

User Manual

5 May 2020

Version 1

Sponsors

Dr. Kiona Ogle

Dr. Michael Fell

Mentor

Isaac Shaffer

TreeViz

Riley McWilliams

Qi Han

Haitian Tang

Daniel Rustrum

Alex Bentley



Contents

1 - Introduction	3
2 - Installation	3
3 - Configuration and Daily Operation	4
4 - Maintenance	4
4.1 - Amazon Web Server	4
4.2 - Firebase	5
5 - Trouble-shooting	7
4.1 - Amazon Web Server	7
4.2 - Visualization	8
6 - Conclusion	9

1 - Introduction

Thank you for working with TreeViz to create A User-friendly platform for visualizing Tree Growth. It is a custom-made system that visualizes the growth of trees over time based on the outputs of the ACGCA model, providing a more user-friendly way to learn about tree growth.. Some of the key highlights include: a user input control panel for easier access to the model, a dynamic render of a 3-dimensional tree and it's rings over time, a user survey for analytical data, and an availability online. The purpose of this document is to help you, the client, successfully install, administer, and maintain the product going forward. Our aim is to make sure that you can expand the audience of the ACGCA model to a broader audience, educating more people about tree growth.

2 - Installation

As part of final delivery, the ACGCA model wrapper should have been installed on a platform of your choice. Over time, however, you may want to move to a new platform or re-install the product. To install the wrapper on your system, you will need Python (<https://www.python.org/downloads/>). If you are on Windows, you will also need a C compiler, such as MinGW (<http://www.mingw.org/>).

Here are the steps for installing the model wrapper:

1. Download the ACGCA_Instance folder from the Github repository.



2. Go to the ACGCA_Instance directory in the command line and type in the following command to enter the src folder. “cd Model/ACGCA/src”
3. Open makefile.mk at “ACGCA_Instance/Model/ACGCA/src” and check if you have ACGCA.dll or ACGCA.so. If you are on Windows, you should have ACGCa.dll. If you are on Mac/Linux, you should have ACCGA.so. If you have the wrong ACGCA type in your makefile.mk, simply change all instances of ACGCA.xx to the one you need. The screenshot below shows all three instances.

```
M makefile.mk x
ACGCA_Instance > Model > ACGCA > src > M makefile.mk
1
2 | ACGCA.dll : excessgrowing.c misc_growth_funcs.c putonallometry.c growthloop.c rebuildstaticstate.c shrinkingsize.c Rgrowthloop.c instance.c
3 | gcc -shared excessgrowing.c misc_growth_funcs.c putonallometry.c growthloop.c rebuildstaticstate.c shrinkingsize.c Rgrowthloop.c instance.c -o ACGCA.dll
4
5 clean :
6 | rm ACGCA.dll
7
```

4. Once you have the correct ACGCA.xx, type “make -f makefile.mk” in the command line to compile the ACGCA model into a dll/so. If you are on Windows, you will need to use a C compiler to run this command, such as MinGW (<http://www.mingw.org/>).
5. Open run.py in “ACGCA_Instance/Model/ACGCA/src” and change the line shown in the screenshot below to your path to ACGCA.dll/ACGCA.so.

```
54 | # lower casex  
55 | mydll = ctypes.CDLL("/Users/<your-path>/ACGCA_Instance/Model/ACGCA/src/ACGCA.dll")  
56 | model = mydll.run_model
```

This line links the wrapper with the compiled model.

6. Now you should be able to run an instance of the ACGCA model. Type “python run.py” to run the instance. You can do this on multiple terminals to run multiple instances. You will need Python installed for this part (<https://www.python.org/downloads/>).

3 - Configuration and Daily Operation

There is no daily operation that needs to be done to keep the system working. Once you have ACGCA model wrapper running on your machine, people will be able to run the visualization on the website. However, you should check on the instances of the ACGCA model every week or so in case they stop running. If this happens, simply start them back up again.

4 - Maintenance

Services have been put into place to help maintenance of the product be as easy as possible. The two services that we used to do this were Amazon’s web services and Google’s Firebase. AWS does a few different services for us including hosting the website, processing API requests, and storing the data that is being passed through the system. Firebase was used to help authenticate the returning users to the website as well as store the information that the users supply in their surveys.

4.1 - Amazon Web Server

One of the benefits of AWS is that there is little to no maintenance needed. However, it would be wise to send an input from the website through and see if the system is still working properly. You will want to do this periodically while the website is being used. It is also good to keep in mind, on saturday and sunday the collection of outputs will be cleared out between the times 1am to 3 am.

4.2 - Firebase

Firebase is a web development platform that was developed by Google and like AWS, offers a few services. The services that were chosen to help the product were the Firebase's authentication and their real-time database. The plan that the development team has chosen was the 'Spark Plan' which is a free plan that has a few limitations in order to keep the cost down.

Firestore Console

The account that the product is linked to is the supplied gmail account that the development team has created for different services. In order to use Firestore you must first access the Firestore console under the gmail that has been supplied.

1. Navigate to firebase.google.com
2. In the upper right corner of the landing page there will be a button that says, 'go to console'. Click on that.
3. After being redirected from step 2, enter the information related to the gmail and click 'sign in'.
4. Once signed in, the next page will be the project page and the relevant project should be the only one, which is called 'ACGCA'.
5. Click on the project that says 'ACGCA' and you will be redirected to ACGCA's project page. This page will contain all of the relevant information to the product.

Authentication

Firestore has an authentication service that the development team has chosen to work with to make maintenance easy. Firestore does not allow there to be too many users to maintain the free plan so the users will need to be manually deleted in order to maintain that free status. Firestore can only hold about 100 accounts in a project before they start to charge for usage. The steps for manually deleting a user are as follows:

1. On the project page for the ACGCA model that is linked to the product there is a menu on the left hand side of the screen. Navigate to this menu and click on the link that says 'Authentication'.
2. Once on the authentication page there will be a few options near the top of the screen. The one that you will be concerned with for maintenance is the users page, which is the first page that is navigated to.
3. If you are under the 'Users' tab in the authentication page of Firestore we will delete the number of users that are listed.
4. Once on the users page, if there are any users that are registered with the product they will show up on this page. It will show their email, when they created the account, when they last signed in, and the unique id that Firestore uses to identify them.
5. Unfortunately, there could be a lot of users that have signed up in this project and there is no way to delete all of them at once so they need to be deleted individually.
6. On the far right of each user's row of information there will be a menu button that is signified by three dots. Click on this.

7. Once you have clicked on this a small menu will appear with the options, 'Reset password', 'Disable account', or 'Delete account'. Click on 'Delete account' to remove them from the list of users.

Database

Firebase has a real time database, which is another service the development team has decided to use for ease of access. Firebase only allows 1 gigabyte of storage in the database for the free plan and will not store any more data unless this database is emptied periodically. To make sure that we do not lose any data we will discuss how to export the database as a JSON for future record keeping as well as how to delete the database.

Export Database as JSON

1. On the project page for the ACGCA model that is linked to the product there is a menu on the left hand side of the screen. Navigate to this menu and click on the link that says 'Database', because a real-time database has already been created once the page has loaded the data will immediately be displayed.
2. The first page that will be loaded will be the 'Data' page which is the only one that you will need to be concerned with for regular maintenance.
3. Near the top of the page underneath the navigation there is a link that says <https://acgca-2396d.firebaseio.com/>. To the right of that there is a menu button that is indicated by three dots. Click on this.
4. A smaller menu will pop up and there will be an option that says to export JSON. When this option is clicked the database will download as a JSON file. Store this for record keeping.

Now that the database has been backed up we can safely delete the contents of the real-time database.

Deleting the Database

1. From the main page in the database section of Firebase, if there is anything in the database there will be a tree with a series of unique ids which will look like the following:

```
acgca-2396d
├── users
│   ├── -M5xbaRzpiOMq-Y-gC0S
│   ├── -M5xc07otsp0-xOxDJ33
│   ├── -M5yGwHtKOQM4IFmYaX-
│   ├── -M5yHF1giXDModmXUjoz
│   ├── -M63MDKVtLrNaa_mV3jk
│   ├── -M66lm4YOucTyTYv7dQv
│   ├── -M67-d84uPMAIa9DBFAn
│   ├── -M67q43KhvkSbH-KUBHR
│   ├── -M68BKCPcROLPLZII6n
│   ├── -M68EWGtNZrlvReoxWyh
│   ├── -M6Cp24V9D4sZGh2wm57
│   ├── -M6RYook7sDfLqtbKaTQ
│   └── -M6RbjTp7Aq5v6wev9LH
```

2. To delete the entire database, hover your mouse over the 'users' collection and a small 'x' will appear next to it. Click on this 'x' to delete the database.
3. If we click on this a warning will appear to alert you that you will be permanently deleting the information. If you are sure you would like to delete the info click 'ok'.

Now that you are familiar with the page if we navigate to the 'Usage' tab that is near the top of the page we will be redirected to the analytics page that will tell us how much storage is being used and how many connections are being made. This is where you will be able to track the amount of data that is being stored and when you should delete the information that is in the database.

5 - Trouble-shooting

This section goes over some common errors that may occur and what to do when they do occur.

4.1 - Amazon Web Server

What's wrong:

Inputs are not being passed.

How to trouble-shoot it:

First ensure there hasn't been any changes AWS has implemented to the following services that changes have been breaking peoples programs. Services:

- API Gateway
- SQS
- DynamoDB
- Lambda

For additional information you can also check the service cloud watch for log information that AWS Logs, this provide insight to not only the services high-level functions, but the internal workings that AWS expose.

Besides that you may need to dive into the documentation that AWS provides(<https://docs.aws.amazon.com/>) and make small incremental changes until the problem has been solved.

What's wrong:

Website doesn't load.

How to trouble-shoot it:

First ensure there hasn't been any changes AWS has implemented to the following services that changes have been breaking peoples programs. Services:

- S3
- Route 53, if in use

For additional information you can also check the service cloud watch for log information that AWS Logs, this provide insight to not only the services high-level functions, but the internal workings that AWS expose.

Besides that you may need to dive into the documentation that AWS provides(<https://docs.aws.amazon.com/>) and make small incremental changes until the problem has been solved.

4.2 - Visualization

There are a couple things that could go wrong when visualizing a tree. If the answer cannot be found here, please go to <https://threejs.org/docs/index.html#manual/en/introduction/Creating-a-scene> for the official Three.JS documentation.

What's wrong:

The visualization looks strange or different from what is expected.

How to trouble-shoot it:

There are several commented-out console.log() statements in the visualization functions. The drawTree() and drawRings() functions are what draw the trees and rings, respectively, on the page. Uncommenting these log statements, by deleting the preceding two backslashes, will display the logs in the console window on a browser (ctrl-shift-i for Windows; command-option-i for Mac). If the log data is inconsistent with the model's output, then there may be a problem with passing data from the model to the website. If the log data *is* consistent with the model's output, then there may be an incorrect use of a variable in the visualization functions. A good place to start checking for this incorrect variable is in the constructor functions for the tree and ring geometries. A commented line is above each constructor, telling exactly which variables should be used to build a certain geometry.

Additionally, in the getData() function, there is a commented-out console.log() statement that prints the entire output of the model after it has been retrieved by the website. This output is saved as this.resultJson, so uncomment the line that prints that to see it.

What's wrong:

The visualization window is black or nonexistent.

How to trouble-shoot it:

This problem is likely within the `initialize()` function or the `animate()` function. `initialize()` sets up the scene and renderer, and `animate()` renders the scene to display objects. If `animate()` is broken, then the window should still be there, but completely black. If `initialize()` is broken, then the window may not even show up. In both of these cases, check to make sure there are no typos. Then, make sure the constructors for the cameras are correct. This can be done by comparing them to the commented line above them, which shows how they are written. The cameras' positions, fields of view, aspect ratios, and view plains should not cause the window to go dark or disappear, but it may mess up how the scene looks. Both cameras should have the following data: `Three.PerspectiveCamera(90, canvasWidth / canvasHeight, 0.1, 1000)`.

6 - Conclusion

With this manual, you have all the details necessary to install and maintain the product. It has been a pleasure working with you on A User-friendly Platform for Visualizing Tree Growth. TreeViz wishes you success for the future of the project and are glad to have been able to help build it.

Best wishes from your developers,

Riley McWilliams

Alexander Bentley

Daniel Rustrum

Haitian Tang

Qi Han