Scripting Report: Morgan Moore s5113911

Tree Generator:

My python script for Maya creates Trees that are designed by the user through a UI

Manual

To run my script:

1. Open Maya
2. Open TreeGenFinal.py and copy the contents into the Maya Script Editor
3. Select all the code and run
4. The window on the right should pop up >>>
5. If necessary, change the submission_path variables (there are 3) which will initially be set to: "/home/xyang/maya_scripts/submission“
   Which is added on to my path:
   e.g FinalTreeGen/iimages/panaram4.jpg

How to use the UI:

- Firstly choose a Treetype from the selection of 7
- Use the colour slider to specify colours for leaves and branches
  Click on colour box for more colour options
- Select the tree material
- Choose parameters for the height angle and iterations
- Click checkbox for fruit/apples to appear on the tree
- Choose the leaf type (maple or default)
- Uncheck the leaves checkbox if you don't want leaves
- Set the leaf material
- Set the scale of the individual leaves
- Set the thickness of the leaves you want in the tree

- Click ‘Apply’ if you’re happy with the above setting
- Click ‘Save’ if you want to save the current tree model
- Click ‘Undo’ if you want to go back to the previous tree generation you had
- Click ‘Cancel’ or the ‘X’ to close the window

External Modules:
In the program I use a some external modules including math, random and functools.
Installed as such:
import math as math, import functools, import random as rand
Math for equations, functools to pass arguments from the UI, and random to alter tree building variable to create a more natural look.

Implementation:
The purpose of this program is to create various trees that can be designed to the users specification.
Attributes such as colour, size, material, and shape will all be adjustable.

The program begins by creating the user interface which contains several user controls.
Once the apply button is pressed the variables holding the values of the user controls are passed to the treegen function.

In the treegen function the controls are queried using the passed variables and the user interface values are obtained.
These values can then either be directly passed into the 'creating' functions iterate, createModel and setMaterial
or used to define variables such as the axiom, rules and material of the tree.

The iterate function uses the basestring (from the chosen treetype/l-system), number of iterations, and the dictionary.
For each iteration the function replaces each symbol in the basestring with its corresponding value from the dictionary,
before then assigning the basestring with this newly created 'replaced' string before the next iteration begins.

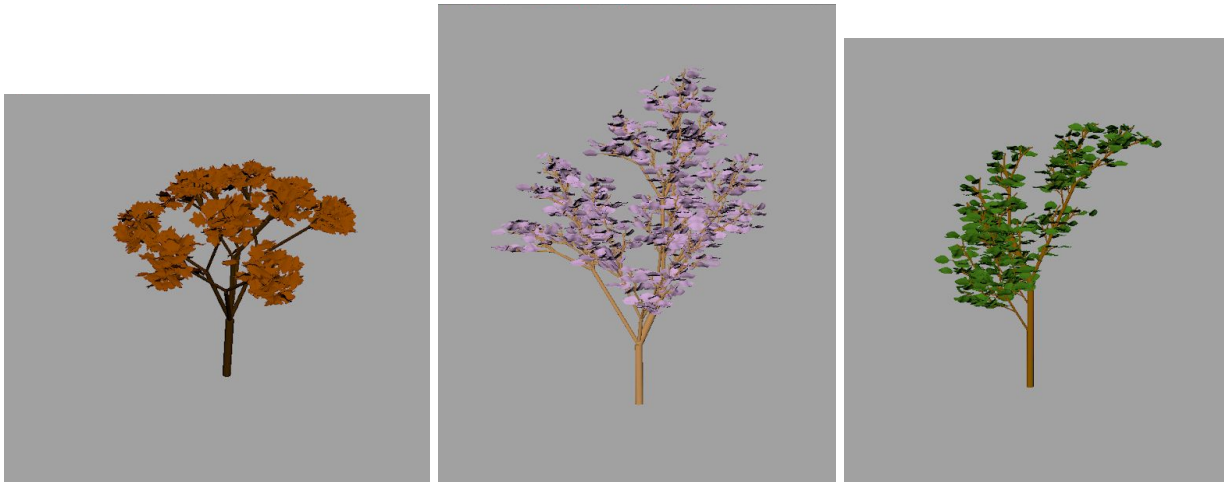
The primary 'creating' function is createModel which controls the building of the tree and leaves.
This function uses the ‘finalstring’ generated in the iterate function along with other necessary variables such as length, turn, and leafsize.
A while loop in the createModel function iterates through the actionstring and checks each symbol against the if/elif cases until a matching case is found, which the program will then enter within these if/elif cases the program calls a number of functions to assist with building the tree including createBranch, createVector, makeleaf and makeapple, and also sets new values for variables such as anleX, angleZ and level,
The cylinders(branches), leaves and apples that are created are then appended to a list which is then used to create groups for each object type that will show up in the outliner of the finished maya scene.

These groups are then returned to the treegen function where they are then passed (along with colour list, and the material types for the tree and leaves) to the set material function.

In the set material function a set and shading node are created, the user specified colour is then attributed to the shading node and a shader list is created, the material is then assigned to the tree,
This 'set material' process is repeated for the leaf with its corresponding colour and material attributes from the user, (assuming the leaves checkbox was on)
The objects have then been set their materials and two tuples (both of two strings) which contain the new shading group name, and the new shader name are returned to the caller

Additional elements of the program include the save undo and cancel tools:
The undo tool loads the maya file from the previous tree generation
the save tool creates a window where the user can specify how to save their tree and then save it
the ‘cancel’ tool calls a the cancelCallback function which closes the main window

Some Example Outcomes:



Overall i am happy with the results and think i believe i achieved what i set out to do. With more time i would work on optimizing to program to improve generation speed. This would improve its usability and also allow it to handle more complex or detailed tree structures. I would also add the ability to add user selected texture images to further improve the appearance of the trees.