Fantasy Adventure Environment

by Staggart Creations

1. Fantasy Adventure Environment

Thank you for purchasing the Fantasy Adventure Environment!

Please consider rating the package through your download list **or leave a review at** the store page **once you're familiar with it.**

Your feedback is instrumental to the future of this package!

Rate   Review

Feedback and suggestions can be made in the forum thread:

https://forum.unity3d.com/threads/486102/

This manual is intended to clarify the use of the package.

This minimum supported version at this time is Unity 5.4

2. Getting started

Importing

Import the Unity package through the Asset Store download manager (Found under Window->Asset Store). Please be sure to download the assets in your current Unity version, as the package has been submitted separately for Unity 5.4 and up, and 2017.1 and up.

You may receive a warning message about obsolete APIs, it is completely safe to run the API updater.
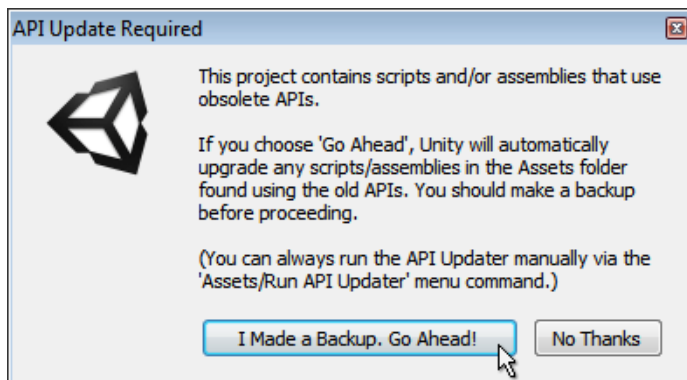
There will be several warnings in the console regarding missing scripts. This is harmless and regards 3rd party scripts used in making the scene. Warnings regarding incompatible lighting or occlusion culling data fall within this as the package hasn't been submitted for every Unity version (it would take several days to do). This merely pertains to the demo scene and has no negative effect on the usage of the package.

Next, open up the included demo scene, which is intended to demonstrate the practical uses of the assets. In order to start building your world, you can browse the prefabs in the */Prefabs* folder and use them as you see fit.

Unity terrain

It's important to note that the vegetation items will not work on the terrain as "Detail objects". This is because they have LODs, which is incompatible. It is also undesirable to use the assets in this way, since the shader would

be overridden by the built-in terrain grass shader.

Instead, add the vegetation items as "Tree Objects".

I highly recommend using tools such as Easy Scatter or Mesh Brush, which have a lot more options for distribution.

Other topics

- Environment set up
- Features
- Customizing assets
- Scripts
- Shaders

3. Environment set up

In order to recreate the demo scene's environmental settings the following steps must be taken:

Go to *Window/Lighting*

- Set the Ambient Source to a white color in the Lighting tab found under *Window/Lighting.*
- Enable Fog, set to *Exponential* with a value of *0.007*
- Assign the *FAE_Skybox* material to the *Skybox* field.
- Disable *Precomputed Realtime GI*

Directional Light:

- On your directional light, assign the *FAE_Lensflare* asset in the *Flare* field if you wish.
- Control the intensity of the sun shaft effects through the Directional Light's color's alpha channel

Post effects:

- Grab the Post Processing Stack package and use the include *FAE_PostProcessingProfile* profile asset if you wish.

Scene:

- Add a WindController object to your scene, for the wind animations on the trees and vegetation to work. For ease of use, you can find a prefab for this in the *Fantasy Adventure Environment/Prefabs/Effects* folder. It should be noted it is merely required to set the initial wind parameters, you can safely remove this object from your scene and the wind will continue to work.
- Select your terrain and add the *PigmentmapGenerator* script. The first time, the pigment map will automatically be generated.
- (Optional) Add a CliffAppearance object to your scene, which will allow you to modify the material properties of all rocks at once. You can find a prefab for this in the *Fantasy Adventure Environment/Prefabs/Effects* folder, which is pre-configured with the default rock materials.
- (Optional) Look for the FoliageBender prefab in the */Prefabs/Effects* and drag it into your player, move the object to the feet/base of the player. This will make the grass and foliage bend away from this position.

More detailed information on these features can be found in the scripts *section.*

4. Support

Should you run into any issues or have questions/feedback, please do not hesitate to reach out! I will do my best to respond as soon as I can.

Unity forums thread: https://forum.unity3d.com/threads/486102/

E-mail: contact@staggart.xyz

Twitter: @JonnyIO

5. Refunds

Please refer to our refund policy

6. Features

6.1. Wind controller

The shaders used for both the trees and the vegetation utilize a custom wind animation method. In order to control this effect a "FAE_WindController" script component must be active in the scene. You can find a preconfigured prefab under the Prefabs/Effects/ folder.

See the scripts section for detailed information about this script.


Once added, this script can be safely removed and the current wind settings will be retained.

6.2. Cliff appearance

The CliffAppearance script allows you to modify multiple cliff/rock materials at once. The settings are are loaded as soon as the object is enabled. So it allows you to quickly apply different settings, or save the object as a prefab to serve as a preset.

Note: The coverage feature is not supported on DirectX 9, the inspector will show an error message.

Essentially this means you can have different looks per scene, while using the same materials.
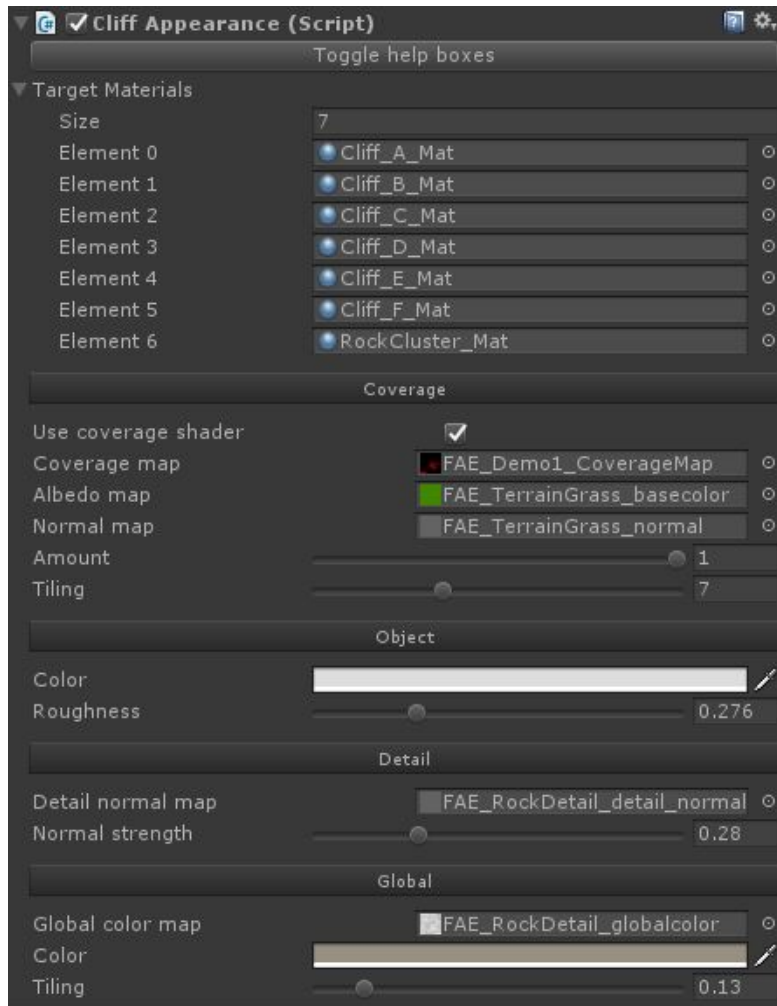
By toggling the "Coverage" shader you can add snow, sand or grass to your rocks. Through the use a coverage map, you can determine where on the terrain this effect should show:

You can make such a map by taking a top-down screenshot of your terrain and bring it into an image editing program. By painting over it with a black/white map (black meaning no coverage, white meaning fully covered) the effect can be centralized.

Currently, it is not possible to have multiple textures on the rocks. For instance grass in one place, and snow in another. As this would require the shader to hold a lot of texture maps, which is taxing on performance. Instead,

you'd have to use a separate CliffAppearance object with different material instances assigned to the cliffs.

6.3. Foliage bending

The shader found under "FAE/Foliage" supports basic player interaction. In order to enable this, the "FoliageBender" script component must be attached to a GameObject in the scene, parented to the player.

Once added, move it to the base/feet of your player:

You can grab the "FoliageBender" prefab found under Prefabs/Effects/.

Materials using the foliage shaders feature a "BendingInfluence" parameter which determines how much influence the Grassbender has on the material.

Note: This functionality is currently limited to one transform and is set to be updated with a more extensive system. Currently, the improved functionality needs some improvements and testing before it is added through an update.

6.4. Fog sheet

The fog sheet is a quad mesh which fades out/in based on its distance from the camera. It can be used to highlight certain areas (push/pull concept) or to enunciate a silhouette.

The shader has an optional texture slot, to add, for instance, a cloud texture.

You can see how it's being used in ABZU in the following GDC
talk: http://www.gdcvault.com/play/1024409/Creating-the-Art-of-ABZU (17:57 mark)

7. Polycount and LODs

Many users might like to know the polycount of the various assets before purchasing, hence this list containing the polycount and consequent LODs.

Trees

Asset name	LOD0	LOD1	LOD2
Tree_A	2497	1871	930
Tree_B	1912	1472	752
Tree_C	2321	1737	952
Birch_A	1539	1251	891
Birch_B	661	370	164
Birch_C	961	347	167
Spruce_A	1240	900	320
Spruce_B	1124	732	264
Spruce_C	1076	708	248
Pine_A	967	596	436
Palm_A	692	372	198

Note: All trees have a billboard after LOD2, which is 8 polygons.

Cliffs

Asset name	LOD0	LOD1	LOD2
Cliff_A	3994	1996	796

Cliff_B	3994	1996	796
Cliff_C	3994	1996	796
Cliff_D	1994	996	396
Cliff_E	996	496	246
Cliff_F	1994	996	496
RockCluster_A	1160	572	278
RockCluster_B	336	160	–
RockCluster_C	268	–	–
RockCluster_D	156	–	–

Vegetation

These items have few to no LODs as the polycount is already low. Flowers are around 12 polygons, whereas their clustered counterparts are around 60 polygons

8. Customizing assets

Branch colors

1. Navigate to the "Substance/Trees" folder and expand the Substance container of the tree you'd like to modify.
2. Select the branch material you'd like to edit and modify the parameters in the inspector

Grass appearance

1. Navigate to the "Substance" folder and expand the "FAE_Vegetation" Substance container.
2. Select the GrassAtlas material and edit its properties.

Note: You cannot use Substance materials directly on the prefabs, since a Substance material requires to use the Standard shader. Otherwise it will not generate its textures, or show the procedural properties, often enough it will even crash the editor. This is why all the assets use regular materials and only utilize the Substance output textures.

Terrain textures

In the *Fantasy Adventure Environment/Terrain/Substance* folder you will find the Substance materials used to generate the terrain textures.

It's handy to add a plane to your scene, and assign such a material to it. This way you'll be able to see the material in your scene, since the preview window uses very simple shading.
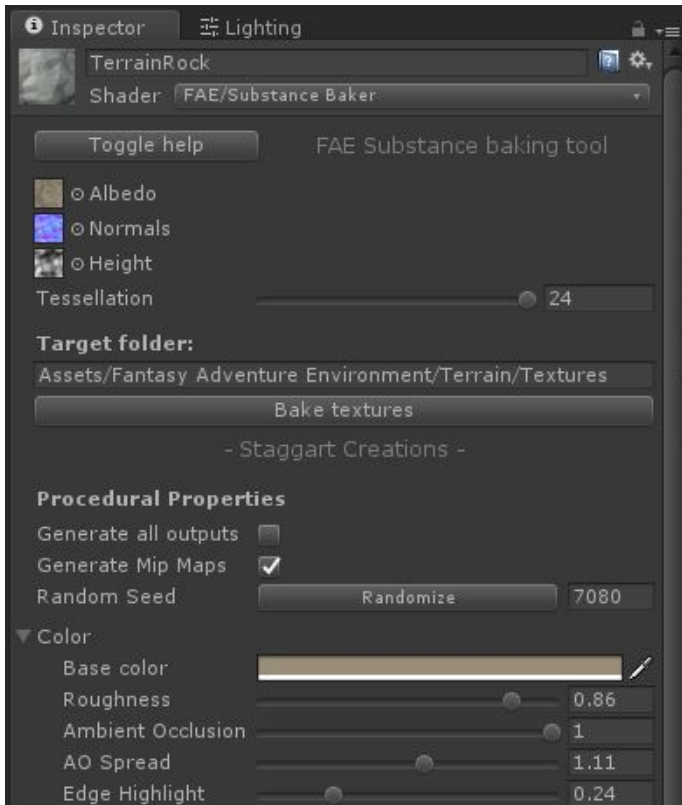
The materials have the *FAE/Substance Baker* shader assigned. This shader is an example PBR shader with a custom inspector. If it is not already assigned, you might have to reassign the shader twice, as it may stay on the Standard shader, this is a bug in the Substance integration plugin.

The last used target folder is saved in the editor settings. By default it points to the FAE textures folder.

It is very important the "Format" is set to "RAW", otherwise the textures will turn out black. The loading behavior is best set to "Do nothing", which will prevent the Substances from re-generating during the building process.
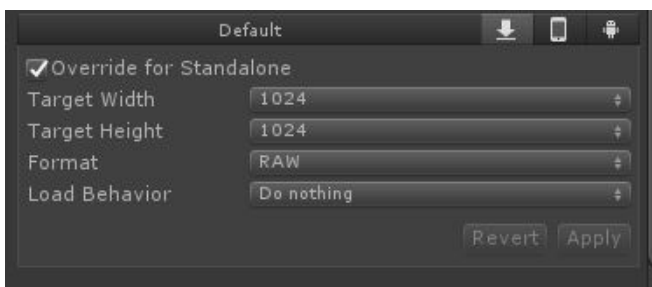
For every material, a sub folder is created, where the texture files will be stored.
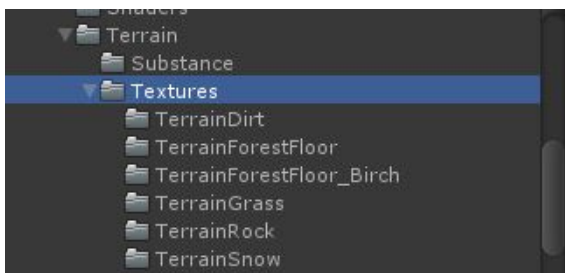
These textures are what you can use on the terrain. The tool overwrites these files, so every time you bake the textures the changes will be reflected on your terrain.

The material feature a heightmap, which can be used for displacement or height-based texture blending. I highly recommend using a third-party terrain shader which will utilize this.

8.1. Material instances

If you'd like to seperate appearances, for the same tree, you are required to make seperate materials.

Every tree type has a Substance container, which generates the textures for the branch, trunk and the billboards for every tree variant. Naturally, if you were to change any of its color properties, these changes will carry through to all other trees using the same textures.

If you'd like to add snowy trees to your scene, but would like to retain a set of regular trees, you are required to make an instance of the Substance material. Which will generate a new set of textures.
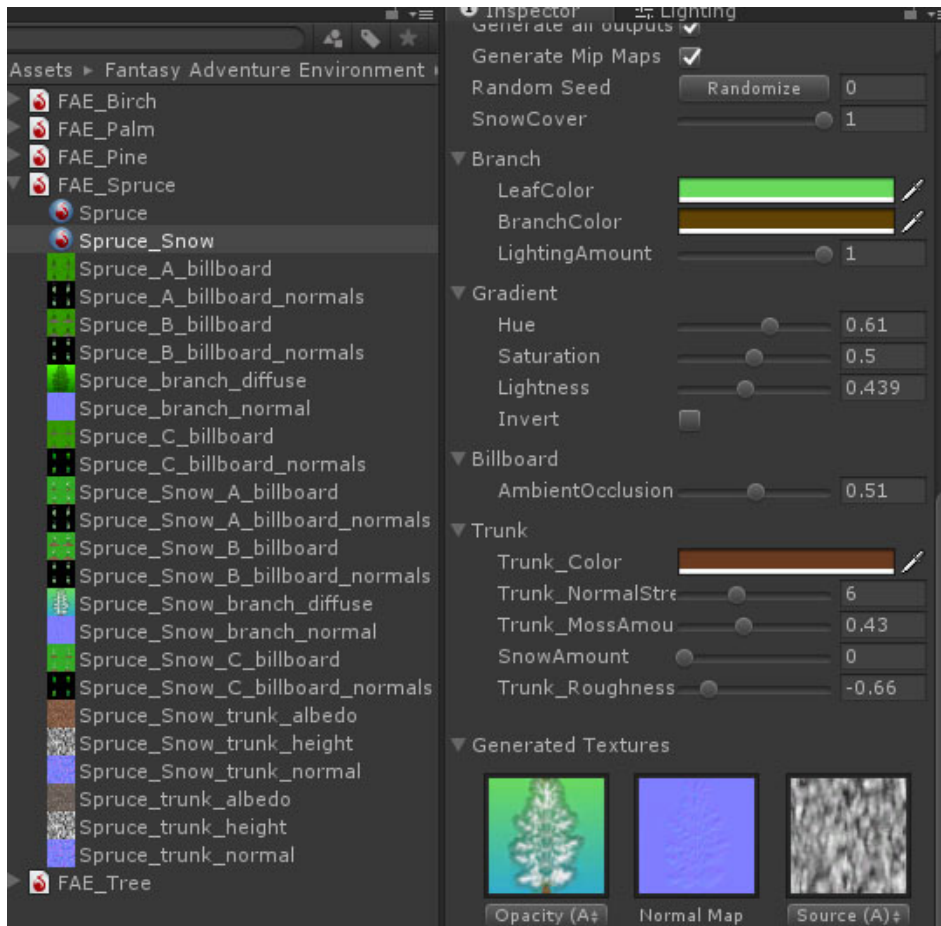
You can do so by following these steps:

1. Select the Substance root, and press the "+" icon (outlined in red) to create a new instance, you can give it a name at this point. You will notice a new set of textures is generated (outlined in blue)
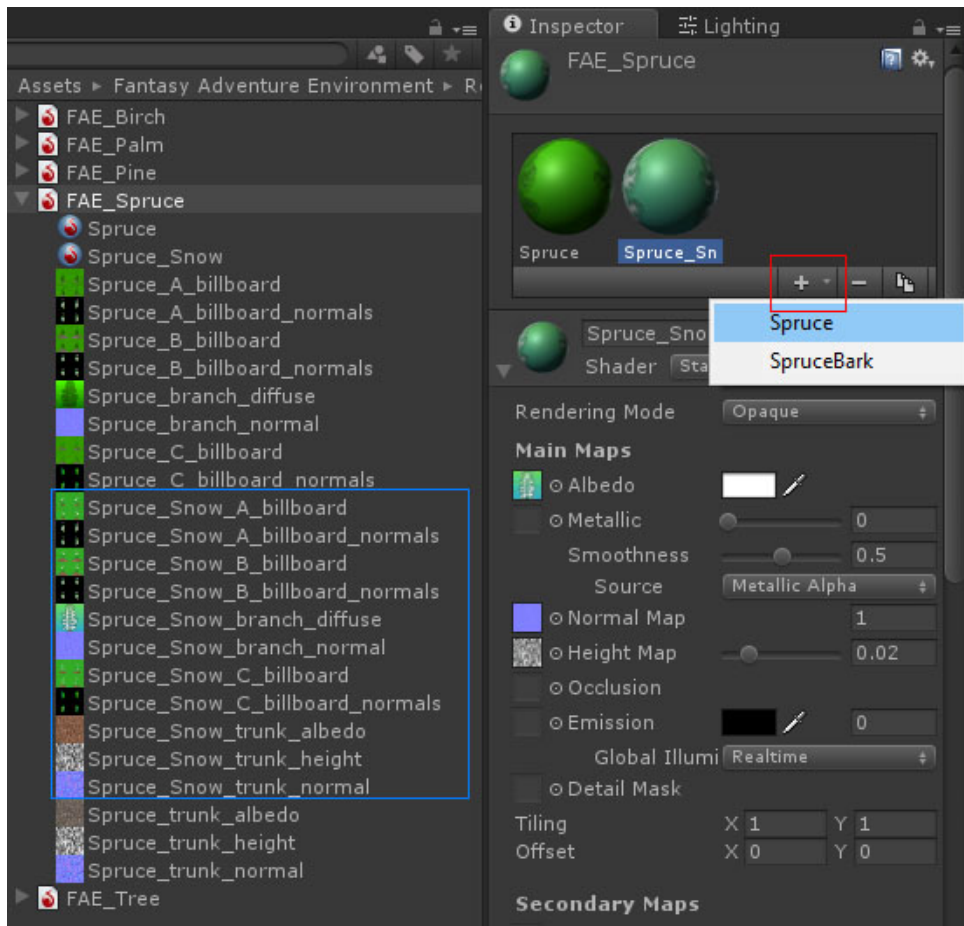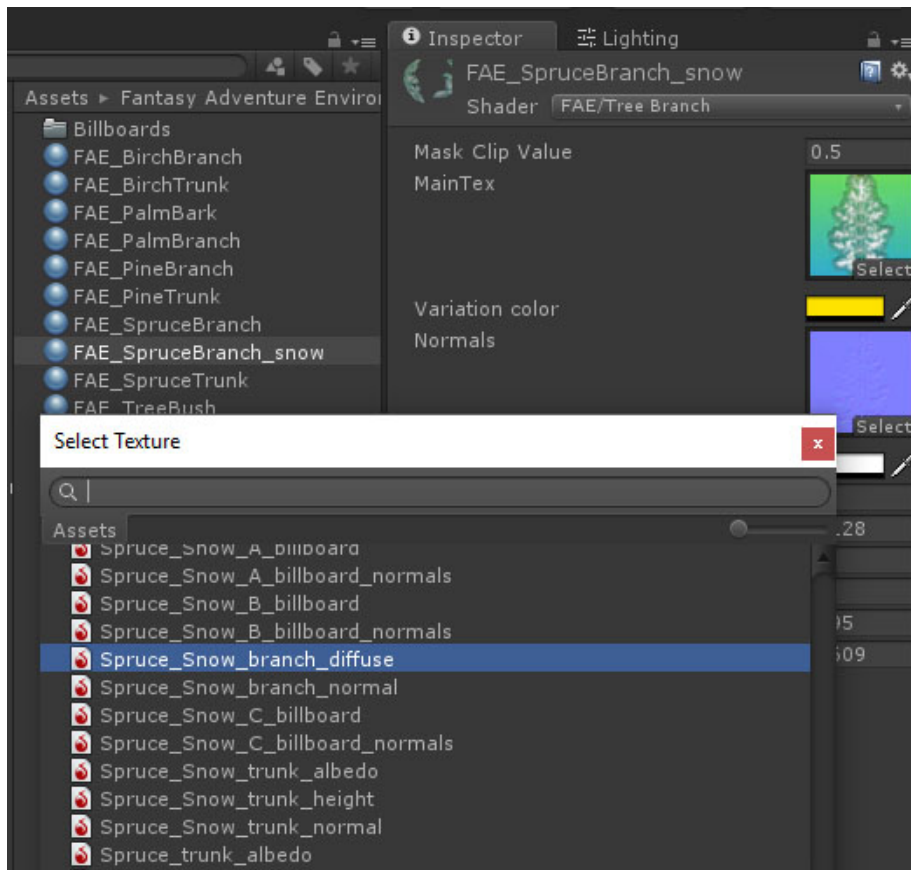
2. Change any properties you like. For example, I'm adding snow. You can come back to these setting at any time.

3. Create a new branch material, and assign the new branch textures to it.

If you're looking to have a different look trunk, make a new trunk material as well.

4. Drag the tree prefab into the scene, and assign your new materials to every LOD mesh

If you're looking to have the billboard of the tree also look snowy, repeat step 3 and 4 for every variant of the tree.

You'll end up with something like this:
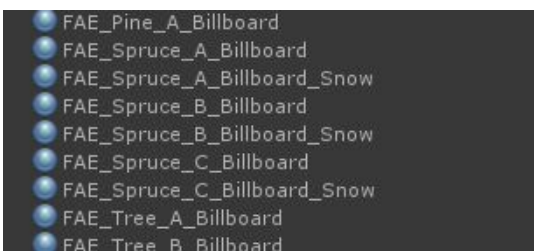
Assign this material to the "billboard" LOD of the tree.

5. Rename the tree, and create a prefab from it.

In summary, the Substance container is responsible for generating textures, and you need materials to utilize these. Unfortunately, it is not possible to use the Substance material directly on the mesh, as they are not able to use custom shaders.

9. Performance Guidelines

This environment pack was produced for desktop/console platforms, as such the visual fidelity does not come very cheap. Development was done on a laptop with an i7-4700HQ and a GTX765M (comparable to a desktop GTX650) where the demo scene performed around 40-70fps. On machine powered by a GTX960 it performed

around the 120fps range.

At this time of writing, the average framerate in the downloadable demo is 95 FPS, across 398 reports.

All shaders support GPU instancing, which is available since Unity 5.6. It sees a huge performance improvement.

It's important to know that Static Batching has been disabled on the tree prefabs. This is so that each tree has its own random wind motion speed and color variation. Since when models are batched they all share the same Transform. And the shaders utilize the model's scale to add an offset to the wind speed.

If you find this negatively impacts your performance, you can enable Static Batching on the prefabs:
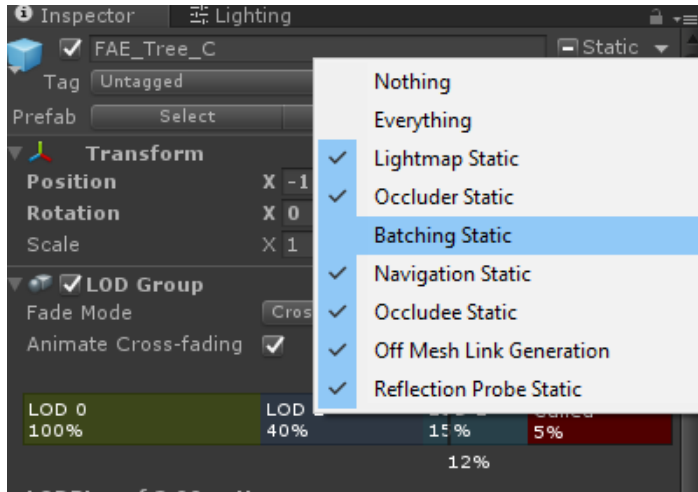
It might be beneficial to set the *Trunk Wind Weight* value to 0 on your *WindController* after doing so. Or else all trees will appear to sway in unison. I hope to circumvent this limitation at some point, but it warrants some experimentation.

Also note how in the demo scene, the line of sight is broken at several points:

This is beneficial for Occlusion Culling, which avoids rendering objects that aren't visible. Thus, reducing the

rendering overhead.

https://docs.unity3d.com/Manual/OcclusionCulling.html

Increasing the "Pixel error" value in your terrain settings, also has a huge impact on performance.

10. Scripts

All scripts are in the "FAE" namespace.

WindController

The shaders used for both the trees and the vegetation utilize a custom wind animation method. In order to control this effect a "FAE_WindController" script component must be active in the scene. You can find a preconfigured prefab under the Prefabs/Effects/ folder.

For ease of use, you can find a prefab for this in the Fantasy Adventure Environment/Prefabs/Effects folder.

By keeping this object in the scene, it is possible to have different wind settings per scene. However, it is safe to remove it, as the settings will be saved in the shaders.

PigmentmapGenerator

This script must be added to your terrain object. It generates a color map texture (pigment map) and heightmap for the *FAE/Grass* shader (see shaders section). Which can be utilized in the material to tint the grass by the terrain's color and have the height taper off where there is no grass.

Attaching this script to your terrain will load it's pigment map on startup. So it's possible to be used in multiple scenes.
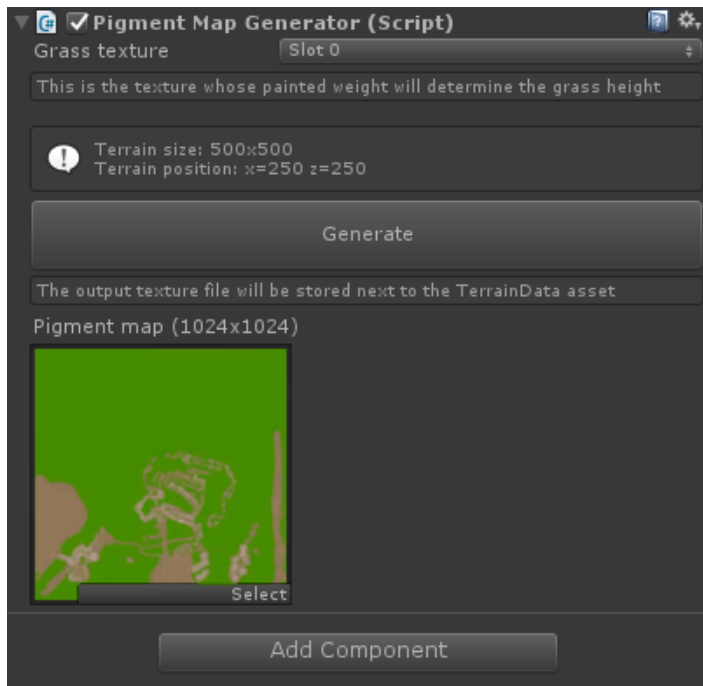
The script also works with mesh terrains. However, like Unity terrain's, the size must be uniform, and the pivot point must be in the corner. If a mesh terrain fails to meet these requirements the pigment map will not line up accordingly.

CliffAppearance

Used to modify the shader properties of the *FAE/Cliff* shader across multiple materials.

Attach this script to any object in your scene and assign the materials using the *FAE/Cliff* shader. The values set in the inspector are applied as you change them, or whenever the object is enabled.

For ease of use, you can find a prefab for this in the *Fantasy Adventure Environment/Prefabs* folder.

This could be used to, for instance, load different settings based on the scene. You could save the object as a prefab to use it as sort of a preset.

FoliageBender

Tells the *FAE/Foliage* shader from which position in the world it should bend away from, and by which strength/radius.

SubstanceBaker

Used internally to bake the terrain materials to PNG files, in order to use them on the terrain

TerrainUVUtil

A internal helper class for the PigmentmapGenerator

11. Shaders

All shaders, except the sunshaft shaders, have been made using [Amplify Shader Editor](), and thus be modified using it. Please make sure you are using the latest version from http://amplify.pt/download/ as this can avoid a lot of headaches.

Please note that I cannot provide support for the use of ASE and modified shaders are no longer subject to user support.

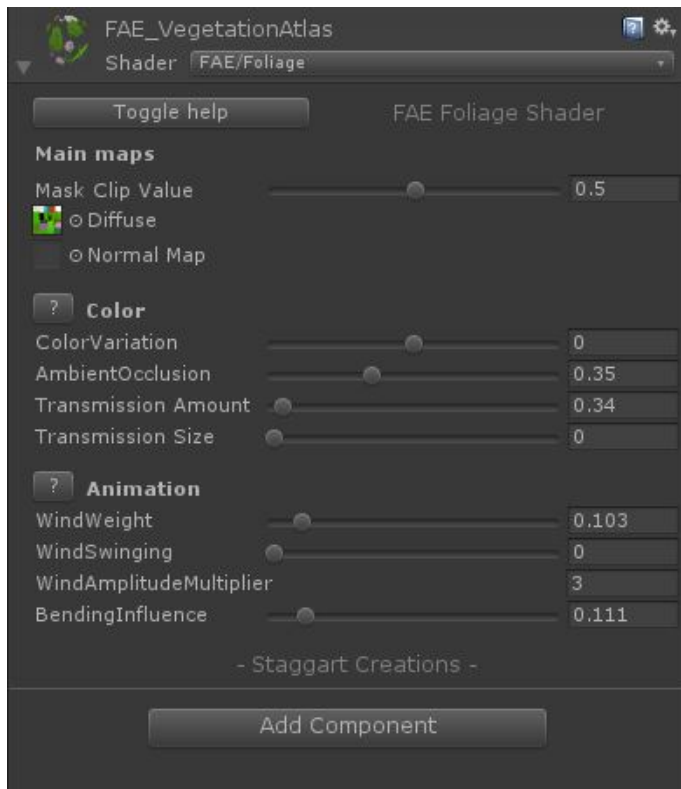The shaders included in the package can be found under the "FAE" section in the shader dropdown list.

Grass

Used by the grass. Features light transmission, wind animations, bending and the heightmap/pigment map functionality.
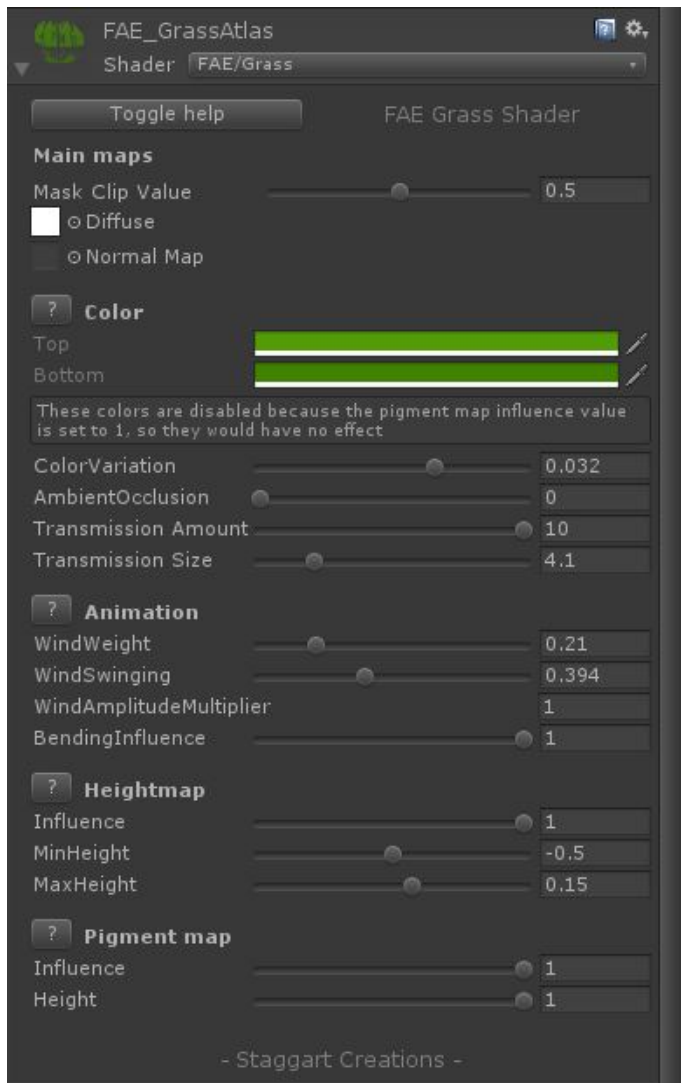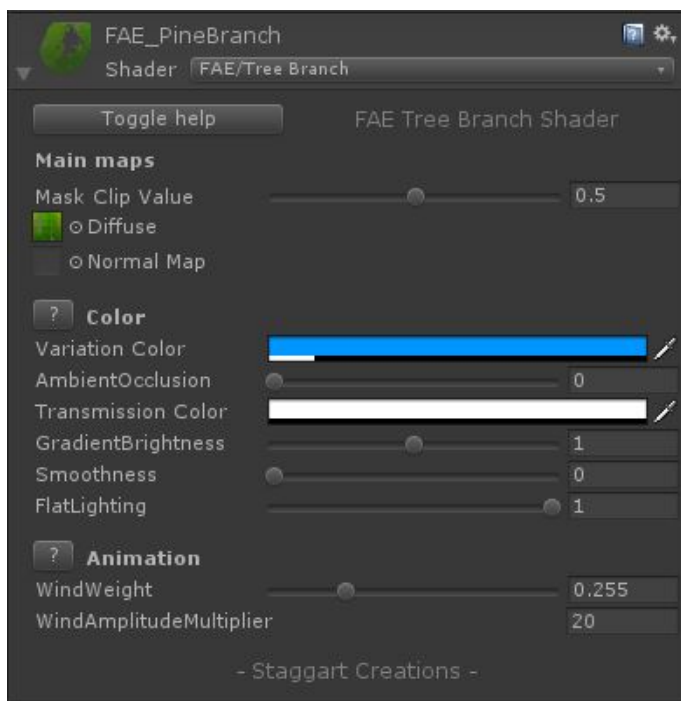
Foliage

Used by the vegetation. Features color variation through wind, bending, light transmission and wind animations

Tree Branch

Shader used on the tree branches, which enables wind animation and stylized rendering.

Tree Trunk

Simple PBR shader, uses the Albedo map's alpha channel for the roughness.

This shader is required on a tree for the global wind motion to work.

Cliff

Features a tri-planar projected color map, and detail normals. Roughness is sampled from the albedo map's alpha channel.

Cliff Coverage

Same functionality as the Cliff shader, but with the addition of a coverage feature

Fogsheet

Opacity is controlled by a distance factor from the surface, to the camera. Optional texture slot.

Sunshaft

Used with the Sunshaft/Long and Sunshaft/Short assets.

Cross-pans a sun shaft texture horizontally. And features distance and intersection fading. The Directional Light's color's alpha channel controls the intensity of the effect.

The Directional Light's color's alpha channel controls the intensity of the effect.

Sunshaft particle

Does much of the same as the Sunshaft shader but is tailored to be used with a ParticleSystem. Where the "Color over lifetime" feature can be used to fade out the sunshaft through the alpha channel.

The Directional Light's color's alpha channel controls the intensity of the effect.

Water

A simplified version of the [Stylized Water Shader](#), considered to be a "lite" version of the complete package. Limited in functionality and customization.

12. Vegetation Studio

The package assets have been tested with [Vegetation Studio](#). Since version 1.0.5 compatibility was added.

Vegetation Studio makes excellent use of GPU Instancing (available since Unity 5.6), which allows you to create extensive dense vegetation at a fraction of the performance cost.

Vegetation Studio is currently in beta, and thus not publicly available yet. The VS settings file will be made available for download in some time. This way you'll be able to recreate this kind of scene with the optimal settings for the assets:

This frame is merely 140 drawcalls.

13. Troubleshooting

Lightmapping

You might experience some issues with lightmaps, such as black spots. Though, this is not directly attributed to the asset. Since LOD1 and up do not support lightmaps, but are rather lit by light probes you may notice visible popping.

It is recommended to not bake lighting but use precomputed GI instead.

Grass is grey

This means the grass material is using a pigment map, but there is none available.

Adding the PigmentmapGenerator script to your terrain will elevate this.

If you don't wish to use this functionality, turn down the "PigmentmapInfluence" parameter on the grass material.

Object appear completely black

This was a shader issue attributed to OSX or DirectX 9, which has been fixed since version 1.0.3

Cliffs turn black when re-importing the package

Ugh, this is such a bummer, I can't quite pinpoint what is causing this. For now, there is a workaround:

Go to the FAE_RockDetail Substance material, right click, and choose "Reimport".

Trees and vegetation turn white in Unity 2017.1 (and newer)

A separate version of the package was submitted for Unity 2017.1, because the shaders require some modification to work. To elevate this issue, re-download the package from the Asset Store in Unity 2017.1

14. Compatibilty

The package was designed for mid-range to high-end PC and consoles. VR has not been tested since I have no access to a headset at the time, but users have reported to have no graphical issues.

Mobile platforms, WebGL 1.0 and DirectX 9 are not supported.

Playstation 4 and Xbox One remain untested since devkits are extremely hard to obtain. If you're developing for these platforms and find yourself having issues, please get in touch as I'd be happy to work towards a solution!

15. Using the shaders with custom assets

Should you like to use the *FAE/Branch* shader on a tree outside of this package it must conform to a requirement, namely the Ambient Occlusion (AO). Which must be baked into the mesh's vertex colors. This is used for shading and animation purposes.

Currently the AO is baked into the RGB channels, but only the Red channel is accessed in the shader. The Green and Blue channels might be overwritten and utilized for different purposes in the future.

The vertex color's alpha channel holds the tree's stiffness, which is used for its global wind motion. This information was baked using a proprietary tool, which is not available to users.

For vegetation assets using the *FAE/Foliage* shader much of the same holds true. Where the AO information is used to determine which parts are not affected by wind or player interaction. As well as several shading functions.
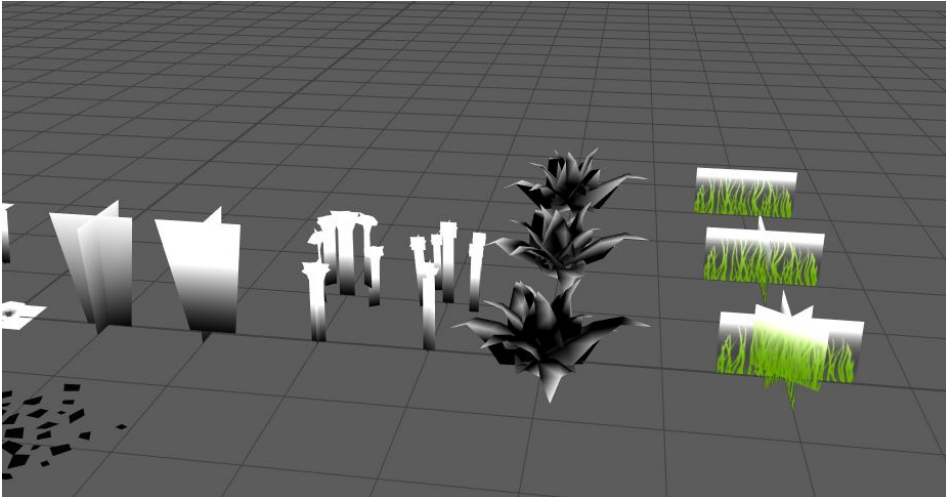
The branch shader will also work with SpeedTree trees, since they also have the AO baked into the vertex colors. Unfortunately, the global wind motion (swaying motion) will not work correctly, since the aforementioned "stiffness" information will be missing. Causing the base of the tree to also move.

Shaders used on the Standard Assets SpeedTree trees

16. Substance materials

The behavior of all Substance material has been configured to bake the Substance to texture when you build your project. So they will not incur a loading time, as they would have to be generated otherwise.

All Substance materials can be found under *Fantasy Adventure Environment/Resources/Substances*.

Source files (.sbs) for the Substances will not be made available.

16.1. Lensflare

Composes a lensflare atlas texture. You can modify the intensity of the individual components.
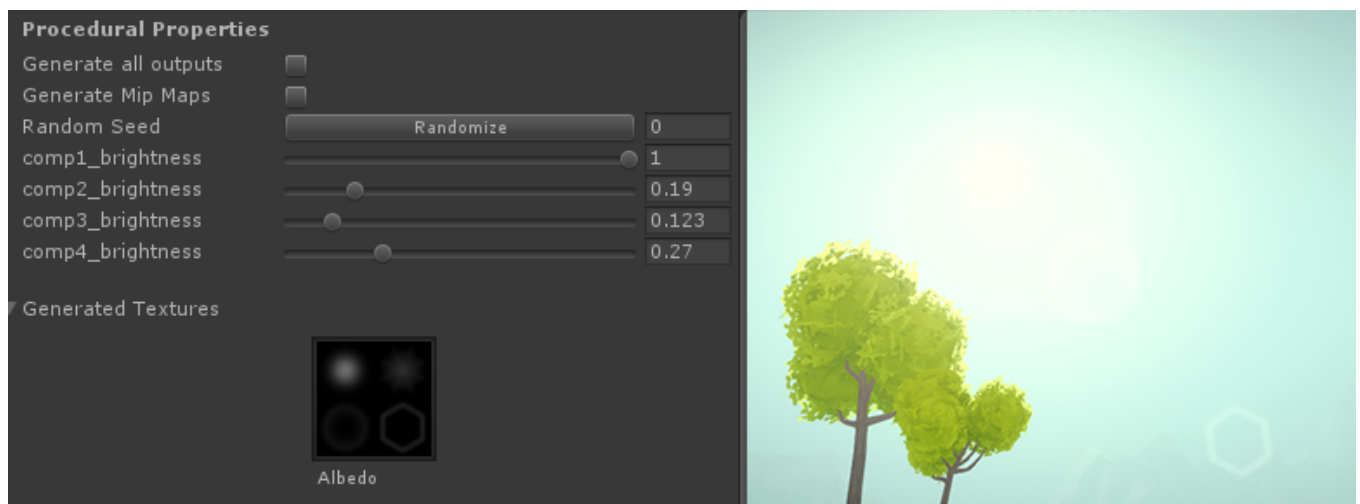
If you'd like to change settings such as the size or the colors, go to the actual lensflare asset found under *Fantasy Adventure Environment/Resources/Effects/Materials*.

16.2. Particles

Used to generate particle sprites, limited options available.

16.3. RockDetails

Generates the global color- and detail normal map for the cliffs. This might be expanded in the future with more kinds of maps.

16.4. Trees

Holds all the texture outputs for the tree branches and trunks. This is where you can modify the colors of the trees.

The snow functionality is a little gimmicky at this point, but I saw no harm in leaving it in ?

16.5. Water

Outputs the maps needed for the water and waterfall. Please do note that the waterfall asset is not officially included, but only produced for the demo scene.

16.6. WindMap

Generates the wind direction vectors. The WindController script drives its parameters, so no need to fiddle with this Substance.