



# GRAPHULATOR: Between Dimensions



# Little History

**Graphing calculators** first came on the market in the 1980s. They replaced basic calculators with more or less numerical function – early calculators and adding machines had a numeric keypad and buttons for operators for addition, subtraction, multiplication and division, and dealt exclusively with whole numbers and fractions.



# Basic Overview

**Graphulator is a graphing calculator.**

**"Plots 3D graphs".**

It comes in handy when you have to visualize graphs of complex equations.

**Graphulator will automate the graphs for you.**

**"Sophisticated than Basic Calculators"**

Main Features:

- User can add custom graphs
- Differentiate negative region
- Scale/Rotate the graph

**Thanks to Desmos for the inspiration,**  
<https://www.desmos.com/calculator>



# Let's see a demo now

To see how it works

“

*The best way to have a  
good idea is to have lots of  
ideas.*

# The concept

First we had the idea of creating a calculator that solves equations and also shows 2D graphs. Later, we left the 2D part and scaled it to 3D only.





# Library, Methods & IDE: OpenGL

- Graphics Library used : OpenGL (GLUT)
- Method used to parse equation : Reverse Polish Notation & Stack Method
- IDE used: Clion(For Linux and Windows ) & Xcode (For macOS)

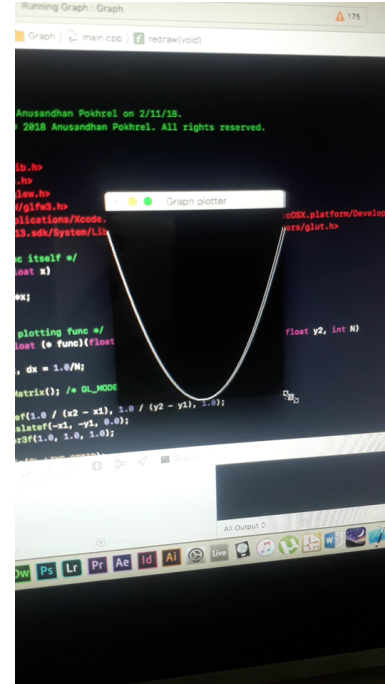
# The development phases



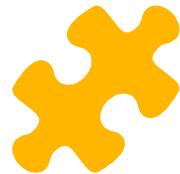
```
Activities Terminal Mon 10:23 PM ashish@CRLannister: ~/Documents/Codes/C++/mini project/project1/project
File Edit View Search Terminal Help

*****
INTEGRATION MENU:
CALCULATES LINEAR EQUATION OF TYPE AX^n+BX^{N-1}...+CONSTANT:
ENTER THE ORDER (n):4
ENTER LOWER LIMIT: (a):5
ENTER UPPER LIMIT: (b):7
ENTER THE COEFF. OF THE Nth ORDER EQUATION:
(a1x^N + a2x^{N-1} + ..... +a(N-1)x + constant
a0 4
a1 5
a2 7
a3 9
a4 2
sh: 1: pause: not found
sh: 1: cls: not found
13786.3137863

*****
MAIN MENU
*****
ENTER 1 FOR ARITHMETIC CALCULATION
ENTER 2 FOR TRIGONOMETRIC CALCULATION
ENTER 3 FOR PERMUTATION AND COMBINATION CALCULATION
ENTER 4 FOR SOLVING EQUATION CALCULATION
ENTER 5 FOR POWER OF A NUMBER CALCULATION
ENTER 6 FOR MATRIX CALCULATOR:
ENTER 7 FOR COMPLEX MANIPULATION:
ENTER 8 FOR INTEGRATION CALCULATION:
ENTER 9 FOR DETERMINANT CALCULATION:
ENTER 10 FOR EXIT
```

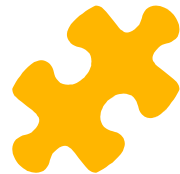






# How we used the graphics library?

The **OpenGL Utility Toolkit (GLUT)** is a library of utilities for OpenGL programs, which primarily perform system-level I/O with the host OS. We used it for functions like window definition, window control, (rendering characters) and monitoring of keyboard and mouse input, which includes almost everything for the project.



# RPN & Stack

## Method (Shunting Yard

### Algorithm)

**Reverse Polish notation (RPN)**, also known as **Polish postfix notation** or simply **postfix notation**, is a mathematical notation in which operators follow their operands.

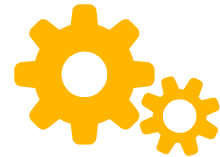
E.g:  $a+b*c$  in RPN is  $a\ b\ c\ *\ +$

# Algorithm

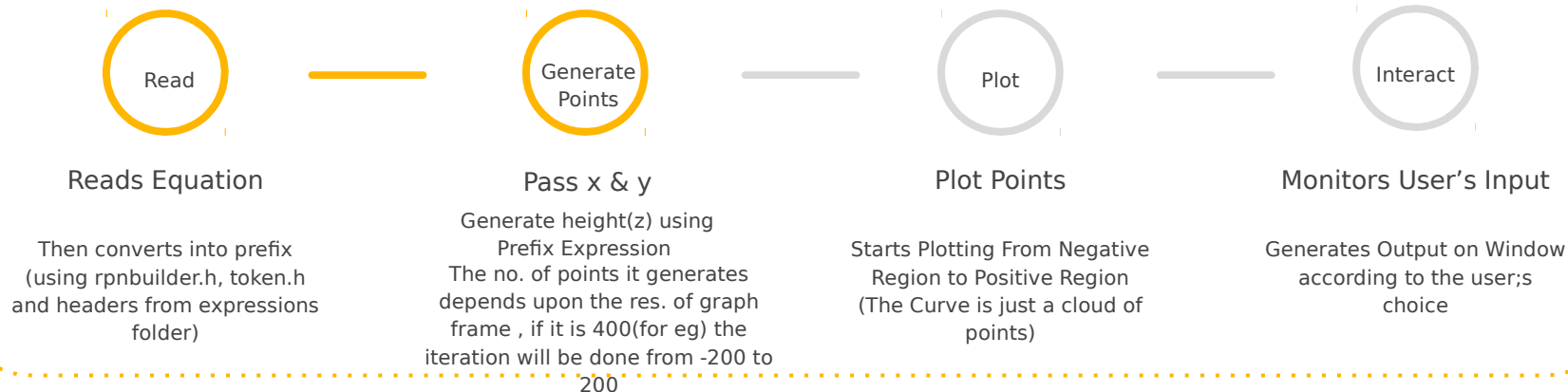
```

15 7 1 1 + - ÷ 3 × 2 1 1 + + - =
15 7 1 1 + - ÷ 3 × 2 1 1 + + - =
15 7 1 1 + - ÷ 3 × 2 1 1 + + - =
15 7 1 1 + - ÷ 3 × 2 1 1 + + - =
15 7      2 - ÷ 3 × 2 1 1 + + - =
15      5 + 3 × 2 1 1 + + - =
      3 3 × 2 1 1 + + - =
      3 3 × 2 1 1 + + - =
        9 2 1 1 + + - =
        9 2 1 1 + + - =
        9 2 1 1 + + - =
        9 2      2 + - =
          9      4 - =
              5 =
              5
    
```

Token	Type	Stack	Actions
15	Operand	15	Push onto stack.
7	Operand	15 7	Push onto stack.
1	Operand	15 7 1	Push onto stack.
1	Operand	15 7 1 1	Push onto stack.
+	Operator	15 7 2	Pop from stack twice (1, 1), calculate (1 + 1 = 2) and push onto stack.
-	Operator	15 5	Pop from stack twice (7, 2), calculate (7 - 2 = 5) and push onto stack.
÷	Operator	3	Pop from stack twice (15, 5), calculate (15 ÷ 5 = 3) and push onto stack.
3	Operand	3 3	Push onto stack.
×	Operator	9	Pop from stack twice (3, 3), calculate (3 × 3 = 9) and push onto stack.
2	Operand	9 2	Push onto stack.
1	Operand	9 2 1	Push onto stack.
1	Operand	9 2 1 1	Push onto stack.
+	Operator	9 2 2	Pop from stack twice (1, 1), calculate (1 + 1 = 2) and push onto stack.
+	Operator	9 4	Pop from stack twice (2, 2), calculate (2 + 2 = 4) and push onto stack.
-	Operator	5	Pop from stack twice (9, 4), calculate (9 - 4 = 5) and push onto stack.



# Work Flow



# Features used to enhance program



Move  
Camera &  
Zoom

Toggle Axis  
Frame &  
Invert  
Negatives

Surf Between  
Different  
Functions

Increase/  
Decrease Point  
Size



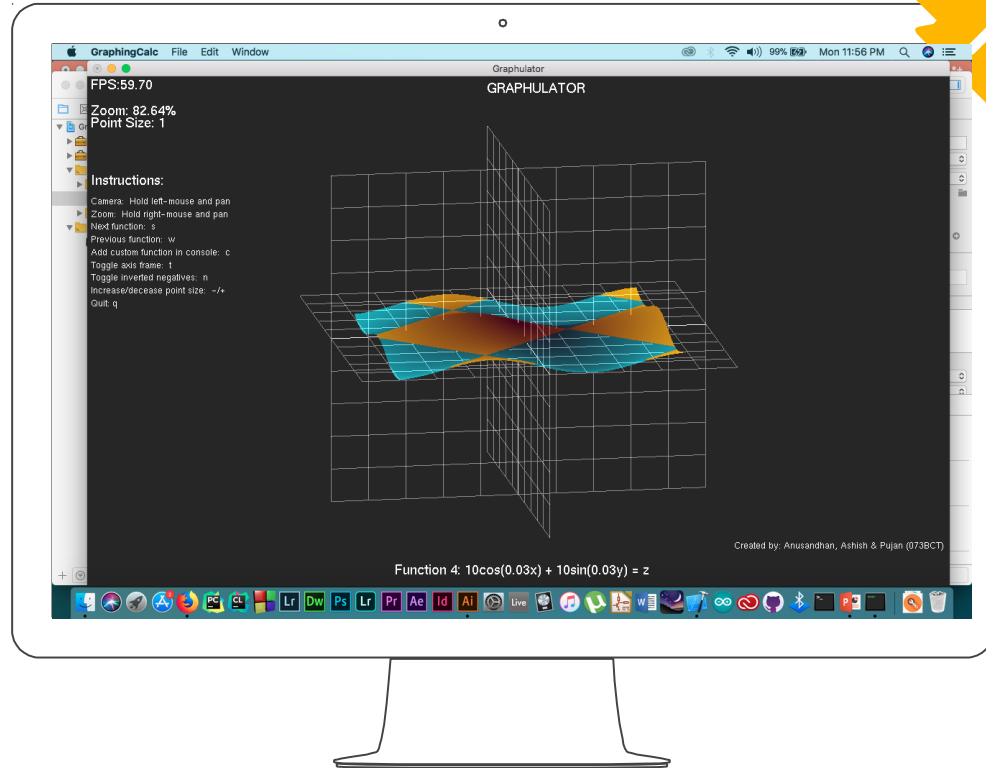
# Limitations of the product

- Should have used GLFW since it is advanced and with which a far better GUI could be developed.
- Still buggy due to lack of proper memory management
- Sometimes problem may arise while plotting some of the custom functions
- Lack of 2D/3D switch mode
- Cannot plot 2 different graphs in a single time (to find intersecting planes, etc)
- Cannot take input function in polar form

# Final Product

Still a prototype.

We wish to develop it  
further in future..





# References

- [https://en.wikipedia.org/wiki/OpenGL\\_Utility\\_Toolkit](https://en.wikipedia.org/wiki/OpenGL_Utility_Toolkit)
- <https://github.com/olegskl/rpn-calculator> (for the RPN algorithm code)
- [https://en.wikipedia.org/wiki/Shunting-yard\\_algorithm](https://en.wikipedia.org/wiki/Shunting-yard_algorithm)
- <https://www.youtube.com/channel/UCiFAmp2Crv66cQA-9SPje1A>
- <https://www.youtube.com/user/sonarsystemslimited>
- <https://www.youtube.com/user/iamdavidwparker>
- And Google ☐



# Thank

**From the developers'  
team.**

you!  
Any questions?

