MANUAL

To run my script:

1. Open Maya
2. Open TreeGenerator.py and copy the contents into the script editor in Maya
3. Enter the Logo Image named 43.png into /home/'yourspace'/maya/2018/prefs/ and under icons
4. Select all the code and run by pressing the Enter key on the keyboard or by the two arrows at the top of the Script Editor
5. The window for the Tree Generator should appear
6. Adjust values to desire
7. Press "Apply" when done
8. Tree will generate

HOW TO USE THE UI

- The first three checkboxes allows the user to choose from three tree types
- The next row of checkboxes allow the user to choose from the type of material they want on the branches and leafs
- The user is able to adjust tree settings such as iteration, steps, and rotation angle using the sliders
- The user is able to adjust the RGB, HSV to as much detail through the input widget
- Apply button generates the tree
- Cancel deletes the window

TIP: To create a more natural looking tree, increase the iterations to make the form more complex, thus adding more leaves and branches.

IMPLEMENTATION

STRUCTURE

I wanted to keep the structure of my program as clean as possible to preserve its readability, therefore I didn't want overly complicated procedures.
My program is based on two main procedures, createModel and buildTree. Here the functions that have been declared elsewhere feeds into these two main procedures. Build tree receives all the values from the user interface and createmodel uses these values.

ALGORITHM EXPLANATION

The algorithm begins by creating a user interface, depending on what values the user changes it accesses the window's input fields from within the applyCallback function. This is then taken to the additional parameters that are applied in the applyCallback function to mirror the sequence of arguments in the partial call. This then accesses all the adjustments by calling the intfFeld command. Through here, this is sent to createModel where the program takes the required values, this is then taken to build tree unless there is any additional changes before.

USER MANUAL, INTERFACE, EXECUTE

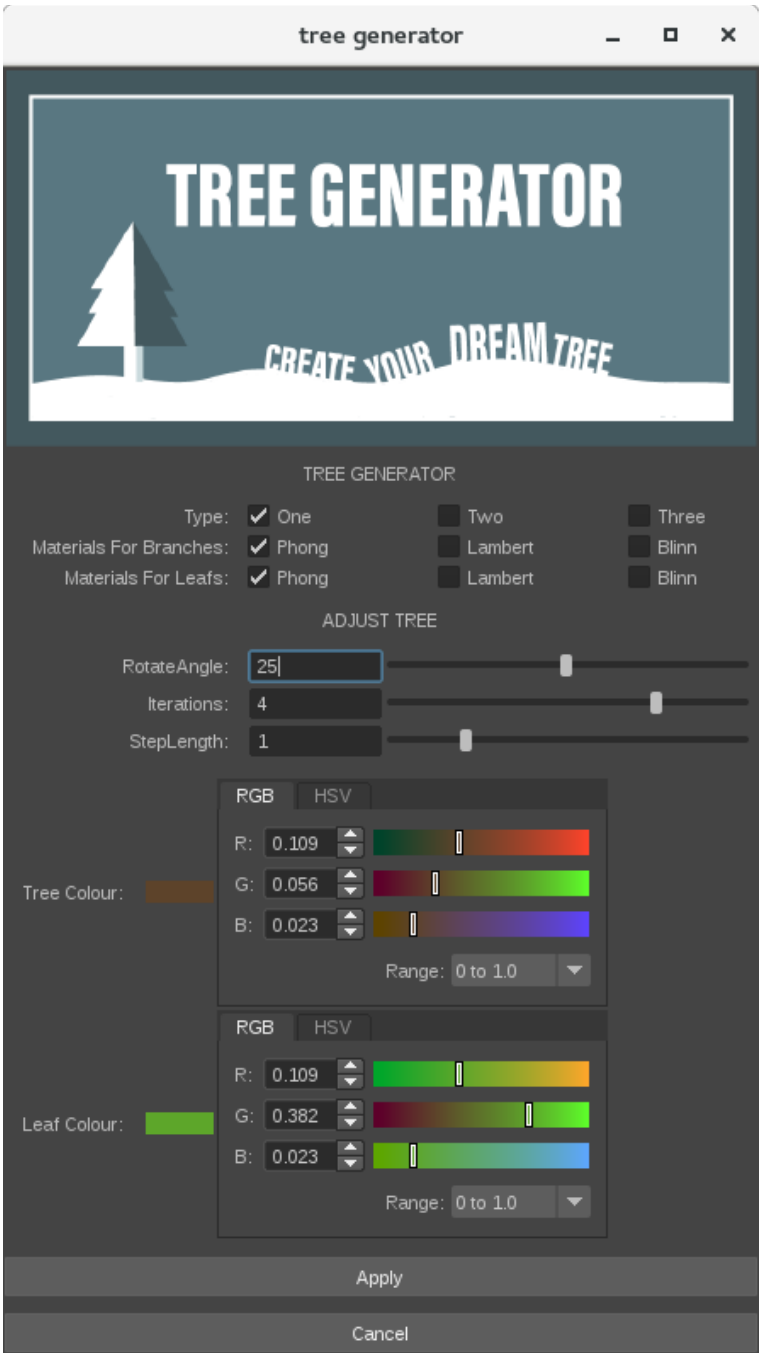
The UI is made by creating a window which displays various input fields that can be used to generate the tree. In my UI, the window allows the user to select and adjust their own input within the range given.
To create the UI, I first had to declare a function which I named "createUI" , within this function I have given two arguments, the title of the function and the callback function. To make the window I called the windows command and inputted the following flags. The flag "resizeToFitChilderen" enabled the window to adjust around the content of the window perfectly,furthermore the "sizable" flag which I set to false to disable the use of the user from resizing the window as by not doing so will make the window less compact. After I created the different settings to change from by including buttons, sliders and checklists. I also added my own logo which I made from scratch in Adobe Illustrator to the UI.

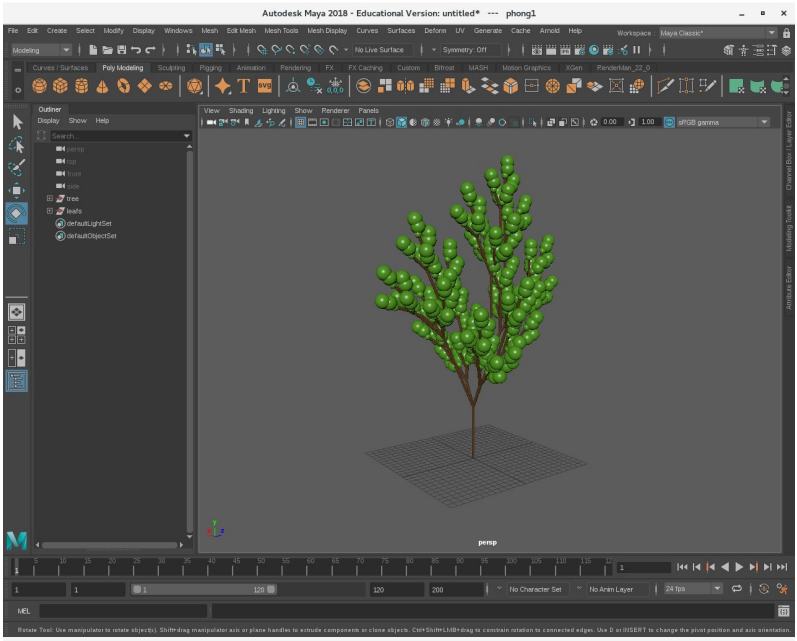
LSYSTEMS

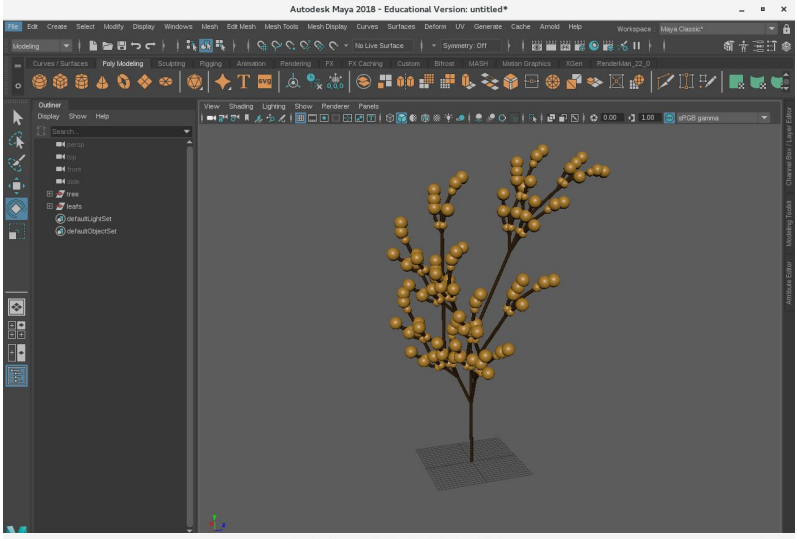
To create my trees through Lsystems, I looked through many resources to better understand the recursive nature that describes it. From this, I was able to create my own Lsystem using this knowledge. To create the tree I must first begin with the axiom string “X” in which to begin the construction. I then created the variables “/” and “|” for the rotation of the angles to generate a 3D model. I also implement leafs using the variable “D”,”S” and “L”, all of which are in varying sizes.

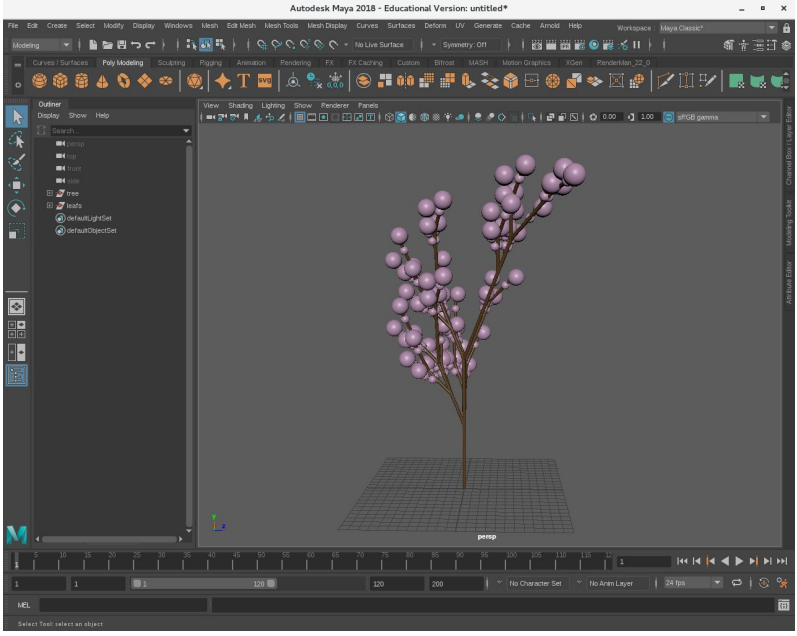
RESULTS

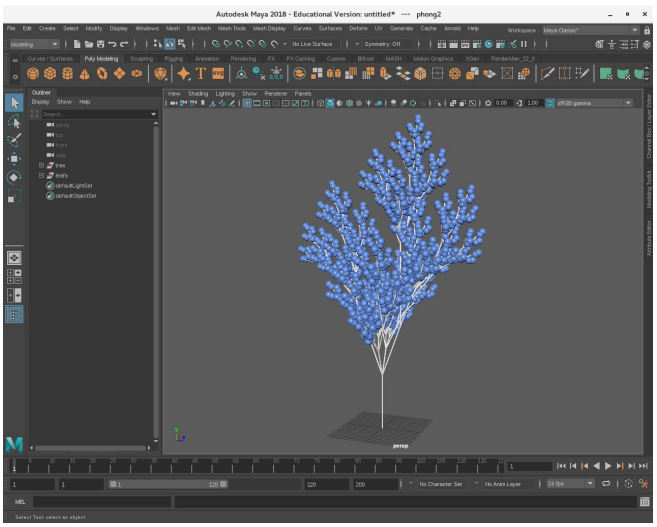
Overall I am pleased with the results and although I find programming difficult, I felt as if I have learnt and enjoyed the progress of learning python. Although it is not that complex, I am happy with the final aesthetics. If I had more time, I would implement some animations into the code with time ranges. Moreover, I would also implement different seasons that the users could pick from as well as variety of leaves types, I thoroughly enjoyed this project very much.

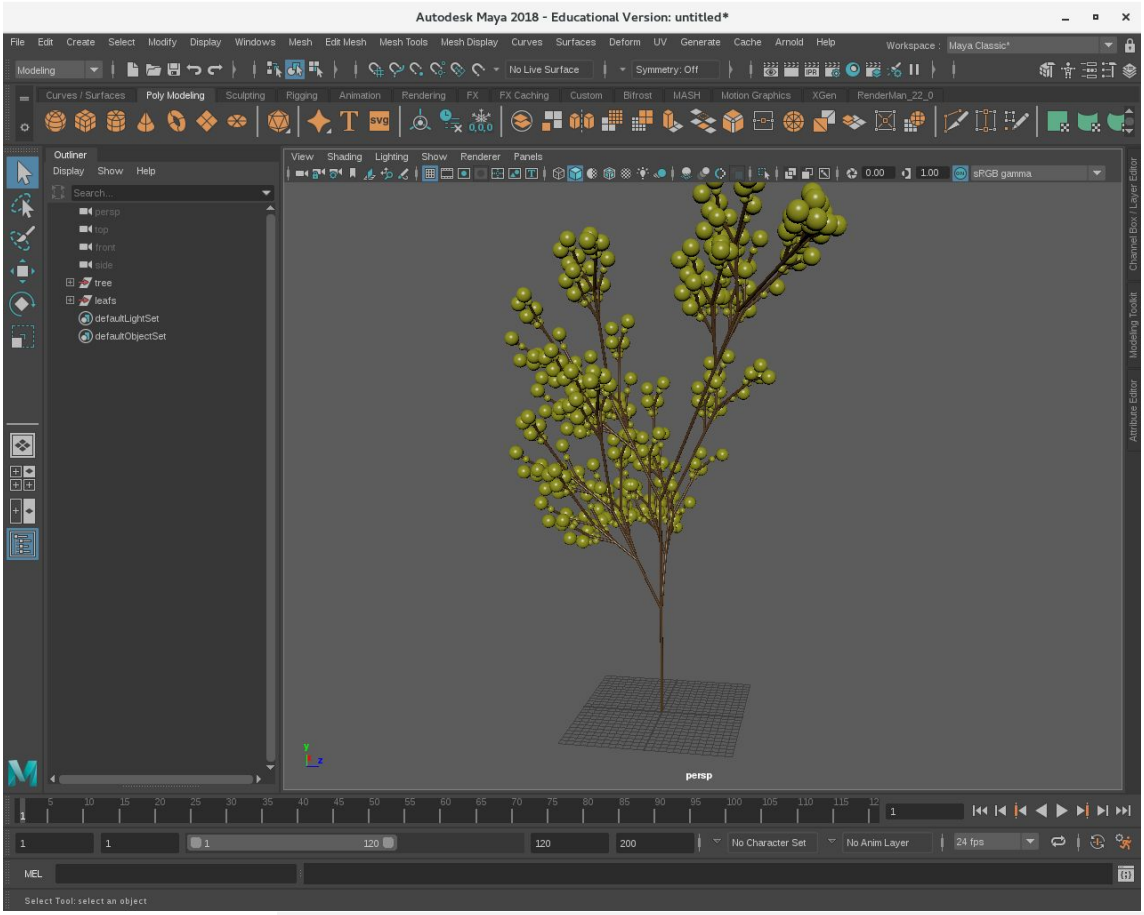

tree generator
TREE GENERATOR
CREATE YOUR DREAM TREE
TREE GENERATOR
Type:
One
Two
Three
Materials For Branches:
Phong
Lambert
Blinn
Materials For Leafs:
Phong
Lambert
Blinn
ADJUST TREE
RotateAngle:
25
Iterations:
4
StepLength:
1
RGB
HSV
R:
0.109
G:
0.056
B:
0.023
Range:
0 to 1.0
Tree Colour:
RGB
HSV
R:
0.109
G:
0.382
B:
0.023
Range:
0 to 1.0
Leaf Colour:
Apply
Cancel

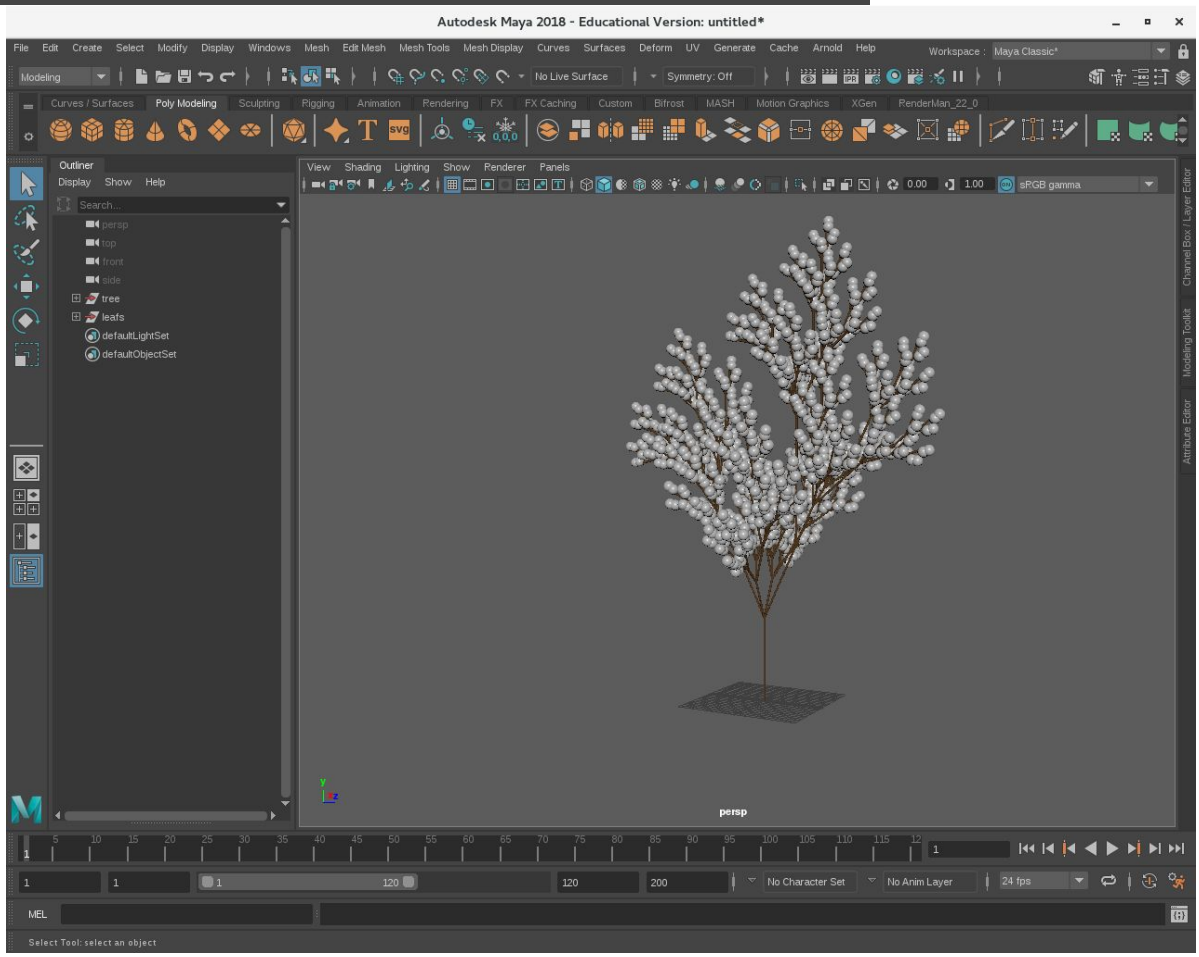

Autodesk Maya 2018 - Educational Version: untitled* --- phong1


Autodesk Maya 2018 - Educational Version: untitled*


Autodesk Maya 2018 - Educational Version: untitled*


Autodesk Maya 2018 - Educational Version: untitled* --- phong2


Autodesk Maya 2018 - Educational Version: untitled*
File Edit Create Select Modify Display Windows Mesh Edit Mesh Mesh Tools Mesh Display Curves Surfaces Deform UV Generate Cache Arnold Help
Workspace : Maya Classic*
Modeling
No Live Surface
Symmetry: Off
Curves / Surfaces Poly Modeling Sculpting Rigging Animation Rendering FX FX Caching Custom Bifrost MASH Motion Graphics XGen RenderMan_22_0
Outliner
Display Show Help
Search...
persp
top
front
side
tree
leafs
defaultLightSet
defaultObjectSet
View Shading Lighting Show Renderer Panels
sRGB gamma
persp
No Character Set
No Anim Layer
24 fps
MEL
Select Tool: select an object


Autodesk Maya 2018 - Educational Version: untitled*
File Edit Create Select Modify Display Windows Mesh Edit Mesh Mesh Tools Mesh Display Curves Surfaces Deform UV Generate Cache Arnold Help
Workspace : Maya Classic*
Modeling
No Live Surface
Symmetry: Off
Curves / Surfaces Poly Modeling Sculpting Rigging Animation Rendering FX FX Caching Custom Bifrost MASH Motion Graphics XGen RenderMan_22_0
Outliner
Display Show Help
Search...
persp
top
front
side
tree
leafs
defaultLightSet
defaultObjectSet
View Shading Lighting Show Renderer Panels
sRGB gamma
persp
No Character Set
No Anim Layer
24 fps
MEL
Select Tool: select an object