Layout

The project contains the following layout:

- Assets
  - AssetStoreTools
  - Code
  - CritiasTreeSystem
    - Code
    - Data
    - Shader
  - Editor
  - Mesh
  - Prefabs
  - Scenes
  - StandardAssets
  - Terrains
  - Trextures

From those only 'CritiasTreeSystem' is important, you only need to copy that for your project. It contains all the required dependencies for the system to work. You can safely clean the 'Data' folder that contains some data specific to this project.

Installation

Copy the folder 'CritiasTreeSystem' anywhere in your project where it suits your needs.

Setup

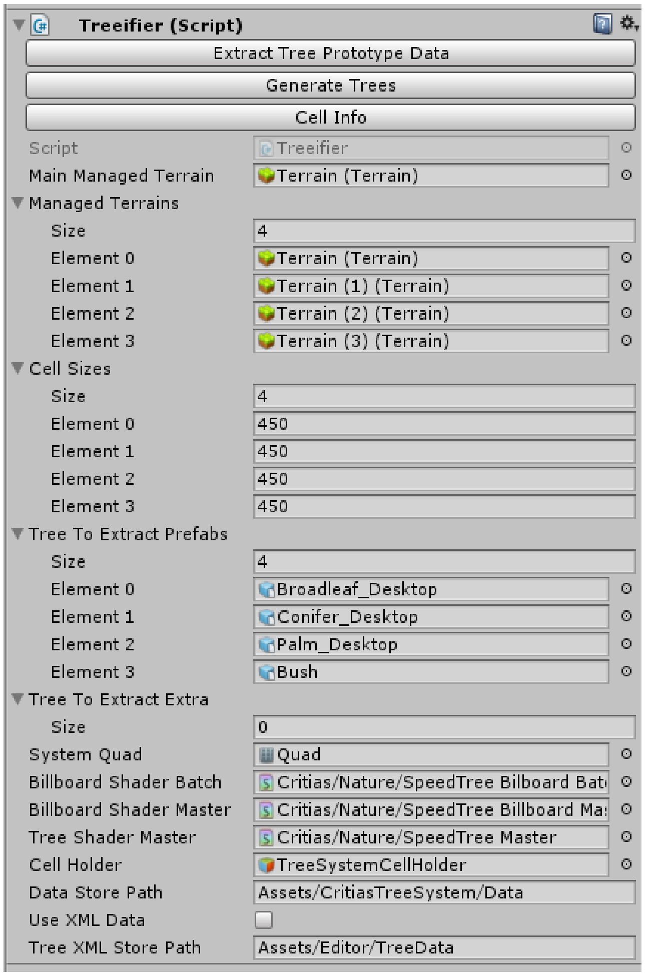
In order to setup the tree system you need to take the necessary preparation steps. After you have taken them, a simple press of a button will do all the work for you, for the rest of your project!

1. Create a game object and add it the 'Treeifier' script. That script is responsible for generating all the offline data. It contains the following fields:
    - **Main Managed Terrain:** The main managed terrain, used for some tree name hash checks. Make sure that it is set to whatever terrain you have that contains all the trees that you'll use.
    - **Managed Terrains:** The secondary managed terrains. Make sure that the main managed terrain IS contained in that array. Add all the terrain that you want to influence the system.
    - **Cell Sizes:** The tree cell size. A larger size will result in more trees tested for drawing but more CPU overhead. A smaller size will result in less trees tested for rendering but more CPU overhead. It is recommended to use larger sizes for larger terrains and smaller sizes for smaller terrains. For a terrain of 2250m a cell size of 450m will be just fine. If your terrain is really big (let's say 10km), consider increasing the size(to something like 2000m). The cell sizes must also be of a fixed size, so if you can't get a

number out of your mind, just use the **'Cell Info'** button that will print to you the possible cell sizes.

- **Tree To Extract Prefabs:** The SpeedTree trees (.spm) that are going to be used inside our system. If you plan on extracting a tree but don't add it to this list, it is not going to be extracted. The must also have 'LODGroups' and 'BillboardRenderers'.
- **Tree To Extract Extra:** The game objects that have as children trees that have as prefab owner the trees added to the **'Tree To Extract Prefabs'** array.
- **System Quad:** The default system quad. Make sure that it's set.
- **Billboard Shader Batch:** The tree billboard batch shader. Make sure that it's set to *'Critias/Nature/SpeedTree Bilboard Batch'*.
- **Billboard Shader Master:** The tree billboard master shader. Make sure that it's set to *'Critias/Nature/SpeedTree Billboard Master'*.
- **Tree Shader Master:** The mesh tree shader. Make sure that it's set to *'Critias/Nature/SpeedTree Master'*.
- **Cell Holder:** The game object that will hold the tree generated billboards and boxes after generating the trees.
- **Data Store Path:** When extracting the trees prototype data, this is the path where the data is going to be stored.
- **Use XML Data:** Only used if you have the original SpeedTree XML which is probably not the case.
- **Tree XML Store Path:** Only use if you have the original SpeedTree XML which is probably not the case.

Default demo setup:



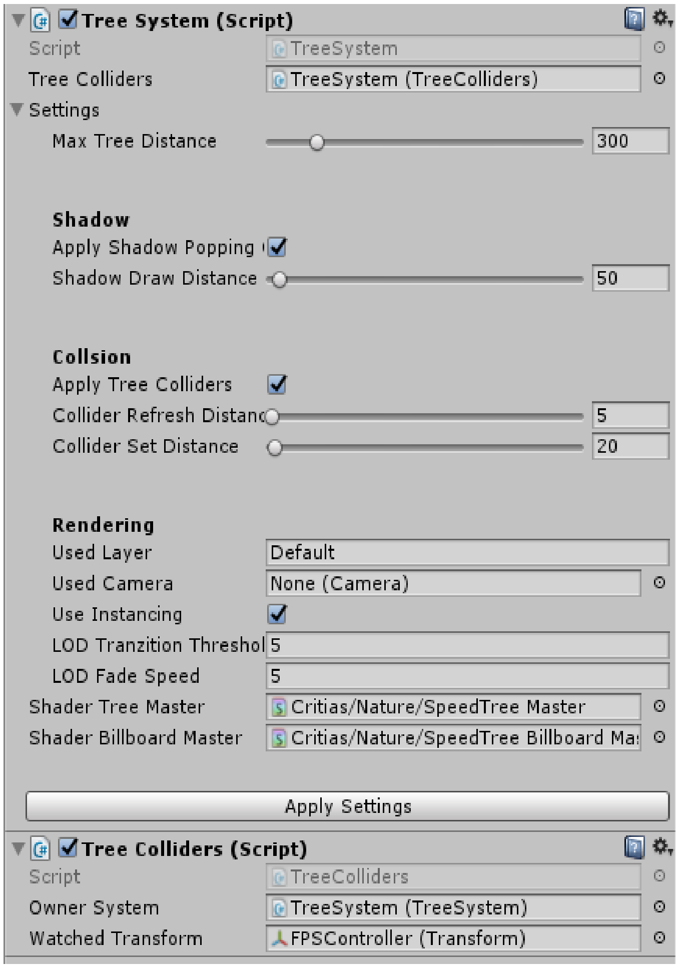
Script	Treeifier
Main Managed Terrain	Terrain (Terrain)
Managed Terrains	
Size	4
Element 0	Terrain (Terrain)
Element 1	Terrain (1) (Terrain)
Element 2	Terrain (2) (Terrain)
Element 3	Terrain (3) (Terrain)
Cell Sizes	
Size	4
Element 0	450
Element 1	450
Element 2	450
Element 3	450
Tree To Extract Prefabs	
Size	4
Element 0	Broadleaf_Desktop
Element 1	Conifer_Desktop
Element 2	Palm_Desktop
Element 3	Bush
Tree To Extract Extra	
Size	0
System Quad	Quad
Billboard Shader Batch	Critias/Nature/SpeedTree Bilboard Bat
Billboard Shader Master	Critias/Nature/SpeedTree Billboard Ma
Tree Shader Master	Critias/Nature/SpeedTree Master
Cell Holder	TreeSystemCellHolder
Data Store Path	Assets/CritiasTreeSystem/Data
Use XML Data	□
Tree XML Store Path	Assets/Editor/TreeData

2. Create a game object and add it the 'Tree System' and 'Tree Colliders' scripts. Those scripts are responsible for generating and drawing the runtime-data. They contain the following fields:
   I. Tree System
      - **Tree Colliders:** Reference to the 'Tree Colliders' script. Make sure that it's set.
      - **Settings**
        - **Max Tree Distance:** Maximum distance at which we'll have mesh trees.
        - **Apply Shadow Popping Correction:** If shadow popping is too bad for your setup use this.
        - **Shadow Draw Distance:** If we use shadow popping correction this is the distance at which we are going to force a tree to be drawn as 'ShadowOnly'.
        - **Apply Tree Colliders:** Use this in order to update tree colliders. Do not use it if you have a terrain that doesn't draw the trees but manages their collisions.
        - **Collider Refresh Distance:** After how much player movement the colliders are going to be refreshed. Use a larger distance for less frequent updates.
        - **Collider Set Distance:** After a collision refresh set the distance at which colliders are going to be set. For example a distance of 100m will have colliders set for trees at 100m around your player.
        - **Used Layer:** Layer on which we are going to draw the trees on. Defaults to the 'Default' layer.
        - **Used Camera:** The camera that we are going to use for drawing the trees and calculating occlusion with the 'CullingGroupAPI'. If it is null, the main camera is going to be used for occlusion and the trees are going to be drawn for ALL the scene's cameras.
        - **Use Instancing:** If we should use GPU instancing. Will tremendously improve FPS, especially if we have a high tree density.
        - **LOD Tranzition Threshold:** The distance over which we are going to apply the SpeedTree's cross-fade.
        - **LOD Fade Speed:** Cross-fade animation speed and tree-to-billboard/billboard-to-tree transition speed
      - **Shader Tree Master:** Shader tree master, make sure it's set just like the 'Treeifier'
      - **Shader Billboard Master:** Shader billboard master, make sure it's set.

   II. Tree Colliders
      - **Owner System:** Owner 'TreeSystem' script, make sure to set it.
      - **Watched Transform:** The transform that we are going to watch in order to trigger a tree collision refresh.

Note: After modifying these values in the inspector, make sure to press 'ApplySettings' if you are in play mode.

Default demo setup:



☑ Tree System (Script)

Script: TreeSystem

Tree Colliders: TreeSystem (TreeColliders)

Settings

Max Tree Distance: 300

Shadow

Apply Shadow Popping ☑

Shadow Draw Distance: 50

Collsion

Apply Tree Colliders ☑

Collider Refresh Distanc: 5

Collider Set Distance: 20

Rendering

Used Layer: Default

Used Camera: None (Camera)

Use Instancing ☑

LOD Tranzition Threshol: 5

LOD Fade Speed: 5

Shader Tree Master: Critias/Nature/SpeedTree Master

Shader Billboard Master: Critias/Nature/SpeedTree Billboard Ma

☑ Tree Colliders (Script)

Script: TreeColliders

Owner System: TreeSystem (TreeSystem)

Watched Transform: FPSController (Transform)

Usage

The usage is quite simple.

1. Follow the 'Setup' described above.
2. Paint your trees on the terrain with your favorite painting/placing method. Add the extra trees on your mesh cliffs, mountains, and any other object that is within a terrain.
3. Press 'Extract Tree Prototype Data' in the 'Treeifier' script. NOTE: You only need to do this once if you did not changed the 'Tree To Extract Prefabs'
4. Press 'Generate Trees' and hit 'Play' in order to see your awesome creation at awesome FPS!
5. Paint/place/tweak your trees again.
6. Go to step '4' again.

Note: When you'll press 'Play' the managed terrain's foliage rendering is turned off. So I would recommend using two terrains, one for the trees and another for the grass.
Note: There's also a script called 'TerrainUtils' that takes care of some of that management for you.