

Wheaton College

Grow-A-Tree

User's Guide

Austin O'Malley and Katie Merrin

Computer Graphics - COMP 365

Prof. Michael Gousie

April 12, 2016

## **Contents of the Grow-A-Tree Program Directory** (folder titled G4)

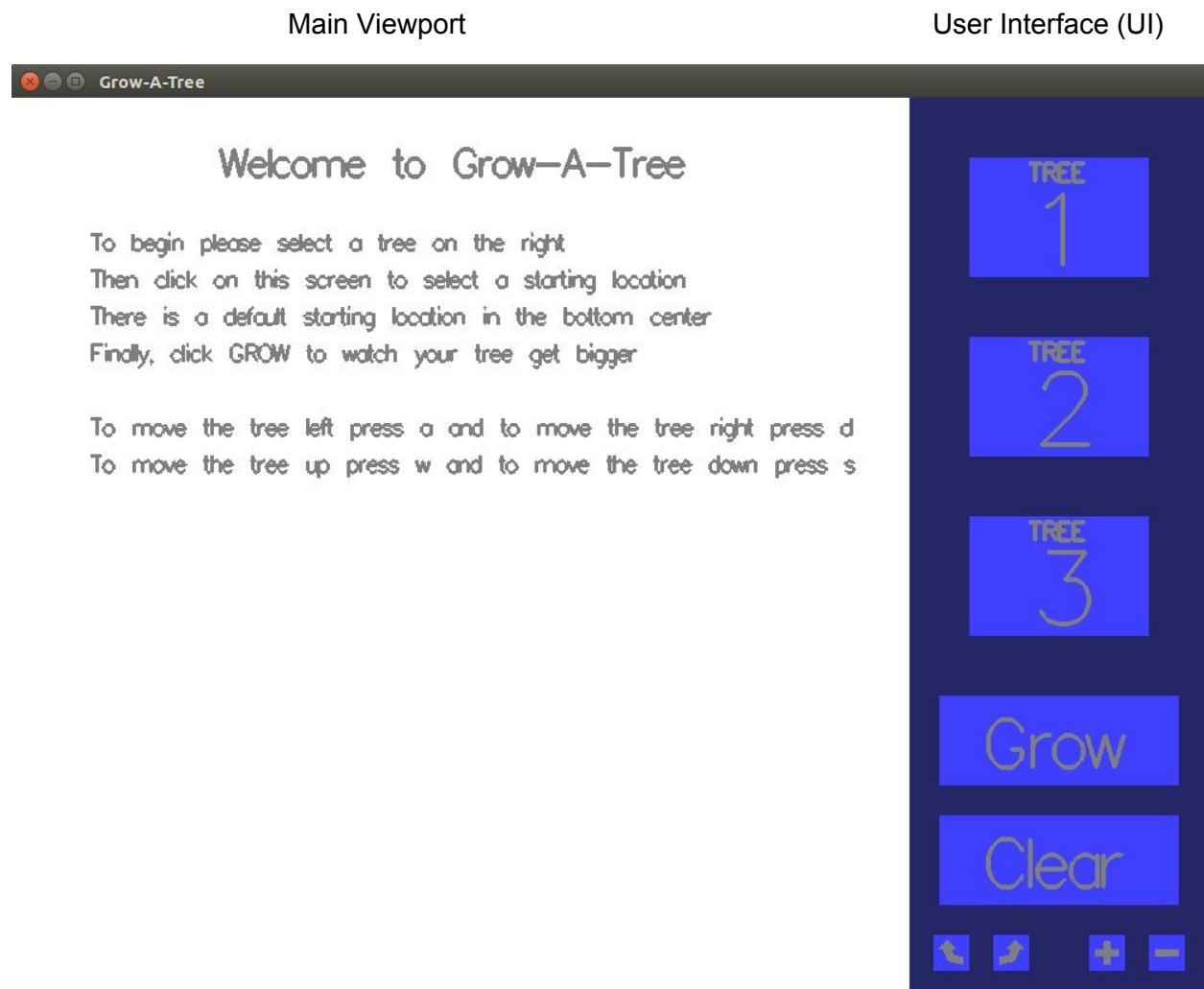
- G4.cpp - main program file to compile and run.
- g4input.txt - contains grammar to dictate tree growth.
- Makefile - contains instructions for the compiler.

## **Compiling and Running Grow-A-Tree**

1. Download and expand the file titled 'Merin\_OMalley\_G4.tar'.
2. Open your terminal or console window and change directory by typing 'cd' followed by a space and the path to the location of your expanded tar bundle.
3. To compile the program on a Linux system, type the command 'make G4' into the terminal and hit Enter on your keyboard.
4. To run the program on a Linux system, type the command './G4' into the terminal and hit Enter on your keyboard.
  - By default, this will use the grammar described in the included g4input.txt file
  - If you desire to use a custom input file, move your file into the G4 program directory, type the command './G4' into the terminal followed by a space and the name of your file (including the extension), and hit Enter on your keyboard.

## Interacting with Grow-A-Tree

Upon running the program, you will be presented with a main viewport on the left for planting and watching your trees grow, accompanied by a simple yet elegant user interface on the right. The first thing to notice are the quick-start instructions on the main viewport.



### Growing Your First Tree

1. Select a tree type by left-clicking one of the 3 tree options on the right.
2. Choose a planting location by left-clicking in the main viewport. The default location is centered in the lower half of the main viewport.

Note: It is advised that you choose a location that is centralized in the lower half to maximize visible growth on the screen.

3. Left-click the 'Grow' button up to seven times to watch your tree get bigger.
4. When you are finished with the current tree and wish to begin another, left-click the 'Clear' button and repeat this process.

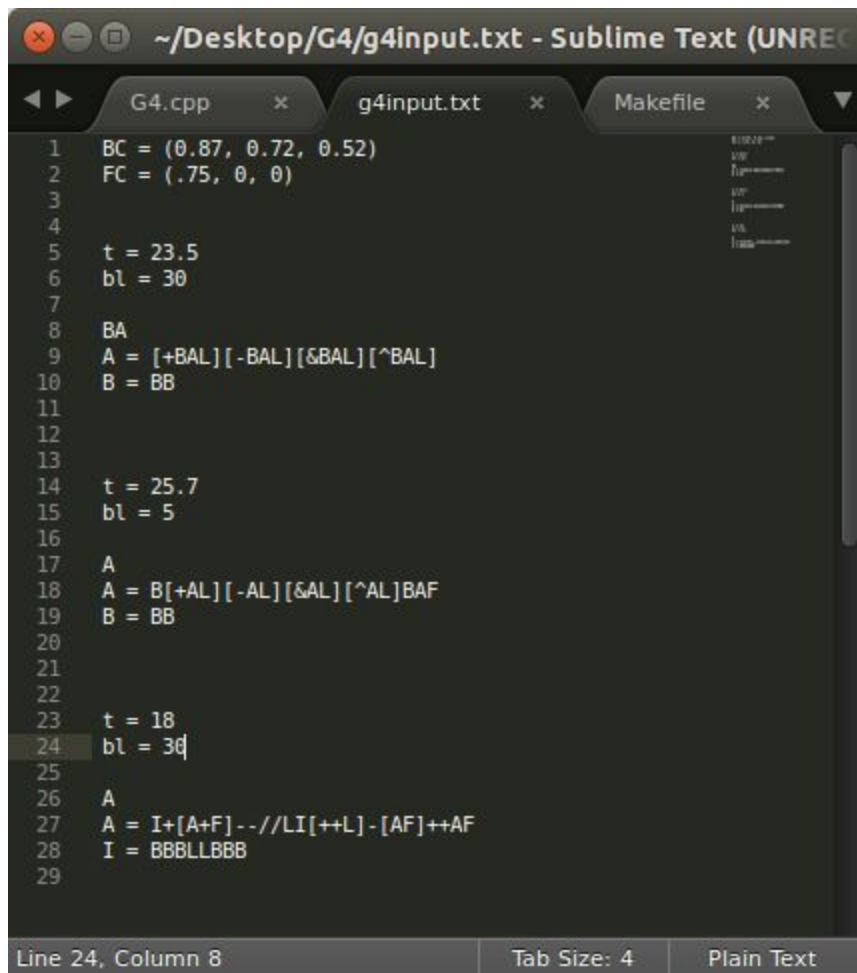
### **Exploring the rest of the User Interface (UI) and User-Interactions**

- In the bottom left of the UI, you will see two buttons: One labeled with a right arrow, and the other by a left arrow. These are used to rotate an active tree to the left or right around its y-axis respectively. This keeps the tree centralized in the viewport.
- In the bottom right of the UI, you will see two buttons: One labeled with a plus symbol (+), and the other by a minus symbol (-). These are used to zoom into and out from the current tree respectively.
- The 'wasd' keys are used to translate the tree around in front of the camera, while maintaining the camera's current distance from the tree.
  - To move left, press 'a' on the keyboard.
  - To move right, press 'd' on the keyboard.
  - To move up, press 'w' on the keyboard.
  - To move down, press 's' on the keyboard.
- The number keys are used to rotate the camera up, down and side-to-side. Note the view volume is recalculated so while the camera is changing linearly, the tree does not appear to change linearly.
  - To rotate the camera up, press '2' on the keyboard.
  - To rotate the camera down, press '8' on the keyboard.

- To rotate the camera left, press '7' on the keyboard.
- To rotate the camera left, press '1' on the keyboard.
- The 'q' key is used to quit the program.

## Editing or Creating a Custom Input File

Included 'g4input.txt' file:



The screenshot shows a Sublime Text editor window with the title bar reading '~/Desktop/G4/g4input.txt - Sublime Text (UNRECOGNIZED)'. The editor has three tabs open: 'G4.cpp', 'g4input.txt', and 'Makefile'. The 'g4input.txt' tab is active, displaying a list of commands and values. The status bar at the bottom indicates 'Line 24, Column 8', 'Tab Size: 4', and 'Plain Text'.

```

1 BC = (0.87, 0.72, 0.52)
2 FC = (.75, 0, 0)
3
4
5 t = 23.5
6 bl = 30
7
8 BA
9 A = [+BAL] [-BAL] [&BAL] [^BAL]
10 B = BB
11
12
13
14 t = 25.7
15 bl = 5
16
17 A
18 A = B[+AL] [-AL] [&AL] [^AL]BAF
19 B = BB
20
21
22
23 t = 18
24 bl = 30
25
26 A
27 A = I+[A+F] - -//LI[++L] - [AF]++AF
28 I = BBBLLBBB
29

```

### Editing 'g4input.txt'

Note: Do not add or delete any unspecified lines or characters.

1. Line 1 defines the branch color for all trees described in the file. Change the RGB number values within the parenthesis to set your desired color (ranging from 0-1).
2. Line 2 defines the flower color for all trees described in the file. Change the RGB number values within the parenthesis to set your desired color (ranging from 0-1).

### Defining a Tree (tree 1: lines 5-10, tree 2: lines 14-19, tree 3: lines 23-28)

3. Line 5 defines the angle of rotation for the branches making up the tree. Change the number following 't = ' your desired angle.
4. Line 6 defines the tree's individual branch length. Change the number following 'bl = ' to your desired branch length (must be a whole number greater than 0).
5. Line 8 defines the start symbol for your CFG (maximum of 10 characters).
6. Line 9 defines the rule for the first symbol of your CFG (maximum of 1 character).  
An '=' with a space on each side separates the symbol from the defining rule (maximum of 50 characters).
7. Line 10 defines the rule for the second symbol of your CFG. An '=' with a space on each side separates the symbol from the defining rule.

Note: A maximum of two symbols, defined by rules, are allowed for each of the three CFG's.

8. If desired, repeat this process for the other two CFG's defined on the lines specified at the beginning of this section.

### **Creating a Custom Input File**

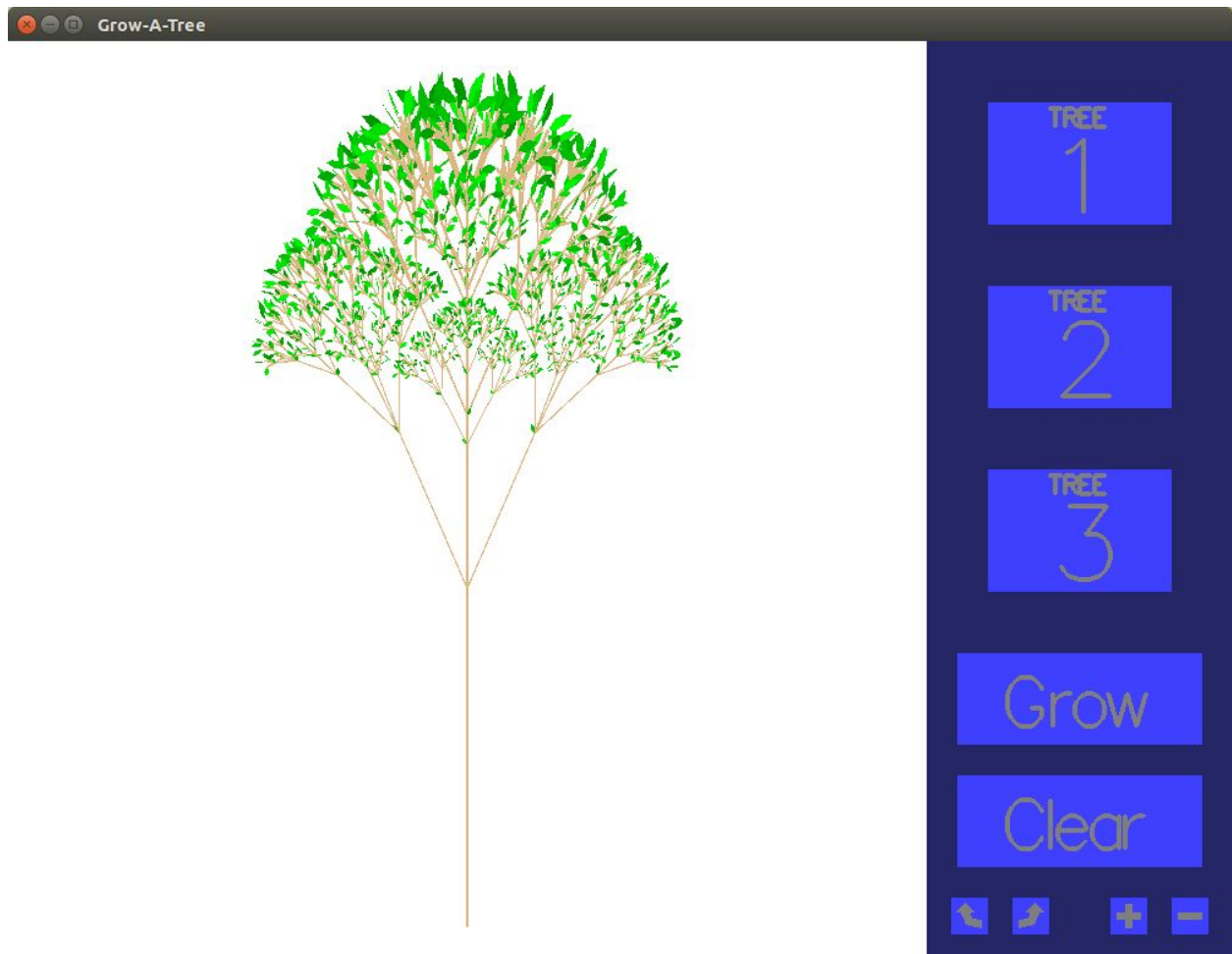
It is advised that you create a new .txt file and simply copy and paste the contents of the included 'g4input.txt' file to begin your own and edit according to the rules specified above.

### **Valid Symbols for Input File CFG**

- B: draw branch segment with color specified on line 1 and length specified on lines 6, 15, and 24 for trees 1, 2, and 3 respectively
- L: draw leaf
- F: draw flower with color specified on line 2
- [ : push matrix onto stack
- ] : pop matrix from stack
- +: rotate left on the z-axis
- -: rotate right on the z-axis
- & : rotate left on the x-axis
- ^ : rotate right on the x-axis
- \$ : rotate left on the y-axis
- % : rotate right on the y-axis

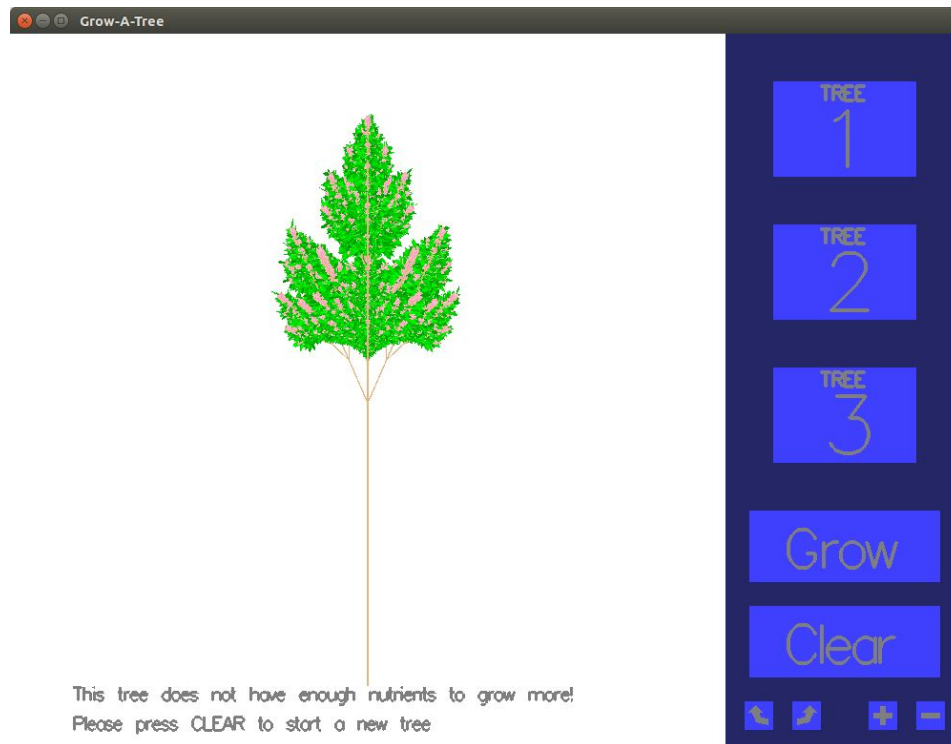
## Example Outputs of Included Trees Grammars

Tree 1:





Tree 2:



Tree 3:

