Project Abacus

Specifications

A User should be able to:

- actually evaluate basic expressions like 3 + 5
- graph functions from $\mathbb{R}$ to $\mathbb{R}$ and restrict its domain
- graph functions in 3D, so $\mathbb{R}^2 \to \mathbb{R}$ and again be able to restrict domains
- graph implicit functions, i.e. functions of the form $f(x, y) = 0$ and $f(x, y, z) = 0$.
- plot multiple graphs on the same set of axes
- do operations on the functions they define, i.e. if they have a function $f(x)$ and $g(x)$, they should be able to plot $f(x) + g(x), f(x) - g(x), f(x)g(x), \frac{f(x)}{g(x)}, f(g(x))$
- zoom in and out of graph
- be able to rotate viewpoint around for 3D graphs

Possible extensions:

- Plotting parametric curves and surfaces
- Plotting complex functions
- Finding derivatives and integrals

Classes

Entities:

- Expression
  - Converts strings to Expression objects that we can manipulate and evaluate (likely using Abstract Syntax Trees)
  - Stores the functions/expressions that the user inputs
- Axes
  - The coordinate system and the space where the graphs are being stored
  - Handles the scales of the axes
- Viewpoint
  - The 'camera' for 3D graphs, handles the orientation, position, distance from graphs/axes

Use cases

- Graph
  - The graph of an expression
  - Uses expression to evaluate points at respective domain

Presenters:

- Renderer
  - Actually used to draw/render the graphs

Gateways:

- UserInterface
    - Converting user input into something our program can understand
    - Although we will be starting out with a command line interface, we ideally want to transition to a graphical user interface