ROTATION, SCALING, TRANSLATION OF MATRICES

GROUP 12

LAU LI XUAN A20EC3002

ABDULLAH NAFIS A20EC4060

LIM ZHEN DONG A20EC0236

INTRO

We have developed a tool that allows user to input the coordinates of a triangle's vertices and select a transformation to observe how the shape changes under the selected transformation.

BACKGROUND

We believe that this tool could help someone to learn the transformation of matrices interestingly so he or she can easily understand the concept.

TOPICS RELATED

- Coordinate System

  - This will enable us to let user to input x, y, z coordinate for us to link up and form a triangle shape

TOPICS RELATED

- Matrices

  - This topic gives us the ability to perform transformation of the triangle shape that created by the user.

  - Transformation that can be done: translation, rotation and scaling

TRANSLATION

- Through key in the moving distance (vector, d) user will be able to see the before and after moving of the triangle.

$$d = d_x = d_y$$

$$\begin{bmatrix} x'_i \\ y'_i \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \end{bmatrix}$$

```
void moveTriangle(double x[],double y[],int gd,int gm)
{
    int d;
    cout<<"\nEnter the moving distance- ";
    cin>>d;

    initgraph(&gd, &gm, (char*)"");

    line(x[0],y[0],x[1],y[1]);
    line(x[1],y[1],x[2],y[2]);
    line(x[2],y[2],x[0],y[0]);

    for(int i=0;i<3;i++)
    {
        x[i]=x[i]+d;
        y[i]=y[i]+d;
    }

    line(x[0],y[0],x[1],y[1]);
    line(x[1],y[1],x[2],y[2]);
    line(x[2],y[2],x[0],y[0]);
}
```

ROTATION

- User can enter a center point and a rotation angle, θ that he want to rotate the created triangle.

$$\begin{bmatrix} x'_i \\ y'_i \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -sin & \theta - p_x\,(cos\theta) + p_y\,(sin\theta) + p_x \\ sin & \theta cos & \theta - p_x\,(sin\theta) - p_y\,(cos\theta) + p_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

```
void rotateTriangle(double x[],double y[],int gd,int gm)
{
    double f,a[3],b[3];
    double x_pivot;
    double y_pivot;

    cout<<"\nEnter the rotation triangle- ";
    cin>>f;
    cout<<"\nEnter X pivot and Y pivot. (Give X and Y pivot = 0 to rotate about origin(0,0)):";
    cout<<"\n\nX pivot- ";
    cin>>x_pivot;
    cout<<"Y pivot- ";
    cin>>y_pivot;

    initgraph(&gd, &gm, (char*)"");

    line(x[0],y[0],x[1],y[1]);
    line(x[1],y[1],x[2],y[2]);
    line(x[2],y[2],x[0],y[0]);

    for(int i=0;i<3;i++)
    {
        double x_shifted = x[i] - x_pivot;
        double y_shifted = y[i] - y_pivot;

        a[i]=x_pivot+(x_shifted*cos(f*3.14159/180)-y_shifted*sin(f*3.14159/180));
        b[i]=y_pivot+(x_shifted*sin(f*3.14159/180)+y_shifted*cos(f*3.14159/180));
    }

    line(a[0],b[0],a[1],b[1]);
    line(a[1],b[1],a[2],b[2]);
    line(a[2],b[2],a[0],b[0]);
}
```

SCALING

- To enlarge or smaller the triangle, user can input the scale factor that he or she needed to scale up or down the triangle.

$$s = s_x = s_y$$

$$S = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$$

```
void scaleTriangle(double x[],double y[],int gd,int gm)
{
    double s;
    cout<<"\nEnter the scale factor- ";
    cin>>s;

    initgraph(&gd, &gm, (char*)"");

    line(x[0],y[0],x[1],y[1]);
    line(x[1],y[1],x[2],y[2]);
    line(x[2],y[2],x[0],y[0]);

    for(int i=0;i<3;i++)
    {
        x[i]=s*x[i];
        y[i]=s*y[i];
    }

    line(x[0],y[0],x[1],y[1]);
    line(x[1],y[1],x[2],y[2]);
    line(x[2],y[2],x[0],y[0]);
}
```

CONCLUSION

We have learned the coordinate system, trigonometry, linear and non-linear equations, differentiation and integration, vectors, matrices, and planes in Mathematics of Computer Graphics. We defined the positions of each vertex of a triangle as coordinates and put each coordinate into a matrix to calculate the result of a transformation. In conclusion, the knowledge we learned in Mathematics of Computer Graphics has helped us creating this program.