

## COMS W4115 Project Proposal: MDraw

*Team: Dongxiang Yan (dy2224)*

*Jingyu Shi (js4151)*

*Huimin Sun (hs2740)*

### *1. Introduction and Motivation*

MDraw means “my draw”, which is designed as a 2D graphics drawing and manipulation language. It enables users to automatically draw and manipulate numerous graphics by taking advantage of this programming language.

Now days, although there are many popular graphics drawing and manipulation software solutions, there is no dedicated text-based language to facilitate and automate this process. Some repeating works are boring and time consuming. The case is even worse for complex graphics. Some graphics are hard to be repeated and scaled manually. By using MDraw, these issues could be resolved with ease.

We plan to fulfill the following key features, but not limited to:

- Functionality of drawing basic graphic shape blocks, like
  - Lines
  - Points
  - Curves
  - Circles, ellipses
  - Triangles, squares, rectangles
  - Polygons etc.
- Functionality of manipulation, like
  - Move (in and between different layers)
  - Rotation
  - Scaling
- Functionality of multiple layers
- Functionality of color
- Functionality of dynamic effect

### *2. Preliminary Examples*

- Comments  
The characters `/*` introduce a comment. The character `*/` terminates the comment.

- Identifiers (Names)
 

The identifiers are consisted by a sequence of letters, digits and the underscore “\_”. The first character must be letter. There is no difference for upper and lower case letters.
- Keywords
 

The keywords are listed as below:  
int, char, sting, float, double, if, else, elseif, for, do, while, switch
- “~” is reserved as a function for drawing points and lines between points. Since all graphics are consisted by points and lines between points, by defining and fulfill this function, we could draw whatever we want.
  - `/* Draw a point (1, 2) */`  
`point_01 = ~(1, 2);`
  - `/* Draw a line between two points (1, 2) and (3, 4) */`  
`line_01 = ~((1, 2), (3, 4));`
  - `/* Draw a triangle (lines between three points) */`  
`triangle_01 = ~((1, 2), (3, 4), (5, 6), (1, 2));`
  - `/* Draw a rectangle */`  
`rectangle_01 = ~((0, 0), (4, 0), (4, 3), (0, 3), (0, 0));`
- Several examples for graphics manipulation: Move, Copy, Rotate, Combine, Delete, Remove, Scale.
  - `/* Move a line to right 3 units and top 5 points at current layer*/`  
`move((3, 5), line_01);`
  - `/* Move a line to another layer*/`  
`move(line_01, layer_02);`
  - `/*Copy a line to another layer to get a new line*/`  
`line_03 = copy(line_01, layer_03);`
  - `/* Rotate a line clock wisely 90 degrees at point (1, 2)*/`  
`rotate(90, (1, 2), line_01);`
  - `/* Rotate a line anti-clock wisely 90 degrees at point (1, 2)*/`  
`rotate(-90, (1, 2), line_01);`
  - `/* Rotate a triangle clock wisely 45 degrees at point (1, 2) */`  
`rotate(45, (1, 2), triangle_01);`

- `/* Combine a line and a triangle as a new graph. This means the new graph contains one line and one triangle*/  
graph_01 = combine(line_01, triangle_01);`
- `/* Delete line_01*/  
delete(line_01);`
- `/* Remove line_01 from graph_01*/  
remove(line_01, graph_01);`
- `/* Scale a square to 5 times of original area with the same central point*/  
scale(square_01, 5);`