

# ECS 175 Project 1

---

“A Simple Two-Dimensional Drawing System”

## Author

---

Name: Tianyang Zhan

SID: 913739374

## Environment

---

- Operating System:Ubuntu 12.04 amd64
- DE:Using Code::Blocks 10.05 for coding, Make for compiling and running
- G++ Version:Ubuntu/Linaro 4.6.3-1ubuntu5

## Prerequisite

---

- OpenGL
- GLUT

## Commands

---

### 1.Compile the program using Makefile

```
$ make
```

### 2. Run the program

```
$ ./graphics
```

### 3. Clean up the folder AFTER program ends

```
$ make clean
```

## Running the program

---

**1. Setup config files (not required)** (Refer to [\*Config Files\*](#))

**2. Run the program** (Refer to [\*Commands\*](#))

```
$ make
```

```
$ ./graphics
```

**3. Use Command Line to set pre-launch settings** (Refer to [\*User Menu\*](#))

**4. Use Keyboard to operate on objects** (Refer to [\*Keyboard Interactions\*](#))

**5. Right Click On Window for Menu to set parameters** (Refer to [\*User Menu\*](#))

**6. Exit the program, outputs save to files automatically**

**7. Clean up the folder** (Refer to [\*Commands\*](#))

```
$ make clean
```

## Config Files

---

**1.Data file used to display shape (default file: data)**

|         |                                 |
|---------|---------------------------------|
| 2       | number of polygons              |
|         | definition of 1st polygon:      |
| 4       | number of points of 1st polygon |
| 0.0 0.0 | coordinates of 1st point        |
| 1.0 0.0 | coordinates of 2nd point        |
| 1.0 1.0 | coordinates of 3rd point        |
| 0.0 1.0 | coordinates of 4th point        |
|         | definition of 2nd polygon:      |
| 3       | number of points of 2nd polygon |
| 3.0 0.0 | coordinates of 1st point        |
| 3.0 1.0 | coordinates of 2nd point        |
| 2.0 0.0 | coordinates of 3rd point        |

## 2. Setting file used to config the window (default file: setting)

|             |                                 |                         |
|-------------|---------------------------------|-------------------------|
| 700 500     | Viewport width and height       | (0~) (0~)               |
| 0 0 0       | Viewport background color R,G,B | (0~255) (0~255) (0~255) |
| 255 255 255 | Line and shape color R,G,B      | (0~255) (0~255) (0~255) |
| dda         | Line Drawing Algorithm          | (dda / bresenham)       |
| 800 600     | Window width and height         | (0~) (0~)               |
| 1 1 1       | Window background color R,G,B   | (0~1) (0~1) (0~1)       |

## Input/Output Files

---

**Inputs (1 OR 2 Files) :** data file (REQUIRED), setting file

- User is able to specify the name of the **data file** when prompted

**Outputs (2 Files) :** data file, setting file

- Program automatically save all changes to output files at exit

## User Menu

---

### 1. Pre-launch Settings (Command Line)

- **Specify input file name:**

Do you want to specify the input data file? (y/n) **y**

Changing Input File to (ex: data.txt) **data1.txt**

- **Specify display window size:**

Do you want to specify the window size? (y/n) **y**

Setting Window Size (ex: 800 600): **1200 800**

## 2. Runtime Menu (Right Click On the Graphic Window)

\*\* For *(Input in Terminal)* items, check command line prompt for custom input \*\*

### **Menu Entries:**

- ❖ **Current Object:** Select object by ID to do operation  
0  
...
- ❖ **Transformation Parameters:** Set Transformation parameters  
Scaling factor (Input in Terminal)  
Translation vector (Input in Terminal)  
Rotation angle (Input in Terminal)  
Set All to Default
- ❖ **Line Drawing Algorithm:** Select the algorithm used to draw lines  
DDA  
Bresenham's
- ❖ **Resize:**
  - ❖ **Resize Window:** Set the Graphics Window size  
800x600  
...  
Custom (Input in Terminal)
  - ❖ **Resize Viewport:** Set the Viewport size  
700x500  
...  
Custom (Input in Terminal)
  - ❖ **Resize Clipping Window:** Set the Clipping Window size  
Set as viewport size  
Custom (Input in Terminal)
- ❖ **Color:**
  - ❖ **Line Color:** Set Line and Shape color  
Red  
...  
Custom (Input in Terminal)
  - ❖ **ViewPort Color:** Set ViewPort background color  
Black  
...  
Custom (Input in Terminal)
  - ❖ **Background Color:** Set Graphic Window background color  
Black  
...
- ❖ **Add New Object (Input in Terminal)** Add new line or polygon to the screen
- ❖ **Exit** Exit the program

# Keyboard Interactions

---

## DEFAULT PARAMETERS:

Translation Vector: 5.0 5.0

Rotation Angle: 0.1

Scaling Factor: 1.1

## FUNCTIONAL KEYS:

### 1. Translation

- **W** : translate current object by **+y** defined in translation vector
- **A** : translate current object by **-x** defined in translation vector
- **S** : translate current object by **-y** defined in translation vector
- **D** : translate current object by **+x** defined in translation vector
- **T** : translate current object by **+x & +y** defined in translation vector

### 2. Rotation

- **Q** : rotate current object **clockwise** by  $\alpha$  defined in rotation angle
- **E** : rotate current object **counterclockwise** by  $\alpha$  defined in rotation angle

### 3. Scaling

- **X** : scale current object by **(1 \* scale)** defined in scaling factor
- **Z** : scale current object by **(1 / scale)** defined in scaling factor

### 4. Reflection

- **R** : reflect current object horizontally

### 5. Rasterizing

- **F** : Rasterize current object

### 6. Object Selection

- **.** : Switch to the next object
- **,** : Switch to the previous object

### 7. Line Drawing Algorithm

- **V** : Switch to the DDA Algorithm
- **B** : Switch to the Bresenham's Algorithm
- **L** : Switch to the Other Algorithm

### 8. Exit Program

- **Esc** : Exit the program

# Project Overview

---

## Required:

|  |  |
|--|--|
| at least three different polygons  | (Statue: <b>Complete</b> ) (Location: <b>Data File</b> )           |
| DDA line drawing Algorithm   | (Statue: <b>Complete</b> ) (Location: <b>[107-125]main.cpp</b> )   |
| Bresenham line drawing Algorithm   | (Statue: <b>Complete</b> ) (Location: <b>[52-104]main.cpp</b> )    |
| CONVEX Polygon Rasterizing Algorithm   | (Statue: <b>Complete</b> ) (Location: <b>[127-181]main.cpp</b> )   |
| 2D Translation Algorithm   | (Statue: <b>Complete</b> ) (Location: <b>[110-115]parser.cpp</b> ) |
| 2D Rotation Algorithm  | (Statue: <b>Complete</b> ) (Location: <b>[117-125]parser.cpp</b> ) |
| 2D Scaling Algorithm   | (Statue: <b>Complete</b> ) (Location: <b>[127-134]parser.cpp</b> ) |
| 2D Line Clipping Algorithm   | (Statue: <b>Complete</b> ) (Location: <b>[149-252]parser.cpp</b> ) |
| 2D Polygon Clipping Algorithm  | (Statue: <b>Complete</b> ) (Location: <b>[254-287]parser.cpp</b> ) |
| User Input Customization   | (Statue: <b>Complete</b> ) (Location: <b>Runtime Menu</b> )        |
| <ul style="list-style-type: none"><li>• <b>ID</b> of the polygon to be manipulated</li><li>• x- and y-extension of a <b>viewport</b> on the screen</li><li>• <b>scaling factor, translation vector, and rotation angle</b></li><li>• min, xmax, ymin, ymax values that define <b>clipping planes</b></li></ul> |  |
| Output written to the input data file  | (Statue: <b>Complete</b> )   |
| Makefile and Manual  | (Statue: <b>Complete</b> )   |

## Extra Credit:

|   |   |
|---|---|
| Separate setting of Window, Viewport, and Clipping Window size            | (Location: <b>Runtime Menu</b> )        |
| Separate setting of Window, Viewport, and background color and Line color | (Location: <b>Runtime Menu</b> )        |
| Allow user to add new shapes  | (Location: <b>Runtime Menu</b> )        |
| Polygon Reflection  | (Location: <b>[138-146]parser.cpp</b> ) |
| Intuitive Keyboard Interactions   | (Location: <b>[240-284]main.cpp</b> )   |
| Allow User to specify input files   | (Location: <b>[469-492]main.cpp</b> )   |
| Save Settings to setting file at exit                                     | (Location: <b>[386-400]parser.cpp</b> ) |