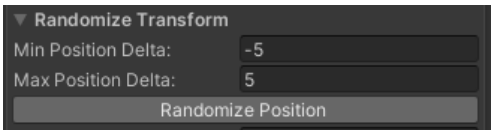
Unity Tools
Francisco Suarez

The following user manual explains how each function works and what to expect after executing them. The API for all the functions developed is annexed at the end of this document.

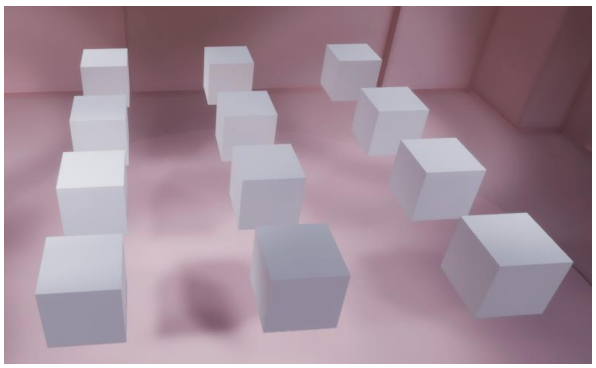
Randomize Transform

This tool consists of three sub tools, one per each element of a transform that can be triggered independently.

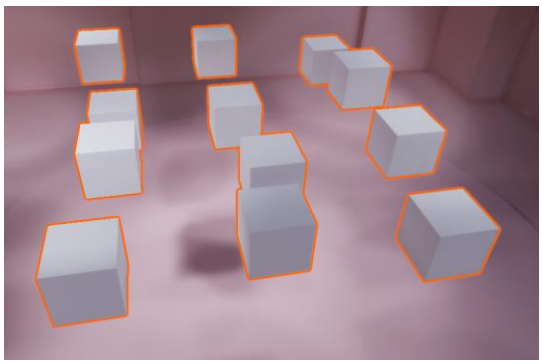
Randomize Position



It takes 2 parameters, *Min Position Delta* and *Max Position Delta*. For each selected object in the scene, three random numbers will be generated between both values and added as a delta to each position axis in the selected object.

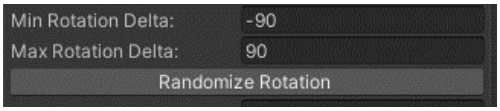


Original Scene

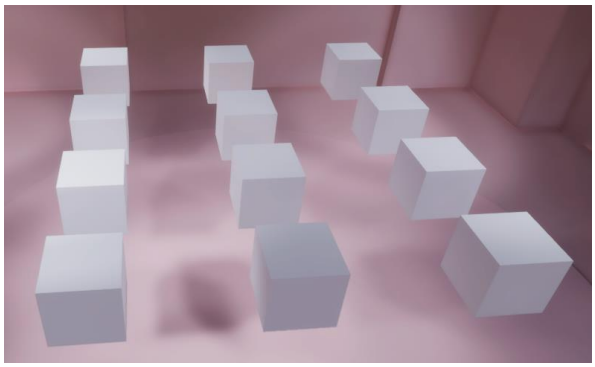


Result using [-0.2, 0.2] as parameters

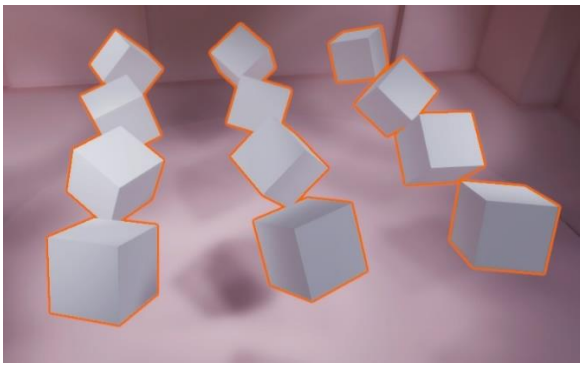
Randomize Rotation



For each selected Object, generates three random numbers between Min Rotation Delta and Max Rotation delta values, and adds them to each rotation axis values.



Original Scene

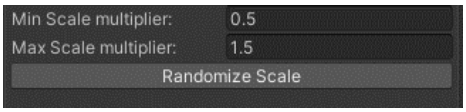


Result using [-90, 90] as parameters

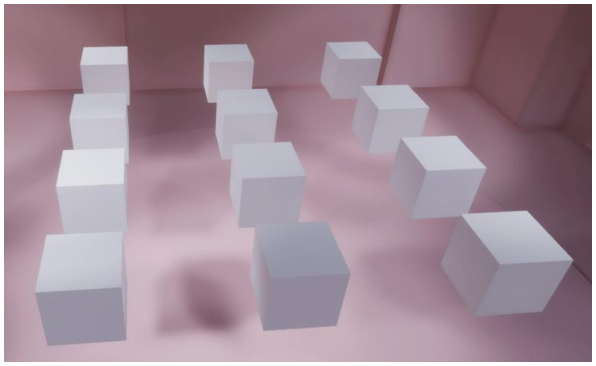
This function uses Euler Angles as the Unity Editor. By doing so, the results of the function in the editor inspector could be a little bit confusing, since Unity uses Quaternions as the internal representation of the rotation, and there is more than one way to represent any given rotation using Euler angles. The values in the inspector may be quite different from the values the function assigned.

Using Euler angles exposes the function to possible problems with Gimbal Lock, but given that the main objective of the function is to randomize the rotation, this shouldn't represent a problem after all and keeps the tool nice and simple.

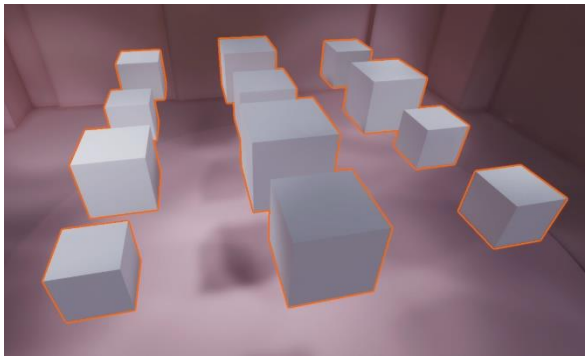
Randomize Scale



As in the previous tools, a random number will be generated for each selected scene object, between Min Scale Multiplier and Max Scale Multiplier, but this time, the same random number will be added to all three scale axis of the object, to avoid deforming it in the process.

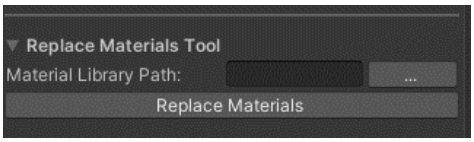


Original Scene



Result using [0.7, 1.4] as parameters

Replace Materials

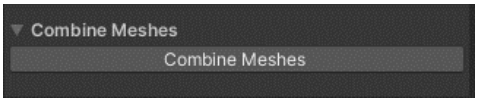


This tool takes as input the location (directory) of a Material Library in the current project. A material Library is just a collection of materials. If a material's name in a selected object matches one existing in the Material Library, the material will be replaced with the one existing in the library.

Original Scene

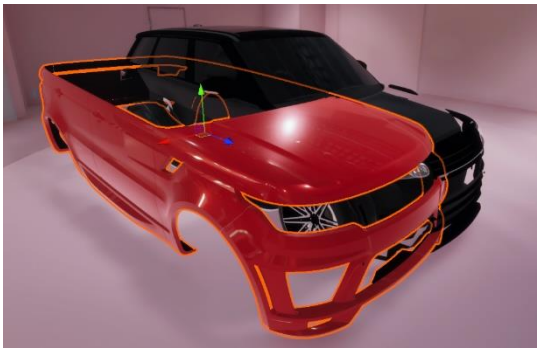
Result after the materials being replaced

Combine Meshes



For all selected objects in the scene, the tool finds all the meshes containing the same material list and merges them into a single mesh.

Original scene with different meshes with the same material assigned (exploded to visualize it better)



Result after Combining the meshes (displaced geometry for visualization purposes)

Useful Links

- Github Repository
- Tool's Flowchart
- Full Unity project download (scene to open: Scenes >> Showroom_01)

API

FranTools_Main.cs

This is the main class for all the tools contained in Fran's Tools. It constructs the main UI. Each tool has its own UI constructor and methods.

Functions:

Name	Type	Inputs	Output	Description
`ShowWindow()`	Public	None	None	Initializes the main UI window
`OnGUI()`	Public	None	None	Renders and handles GUI events
`OnEnable()`	Private	None	None	Instantiate objects for the tools.

Transform Randomizer.cs

Editor class to randomize values in the selected GameObjects transforms

Name	Type	Inputs	Output	Description
TransformRandomizer_UI()	Public	None	None	Main UI for the Transform Randomizer Tool.
PositionRandomizer_UI()	Private	None	None	UI section for the Position Randomizer tool.
RotationRandomizer_UI()	Private	None	None	UI section for the Rotation Randomizer tool.
ScaleRandomizer_UI()	Private	None	None	UI section for the Scale Randomizer tool.
RandomizePosition()	Private Static	float minPositionDelta float maxPositionDelta	None	Randomize the position of all the selected objects in the scene adding a random value to its position in all 3 axis.
RandomizeScale	Private Static	float minMultiplier float maxMultiplier	None	Randomize the scale of all selected objects in the scene, multiplying all 3 scale axis by a random number between minMultiplier and maxMultiplier.
RandomizeRotation	Private Static	float minRotationDelta float maxRotationDelta	None	Randomize the rotation of all the selected objects in the scene adding a different random value to its rotation in each axis.

MaterialsReplacement.cs

If a material's name in the selected object matches one existing in the Material Library, this class will replace the material in the object with the one existing in the library.

Functions:

Name	Type	Inputs	Output	Description
`ReplaceMaterials_UI()`	Public	None	None	UI section for the Replace Materials Tool.
`ReplaceMaterials()`	Private Static	string MaterialLibraryPath <optional parameter, default: "Assets/MaterialsLibrary">	None	Replace the materials in all selected objects with matching names materials in the Assets/MaterialsLibrary folder.

MeshCombiner.cs

Editor class to combine GameObject meshes that share the same Materials.

Name	Type	Inputs	Output	Description
MeshCombiner_UI()	Public	None	None	UI section for the Combine Meshes Tool.
MergeSameMaterialMeshes()	Private Static	None	None	For all selected objects in the scene, find all the meshes containing the same material list and merge them into a single mesh. The result is placed in the scene.
GetObjectsToMerge()	Private Static	None	List<List< GameObject >>	From the selected objects in the scene, finds which ones share the same materials list. These objects can be safely merged to lower drawcalls. Returns a list of lists of objects sharing the same materials list. E.G.:{ {BlueSphere, BlueCube} , {PurpleCone, PurpleTorus, PurpleCylinder} }
MergeMeshes()	Private Static	List<GameObject> objectsToMergeList string newObjectName	None	Merges meshes containing the same material list in a new object.
MaterialListContains()	Private Static	List<List<Material>> materialsListOFLists List<Material> MaterialListToCheck	bool	Checks if a material list from an object is contained in a list of lists of materials
MaterialListEquals()	Private Static	List<Material> materialListA List<Material> materialListB	bool	Checks if two material lists are exactly the same. Materials must be in the same order to return true.

Known issues

- Merging Meshes with multiple materials results in a new mesh with only one material slot.
- Merging meshes in the Range Rover FBX example without replacing the materials first, will cause all meshes to merge together.