

# **Programming Project Report**

Pablo Guerra  
010849976

## **Problem Statement:**

For our first assignment we were tasked with the job of taking a file containing data points provided by the user, generating a list of commands using those data points and using those commands to create charts. In order to do this, two separate programs were created that both provide different functionality. The first program is intended to request a file from the user containing data points as well as requesting the chart type to be created. It will then parse all the data points in the file and generate a list of drawing commands based on the selected chart type. These commands are written and saved to a file. The second program is intended to request a file from the user that has the generated draw commands and parses the data to display the graph using OpenGL.

## **Design:**

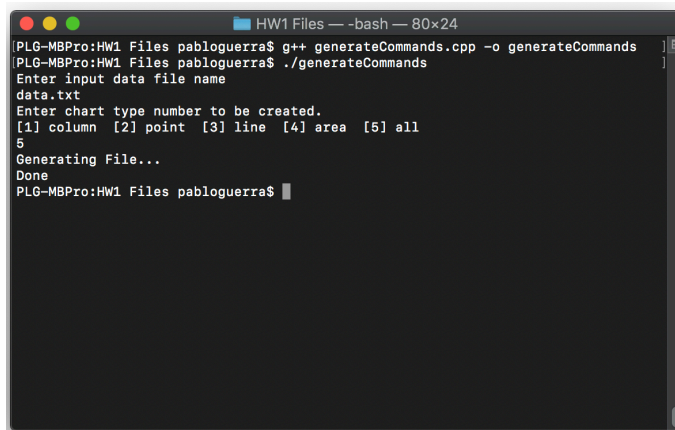
In order to make these programs work effectively certain design decisions had to be made. The first major decision made was how to parse the data points provided by the user. This can be done in many different ways, but the main differences are the choice of delimiter. Originally commas were used as the delimiter but after looking at the sample data provided this was changed to spaces. This is important because if any other delimiter is used with this program then it will not work as intended. Another important design choice made was to use the vector data structure since it is impossible to tell how many data points or commands will be provided and used. This allows us to generate as many commands as needed.

## **Implementation:**

The first step of creating these programs was to have a basic program in place with minimal functionality. This included simply printing to the command line, getting user inputs and opening and reading files. With this basic functionality ready, the core features were ready to be implemented as individual functions that would be called when needed. In our first program, the main important functions include opening a file and appending lines, creating a basic graph which consists of the axis and tick marks, and four chart functions which generate chart specific commands. In our second program, every possible draw command was created into separate functions and a file reading function was included to retrieve the commands.

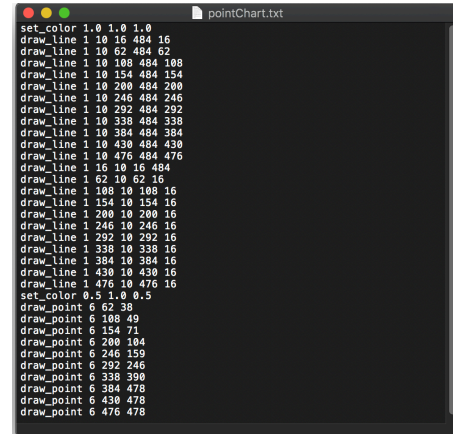
## Testing:

In order to test that these programs were working as intended, the provided sample data points was used to compare the output of the commands and graphs to the commands and images provided. The first step is to run the first program and provide the data points file name as well as the chart type. An image of this can be seen below.



```
HW1 Files — -bash — 80x24
PLG-MBPro:HW1 Files pabloguerra$ g++ generateCommands.cpp -o generateCommands
PLG-MBPro:HW1 Files pabloguerra$ ./generateCommands
Enter input data file name
data.txt
Enter chart type number to be created.
[1] column [2] point [3] line [4] area [5] all
5
Generating File...
Done
PLG-MBPro:HW1 Files pabloguerra$
```

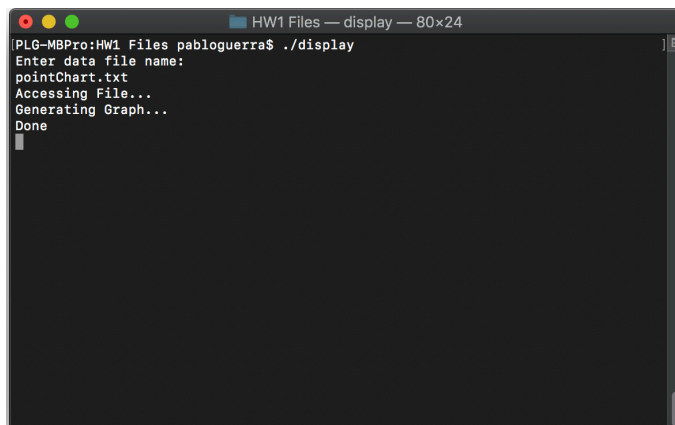
First program execution



```
pointChart.txt
set_color 1.0 1.0 1.0
draw_line 1 10 16 484 16
draw_line 1 10 62 484 62
draw_line 1 10 108 484 108
draw_line 1 10 154 484 154
draw_line 1 10 200 484 200
draw_line 1 10 246 484 246
draw_line 1 10 292 484 292
draw_line 1 10 338 484 338
draw_line 1 10 384 484 384
draw_line 1 10 430 484 430
draw_line 1 10 476 484 476
draw_line 1 16 10 484
draw_line 1 62 10 484
draw_line 1 108 10 484
draw_line 1 154 10 484
draw_line 1 200 10 484
draw_line 1 246 10 484
draw_line 1 292 10 484
draw_line 1 338 10 484
draw_line 1 384 10 484
draw_line 1 430 10 484
draw_line 1 476 10 484
set_color 0.5 1.0 0.5
draw_point 6 62 38
draw_point 6 108 49
draw_point 6 154 71
draw_point 6 200 104
draw_point 6 246 159
draw_point 6 292 246
draw_point 6 338 390
draw_point 6 384 478
draw_point 6 430 478
draw_point 6 476 478
```

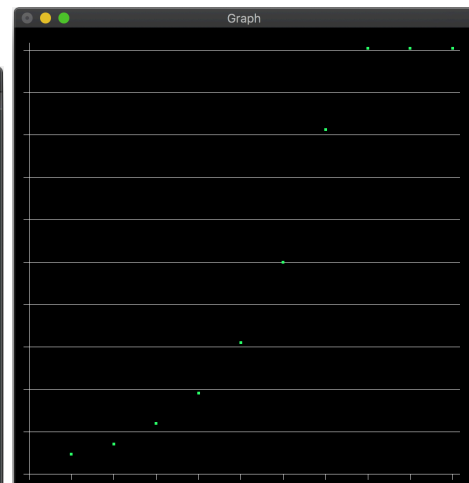
Sample output file

The second step is to run the second program and enter the chart file name to be displayed. The result of this is shown in the images below.



```
HW1 Files — display — 80x24
PLG-MBPro:HW1 Files pabloguerra$ ./display
Enter data file name:
pointChart.txt
Accessing File...
Generating Graph...
Done
```

Second program execution



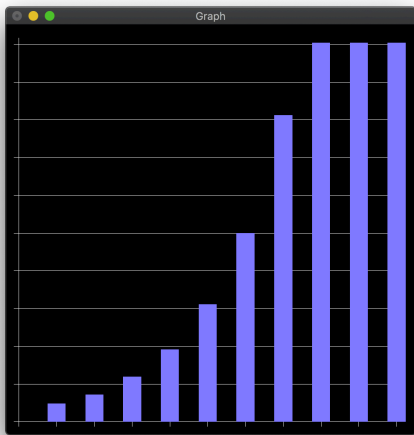
Point Chart

As seen from the above images, the programs do produce the desired results and match those provided in the sample data and images.

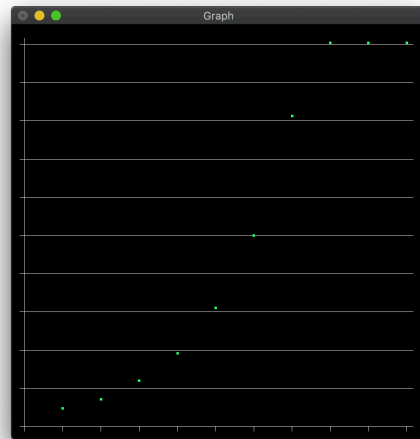
## Conclusions:

Overall, the programs work as intended and they provide the expected results. The assignment was a little challenging to determine certain values and equations to use in order to obtain the right points but the actual implementation of using OpenGL was straight forward. This was a great introduction to using the OpenGL library and provided great insight on the functions needed to get everything working right.

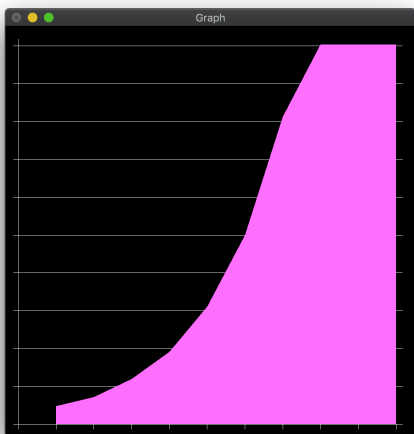
## Images:



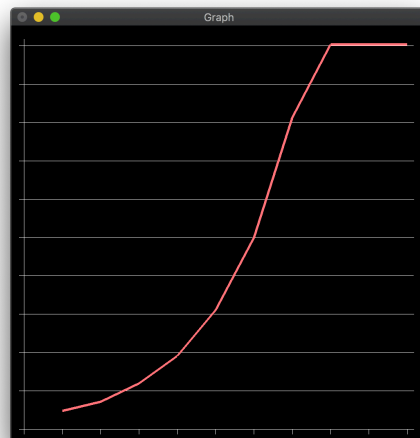
Column Chart



Point Chart



Area Chart



Line Chart