Voxel Terrain Editor

Example Scene: Open scene file Assets/VoxelTerrainEditor/SceneExample/VoxelTerrainScene.scene

Setting Voxel Terrain Size and Textures: Click on the VTerrain object in the Hierarchy and in the inspector you can set the terrain size(width, height and depth). Textures can be changed on material Resources/TerrainMaterial. To change size create a new terrain by assigning new TerrainName or delete current terrain with that name in StreamingAssets/voxelLevels/ so it gets recreated.

There is no limit to the size of the terrain, the only limit is your memory. If you want big terrain do that smart, like 1024x64x1024 not 1024x1024x1024. Voxel terrain memory consumption is much higher compared to standard terrains created from heightmap. Voxel use 3d arrays, so set the terrain size that your computer or device can support.

RES: Means resolution(chunk size) . You can tune size and chunk sizes to get best result;

Save And Load Terrain:

```
VoxTerrain.Instance.saveMap("TerrainName");    // save terrain, set your terrain name
VoxTerrain.Instance.loadMap("TerrainName");   //  load terrain, after this call ReBuildCollider();
VoxTerrain.Instance.ReBuildCollider();
SaveLoad.getSavedMaps();    // returns string[] with list of all saved terrains
```

All saved terrains are in StreamingAssets folder in binary .bytes format.

To load other saved terrain to the editor click on the Window/Voxel menu than VoxelEditorLoader and select level you want to load.

Terrains saved on the mobile devices are saved to the PersistentDataPath instead since StreamingAssetsPath is not writable on mobile platforms.

Draw:

```
VoxTerrain.Instance.Draw3D(Vector3 position, Vector3 scale, OBJ obj, EFFECT BrushEffect, Color color,bool
doSculpt,bool doPaint)

// OBJ is VoxTerrain.Instance.OBJ.SPHERE, CUBE, RANDOM,PLANE...
// EFFECT is VoxTerrain.Instance.EFFECT.ADD, SUB, PAINT

// textureID argument: define textures and pass it as  textureID argument when calling Draw3D

        static short grass = 0; //texture 1
        static short  stone = 1; //texture 2
        static short  ground = 2; //texture 3
        static short  white = 3; //White or no texture
```

After Draw3D call VoxTerrain.Instance.ReBuildCollider(); // if BrushAndGUI disabled

BrushAndGUI: GUI and Draw on click(Example script). If this script is disabled call VoxTerrain.Instance.ReBuildCollider(); in your script.

In GUI You have 3 modes: sculpt, paint and combined. Sculpt is only drawing, paint is only texture paint, combined is both.You can switch between textures with numbers on your keyboard. Use mouse to navigate or Fly with camera on WSAD and rotate on Arrows. Brush size is size of your brush.

Negative values remove Voxels. Try all brush sizes...

To adjust camera movement speed or mouse sensitivity open BrushAndGUI.cs and change variables named: dragCameraSpeed, rotateCameraSpeed, scrollCameraSpeed and flyCameraSpeed.

VersionOnePointZeroPointFour.zip ? If you want to learn how Voxels works it's easier to learn from that version. It's runitime version only, try it...

Interaction with the terrain: Terrain chunks have mesh colliders, standard raycast can be used to get a point on the mesh.

Soldier(Available in previous versions, check PreviousVersions folder): Soldier model with SoldierScript is here to see how to use character in your game. For mobile devices use some model with less polygons. Right click to move soldier. If you use Unity 4+ and get errors about animation click Soldiers Rig in soldier prefab and set the animation Type to Legacy. Than add new soldier to the scene and attach SoldierScript and character controller to him. Set first animation to Idle and second to Run. Finally set position on terrain higher so it can not fall through terrain.

Saving Voxel Terrain to Prefab(Planned for future updates): If you don't need destructible terrain in your game then you can save the Terrain object to Prefab and use it as standard game object to get performance and loading times.

Export To OBJ: Export OBJ in Window/Voxel menu is a script that exports Voxel Terrain or other object to OBJ Format that can be used in 3d modeling programs as Blender. Blender Obj importer does not support Vertex Colors so it's not possible to export Texture blends. I will look to add more formats and solutions to export terrain from Unity. For info how to make Tri planar node setup in Blender look: http://blenderartists.org/forum/showthread.php?265996-Cycles-Tri-planar-Texturing&s=d45e53288b8a22d5f24c330ec8175150

Feel free to ask everything

email: bravecharlie101@gmail.com

http://www.youtube.com/watch?v=YJhk5-eV1nc&feature=youtu.be
http://www.youtube.com/watch?v=WNmmjDNCQMQ