RAMAIAH UNIVERSITY OF APPLIED SCIENCES

ASSIGNMENT

Course Code	19CSC309A
Course Name	Computer Graphics
Programme	B.tech
Department	CSE
Faculty	FET

Name of the Student	Sudhanshu shekhar
Reg. No	17ETCS002213
Semester/Year	6th/3RD
Course Leader/s	Dr. Subarna Chatterjee

Declaration Sheet				
Student Name	Sudhanshu shekhar			
Reg. No	17ETCS002213			
Programme	B.tech	Semester/Year	6^{th} / 3^{rd}	
Course Code	19CSC309A			
Course Title	Computer Graphics			
Course Date	5/01/2020	to	19/04/2020	
Course Leader	Dr. Subarna Chatterjee			

Declaration

The assignment submitted herewith is a result of my own investigations and that I have conformed to the guidelines against plagiarism as laid out in the Student Handbook. All sections of the text and results, which have been obtained from other sources, are fully referenced. I understand that cheating and plagiarism constitute a breach of University regulations and will be dealt with accordingly.

Signature of the Student		Date	19/04/2020
Submission date stamp (by Examination & Assessment Section)			

Signature of the Course Leader and date	Signature of the Reviewer and date

1.1 Introduction to the problem

Approach for solving question:

Here I have taken triangle as a polygon for the upcoming RUAS graphics elevation, the triangle is transformed by some degree, I have taken 45 degree as transformation angle and its given that transformation is scout the organ in OpenGL.Thereafter we have used the glTranslatef function in our cpp program to translate the transformed triangle 4,8 unit in X and Y direction respectively as shown in line number 15 in source code 1 below also we have used the glscalef function and coordinate is taken as x = 5, y = 6 and z = 0 as given in question and implemented in line 16 in source code1.

Let's understand what are the transformations of a 2D object:

Rotation: 2D rotation aims for rotating any object by some angle θ about an arbitrary axis, involves several rotational and translation transformations. When we rotate an object about the origin (in 2-D), we in fact rotate it about the z-axis. Every point on the object rotates along a circular path, with the centre of rotation at the origin.

The function glRotatef(θ, x, y, z) in openGL performs rotation of the object where θ is the angle of rotation in degree and x, y, z are the axes on which the rotation should take place.

1. Rotation refers to rotating a point.
   Formula: X = xcosA - ysinA

   Y = xsinA + ycosA,

   A is the angle of rotation.

   The above formula will rotate the point around the origin.

   To rotate around a different point, the formula:

   X = cx + (x-cx)*cosA - (y-cy)*sinA,

   Y = cx + (x-cx)*sinA + (y-cy)*cosA,

   cx, cy is centre coordinates,

   A is the angle of rotation.

The OpenGL function is **glRotatef (A, x, y, z).**

Translation: Transformation is a process of modifying and re-positioning the existing graphics. For the 2D objects, translation is performed by changing the x and y directions of the object.

The function glTranslatef(x, y, z) in openGl performs translation of an object/polygon where x, y and z are the directions in float where the transformation should be performed.

1. **:** It refers to moving an object to a different position on screen.
   Formula: X = x + tx

   Y = y + ty

   Where tx and ty are translation co-ordinates.

   The OpenGL function is **glTranslatef(tx, ty, tz);**

Scaling: scaling is a process of modifying or altering the size of objects. Scaling may be used to increase or reduce the size of object. the coordinate points of the original object to change. Scaling factor determines whether the object size is to be increased or reduced.

The function glScalef(x, y, z) in openGL performs scaling on an object/polygon where x, y and z are the scaling factors.

Scaling refers to zooming in and out an object in different scales across axes.
Formula: X = x*sx

Y = y*sy, sx, sy being scaling factors.

The OpenGL function is **glScalef(float x, float y, float z)**

1.2 Implementation of transformation

Source Code 1:

graphics assignment (Global Scope)

```
1   #include<GL/glut.h>
2   float angle = 45;
3   void myinit(void)
4   {
5       glClearColor(1.0, 1.0, 1.0, 0.0);
6       glMatrixMode(GL_PROJECTION);
7       gluOrtho2D(0.0, 600.0, 0.0, 600.0);
8       glEnable(GL_COLOR_MATERIAL);
9   }
10
11  void polySegment(void)
12  {
13      glClear(GL_COLOR_BUFFER_BIT);
14      glColor3f(0.0f, 0.6f, 0.1f);
15      glTranslatef(4, 8, 0);
16      glScalef(5, 6, 0);
17      glRotatef(angle, 0.0, 0.0, 1.0);
18      glBegin(GL_POLYGON);
19      int p1[] = { 70,50 };
20      int p2[] = { 50,50 };
21      int p3[] = { 20,10 };
22
```

Fig 1.1: OpenGl program to perform the given transformations in a triangle

We have included the library gl/glut.h in line 1 and we have taken 45 degree as our angle and used the data type float, from line 4 to line 9 we have defined the body color by using the `glClearColor(1.0, 1.0, 1.0, 0.0);`command and x,y,z is clearly defined ,we projected our triangle by defining `glMatrixMode(GL_PROJECTION);`command, the shape here is defined by the command as `gluOrtho2D(0.0, 600.0, 0.0, 600.0);` from line 11 to line 21 in polySegment(void) we have defined the color of triangle as glColor3f(0.0f, 0.6f, 0.1f);

we have used our coordinate as x =4, y = 8 ,z = 0 to translate `glTranslatef(4, 8, 0);`command is used and to scale we have used the co-ordinate x = 5,y = 6,z = 0 and the command used is `glScalef(5, 6, 0);`to rotate the triangle we have used the glRotatef(angle, 0.0, 0.0, 1.0);command. Now to determine the vertex of the triangle we have define the varable p1,p2 and p3 and given the data type as int as `int p1[] = { 70,50 }; int p2[] = { 50,50 };` and `int p3[] = { 20,10 };`

Source code 2:

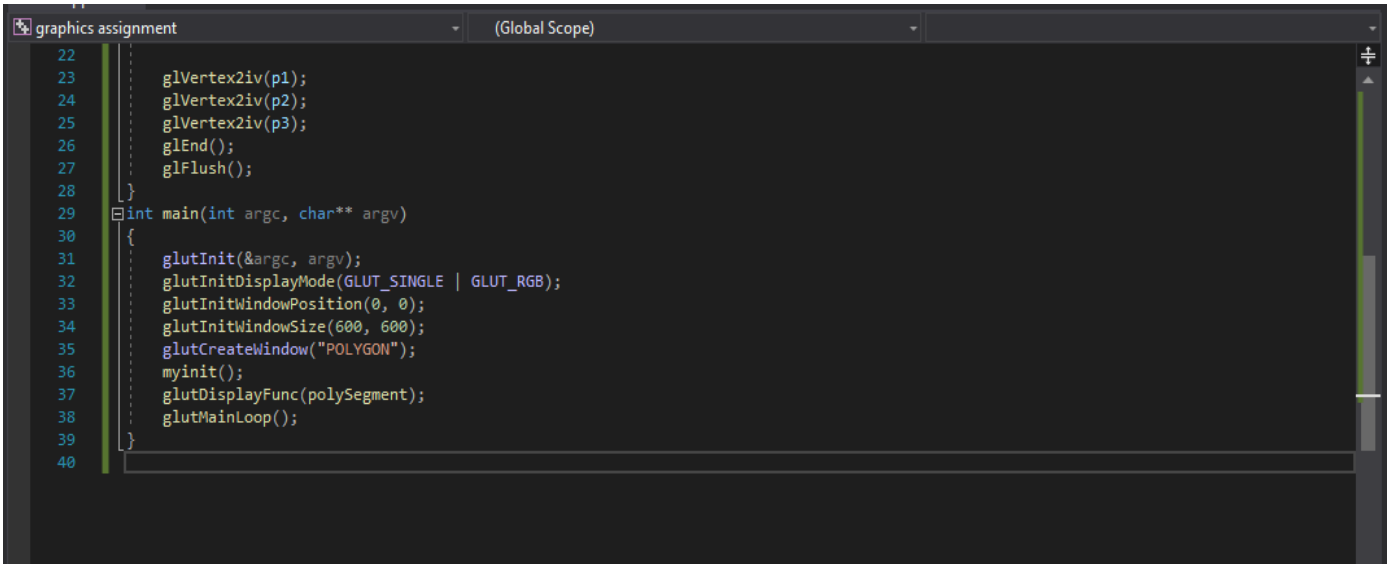

Fig 1.2: OpenGl program to perform the given transformations in a triangle

From line 22 to line 27 we are calling the vertex p1,p2,p3 whose coordinate we have already define in the above line of code. For calling the vertex p1,p2,p3 we use `glVertex2iv(p1);` `glVertex2iv(p2);` and `glVertex2iv(p3);` respectively, after this we have end our gl by specifying `glEnd(); glFlush();` command

now from line 30 to 40 we have defined the display mode by `glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);` command, we have set the windows position by `glutInitWindowPosition(0, 0);` command we have determine the windows size by `glutInitWindowSize(600, 600);` command creating the windows by `glutCreateWindow(" POLYGON");` commandand at last we need to display the function by using `glutDisplayFunc(polySegment);`command and then we end our loop in this program.

1.3 Results with screenshots and discussion

Output comes from above program:

FIG1.3 The above image is Before the rotation of triangle or this

Image can also be considered as the initial image.

FIG1.4 The above image is After rotation is done by 45 degree.

The rotated triangle is then performed the translate operation. Which is clearly shown in the above snippet. Translation means Re-positioning an object along a straight-line path from one coordinate location to another. In the above figure red colored triangle is the actual square (before transformation). And the yellow colored square (after transformation) is the transformed square.

FIG1.5 The above image is considered as the image after Translation and

Is also called the translation image.

FIG1.6

The above snippet is the final output after performing the three operations of transformation as mentioned in the question. That's is rotation, translating and scaling is performed on the square.

Here scaling is the property where the size of the image will be resized that could be seen clearly in the above snippet.