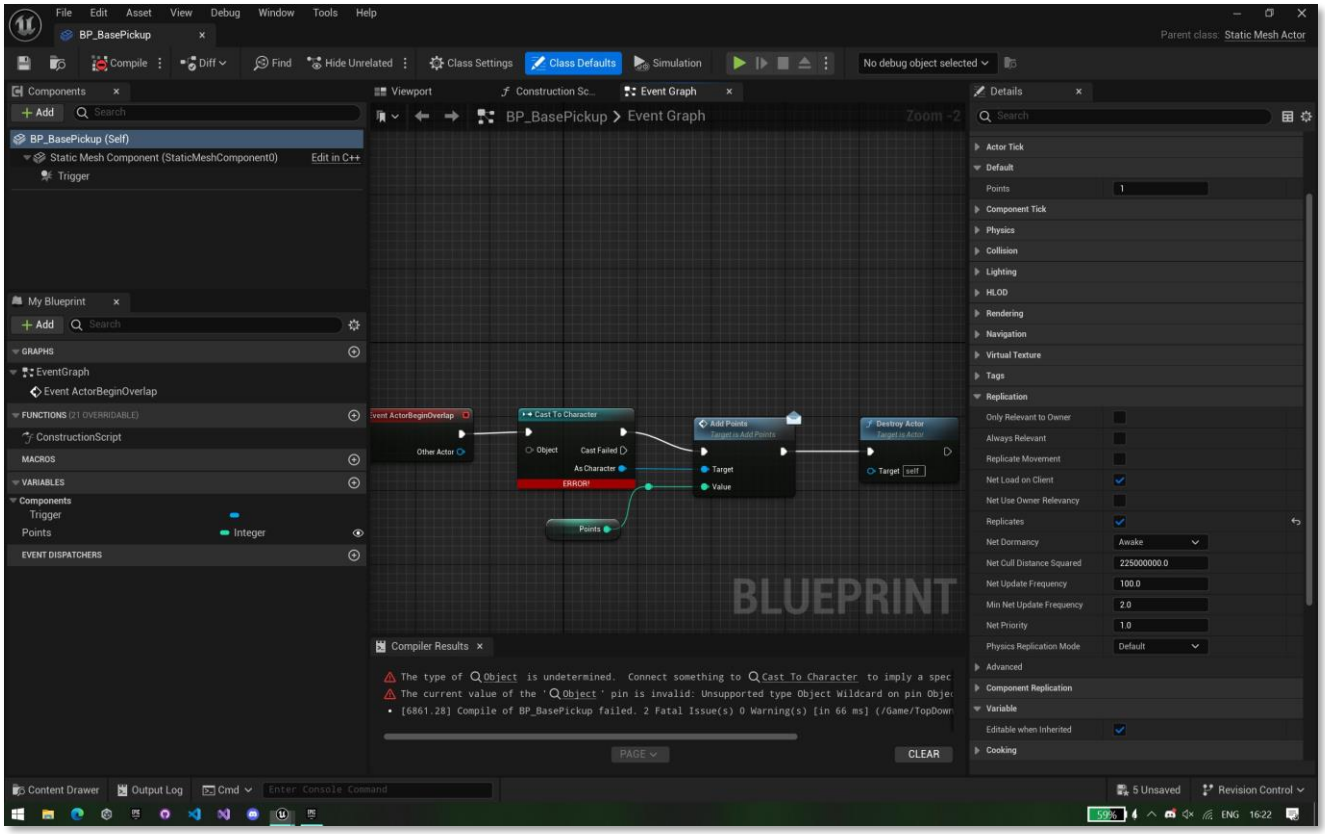

Fig 4.1.5: Screenshot showing enabling of replication.

4.2 Adding pickup variants

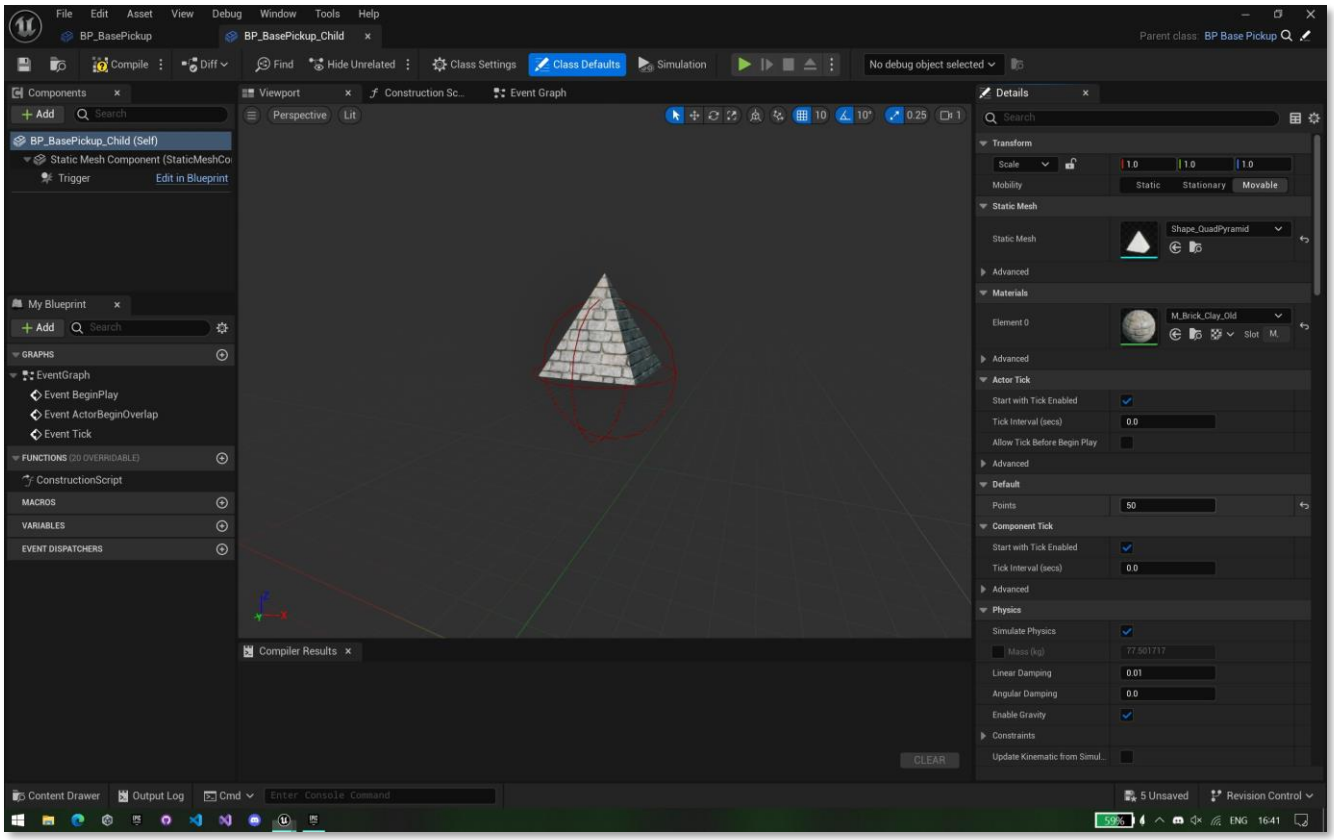

Fig 4.2.1: Screenshot showing a variant of a pickup.

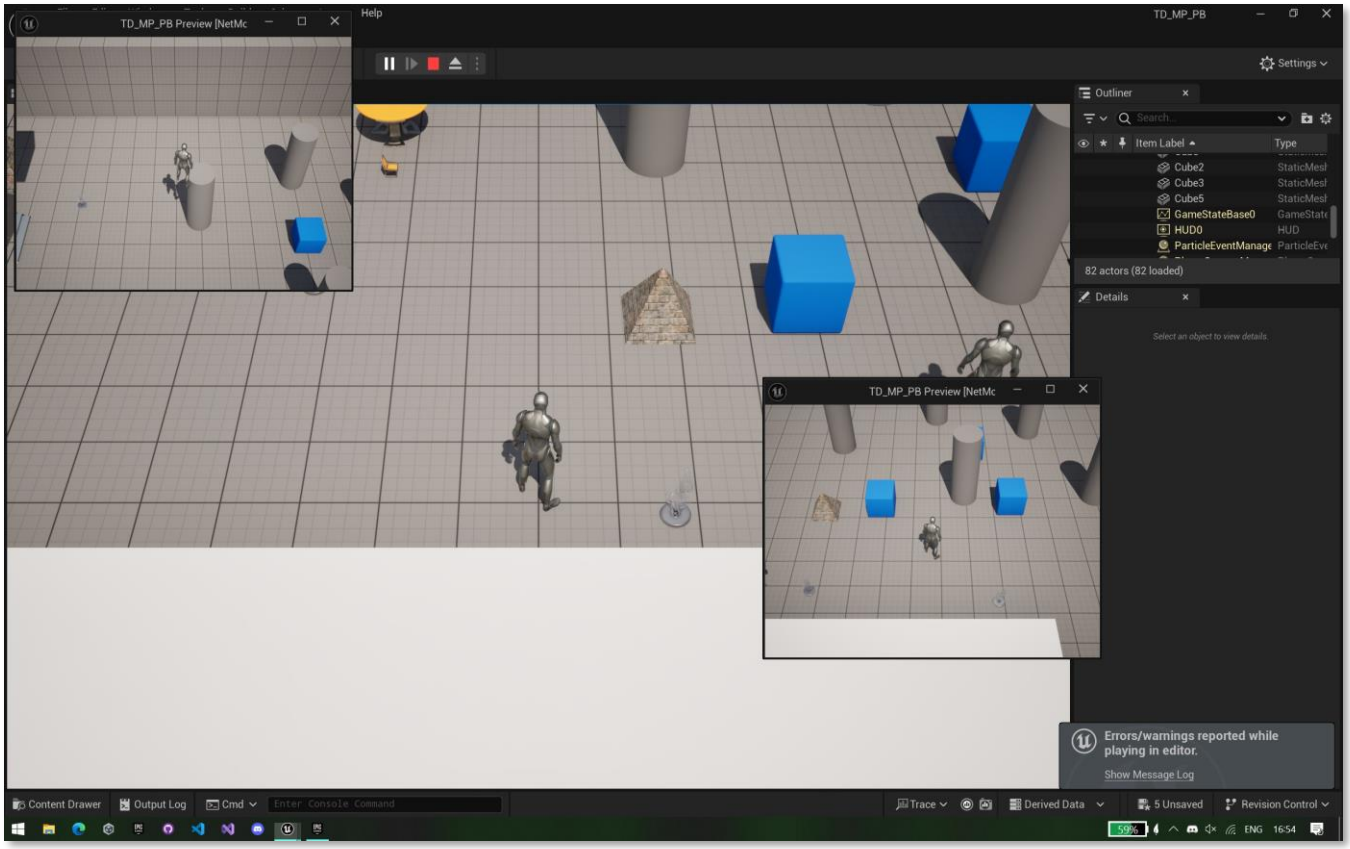

Fig 4.2.2: Screenshot showing a playtest variant of various pickup.

4.3. Adding a points counter to the characters

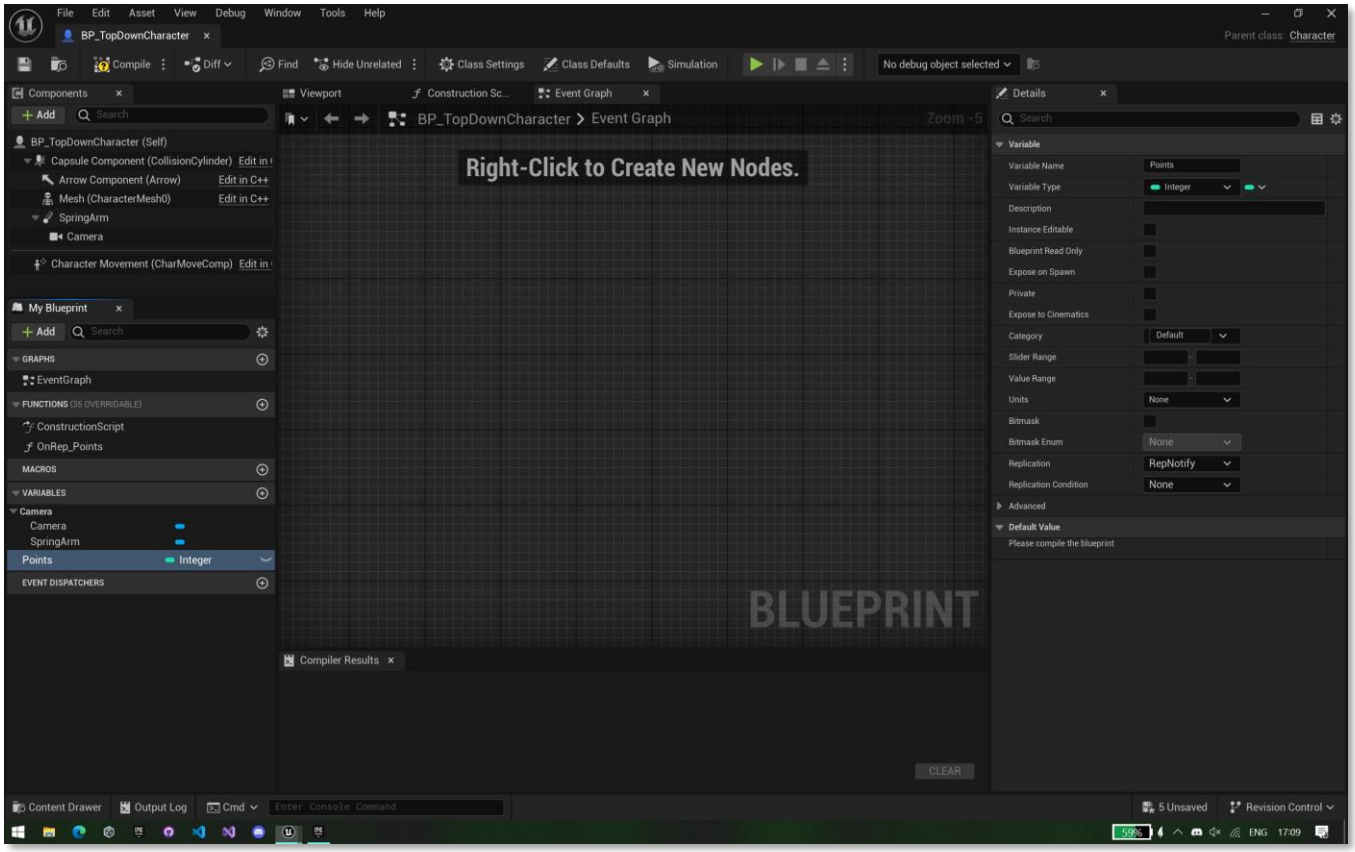

Fig 4.3.1: Screenshot showing a RepNotify of variable Points.

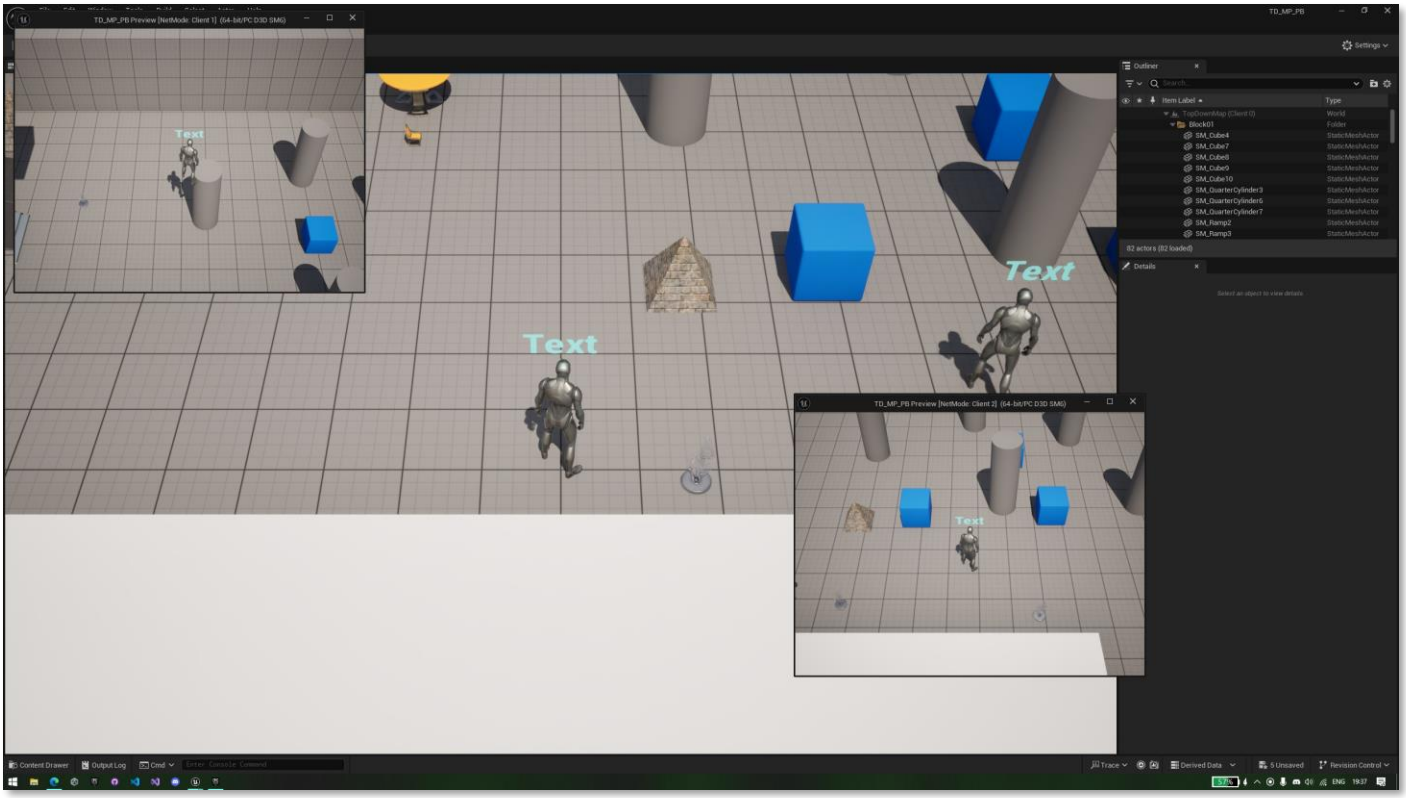

Fig 4.3.2: Screenshot showing a Text on head.

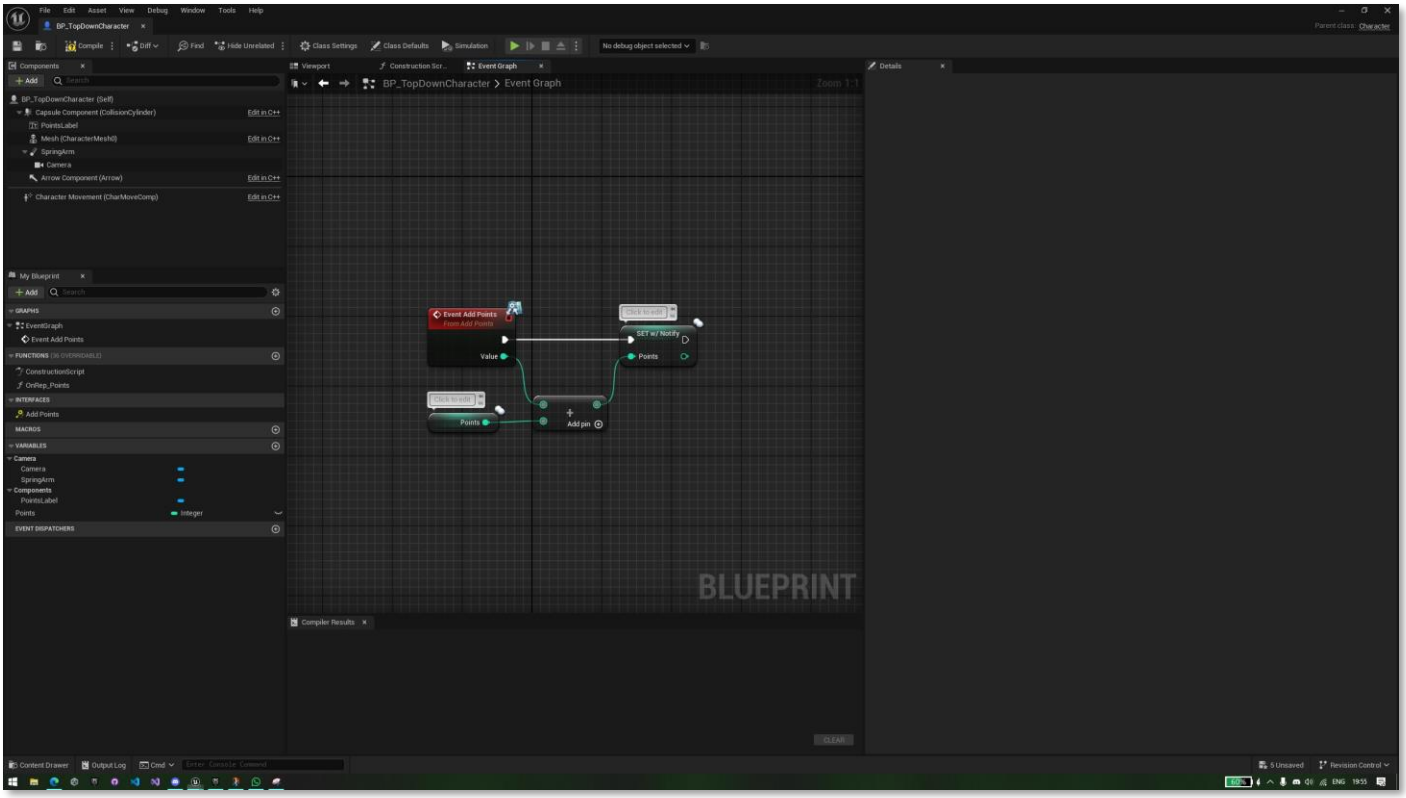

Fig 4.3.3: Screenshot showing add operation for updating score.

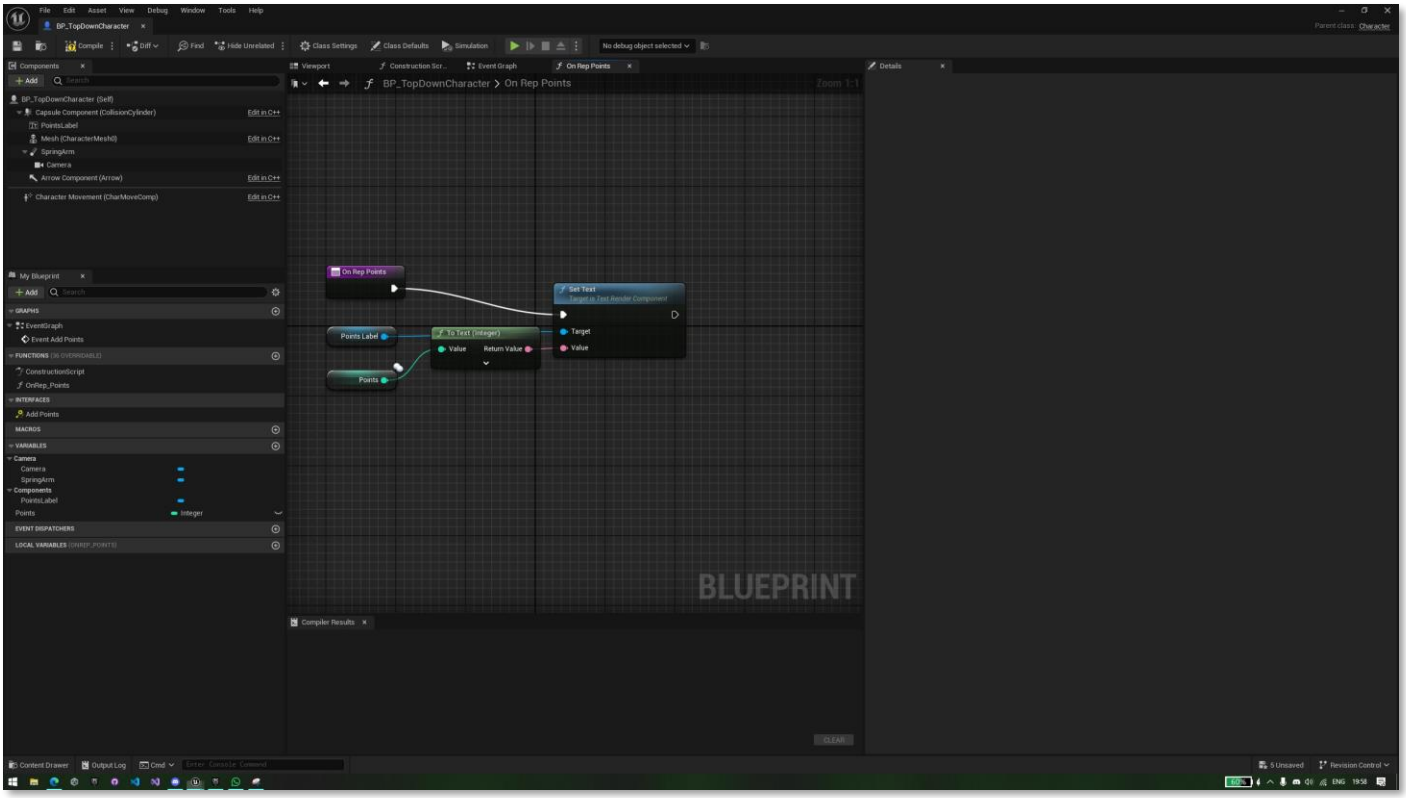

Fig 4.3.4: Updating the Text component

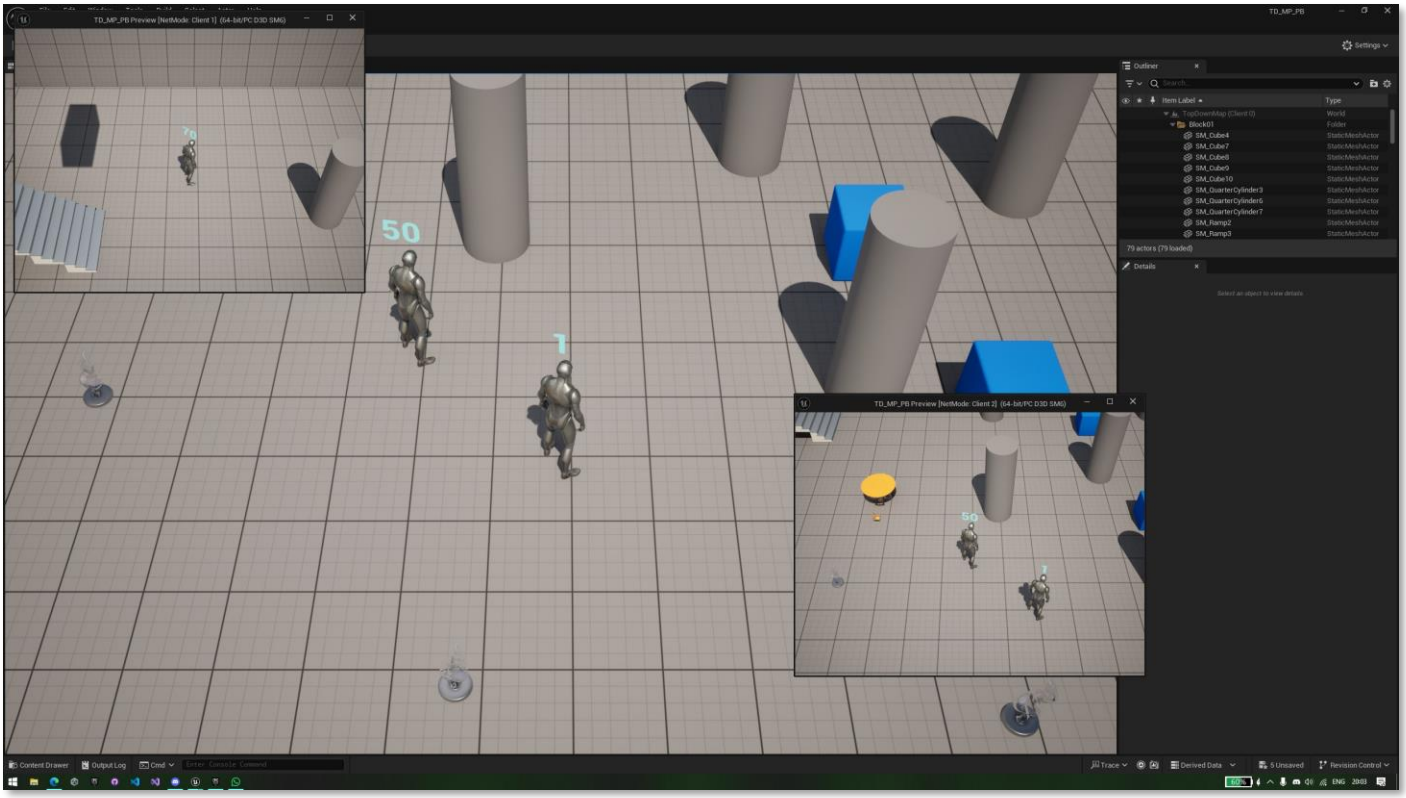

Fig 4.3.5: Screenshot showing that the text elements are updated

5. Executing functions over the network

5.1.1 Choosing a random position

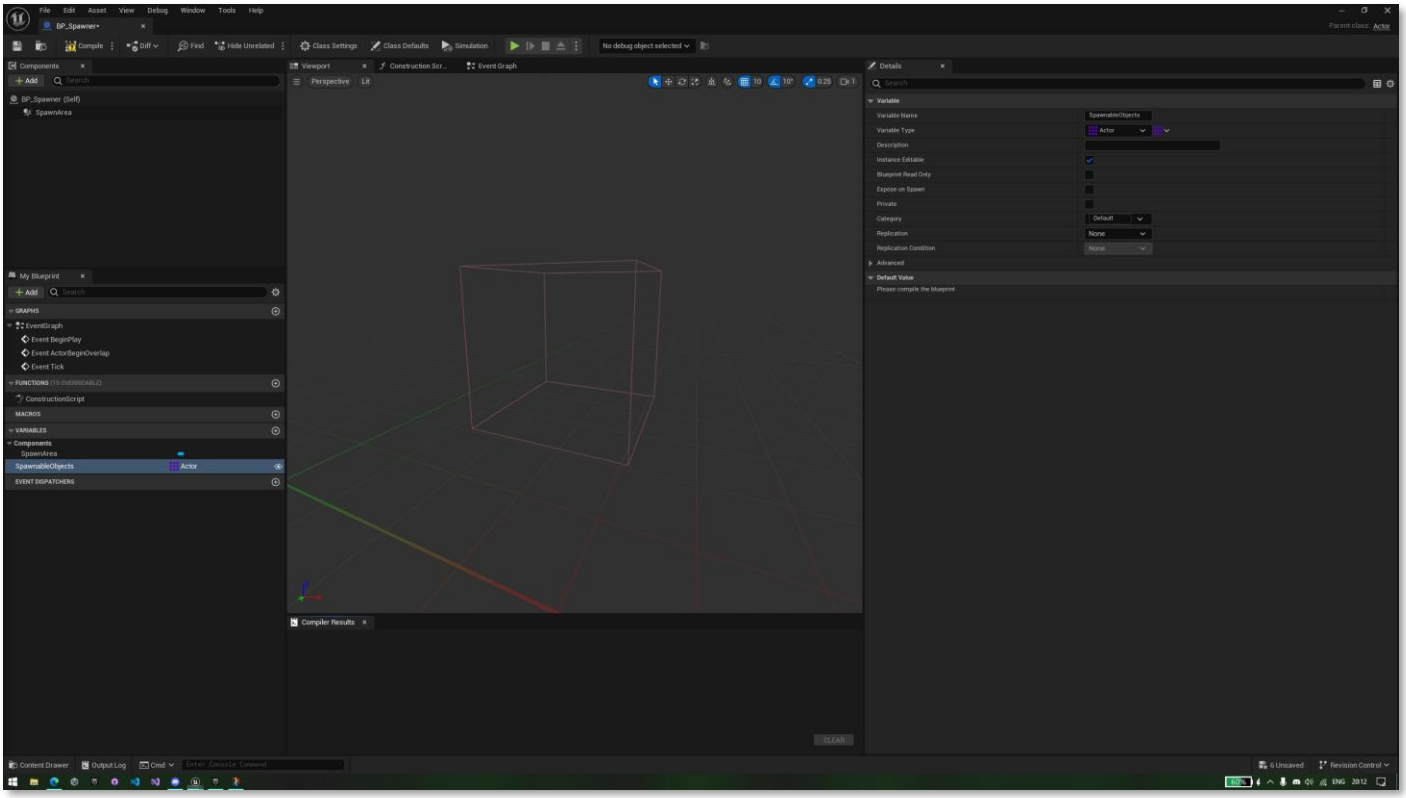

Fig 5.1.1.1: Screenshot showing spawnable objects array for random spawning

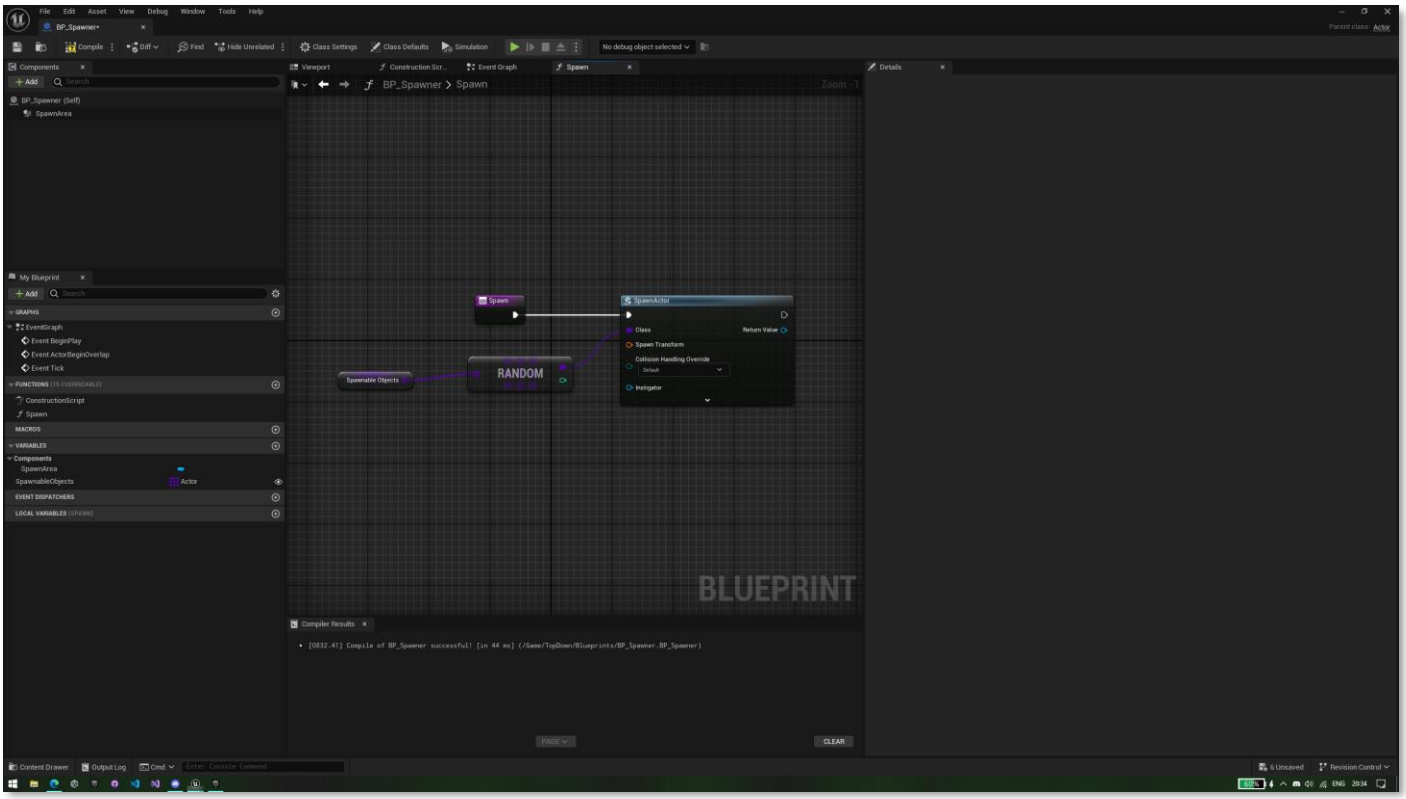

Fig 5.1.1.2: Screenshot showing configuration in event graph

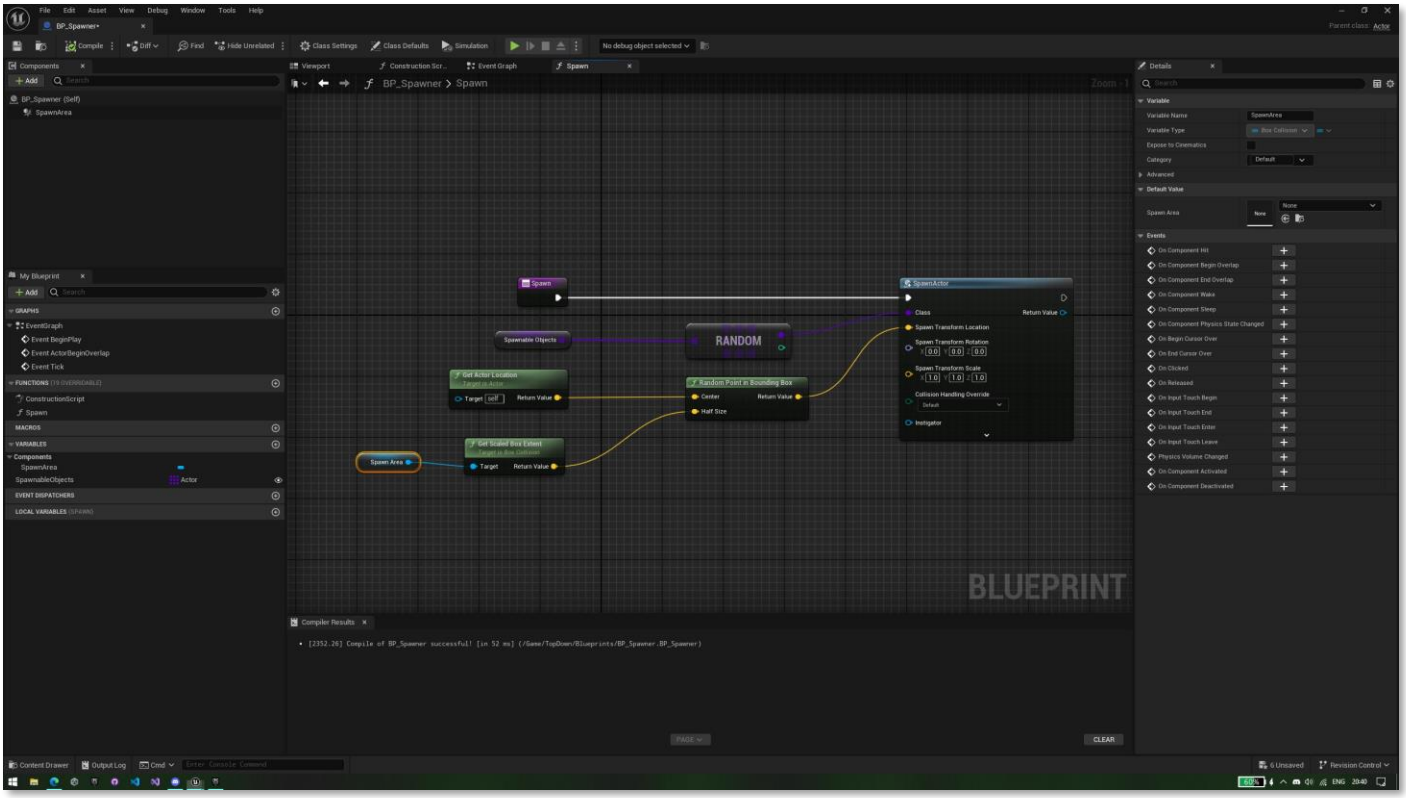

Fig 5.1.1.3: Screenshot showing configuration in event graph 2

5.1.2 Spawn random pickups at predefined intervals

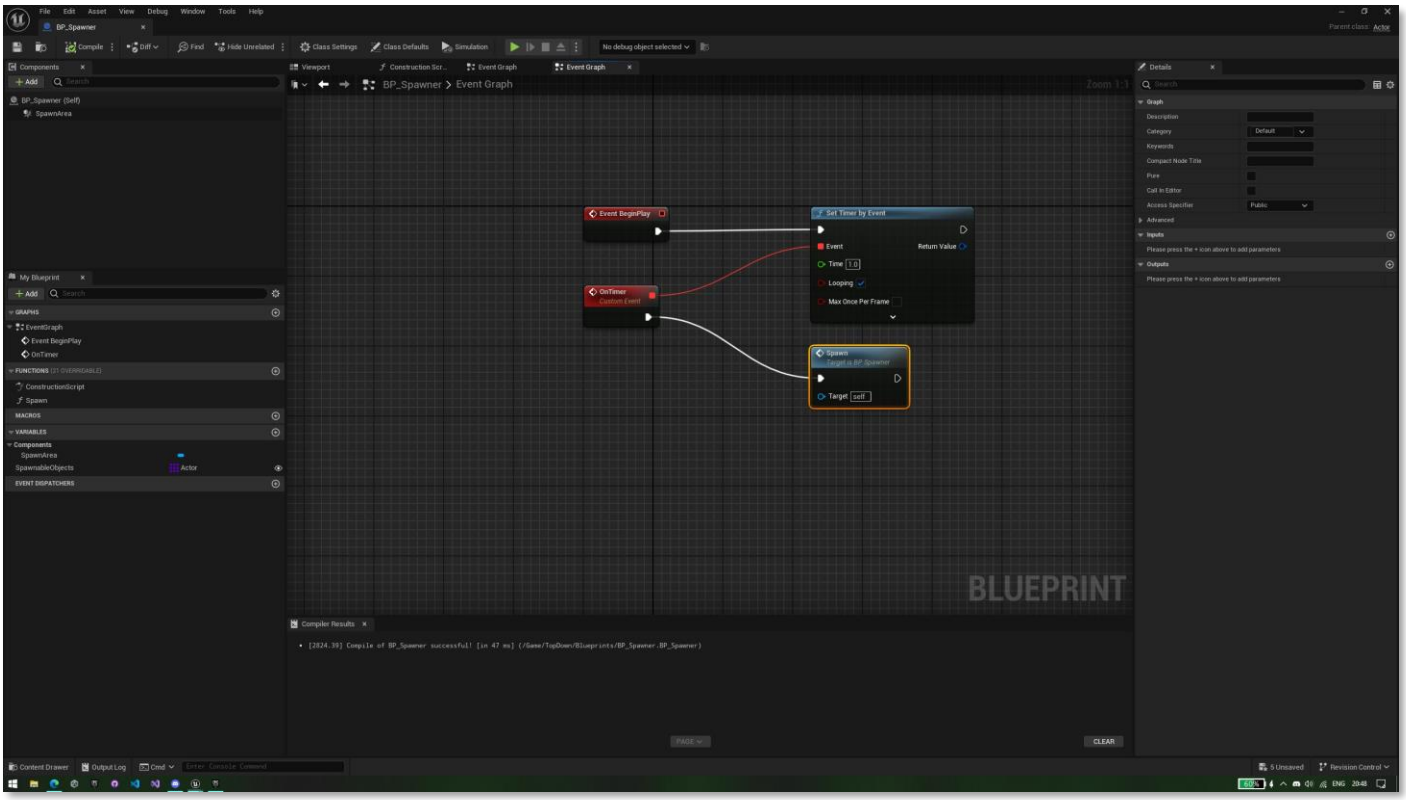

Fig 5.1.2.1: Screenshot showing configuration in event graph interval spawn

Fig 5.1.2.2: Screenshot showing random spawning of pickups

5.1.3 Using Actor authority to correctly spawn pickups

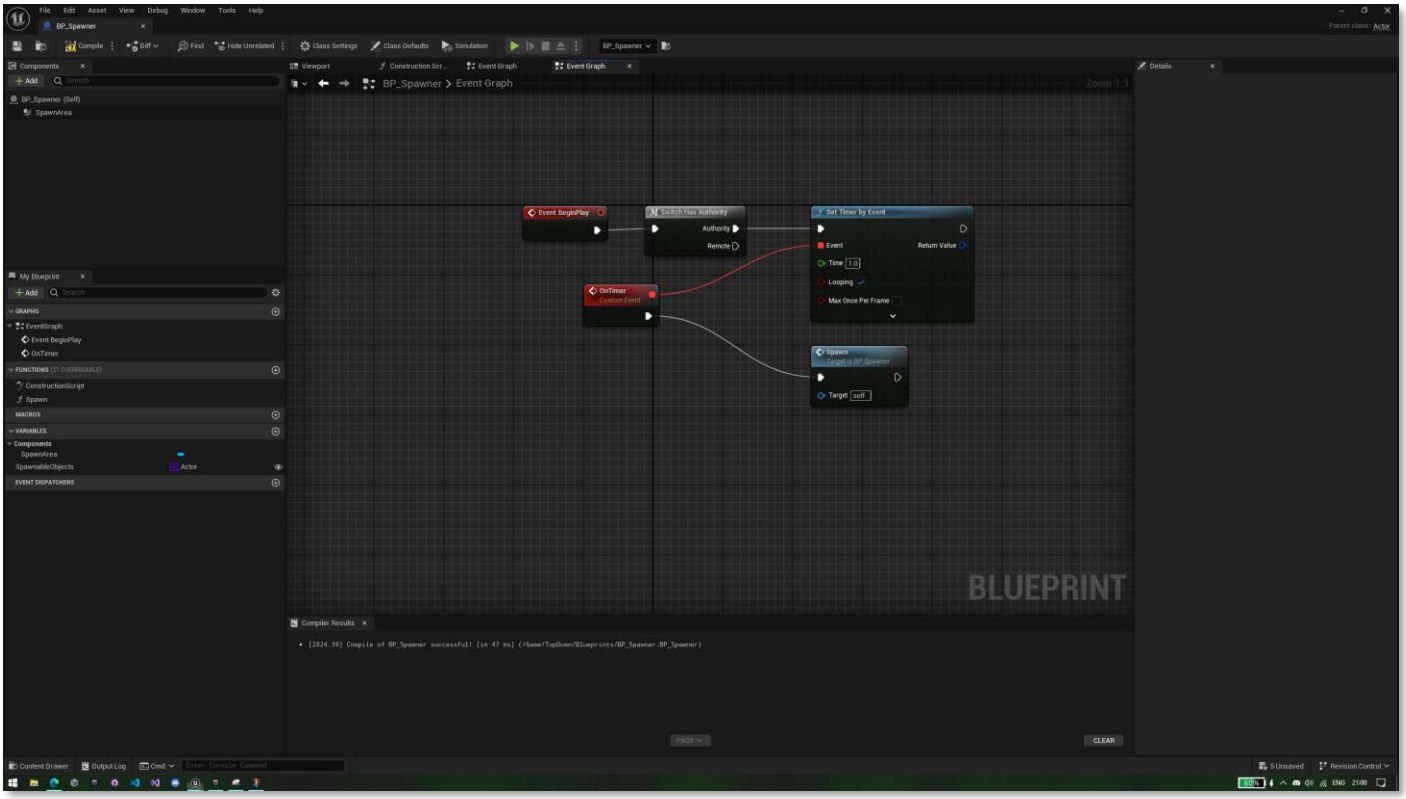

Fig 5.1.3.1: Screenshot showing addition of Switch Has Authority

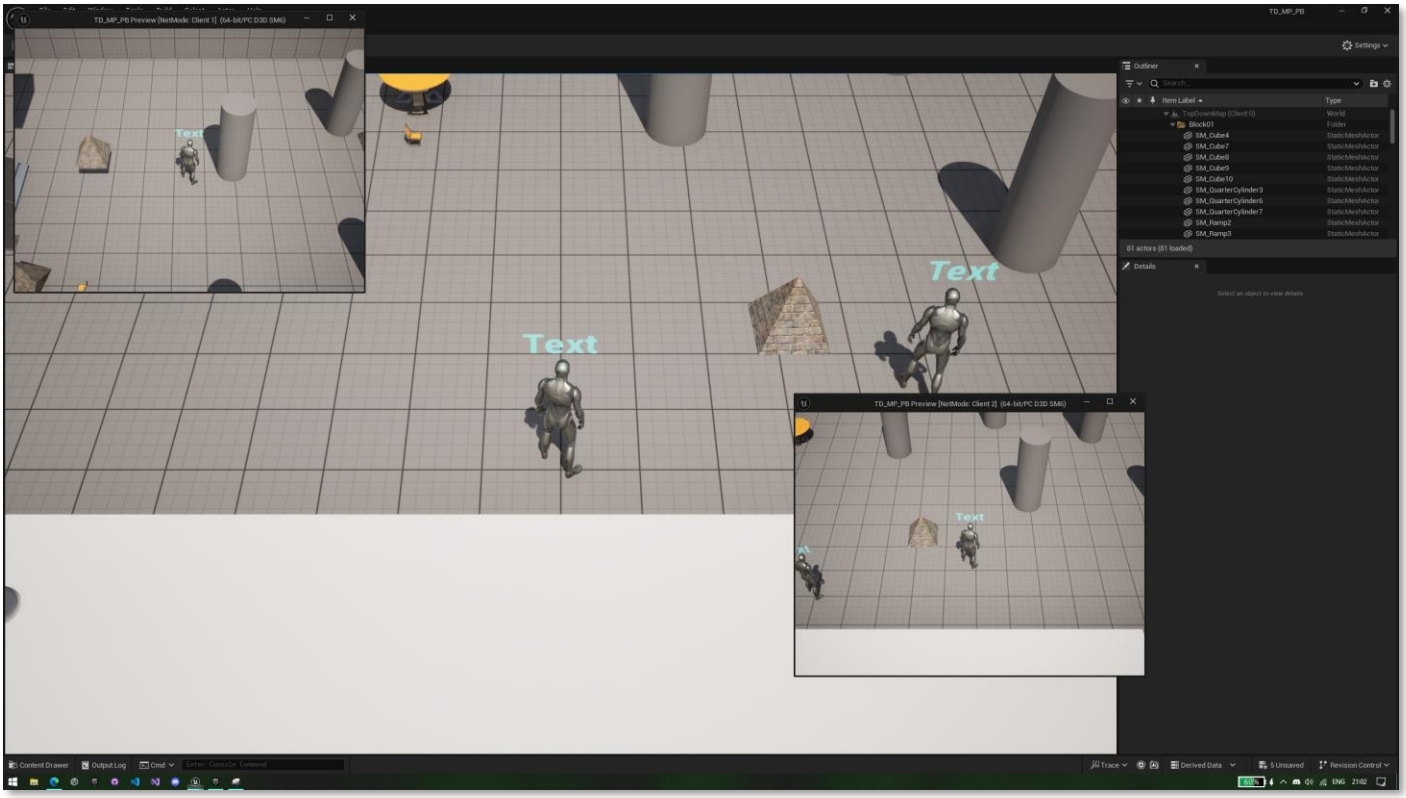

Fig 5.1.3.2: Screenshot showing play behaviour after addition of Switch Has Authority

5.2. Skinning characters

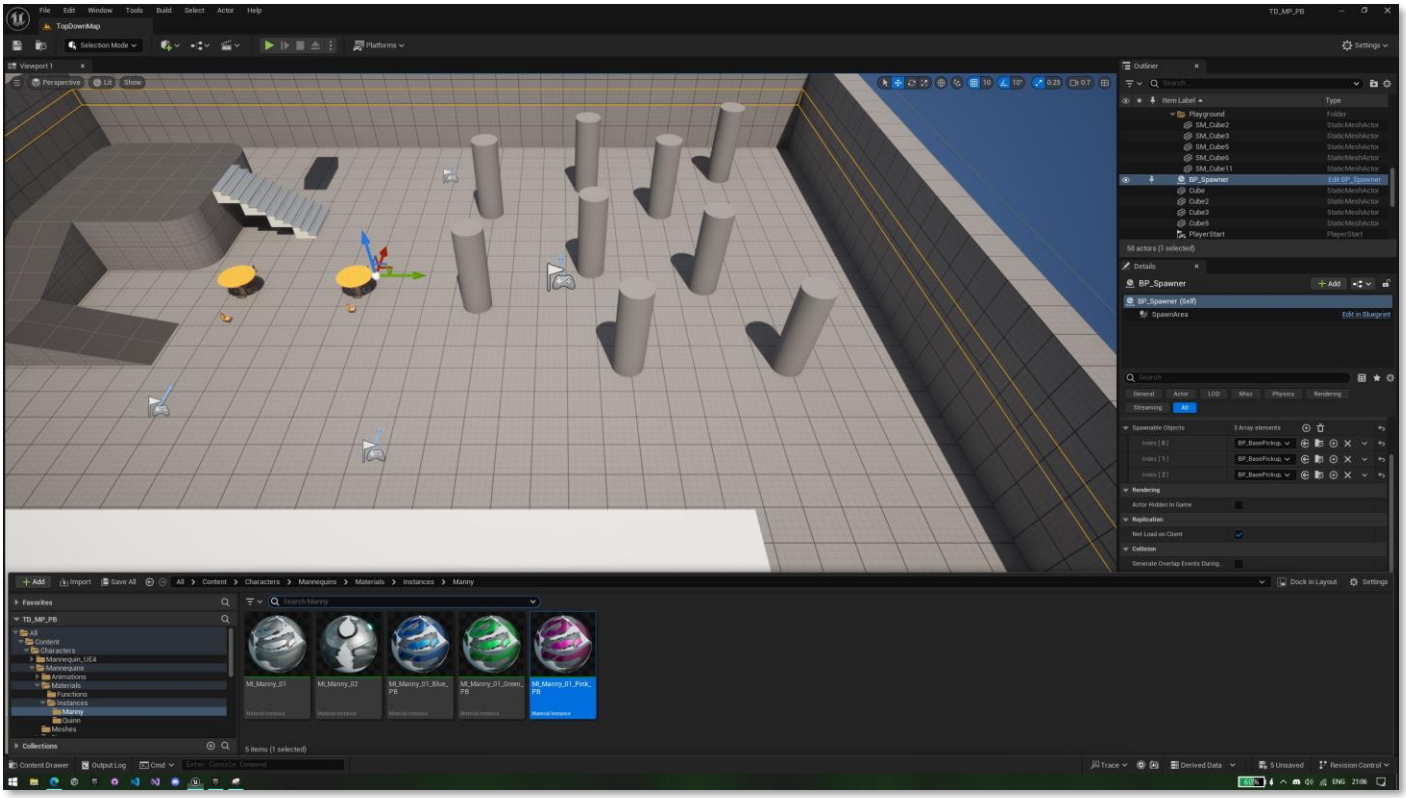

Fig 5.2.1: Screenshot showing new colours for Manny's Material (Tint)

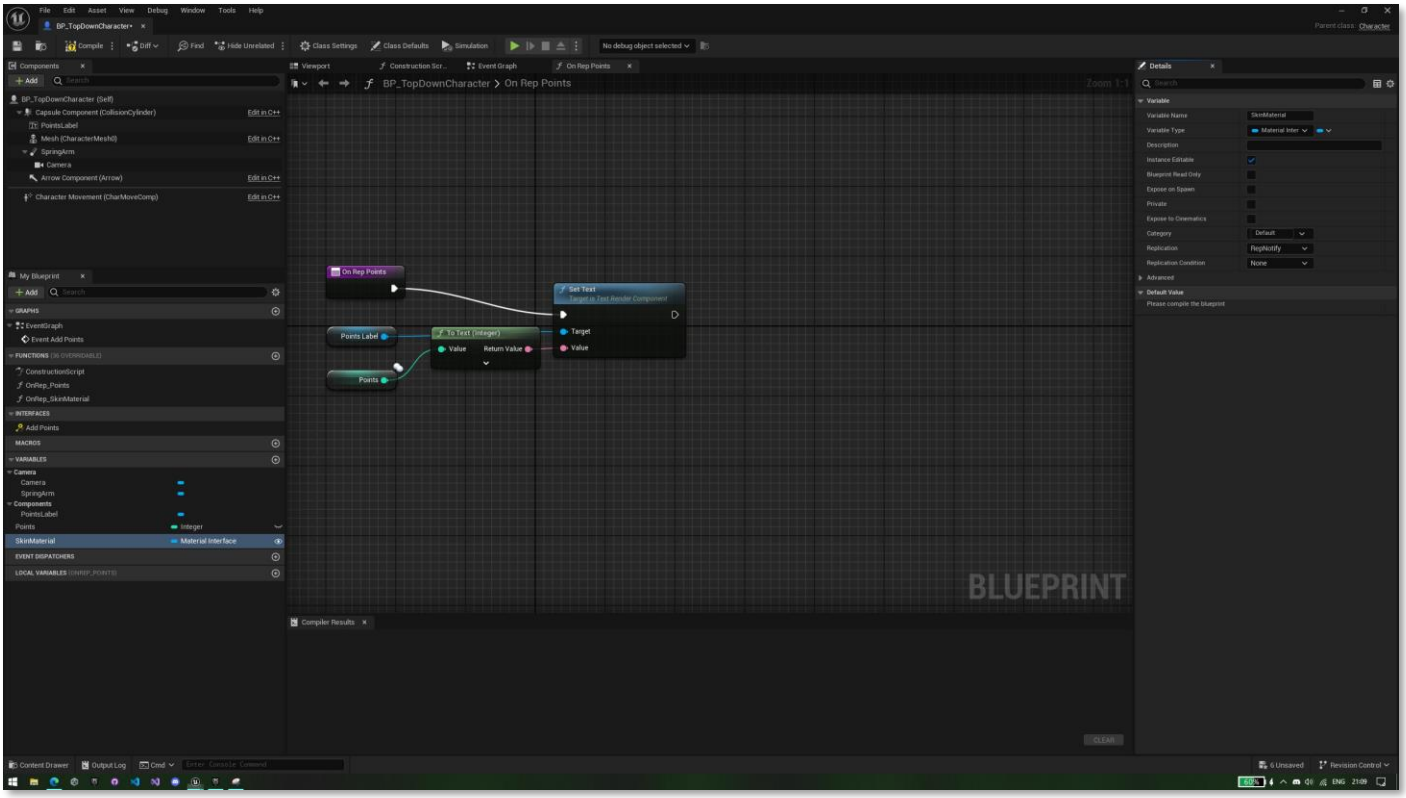

Fig 5.2.2: Screenshot showing addition of SkinMaterial with RepNotify

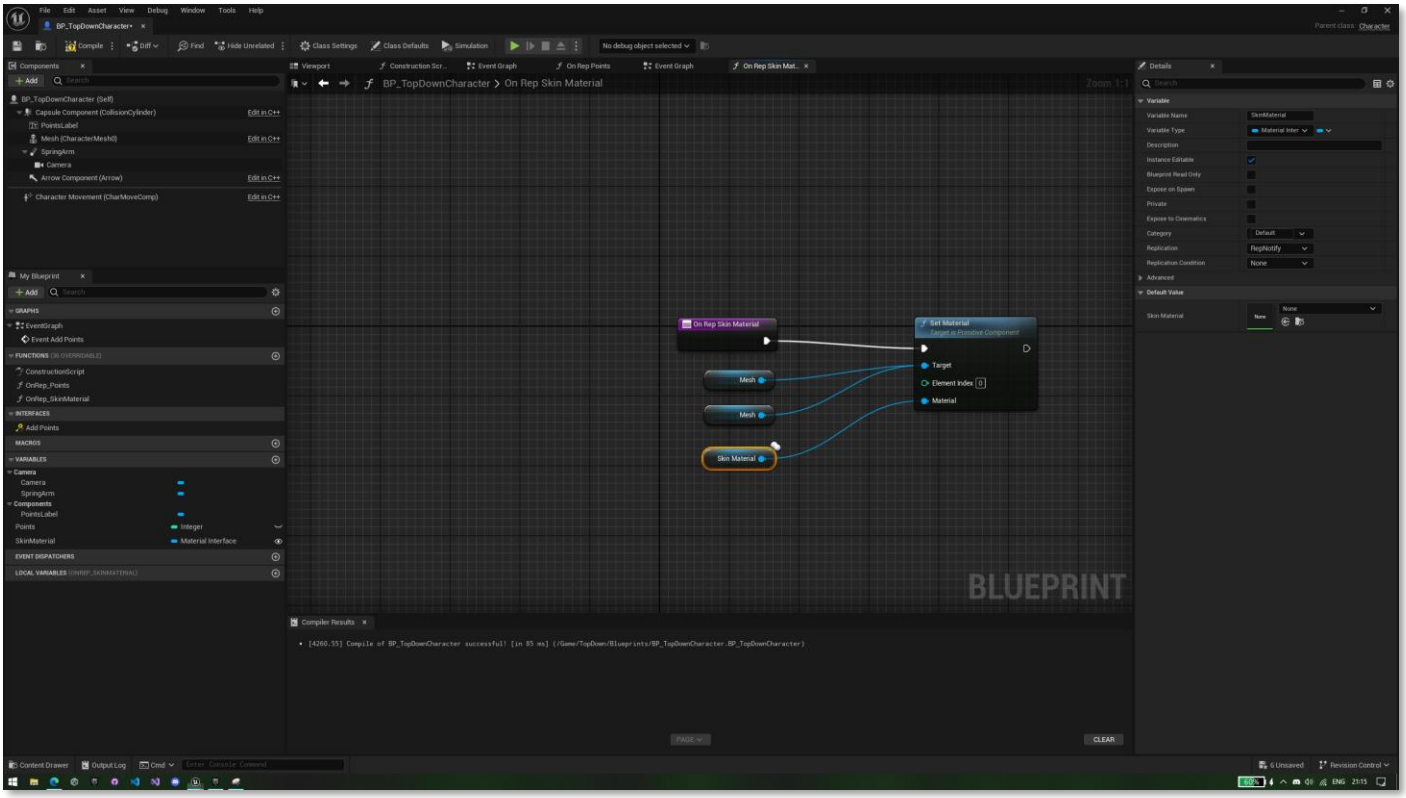

Fig 5.2.3: Screenshot showing material change in event graph in OnRepSkinMaterial

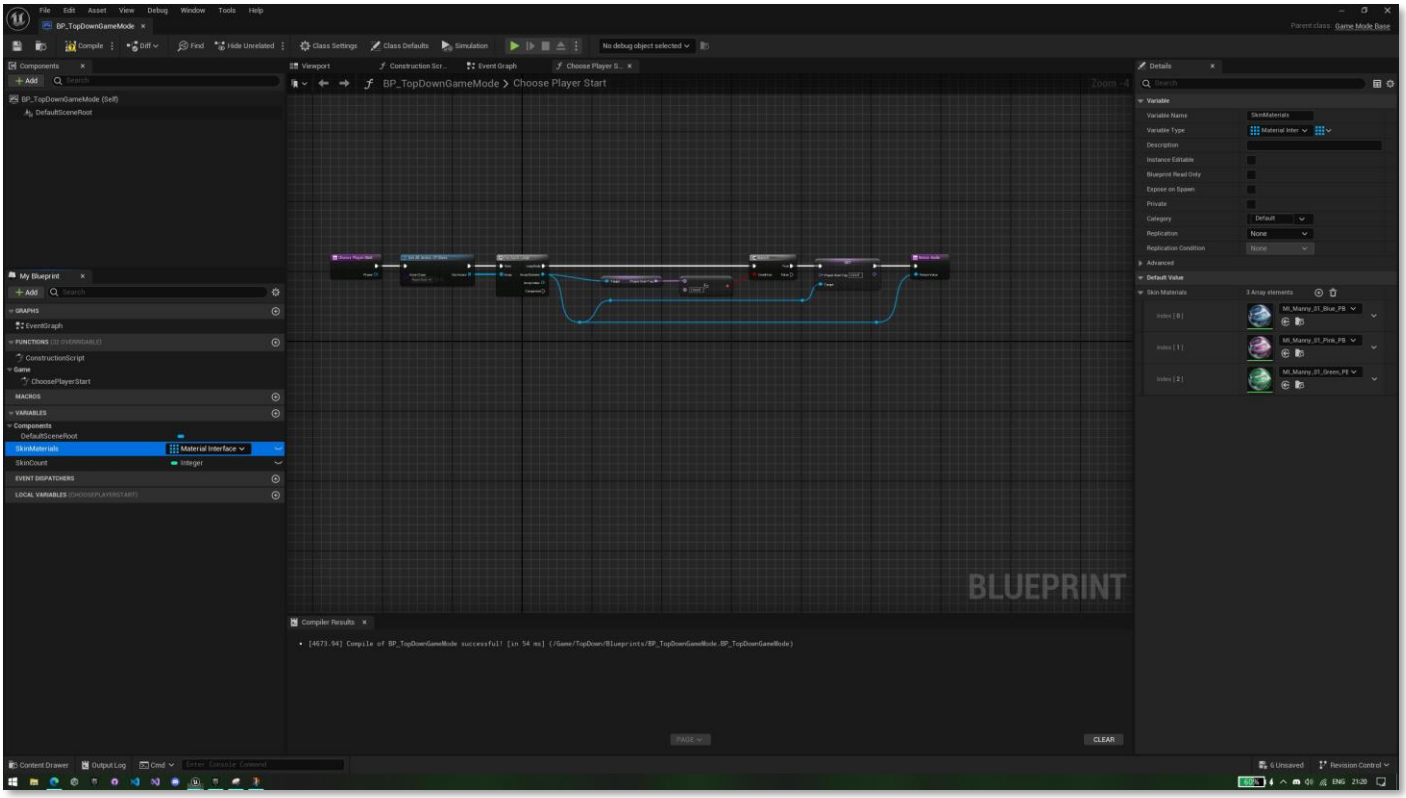

Fig 5.2.4: Screenshot showing material array and index

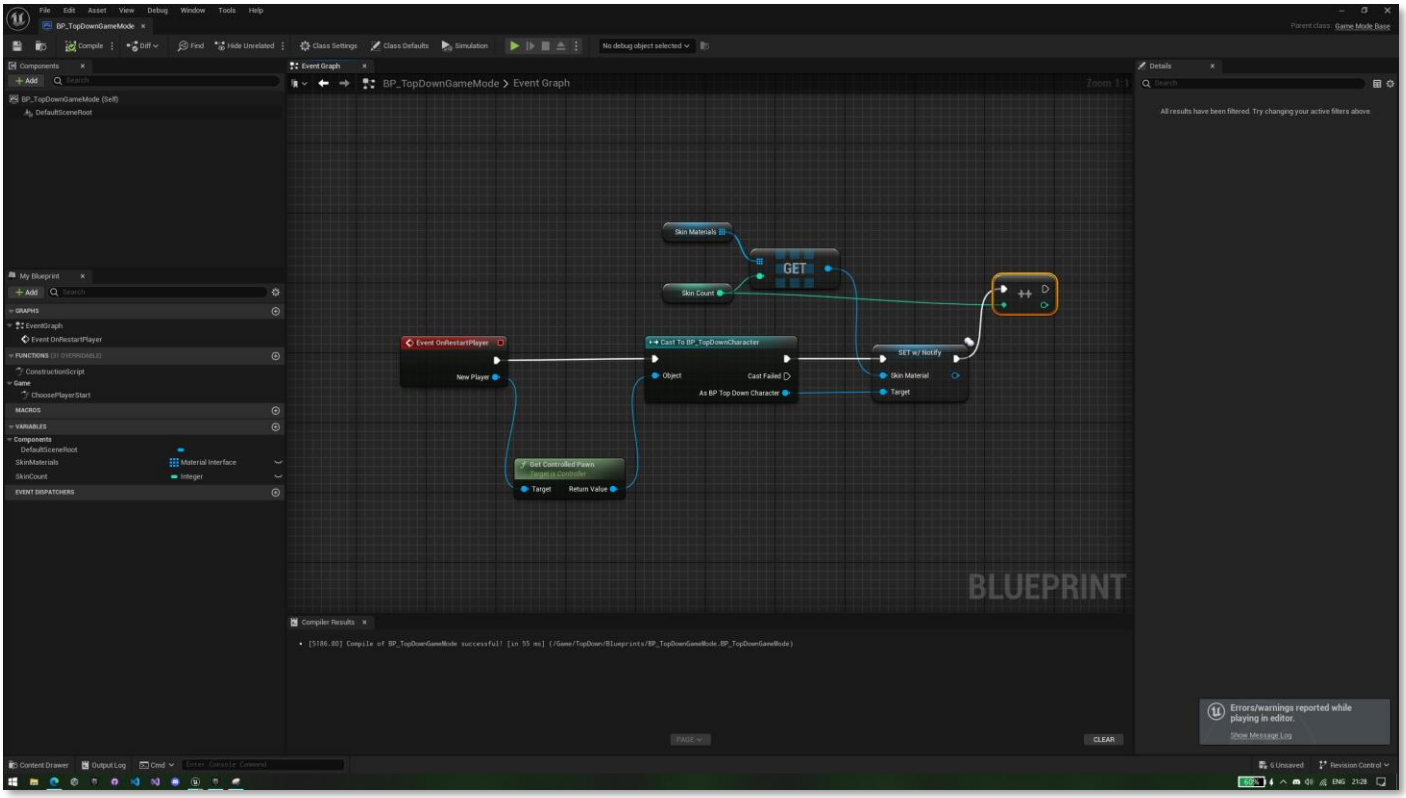

Fig 5.2.4: Screenshot showing switching

5.3. Testing the Game

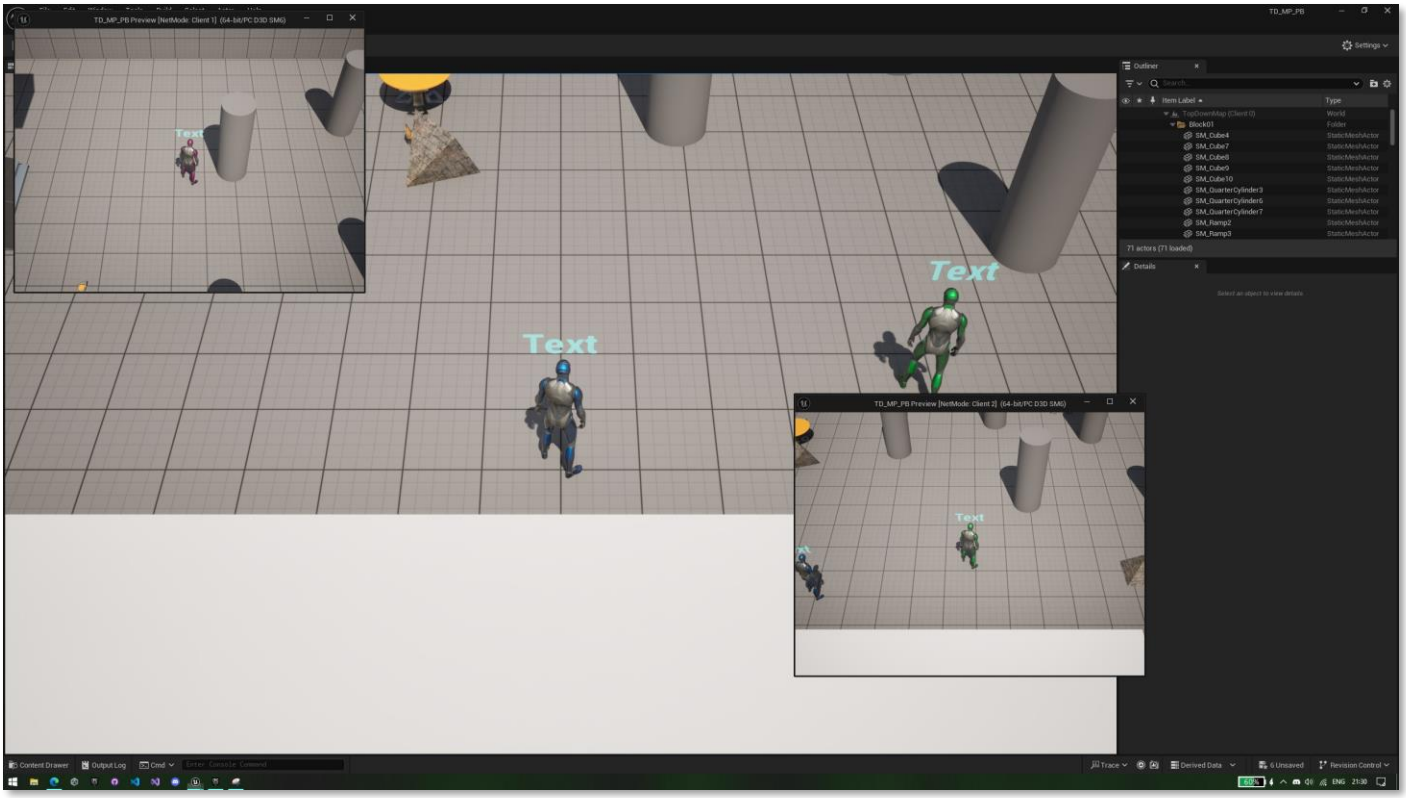

Fig 5.3.1: Screenshot of running the game with different skins.

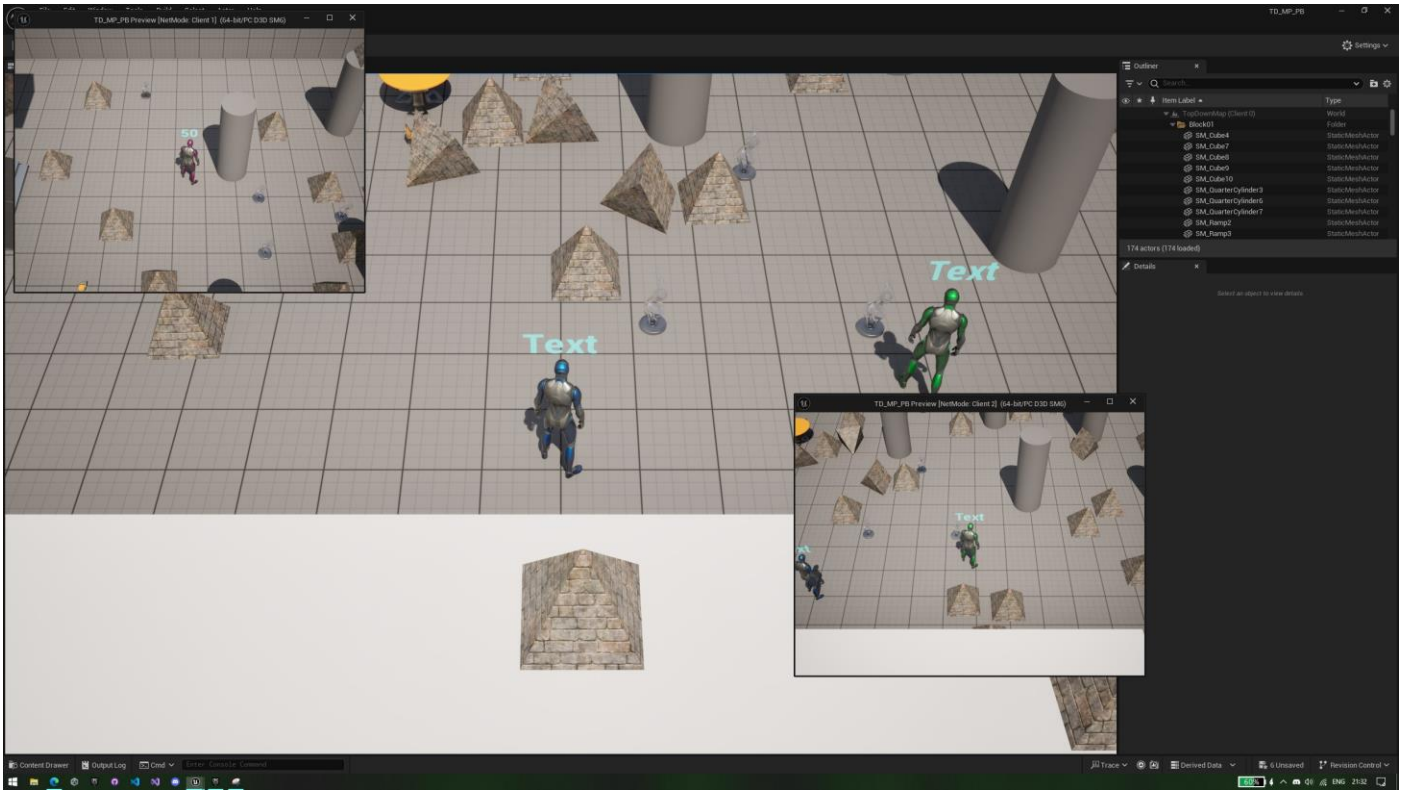

Fig 5.3.2: Screenshot of running the game with different skins and pickup props