Program 9

Aim: Perform the following 3D Transformation operation:

a) Translation

b) Rotation

c) Scaling

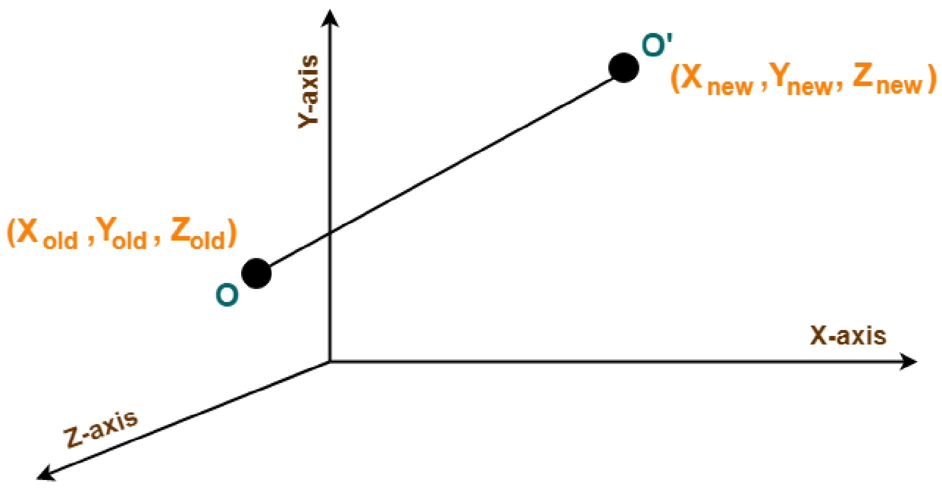
d) Sheering

Theory:

Translation

Consider a point object O has to be moved from one position to another in a 3D plane. Given a Translation vector (T_x, T_y, T_z)-

- T_x defines the distance the X_{old} coordinate has to be moved.
- T_y defines the distance the Y_{old} coordinate has to be moved.
- T_z defines the distance the Z_{old} coordinate has to be moved.



This translation is achieved by adding the translation coordinates to the old coordinates of the object as-

- $X_{new} = X_{old} + T_x$ (This denotes translation towards X axis)
- $Y_{new} = Y_{old} + T_y$ (This denotes translation towards Y axis)

- $Z_{new} = Z_{old} + T_z$ (This denotes translation towards Z axis)

In Matrix form, the above translation equations may be represented as-

$$\begin{bmatrix} X_{new} \\ Y_{new} \\ Z_{new} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{old} \\ Y_{old} \\ Z_{old} \\ 1 \end{bmatrix}$$

Rotation

Consider a point object O has to be rotated from one angle to another in a 3D plane. Let-

- Initial coordinates of the object O = (X_{old}, Y_{old}, Z_{old})
- Initial angle of the object O with respect to origin = Φ
- Rotation angle = θ
- New coordinates of the object O after rotation = (X_{new}, Y_{new}, Z_{new})

In 3 dimensions, there are 3 possible types of rotation-

- X-axis Rotation
- Y-axis Rotation
- Z-axis Rotation

Rotation along X-direction: This rotation is achieved by using the following rotation equations-

- $X_{new} = X_{old}$
- $Y_{new} = Y_{old} \times \cos\theta - Z_{old} \times \sin\theta$
- $Z_{new} = Y_{old} \times \sin\theta + Z_{old} \times \cos\theta$

In Matrix form, the above rotation equations may be represented as-

$$\begin{bmatrix} X_{new} \\ Y_{new} \\ Z_{new} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{old} \\ Y_{old} \\ Z_{old} \\ 1 \end{bmatrix}$$

Rotation along Y-axis: This rotation is achieved by using the following rotation equations-

- $X_{new} = Z_{old} \times \sin\theta + X_{old} \times \cos\theta$
- $Y_{new} = Y_{old}$
- $Z_{new} = Y_{old} \times \cos\theta - X_{old} \times \sin\theta$

In Matrix form, the above rotation equations may be represented as-

$$\begin{bmatrix} X_{new} \\ Y_{new} \\ Z_{new} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{old} \\ Y_{old} \\ Z_{old} \\ 1 \end{bmatrix}$$

Rotation along Z-axis: This rotation is achieved by using the following rotation equations-

- $X_{new} = X_{old} \times \cos\theta - Y_{old} \times \sin\theta$
- $Y_{new} = X_{old} \times \sin\theta + Y_{old} \times \cos\theta$
- $Z_{new} = Z_{old}$

In Matrix form, the above rotation equations may be represented as-

$$\begin{bmatrix} X_{new} \\ Y_{new} \\ Z_{new} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{old} \\ Y_{old} \\ Z_{old} \\ 1 \end{bmatrix}$$

Scaling

- Scaling may be used to increase or reduce the size of object.
- Scaling subjects the coordinate points of the original object to change.
- Scaling factor determines whether the object size is to be increased or reduced.
- If scaling factor > 1, then the object size is increased.
- If scaling factor < 1, then the object size is reduced.

Consider a point object O has to be scaled in a 3D plane. Let-

- Initial coordinates of the object O = (X_{old}, Y_{old},Z_{old})
- Scaling factor for X-axis = S_x
- Scaling factor for Y-axis = S_y
- Scaling factor for Z-axis = S_z
- New coordinates of the object O after scaling = (X_{new}, Y_{new}, Z_{new})

This scaling is achieved by using the following scaling equations-

- $X_{new} = X_{old} \times S_x$
- $Y_{new} = Y_{old} \times S_y$
- $Z_{new} = Z_{old} \times S_z$

In Matrix form, the above scaling equations may be represented as-

$$\begin{bmatrix} X_{new} \\ Y_{new} \\ Z_{new} \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{old} \\ Y_{old} \\ Z_{old} \\ 1 \end{bmatrix}$$

Shearing

In a three-dimensional plane, the object size can be changed along X direction, Y direction as well as Z direction. So, there are three versions of shearing-

1. Shearing in X direction

2. Shearing in Y direction
3. Shearing in Z direction

Consider a point object O has to be sheared in a 3D plane. Let-

- Initial coordinates of the object O = (X_{old}, Y_{old}, Z_{old})
- Shearing parameter towards X direction = Sh_x
- Shearing parameter towards Y direction = Sh_y
- Shearing parameter towards Z direction = Sh_z
- New coordinates of the object O after shearing = (X_{new}, Y_{new}, Z_{new})

Shearing in the X-direction: Shearing in X axis is achieved by using the following shearing equations-

- $X_{new} = X_{old}$
- $Y_{new} = Y_{old} + Sh_y \times X_{old}$
- $Z_{new} = Z_{old} + Sh_z \times X_{old}$

In Matrix form, the above shearing equations may be represented as-

$$\begin{bmatrix} X_{new} \\ Y_{new} \\ Z_{new} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ Sh_y & 1 & 0 & 0 \\ Sh_z & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{old} \\ Y_{old} \\ Z_{old} \\ 1 \end{bmatrix}$$

Shearing in the Y-direction: Shearing in Y axis is achieved by using the following shearing equations-

- $X_{new} = X_{old} + Sh_x \times Y_{old}$
- $Y_{new} = Y_{old}$
- $Z_{new} = Z_{old} + Sh_z \times Y_{old}$

In Matrix form, the above shearing equations may be represented as-

$$\begin{bmatrix} X_{new} \\ Y_{new} \\ Z_{new} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & Sh_x & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & Sh_z & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{old} \\ Y_{old} \\ Z_{old} \\ 1 \end{bmatrix}$$

Shearing in the Z-direction: Shearing in Z axis is achieved by using the following shearing equations-

- $X_{new} = X_{old} + Sh_x \times Z_{old}$
- $Y_{new} = Y_{old} + Sh_y \times Z_{old}$
- $Z_{new} = Z_{old}$

In Matrix form, the above shearing equations may be represented as-

$$\begin{bmatrix} X_{new} \\ Y_{new} \\ Z_{new} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & Sh_x & 0 \\ 0 & 1 & Sh_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{old} \\ Y_{old} \\ Z_{old} \\ 1 \end{bmatrix}$$

Code:

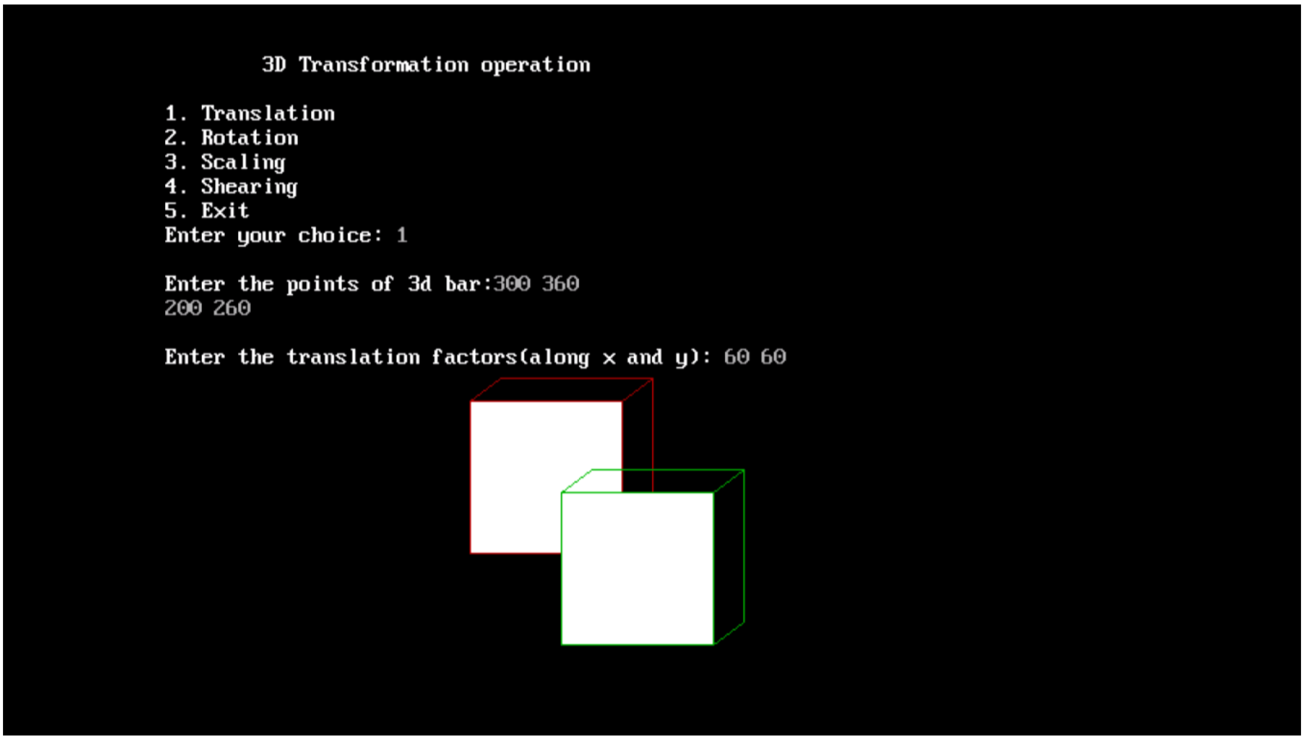
```
#include <iostream.h>
#include <graphics.h>
#include <process.h>
#include <stdio.h>
#include <conio.h>
#include <math.h>

int main(){
    int x1, x2, y1, y2, nx1, nx2, ny1, ny2, c, s, constant;
    int sx, sy, sz, depth = 20, xt, yt, zt, r, sh, shx, shy, shz;
    float t;
    int gd = DETECT, gm;
```

```
    initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");
    printf("\n\n\t3D Transformation operation");
    printf("\n\n1. Translation \n2. Rotation \n3. Scaling \n4. Shear
ing \n5. Exit");
    cout<<"\nEnter your choice: ";
    cin>>c;
    switch(c){
        case 1:
            printf("\nEnter the points of 3d bar:");
            scanf("%d %d %d %d", &x1, &y1, &x2, &y2);
            cout<<"\nEnter the translation factors(along x and y): "
;
            scanf("%d %d", &xt, &yt);
            nx1 = x1 + xt;
            ny1 = y1 + yt;
            nx2 = x2 + xt;
            ny2 = y2 + yt;
        break;
        case 2:
            cout<<"\nEnter the points of 3d bar:";
            scanf("%d %d %d %d", &x1, &y1, &x2, &y2);
            cout<<"\nEnter the rotating angle: ";
            scanf("%d", &sx);
            cout<<"\nRotating along z axis";
            nx1 = abs(x1 * cos(sx * 3.14 / 180) - y1 * sin(sx * 3.14
 / 180)) + 200;
            ny1 = abs(x1 * sin(sx * 3.14 / 180) + y1 * cos(sx * 3.14
 / 180)) + 200;
            nx2 = abs(x2 * cos(sx * 3.14 / 180) - y2 * sin(sx * 3.14
 / 180)) + 200;
            ny2 = abs(x2 * sin(sx * 3.14 / 180) + y2 * cos(sx * 3.14
 / 180)) + 200;
        break;
        case 3:
            cout<<"\nEnter the points of 3d bar:";
            scanf("%d %d %d %d", &x1, &y1, &x2, &y2);
            cout<<"\n Enter the scaling factor(for x,y,z): ";
            scanf("%d %d %d", &sx, &sy, &sz);
            nx1 = x1 * sx + 100;
            ny1 = y1 * sy + 100;
            nx2 = x2 * sx + 100;
            ny2 = y2 * sy + 100;
            depth = depth * sz;
        break;
        case 4:
```

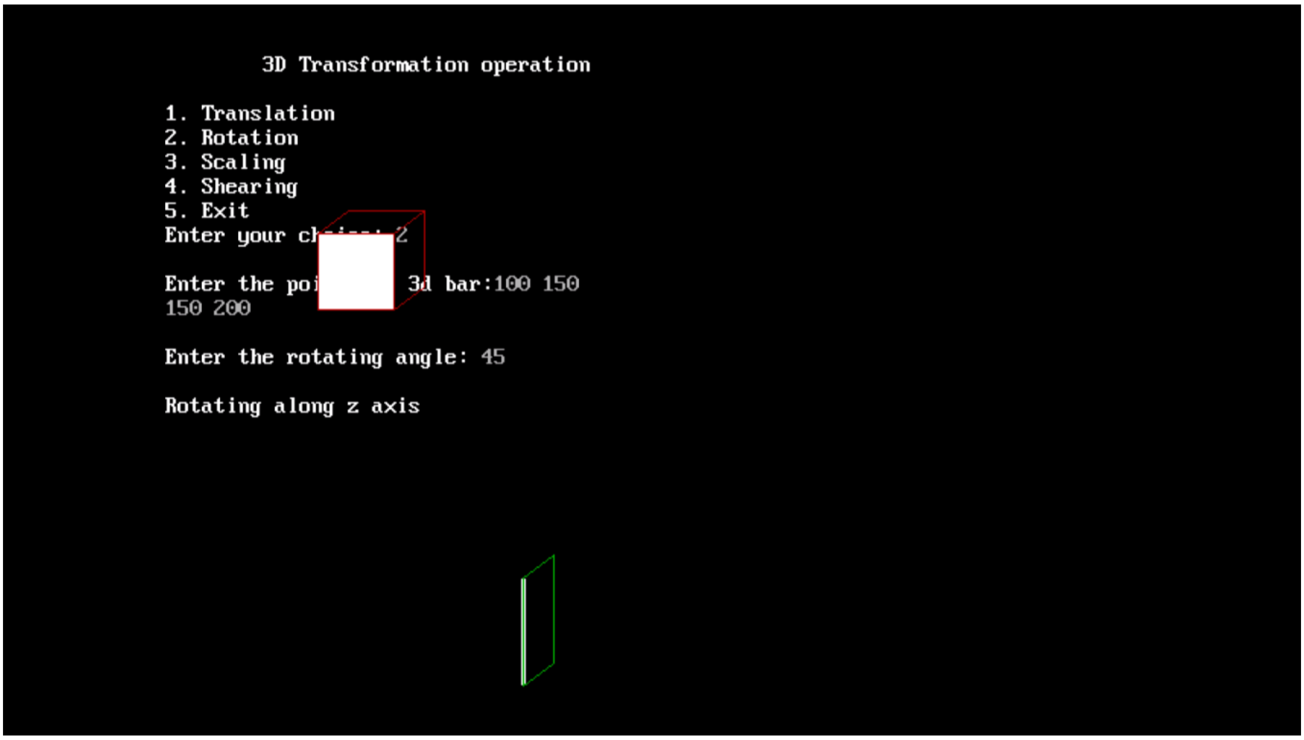
```
            cout<<"\nEnter the points of 3d bar:";
            scanf("%d %d %d %d", &x1, &y1, &x2, &y2);
            cout<<"\nShear along: \t(1)x-axis \t(2)y-axis \t(3)z-
axis";
            cout<<"\nEnter choice (1 or 2 or 3): ";
            scanf("%d", &s);
            cout<<"\nEnter shearing parameter: ";
            cin>>sh;
            switch(s){
                case 1:
                    nx1 = x1 + sh * y1;
                    ny1 = y1;
                    nx2 = x2 + sh * y2;
                    ny2 = y2;
                break;
                case 2:
                    nx1 = x1;
                    ny1 = y1 + sh * x1;
                    nx2 = x2;
                    ny2 = y2 + sh * x2;
                break;
                case 3:
                    nx1 = x1 + sh * y1;
                    ny1 = y1 + sh * x1;
                    nx2 = x2 + sh * y2;
                    ny2 = y2 + sh * x2;
                break;
            }
        break;
        case 5:
            cout<<"\nExiting...";
            exit(0);
        default:
            cout<<"\nEnter the correct choice!!";
        break;
    }
    setcolor(RED);
    bar3d(x1, y1, x2, y2, 20, 1);
    getch();
    setcolor(GREEN);
    bar3d(nx1, ny1, nx2, ny2, depth, 1);
    getch();
    closegraph();
    return 0;
}
```

Output:

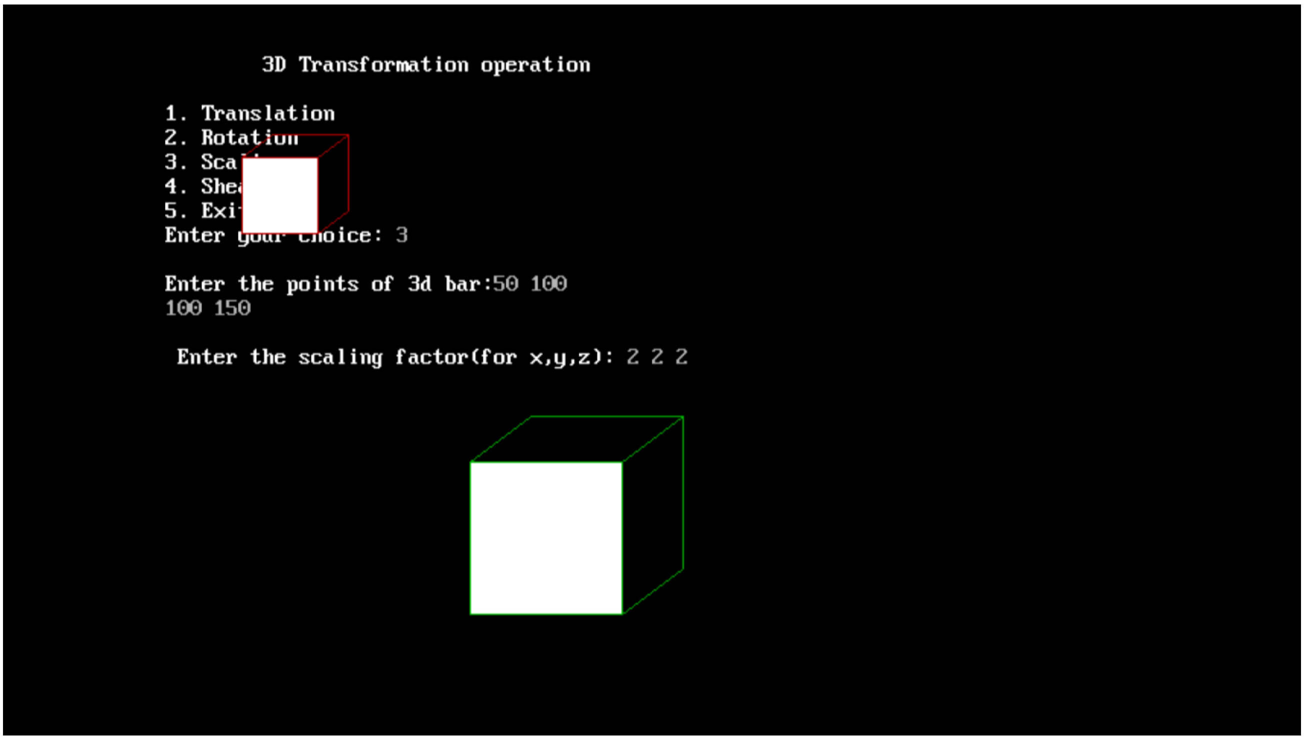
Translation



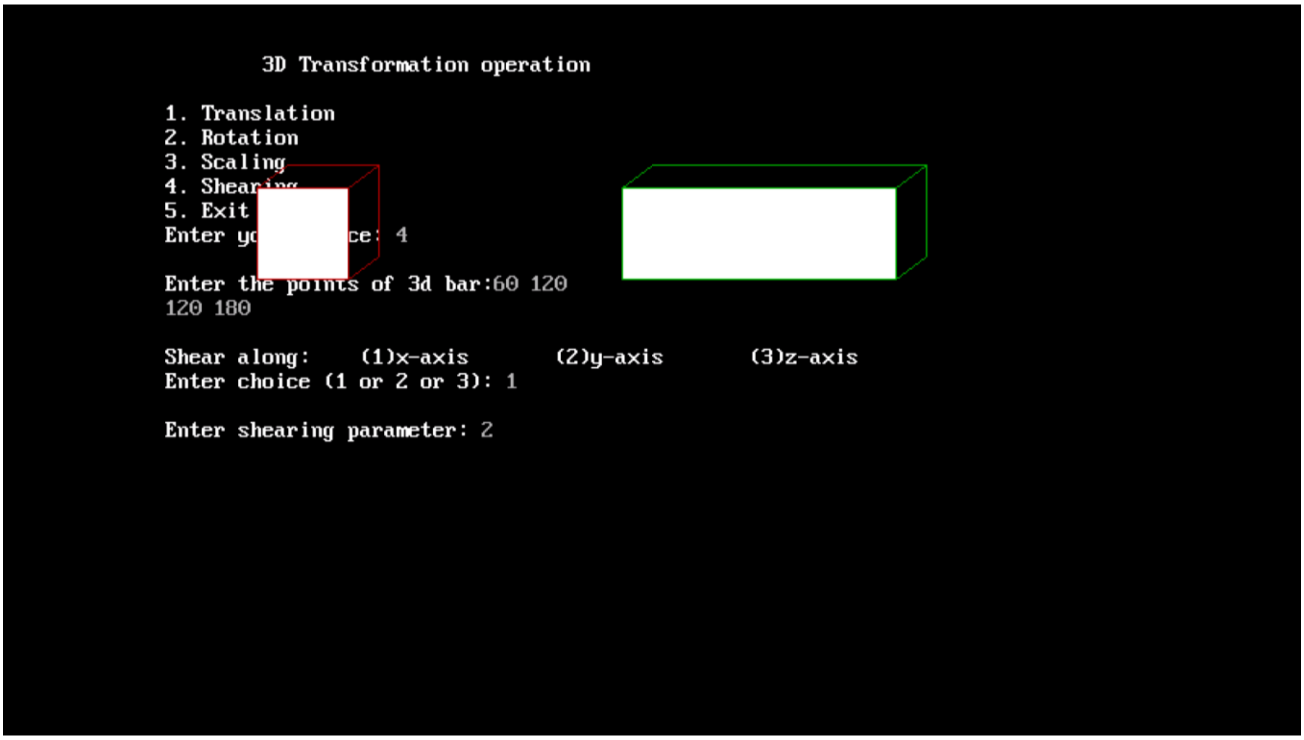
Rotation



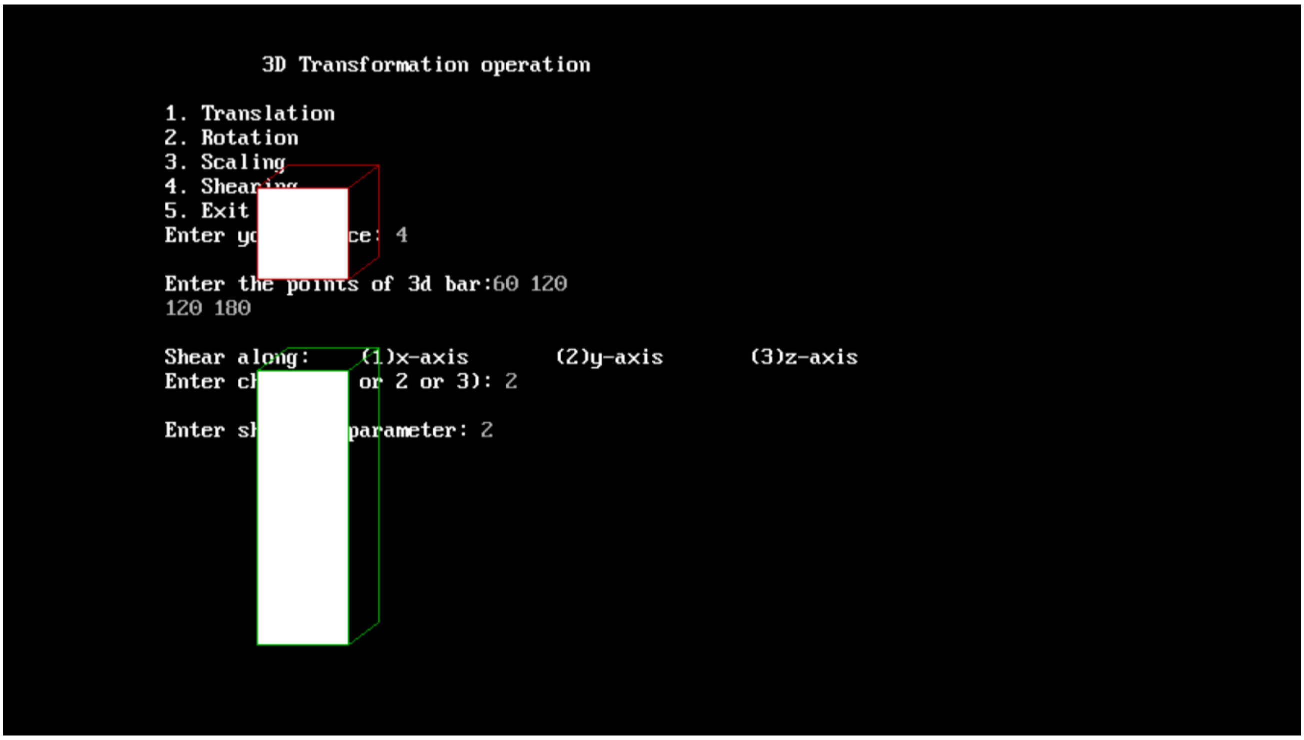
Scaling



Shearing (x-shear)



Shearing (y-shear)



Shearing (z-shear)