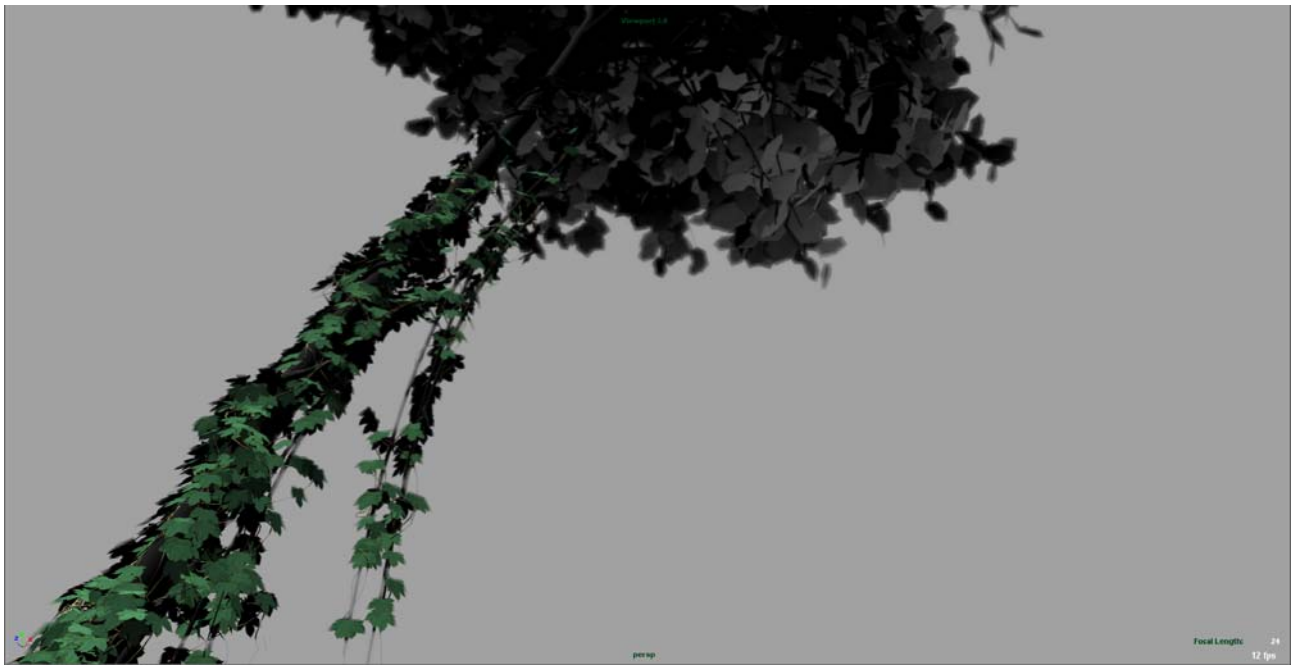
Wai On Ryan Chan - MPC Vegetation Masterclass



Fabric:Splice Node *(tested against 1.13.0)*

Tool Setup
The vegetation tool is written in Kale using *Fabric Engine*. It is incorporated into *Autodesk Maya 2014* using the Fabric:Splice plug-in, and thus should be portable to other DCC's with ease with the exception of the accompanying python interface.

Upon extracting the folder from the handin file, you should have the following files (key files are highlighted in green):

- *This readme*;
- FabricVegetation_files
  - demos
  - FabricVegetation
    - documentation
    - ext
    - images
    - FabricVegetation.splice
    - vegetationOp.kl
  - scenes
  - FabricVegetation.py

The *FabricVegetation.py* file and the whole *FabricVegetation* folder should be moved to the maya/2014-x64/scripts folder so that it appears like so:

- 2014-x64

    - scripts
        - FabricVegetation.py
        - FabricVegetation
            - ...

    (DOWNLOAD FROM HERE - http://dist.fabric-engine.com/FabricEngine/)
- Before using the tool, installation of Fabric 1.13.0 is required. The current version, 1.15.0 should be available at http://fabricengine.com/get-fabric/.
- After installation, you need to source the *environment.sh* file included in the *FabricEngine* directory.
- Then to add the Fabric:Splice plug-in to Maya you need to add the appropriate folder from the *FabricEngine* directory to the MAYA_MODULE_PATH env variable.
- You also need to add the *ext* folder from the *FabricVegetation* folder to the FABRIC_EXTS_PATH env variable in order to have a custom extension working for the tool.

After setting all of this up, it should simply be a matter of starting up Maya with the new environment variables and setting up a shelf button with the given icon from the *images* folder in *FabricVegetation*. The code attached to the shelf button should be:

```
import FabricVegetation
FabricVegetation.FVegetationUI()
```

If pressing this button brings up a UI and importing the node, etc. works as shown in the usage demo video the tool is working.

There are 2 of the speedtree scenes set-up with the node and a grown mesh as examples, the UI should work with the right off the bat.

User Interface

Many of the values follow the paper also included in the files however a brief explanation is given for each slider:

- *Distribution angle* – The conical angle in which the bud can potentially grow to each step
- *Distribution samples* – The number of random positions the bud will evaluate against each step, the higher the more accurate the growth
- *Bud chance* – The chance of an inactive lateral bud being grown along a branch for traumatic reiteration-sakes
- *Bud active chance* – The chance of one of those inactive buds being active upon being created so as to achieve branching
- *Leaf chance* – The chance of a leaf being created per step, per branch
- *Min increment size* – The minimum distance each active bud has to grow per step
- *Max increment size* – The maximum distance each active bud can grow per step
- *Increment steps* – The number of iterations of growth to go through

- *Max thickness* – The maximum radius of the branching
- *Leaf size* – The size of each plane for the leaves
- *Quadratic falloff lights* – Whether the lights should have a quadratic falloff or not (doesn't work too well at the moment)

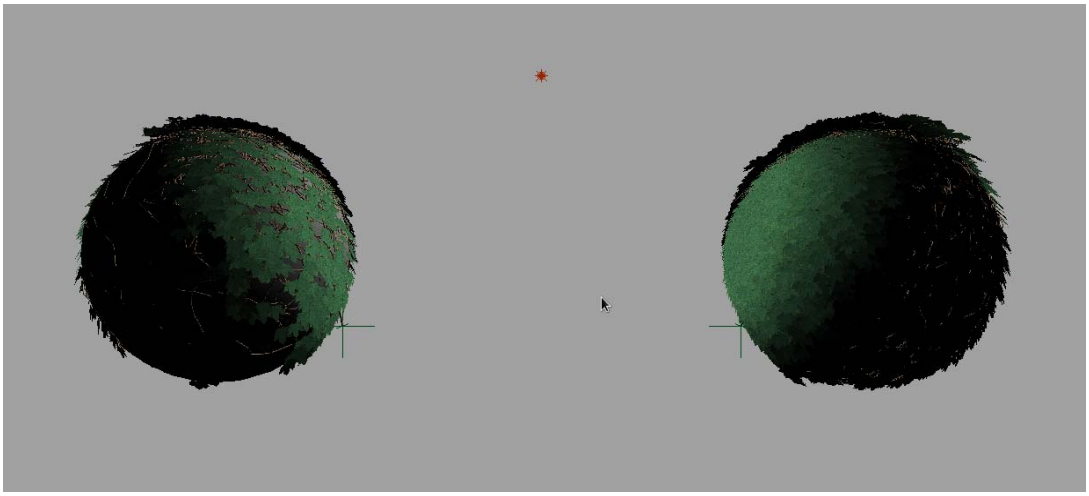
These values are also referenced in the documentation although under slightly different names.
The rest of the UI is self explanatory or explained in the usage demo.

Current Issues

- Sunlight doesn't work very well, neither is the UI very prepared for artists to use it yet as there is no clear option and it doesn't check whether a sun already exists
- There is a bug mentioned in the documentation with the clear seeds and clear lights buttons, both of them disconnect all seeds and lights but don't clean the *fabricVegetation* node's multi-array indexes, making any new seeds/lights have bad naming convention and wrong connections
- The leaves are generally in the correct direction but there are cases where they may seem to be less random than desired, i.e. on a large flat surface, the sizes also need to be randomized
- There are occasional collisions with the environment geometry and the vegetation itself, however most of this should be fixed by now although the leaves do not have collision detection

Interesting Artifacts

- The number of polys detailing the environment makes an impact on the growth of the vegetation, it appears the higher-poly the mesh the more the vegetation grows, giving a more dense appearance *(image left: lower-poly sphere*)



- When there are no lights in the scene there is a strange bug that makes the vine grow wildly, nearly disregarding the environment geometry
- I checked with real-life reference of ivy and with images and they seem to

match the branching pattern quite similarly in this implementation

Speed Comparisons

Below is a small chart demonstrating the different speeds at which the tool should be expected to run. Naturally there are many other attributes that affect the running speed but they were kept as close as possible (i.e. seed position, etc.)

		Octree Build	Speed (secs)	
Steps	Environment Polycount	(Approx. secs)	4 Cores	8 Cores
300	6	1	2.11	1.78
300	1174686	8.5	3.59	2.57
700	412	0.5	1.67	1.03
700	673464	8	4.02	1.78
1200	6	3	31.9	14.36
1200	1174686	9	31.61	10.6

References

Benes, B. and Millan, E. (2002). Virtual climbing plants competing for space. *Proceedings of Computer Animation 2002 (CA 2002)*.

CGTextures, (n.d.). *Leaf texture*. [image] Available at: http://www.cgtextures.com/texview.php?id=8016&PHPSESSID=mrifflf7oj7k9kdjn847bkj7k2 [Accessed 12 Feb. 2015].

The dummy ivy texture image was provided from the above website.