Voxeliser Documentation

The vocaliser is a compute shader used to generate a voxel-esque model/effect from any generic model.
It should be noted that this shader is intensive, computation is all CPU based and having voxels set too small will cause lag.
This shader requires two parts: Mesh Setup, Voxel Setup

Mesh Setup

This shader effects a single model, so characters with several models to build clothes/ body/ hair will require the same steps on each model.

All models intended to be effect require the following: A Mesh Renderer or a Skinned Mesh Renderer attached to the parent or any of the children(Only the first child instance will be affected).

Secondly each model needs the following scripts: VoxeliserCompanion.cs (or derived script) and VoxeliserHandler.cs (or derived script)

Companion Script Details

Options(Only choose one script)

VoxeliserCompanion.cs
Used for a model which contain a Mesh Renderer and Mesh Filter.

VoxeliserCompanion_SkinnedMeshRenderer.cs
Used for a model with a Skinned Mesh Renderer.

Variables

VOXEL_COUNT:

This determines how many voxels will be instantiated at start. Allowing for object pooling, a technique to remove lag.

Setting this too low will cause voxels to be missing from the model at run time. Too high will cause additional lag.

UPDATE_N_FRAMES:

How often the voxeliser will update the voxel positions. For optimisation purposes it is N + 2 Frames. Example at a value of 0, the voxeliser will still only update every 2 frames.

Setting this higher will reduce game lag, however increase visual lag.

Handler Script Details

Options(Only choose one script)

VoxeliserHandler.cs

Used to allow voxels to be updated every frame, useful for any mesh that will move/rotate.

VoxeliserHandler_Trail.cs

Same as VoxeliserHandler.cs, however when a voxel is removed it will trail away from its original position.

VoxeliserHandler_Static.cs

This has two methods of use. First being a static object in the world. Voxels will be placed in the correct position at runtime and converted into a single mesh. Useful for environmental objects. Second use is mesh generation. If SaveMesh is set to true at runtime, a mesh will be generated and saved into the Assets base folder. You are then able to remove the original object and use the new mesh as desired.

Variables

Voxel

Gameobject prefab of a voxel. More detail about Voxel details are listed further down.

Voxel Size

This number alters the voxel scale. For example a voxel of 1, with a scale of 0.2, will be placed with a size of 0.2.

Chance Of Explode

When a voxelised model is disassembled, an explosion effect occurs to a random number of the voxels. Chance of a voxel exploding is this value out of 100. In the case of wanting no explosion desired set to 0.

Explosion Factor

When a voxel is exploded, the power or original velocity is determined by this value.

Shrink Time

When a voxel is exploded, it shrinks over this period of time till no longer visible.

Chance Of Trailing

As voxels are removed as they no longer fit the original mesh position there is a chance for a voxel to trail away. Chance of a voxel trailing is this value out of 100. In the case of wanting no trial desired set to 0.

Trailing Shrink Time

As a voxel trails away, it shrink over this period of time till no longer visible.

Trailing Speed

As a voxel trails away, it moves at this velocity.

Trailing Type

Trailing voxels can have 3 different effects: Backwards, Fixed, Local_Fixed

Backwards: All voxels will trail behind the models forwards direction. Example while running voxels trail behind the mesh

Fixed: Voxels will trail in the direction set by the Trail Direction variable regardless of the mesh's rotation. Example wind carrying voxels in a single direction.

Fixed Local: Voxels will trail in the direction set by the Trail Direction variable taking into account the mesh rotation. Example Backwards effect can be recreated by setting -1 on z axis of the Trail Direction.

In the case of wanting a more custom effect, for example a variable wind a derived script can be used to override the methods GetTrailDirection() and GetChanceOfTrailing().

Trail Direction

As a voxel trails away, it moves in this direction when using any Fixed or Fixed Local modes.

Save Mesh On Play

This causes a mesh to be saved into the assets base folder on play. When enabled a dialogue box will appear asking for a name.

Voxel Setup

A voxel is required to be a prefab containing any type of mesh, however the basic cube mesh is recommended.

Each voxel prefab requires the Voxel.cs(Or derived script) script attached.

Finally it is recommended that the only components attached are the script and the mesh renderer/filter due the large amount of voxels used. Having a box collider attached will cause high levels of lag.

Voxel Script Details

Options(Only choose one script)

Voxel.cs

Used for a basic voxel

Voxel_Trailing.cs

Used for trailing effect, must be used with VoxeliserHandler_Trail.cs or no effect will occur.