

**Use of Matrices in Computer Graphics**

ADVDISC S18 Group Project #1

1st Term AY 2015-2016

A documentation submitted to the faculty of the

College of Computer Studies

In partial fulfillment for the requirements

for the course ADVDISC

Submitted to:

Mr. Rigan Ap-apid

Submitted by:

Matthew Seaver Choy

Jan Allen Gapuz

Kingston Anthony Koa

Juan Lorenzo Simeon

October 26, 2015

**Contributions**

*Matthew Seaver Choy*

-Plotting Parabola

-Plotting Polygon

-Implement multiple figures

-Implement shearing operation

-Display of coordinates of the figures before and after transformation

*Jan Allen Gapuz*

-Plotting Point

-Plotting Line Segment

-Implement translate operation

*Kingston Anthony Koa*

-Plotting Vector

-Constructed all the GUI components of the program

-Implement reflect operation

-Implement rotate operation

*Juan Lorenzo Simeon*

-Plotting Hyperbola

-Plotting Ellipse

-Implement deletion of figures

-Implement the highlighting (change of color) of the current figure selected

-Plotting of the transformed figures

**Introduction**

In our subject of Advanced Discrete Structures(ADVDISC), we were tasked to come up with a system that will allow the user to input a 2D object which can be one or more points, line segments, vectors, polygons (rectangle, square, etc.), conics (circle, ellipse, parabola, hyperbola). After displaying the figure that the user wanted, the system must allow the user to choose what operation to do with the object which are the following: translate (change in position), rotate (change in orientation), shear (change in shape), uniform or non-uniform scale (change size), dilate, contract, reflect.

**Design**

Flow

First the system displayed the GUI with the required buttons and operation options with a Cartesian plane to display the figures desired by the user. The user can choose which figure to create, this will open a dialog box where the user may input the specifications of the figure. The figure will then be added to the figure list at the bottom of the screen and the figure will be drawn on the graph.

In the figure list, the user may select on which figure he wants to select for deletion or transformation. The selected figure is color *red* on the graph while the other unselected figures are color *blue*. The frame also has a table that displays the result or the coordinates of the figure selected by the user. The selected figure’s X and Y coordinates are displayed in the panel on the bottom right.

When the user wants to transform a figure, he may select the type of operation from the combo box (popup in MATLAB). When the user chooses a transformation, the program will open a dialog box where the values regarding the transformation may be specified. The transformed figure will be in color *red* while the previous figure before the transformation is in color *magenta*. The previous figures of unselected figures will be in color *cyan*. The current and previous figures’ X and Y coordinates are displayed in the panel on the bottom right.

Data Structures

To store the figures, we stored the X coordinates and Y coordinates of each figure. A point in the figure is a column in the matrix of the figure.

Example of a line segment with endpoints (0,1) and (0,25):

obj.xCoor = [0, 0]

obj.yCoor = [1, 25]

Operations

Scale

The matrix is multiplied by a vector [x, y] where the x coordinates of the figure are scaled by x and the y coordinates of the figure are scaled by y.

Shear

The matrix is multiplied with 2x2 matrix

1 B

0 1

Where B is tan(a) = aH/H, a being the shearing angle (clockwise)

Reflect

When reflecting over the x-axis, the y coordinates of the figure will be multiplied by -1, while the x coordinates are retained. When reflecting over the y-axis, the x coordinates of the figure will be multiplied by -1, while the y coordinates are retained.

Translate

When translating the figure by x, y, the x coordinates of the figure will be added with x and the y coordinates of the figure will be added with y.

Rotate

The matrix is multiplied with 2x2 matrix

cos(a) sin(a)

-sin(a) cos(a)

Where a is the angle of rotation (clockwise)

**Implementation**

We have decided to use MATLABR2011a for this project. We used functions that are related to the GUI, which are the following:

|  |  |
| --- | --- |
| **Function** | **Description** |
| pointPushButton\_Callback() | listener to “Point” button |
| lineSegmentPushButton\_Callback() | listener to “Line Segment” button |
| parabolaPushButton\_Callback() | listener to “Parabola” button |
| hyperbolaPushButton\_Callback() | listener to “Hyperbola” button |
| polygonPushButton\_Callback() | listener to “Polygon” button |
| elipsetPushButton\_Callback() | listener to “Elipse” button |
| vectorPushButton\_Callback() | listener to “Vector” button |
| operationsComboBox\_Callback() | ComboBox that contains options of operations |
| figuresListBox\_Callback() | ListBox that contains previously created figures. |
| deletePushButton\_Callback() | delete a selected figure from the ListBox |
| transformPushButton\_Callback() | transform a figure given the selected operation from the ComboBox |

The following are additional functions which uses the user’s input to manipulate and display the figures.

|  |  |
| --- | --- |
| **Function** | **Description** |
| drawAxis() | draws the x and y axis on the plane |
| plot() | Built-in function that plots specific figures given the coordinates |
| updateGraph() | refreshes the plane |
| translate(x, y) | translates a figure by x, y and produce the new values. |
| shear(a) | shears a figure given a as the angle of shearing, clockwise, and produce the new values. |
| rotate(a) | rotates a figure given a as the angle of rotation, clockwise, and produce the new values. |
| scale(x, y) | scales a figure by x, y and produce the new values. |
| reflectOverX() | reflects a figure over the x axis and produce the new values. |
| reflectOverY() | reflects a figure over the y axis and produce the new values. |

**Conclusion**

Given the specifications, we used matrices to implement the different operations that can be done to different figures. With any given figure, we would just manipulate their coordinate matrix, and we would have successfully implemented the operations required. We found the teachings of ADVDISC useful since we were able to learn how to multiply matrices and add matrices. By using the rotation matrix, shearing matrix, or scaling vectors, it was simple to perform the specific operations given our background in matrix row operations.