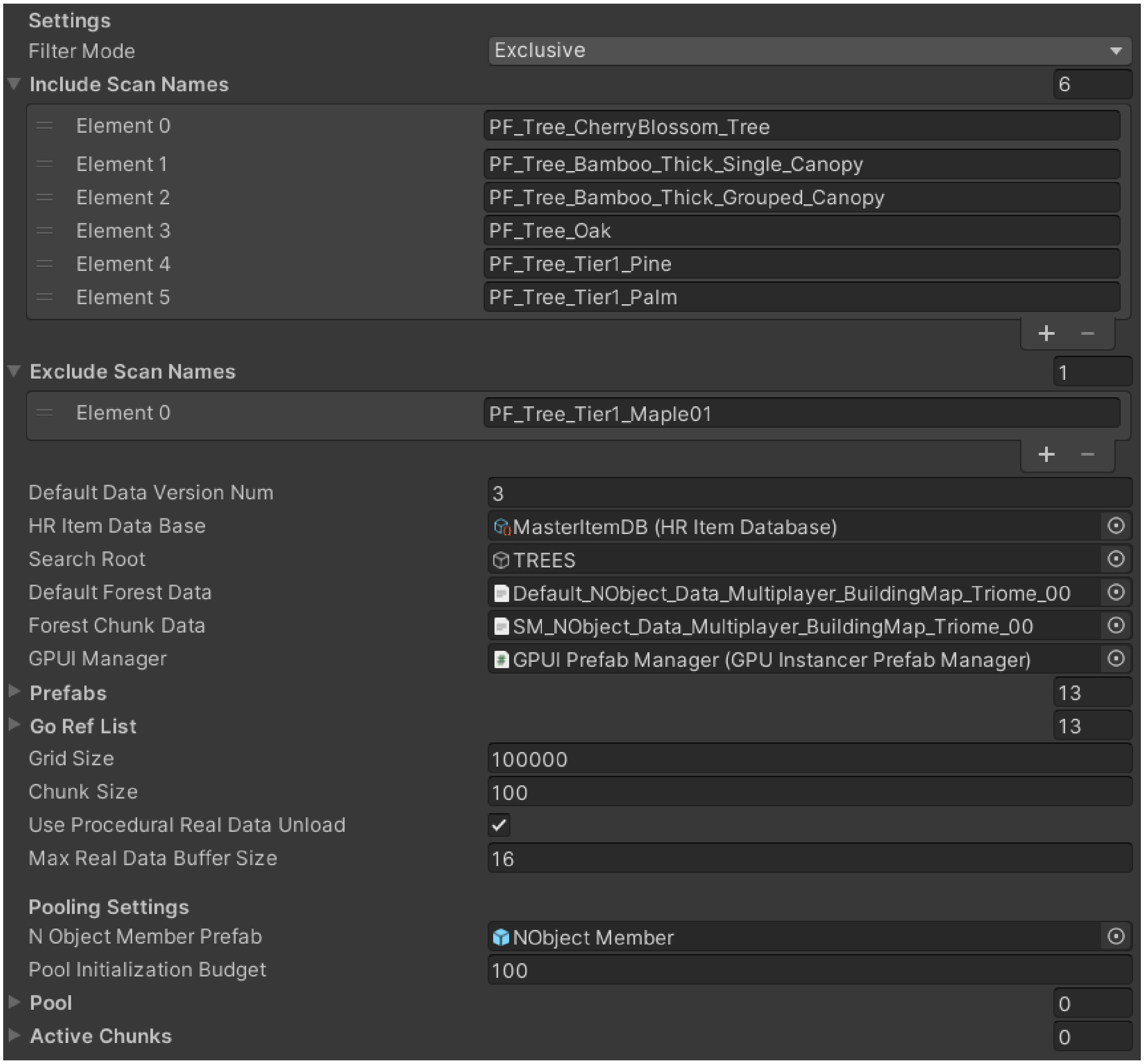
No GameObject Workflow

Editor Window:

1. NObjectManagerEditor (Attached to NObject Manager)



Settings

Filter Mode	Exclusive
Include Scan Names	6
Element 0	PF_Tree_CherryBlossom_Tree
Element 1	PF_Tree_Bamboo_Thick_Single_Canopy
Element 2	PF_Tree_Bamboo_Thick_Grouped_Canopy
Element 3	PF_Tree_Oak
Element 4	PF_Tree_Tier1_Pine
Element 5	PF_Tree_Tier1_Palm
Exclude Scan Names	1
Element 0	PF_Tree_Tier1_Maple01
Default Data Version Num	3
HR Item Data Base	MasterItemDB (HR Item Database)
Search Root	TREES
Default Forest Data	Default_NObject_Data_Multiplayer_BuildingMap_Triome_00
Forest Chunk Data	SM_NObject_Data_Multiplayer_BuildingMap_Triome_00
GPUI Manager	GPUI Prefab Manager (GPU Instancer Prefab Manager)
Prefabs	13
Go Ref List	13
Grid Size	100000
Chunk Size	100
Use Procedural Real Data Unload	☑
Max Real Data Buffer Size	16

Pooling Settings

N Object Member Prefab	NObject Member
Pool Initialization Budget	100
Pool	0
Active Chunks	0

BaseNObjectManagerEditor:

Settings:

Filter Mode: Inclusive (only include scans with tree names in the Include Scan Names array), Exclusive (excludes scanned tree names in the Exclude Scan Names array), Off (Scans in everything under the TREES Sectr layer that has pf_tree in its name)

Default Data Version Num: Whenever a new NObjectData generation is completed, this number is incremented by one to trigger the AutoDiffAndUpgradeData operation on players when they load. (ie. new tree is added by artist, players can have new tree in their data without losing previous save data)

Search Root: The Sectr root to search for trees

Default Forest Data: This data is baked in editor time to serve as the template for player data. Player will make a copy of this data on first launch.

Forest Chunk Data: This data is baked in editor time during spatial grid calculation for chunk member pooling and streaming.

Prefabs: The array holding reference to GPUIPrefabPrototypes

GoRefList: The array holding reference to the Prefab of real objects

Grid Size: For spatial grid (chunk) calculation. Leave at 100000, if edge chunks are missing, increase Grid Size.

Chunk Size: Increase to have each chunk include more trees, decrease to have less. Grid Size must be divisible by Chunk Size.

Use Procedural Real Data Unload: Automatically unload real trees back to NObjectData once the amount of real trees reach *MaxRealDataBufferSize.* Recommend to leave this on to prevent worst case scenario when player interacts with a lot of trees on the map.

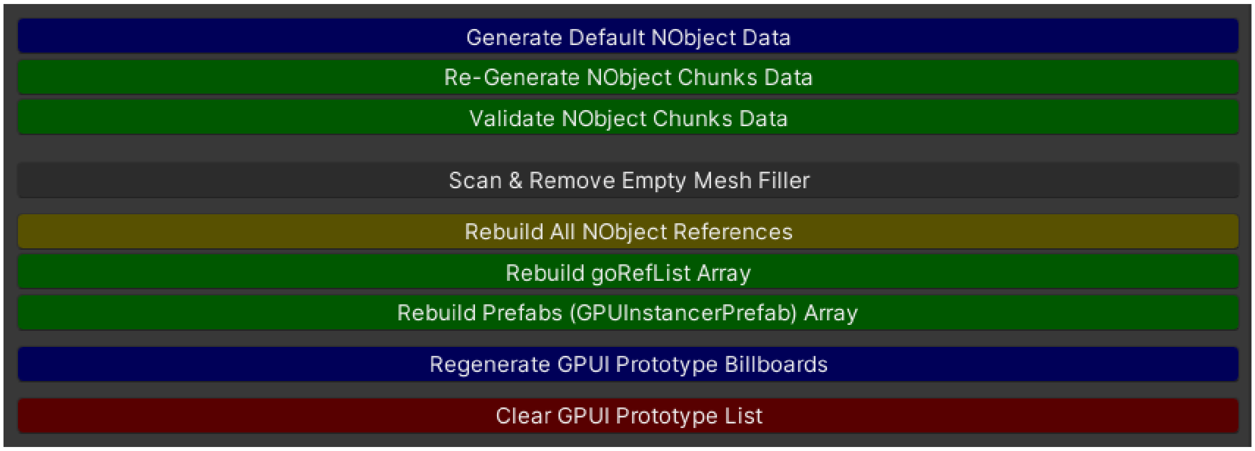
Max Real Data Buffer Size: Every time real data is loaded (NObjectData to Real Prefab), the data is added to Real Data Buffer. Increase buffer size to allow more real data present in scene.

Pooling Settings:

NObjectMemberPrefab: The Prefab for pooled members that serve as the connection between player interaction and NObjectData.

Pool Initialization Budget: How many members to allocate on start. If pool space runs out, more is automatically allocated. However, if space runs out fast, consider increasing the initialization budget.

Buttons:


Generate Default NObject Data
Re-Generate NObject Chunks Data
Validate NObject Chunks Data
Scan & Remove Empty Mesh Filler
Rebuild All NObject References
Rebuild goRefList Array
Rebuild Prefabs (GPUInstancerPrefab) Array
Regenerate GPUI Prototype Billboards
Clear GPUI Prototype List

Generate Default NObject Data: Creates default NObject data based on loaded Sectr trees under *Search Root* and *Scan Filter* mode. This data needs to be regenerated when new trees are added/placed by artists.

Re-Generate NObject Chunks Data: Recalculates spatial grid for all entries in NObjectData and group them to chunks. This data is automatically created by the above function, only regenerate if *Forest Chunk Data* corruption is suspected.

Validate NObject Chunks Data: Validates spatial grid calculation, passes if every data is assigned in a chunk correctly.

Scan & Remove Empty Mesh Filler: Scans all Prefabs in the GoRefList and removes MeshFillers containing null Mesh. This function was not used as correct tree Prefabs should not contain null mesh.

Rebuild All NObject References: Rebuilds all NObject reference data based on *Default Forest Data*. Automatically scans workspace for Prefab references and rebuild its respective GPUI Prototype. This will clear GPUI Prototype List and reattach GPUIPrefab script on Prefabs, so reimport of the prefabs will be triggered. Only neccessary if a new type of data is scanned in (ie. new type of tree variant)

Rebuild GoRefList: Rescans workspace for Prefab references based on *Default Forest Data* and store references in the GoRefList GameObject array.

Rebuild Prefabs Array: Rebuilds the Prefabs GPUInstancerPrefab array based on the GoRefList array.

Re-Generate GPUI Prototype Billboards: Regenerates billboards for all current prototypes in the Prefabs array. Billboard setting changed through GPUIPrefabManager will be lost.

Clear GPUI Prototype List: Removes GPUIPrefab component from all referenced prefabs in the Prefabs array, and clear GPUIPrefabManager's prototype array. No Undo, reimport of prefabs will be triggered, and prototype will need to be redefined through Rebuild All or manual define in GPUIPrefabManager.

QuickStart Tips:

- **Rebuild All NObject Reference is only needed if a new type of Prefab is added to Sectr trees, which will also need to regenerate NObjectData.**
- **Clear GPUI Prototype List before Rebuild All, if GPUIPrefabManager still shows leftover protypers, click again until no leftover is shown.**
- **To determine if GPUI is working correctly, load in game and see if registered instance count is more than 0 on GPUIPrefabManager. If all instance count show 0, GPUI is not working.**
- **GPUIPrefabManager settings store automatically even during runtime, so make sure to undo them if altering for testing purposes.**

Note: Some GPUI utility and editor functions are altered to automate NObject tools for this project, updating GPUInstancer may bring errors. For questions please ping me or DM on Discord - Kevin.