UTM

UNIVERSITI TEKNOLOGI MALAYSIA

GROUP PROJECT - "COVID-19 STAY AT HOME"

SECV1113-01 : MATHEMATICS FOR COMPUTER GRAPHICS

GROUP 2

LECTURER

DR. SURIATI BINTI SADIMON

PREPARED BY

AQILAH HANIM BINTI MOHD TAUFIK (B19EC0006)

ABDUL AZIM BIN ABDUL MALIK (A19EC0002)

PRASANT KARUNAMURTHI (A19EC0148)

CHONG HONG LEI (A19EC0035)

AIMI BINTI RUSDI (B19EC0001)

Bachelor of Computer Science (Computer Graphics and Multimedia Software)

TABLE OF CONTENTS

1. INTRODUCTION

A project was assigned to create a graphical card based on the theme that was given which was “COVID-19 Stay At Home” for a subject course entitled Mathematics for Computer Graphics SECV1113. It was instructed that an application of 2D transformation such as translating, rotating and scaling should be applied as well as the use of interpolation in the graphic library of C++ language. The graphic design card was programmed using a coding application DEV C++ with installation of graphic.h library package.

It is understood that the following title was assigned as a suitable representation of our current situation of a global pandemic due to a virus. Therefore, we designed specifically the shapes that represented the situation well as what we should do during a pandemic. In this project, we will implement a graphic library in order to generate basic shapes using graphic function to create a picture. Graphic functions that are used are as of Circle(), Line(), Rectangle() and more.

Not only that, trigonometric and linear interpolation functions were also applied in the coding as per instruction along with 2D transformation such as rotation, translation and scaling to create our desired graphic image card. In this project report, we will explain further on the implementation of the transformation, interpolation, graphic library functions as well as the task distribution for each member.

2. 2D TRANSFORMATION AND FUNCTIONS

It is discovered that transformation here in computer graphics is widely used to modify and reposition the graphics or shapes created in the program. To be more specific, a two dimensional transformation is helpful in changing the size, shape and position of the object. There are five types of transformation that can be found and applied in computer graphics which includes, translation, scaling, rotation, reflection and shear. The main transformation that is to be focused on and explained in detail will only be for translation, scaling and rotation talong with the calculation and graphical function used.

2.1 TRANSLATION

A translation process is functioned to move an object or point from one position to another on a 2D plane without changing the size of the object shape. It is a process of moving the initial coordinates labelled (X, Y) to a new coordinates labelled (X', Y') through a vector shift (t_x, t_y). When the straight line is translated, it will be drawn using endpoints.

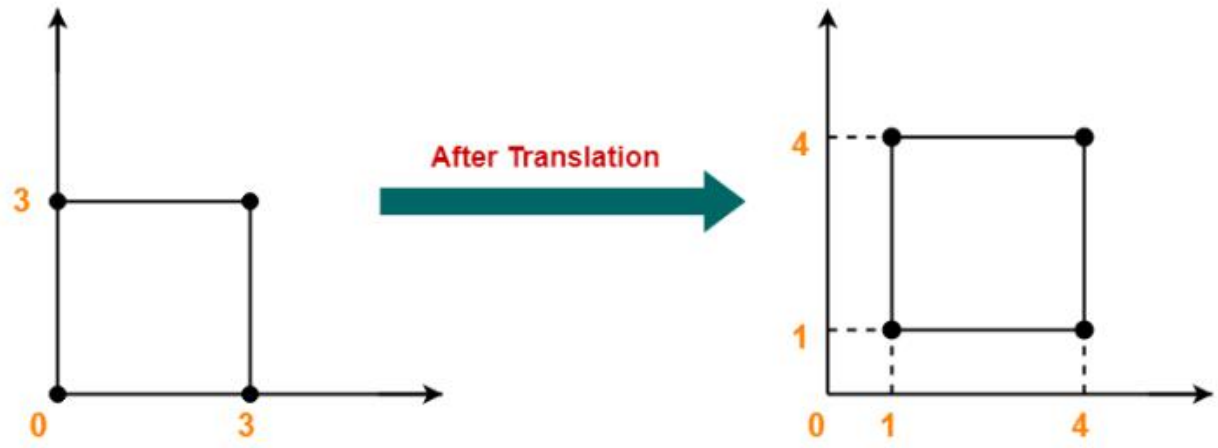


Figure 1. 2D plane of before and after translation

It is understood that each coordinate represents each other. As for an example:-

$X' = X + t_x$

$Y' = Y + t_y$

$$P = \frac{[X]}{[Y]} \qquad p' = \frac{[X']}{[Y']} \qquad T = \frac{[t_x]}{[t_y]}$$

We can write it simply with the equation of P'= P + T.

To implement this function based on our graphic project, translation was applied to the line function that was placed at the shape of a house to do the vertical lines as the walls. This was done to see how the lines can be created longer by moving the coordinates to form the designated shapes.

2.2 SCALING

A scaling transformation is when an alteration occurs to a shape or object which changes its size to either an expansion or compression of the object. The changes of size of the object is done by a multiplication of the scaling factors Sx and Sy. It is decided when the scaling factor S is less than 1, the size of the object will reduce while expansion occurs when the scaling factor S is more than 1. It can be represented mathematically as below :-

$$X' = X \,.\, S_x \quad \text{and} \quad Y' = Y \,.\, S_y$$

$$\begin{bmatrix} X_{new} \\ Y_{new} \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \times \begin{bmatrix} X_{old} \\ Y_{old} \end{bmatrix}$$

Scaling Matrix

The picture above represents the Matrix form of scaling for the calculation to get the size value.

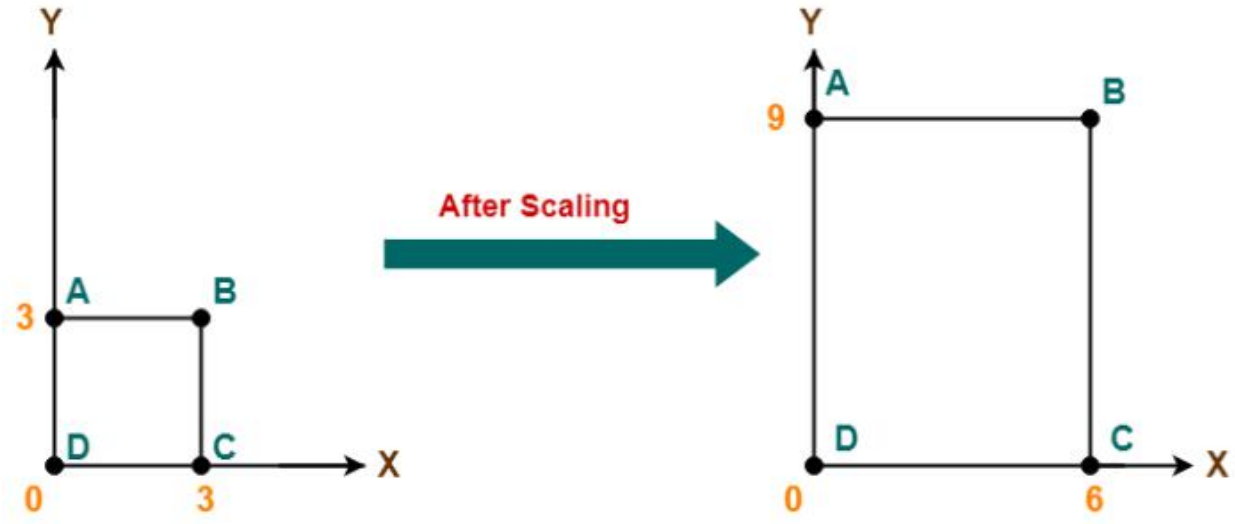


Figure 2. Before and after process of scaling transformation

Implementation of scaling for our coding was applied by inserting functions such as findNewCoordinate. This is to create or scale the new object away from the initial object. Overlapping may occur if we do not insert a new coordinate as it is similar to only enlarging a new image above the initial image. Example of findNewCoordinates coding are usually stated as below :-

```
void findNewCoordinate(int s[][2], int p[][1])
{
    int temp[2][1] = { 0 };

    for (int i = 0; i < 2; i++)
        for (int j = 0; j < 1; j++)
            for (int k = 0; k < 2; k++)
                temp[i][j] += (s[i][k] * p[k][j]);

    p[0][0] = temp[0][0];
    p[1][0] = temp[1][0];
}
```

2.3 ROTATION

Rotation is a method of changing an the angle of an object. Rotation can be done either clockwise or anti-clockwise direction. For a successful rotation of an object, the rotation angle and rotation point have to be determined. Rotation point is also named as a pivot point. The positive pivot point meaning (rotation angle) rotates an object in a counter clockwise direction (anti-clockwise) while the negative value of the pivot point (the rotation angle) rotates the object in the clockwise direction.

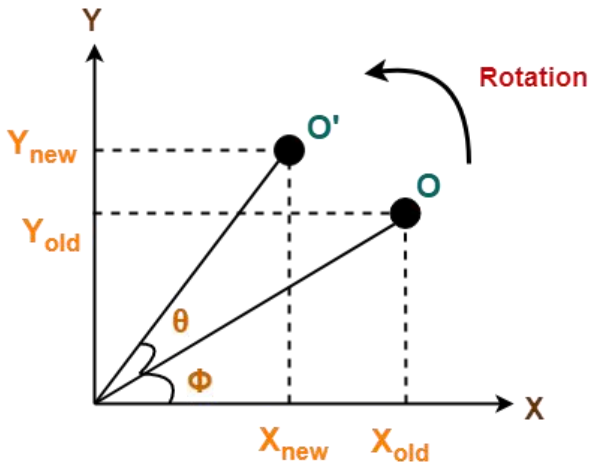


Figure 3. Before and after process of rotation transformation

The matrix for rotating a point in a 2D plane about its origin is defined as:

$$R_\beta = \begin{bmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{bmatrix}$$

A 2D vector in a Cartesian plane on the other hand, is rotated in the following way:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

3. INTERPOLATION

Interpolation may be a technique very widely used. Most commonly data is defined on a regular grid (values are written at the vertex of a 2D or 3D grid) or on a line (in the 1D case) so the software must calculate values at random on the grid. If the object is located on a vertex of the grid then we will actually use the value contained there.

An interpolant is a way to transform one number into another one. We add 2 for example to change 2 into 4. An interpolant has the actual function of modifying one number. An interpolant is a way to transform one number into another one.

As an example, we add 2 for example to change 2 into 4. An interpolant has the actual function of modifying one number. In a number of equivalent or unequal moves, in another. If we started with 2 and added 0.2 repeatedly, that would generate a 2.2, 2.4, 2.6, 2.8, 3.0, 3.2, 3.4, 3.6 sequence. Then such numbers should be used to translate, scale, rotate an object, the virtual camera moves or the position changes, virtual light source colour, or brightness.

Linear Interpolation:

$$n = n_1 + t(n_2 - n_1)$$

Can also be expressed as

$$n = n_1 + t(n_2 - n_1)$$
$$n = n_1 + tn_2 - tn_1$$
$$n = n_1(1-t) + n_2 t$$

Trigonometric Interpolation :

$$\sin^2(\beta)+\cos^2(\beta)=1$$

If β varies between 0 and $\pi/2$

$\cos^2(\beta)$ varies between 1 and 0
$\sin^2(\beta)$ varies between 0 and 1

which can be used to modify the two interpolated values n_1 and n_2 as follows

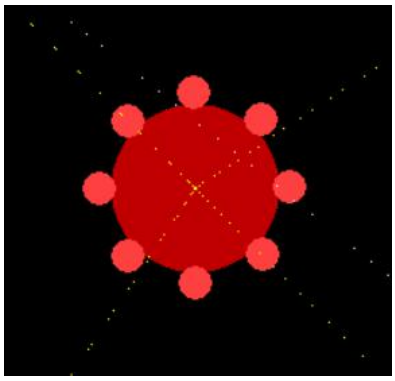
$$n = n_1 \cos^2(t) + n_2 \sin^2(t)$$

for $0 \le t \le \pi/2$

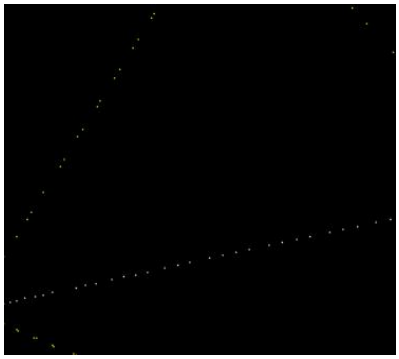
4. OUTPUT DESCRIPTION

The output of our project is as given in the table below.

Shape	Function

Virus
 
	Rotation : Small Viruses
	`void rotation(int _x, int _y, int x_pivot,int y_pivot,int n,int sec)`
	`{`
	`for(int angle=0;angle<360;angle+=45)`
	`{`
	`int x=(COS(angle)*(_x-x_pivot)) + (SIN(angle)*(y_pivot-_y)) + x_pivot;`
	`int y=(SIN(angle)*(_x-x_pivot)) + (COS(angle)*(_y-y_pivot)) + y_pivot;`
	`setcolor(LIGHTRED);`
	`circle(x,y,n);`
	`setfillstyle(SOLID_FILL,LIGHTRED);`
	`floodfill(x,y,LIGHTRED);`
	`delay(sec);`
	`}`
	`}`
	Scaling : Big Virus
	`void scaling(int x,int y,float sx,float sy,int r)`
	`{`
	`int s[2][2] = {sx,0,0,sy};`
	`int p[2][1];`
	`p[0][0] = sx*x;`
	`p[1][0] = sy*y;`
	`r = r*sqrt(pow(sx,2)+pow(sy,2));`
	`circle(p[0][0],p[1][0],r);`
	`setfillstyle(SOLID_FILL,RED);`
	`floodfill(p[0][0],p[1][0],RED);`
	`rotation((p[0][0]+r-30),(p[1][0]+r-30),p[0][0],p[1][0],20,200);`
	`}`

Dotted Lines (Rays)



Trigonometric Interpolation

```
void trigointerpolation(float x1[],float y1[],float x2[],float
y2[],int n, int c)
{
        float x,y;
        float no_post=500;
        float m=no_post-1;
        float step=90/m;
        float t=0;

        do{
        for(int i=0;i<n;i++)
        {
                x = (x1[i]*pow(cos(t),2)) + (x2[i]*pow(sin(t),2));
                y = (y1[i]*pow(cos(t),2)) + (y2[i]*pow(sin(t),2));
                t+=step;
                putpixel(x,y,c);
        }
        }while(t<90);
}
```

Text

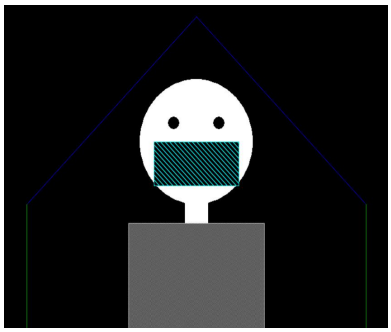
Message

```
setcolor(MAGENTA);
        settextstyle(10,HORIZ_DIR,6);
        outtextxy(500,50, "COVID-19 KILLS");

        settextstyle(10,HORIZ_DIR,3);
        outtextxy(500,100, "SOCIAL DISTANCE YOURSELF");

        for(int i =0;i<10;i++){
                delay(500);
                setcolor(i);
                settextstyle(10,HORIZ_DIR,2);
                outtextxy(505,150, "#STAYATHOME");
                continue;
        }
```

House



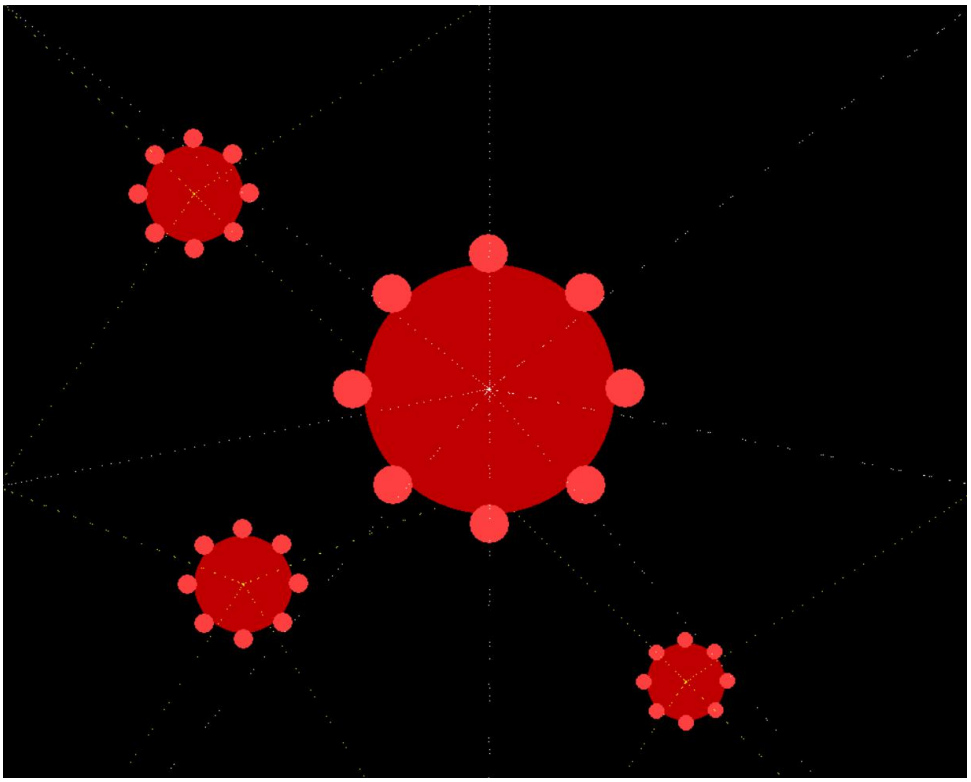
Reflection : Roof

```
void reflection(int xp, int yp, int x, int y)
{
        int diff = xp - x;
        x = (2*diff) + x;
        line(xp, yp, x, y);
}
```

Translation: Walls

```
void translation(int x1, int y1, int x2, int y2, int tx, int ty)
{
        x1 += tx;
        x2 += tx;
        y1 += ty;
        y2 += ty;
        line(x1,y1,x2,y2);
}
```

Final Output :




COVID-19 KILLS
SOCIAL DISTANCE YOURSELF
#STAYATHOME

5. TASK DISTRIBUTION AMONG MEMBERS

Name	Matrics	Task
Aqilah Hanim Binti Mohd Taufik	(B19EC0006)	Explanation on Rotation Explanation on Interpolation Output Description Task Distribution
Abdul Azim Bin Abdul Malik	(A19EC0002)	Coding on Translation Coding on Rotation Main Coding
Prasant Karunamurthi	(A19EC0148)	Output Description Check Coding Finalize Final Report
Chong Hong Lei	(A19EC0035)	Coding on Interpolation Coding on Scaling Main Coding
Aimi Binti Rusdi	(B19EC0001)	Introduction to Transformation Explanation on Translation Explanation on Scaling