Hw1 Program Manual

Sahil Faruque, ID: 999640034, ssfaruque@ucdavis.edu

October 15, 2018

1 Introduction

This is a brief manual on how run the program for homework 1.

2 Building the program

In order to build the program, type **make** in the directory containing the Makefile. The output should be an executable called **out**. The program is compatible with both Linux and Mac OS.

3 How to use the program

3.1 Command line parameters

In order to run the program, you need to add 2 extra parameters to the command line which are the window's width and the window's height.

```
// syntax for running the program
./out windowWidth windowHeight

// Example run of the program.
// This runs the program with a window width of 1080 and height 720
./out 1080 720
```

3.2 Commands

Note: All commands in this program uses a single space in order to separate the arguments. In order to see a list of all of the possible commands that you can enter, type **help** into the command prompt. Output of the **help** command can be seen in the figure below.

The **color** parameter in some of the commands is a 32 bit integer in hexadecimal form where the first most significant byte is the red amount, the second byte is the green amount, the third byte is the blue amount, and the fourth byte is the alpha amount. The alpha amount should always be set to **FF**.

```
// color parameter value that contains max red amount
0xFF0000FF

// color parameter value that contains max green amount
0x00FF00FF

// color parameter value that contains max blue amount
0x0000FFFF
```

```
>> help
---------- LIST OF COMMANDS ----------
load: 'load fileName'
save: 'save fileName'
setLineDrawingMode: 'setLineDrawingMode id drawWithBresenham'
setClippingWindow: 'setClippingWindow xmin xmax ymin ymax'
rasterize: 'rasterize id color'
wireFrame: 'wireFrame id color'
dda: 'dda x1 y1 x2 y2 color'
bresenham: 'bresenham x1 y1 x2 y2 color'
polygon: 'polygon numVertices vertices... color drawWithBresenham rasterize'
translate: 'translate id xTrans yTrans'
scale: 'scale id xScale yScale'
rotate: 'rotate id angle'
remove: 'remove id'
displayContents: 'displayContents'
setViewport: 'setViewport x y width height'
exit: 'exit'
--------------------------------------
```

Figure 1: Output of **help** command

3.3 Loading and saving scenes

3.3.1 Loading scenes

In order to load a scene, simply enter the command: "load fileName" where **fileName** is the path to the file that you want to load

```
// syntax for running the command
load fileName

// Example of loading the sample file
load input/sample.txt
```

3.3.2 Saving scenes

In order to save a scene, simply enter the command: "save fileName" where **fileName** is the path to the file that you want to save

```
// syntax for running the command
save fileName

// Example of saving the scene to a file
save input/sample.txt
```

3.4 Displaying Scene contents

In order to see a description of all of the lines and polygons in the current scene, enter the command **displayContents**

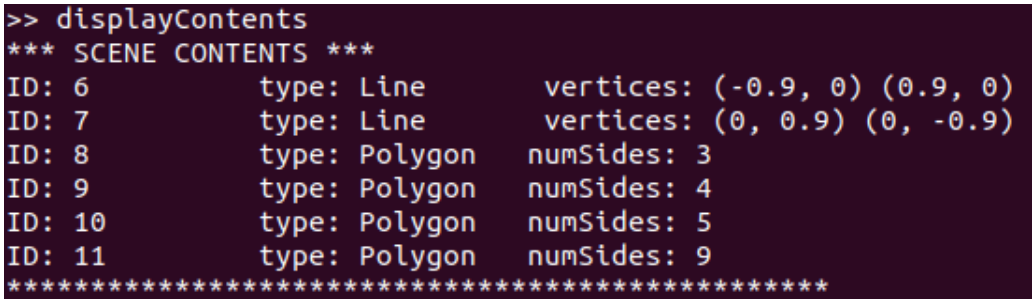

Figure 2: Example output of **displayContents**

3.5 Setting the line drawing mode

In order to set the drawing mode of a particular line or polygon to use DDA or Bresenham algorithms, enter the command: "setLineDrawingMode id drawWithBresenham" where **id** is the id of the designated line or polygon and **drawWithBresenham** is either a 1 or 0 value (if 1 it uses Bresenham and if 0 it uses DDA). If id is set to **-1**, then the change in line drawing algorithm will be applied to all lines and polygons in the scene.

```
// syntax for running the command
setLineDrawingMode id drawWithBresenham

// Example of setting the line drawing status of a particular line of polygon
// The line or polygon with id 7 will be drawn with DDA algorithm
setLineDrawingMode 7 0

// All lines and polygons in the scene will be drawn with bresenham algorithm
setLineDrawingMode -1 1
```

3.6 Setting the clipping window

By default, the clipping window is set such that xmin = -1, xmax = 1, ymin = -1 and ymax = 1. In order to set the clipping window to something else, use the command "setClippingWindow xmin xmax ymin ymax".

```
// syntax for running the command
setClippingWindow xmin xmax ymin ymax

// Example of setting the clipping window with xmin = -10, xmax = 10, ymin = -10, and ymax = 10
setClippingWindow -10 10 -10 10
```

3.7 Rasterization

Rasterizing a polygon can be done with the command: "rasterize id color" where **id** is the id of the polygon to rasterize and **color** is a 32 bit value in hexadecimal form.

```
// syntax for running the command
rasterize id color

// Example of rasterizing polygon with id 8 and red color
rasterize 8 0xFF0000FF
```

3.8 Wireframe

The wireframe command gets rid of the rasterization of a polygon and can be done through the command: "wireFrame id color" where **id** is the id of the polygon to rasterize and **color** is a 32 bit value in hexadecimal form.

```
// syntax for running the command
wireFrame id color

// Example of wire framing a polygon with id 5 and green color
wireFrame 5 0x00FF00FF
```

3.9 Drawling lines with DDA

The DDA command draws a line with the DDA algorithm and can be done with the command: "dda x1 y1 x2 y2 color" where **x1** and **y1** are the x and y values of the starting vertex, **x2** and **y2** are the x and y values of the ending vertex, and **color** is a 32 bit value in hexadecimal form.

```
// syntax for running the command
dda x1 y1 x2 y2 color

// Example of drawing a purple diagonal line
dda 0.0 0.0 0.5 0.5 0xFF00FF00
```

3.10 Drawing lines with Bresenham

The Bresenham command draws a line with the Bresenham algorithm and can be done with the command: "bresenham x1 y1 x2 y2 color" where **x1** and **y1** are the x and y values of the starting vertex, **x2** and **y2** are the x and y values of the ending vertex, and **color** is a 32 bit value in hexadecimal form.

```
// syntax for running the command
bresenham x1 y1 x2 y2 color

// Example of drawing a blue horizontal line
bresenham -0.5 0 0.5 0 0x0000FFFF
```

3.11 Drawing Polygons

The command to draw a polygon of n sides is the command: "polygon numVertices vertices... color drawWithBresenham rasterize" where **numVertices** is the number of vertices of the polygon you want to draw, vertices.. is of the form x1 y1 x2 y2 x3 y3..., **color** is a 32 bit value in hexadecimal form, **drawWithBresenham** is a 0 or 1 value where 0 draws with dda and 1 draws with bresenham, and rasterize is 0 or 1 where 1 is the polygon is rasterized and 0 is the polygon is not rasterized

```
// syntax for running the command
polygon numVertices vertices... color drawWithBresenham rasterize

// Example of drawing a triangle rasterized with the color red and lines draw
// with the bresenham algorithm
polygon 3 0.0 0.0 0.5 0.0 0.0 0.5 0xFF0000FF 1 1
```

3.12 Transformations

3.12.1 Translation

The command to translate a line or polygon is the command: "translate id xTrans yTrans" where **id** is the id of the line or polygon to translate, **xTrans** is the amount to translate in the x direction, and **yTrans** is the amount to translate in the y direction,

```
// syntax for running the command
translate id xTrans yTrans

// Example of translating a line or polygon with id of 0 by 0.4 in x direction
// and -0.3 in y direction
translate 0 0.4 -0.3
```

3.12.2 Scale

The command to scale a line or polygon is the command: "scale id xScale yScale" where id is the id of the line or polygon to scale, **xScale** is the amount to scale in the x direction, and **yScale** is the amount to scale in the y direction,

```
// syntax for running the command
scale id xScale yScale

// Example of scaling a line or polygon with id of 1 by 2 in x direction
// and 1.5 in y direction
scale 1 2.0 1.5
```

3.12.3 Rotation

The command to rotate a line or polygon about its centroid is the command: "rotate id angle" where **id** is the id of the line or polygon to rotate and **angle** is the amount to rotate by.

```
// syntax for running the command
rotate id angle

// Example of rotating a line or polygon with id of 2 by 45 degrees
rotate 2 45.0
```

3.13 Removing lines or polygons in the scene

In order to remove a specific line or polygon from the scene, use the command: 'remove id' where **id** is the id of the line or polygon that you want to remove.

```
// syntax for running the command
remove id

// Example of removing a line or polygon with id 8
remove 8
```

3.14 Setting the viewport

In order to set the viewport, use the command: 'setViewport x y width height' where **x** is the x value of the point to start in window coordinates, **y** is the y value of the point to start in window coordinates, **width** is the width of the viewport, and **height** is the height of the viewport.

```
// syntax for running the command
setViewport x y width height

// Example of setting the viewport to start at (100, 100) in window coordinates with a width of
    640 and height of 480
setViewport 100 100 640 480
```

3.15 Exiting the program

In order to exit the program, simply type the command: 'exit'

```
// syntax for running the command
exit
```

3.16 File Format

The files to load from and save to should be in the following format

```
3   // number of lines/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
    //definition of line
2
-0.9 0 // coordinates of 1st point
0.9 0  // coordinates of 2nd point
```

4 Notes

All lines in the scene are clipped with Cohen-Sutherland while all of the polygons are clipped with Sutherland-Hodgman. The clipping window is set to xmin = -1, xmax = 1, ymin = -1, and ymax = 1 by default. An example file called input/sample.txt can be loaded if desired. Any input file should be located inside the **input** directory for better organization, but not necessary.

5 Algorithm Implementation locations

5.1 DDA

File: graphics.cpp
Lines: 72-92

5.2 Bresenham

File: graphics.cpp
Lines: 95-115

5.3 Rasterization

File: graphics.cpp
Lines: 270-319

5.4 Cohen-Sutherland Clipping

File: graphics.cpp
Lines: 347-403

5.5 Sutherland-Hodgman Clipping

File: graphics.cpp
Lines: 453-464

5.6 Translation

File: math.cpp
Lines: 5-11

File: entity.cpp
Lines: 23-33

5.7 Scale

File: math.cpp
Lines: 14-20

File: entity.cpp
Lines: 36-49

5.8 Rotation

File: math.cpp
Lines: 23-36

File: entity.cpp
Lines: 52-64

6 Extra features

- Draw and rasterize concave polygons
- Dynamically load and save scene data from and to files
- Draw any number of polygons and lines dynamically