

Computer Graphics Assignment-6

Projections

Name-Ashish Goyal

Id-2016ucp1100

Batch-A (1, 2)

Ques 1: How does the image change if you make the plane of projection as the principal planes and the center of projection is at infinity in the direction perpendicular to the plane of projection?

Code:

```
from graphics import *
import math
clr1="red"
clr2="green"
f=open("file.txt","r")
c=int(f.readline())
vertex1=[]
vertex2=[]
for i in range(c):
    number_string=f.readline().split(' ')
    number_Array=[int(i) for i in number_string]
    point=[]
    point.append(number_Array[0])
    point.append(number_Array[1])
    point.append(number_Array[2])
```

```
vertex1.append(point)
point=[]
point.append(number_Array[3])
point.append(number_Array[4])
point.append(number_Array[5])
vertex2.append(point)
```

```
win_obj=GraphWin("Ortho Projection",900,900)
win_obj.setBackground("Yellow")
win_obj.setCoords(-300,-300,600,600)
x_axis=Line(Point(-300,0),Point(600,0))
y_axis=Line(Point(0,-300),Point(0,600))
z_axis=Line(Point(300,300),Point(-300,-300))
```

```
x_axis.setOutline("Black")
y_axis.setOutline("Black")
z_axis.setOutline("Black")
x_axis.setArrow('both')
y_axis.setArrow('both')
z_axis.setArrow('both')
x_axis.draw(win_obj)
y_axis.draw(win_obj)
z_axis.draw(win_obj)
```

```
info_x=Text(Point(580,-10),"+x axis")
info_x.draw(win_obj)
```

```

info_x=Text(Point(-280,-10),"-x axis")
info_x.draw(win_obj)
info_y=Text(Point(-10,580),"+y axis")
info_y.draw(win_obj)
info_y=Text(Point(-10,-280),"-y axis")
info_y.draw(win_obj)
info_ny=Text(Point(-280,-280),"+z axis")
info_ny.draw(win_obj)
info_ny=Text(Point(280,280),"-z axis")
info_ny.draw(win_obj)
origin=Text(Point(-10,-10),"origin")
origin.draw(win_obj)

```

Projection

```
def drawLine(x0,y0,z0,x1,y1,z1,color):
```

```

    ax, ay = x0-(z0*0.3), y0-(z0*0.3)
    bx, by = x1-(z1*0.3), y1-(z1*0.3)
    #print(ax,"",ay)
    #print(bx,"",by)
    line=Line(Point(ax,ay),Point(bx,by));
    line.setFill(color)
    line.setWidth(3)
    line.draw(win_obj)

```

```
def drawSolid(vertex1,vertex2,clr):  
    for i in range(c):  
        x0 = vertex1[i][0]  
        y0 = vertex1[i][1]  
        z0 = vertex1[i][2]  
        x1 = vertex2[i][0]  
        y1 = vertex2[i][1]  
        z1 = vertex2[i][2]  
        drawLine(x0,y0,z0,x1,y1,z1,clr)
```

```
drawSolid(vertex1,vertex2,clr1)
```

```
tx=int(input("Enter Tx:"))
```

```
ty=int(input("Enter Ty:"))
```

```
tz=int(input("Enter Tz:"))
```

```
vertex3=[]
```

```
vertex4=[]
```

```
for i in range(c):
```

```
    point=[]
```

```
    x0 = vertex1[i][0]+tx
```

```
    point.append(x0)
```

```
    y0 = vertex1[i][1]+ty
```

```
    point.append(y0)
```

```
    z0 = vertex1[i][2]+tz
```

```
    point.append(z0)
```

```
    vertex3.append(point)
```

```
    point=[]
```

```
x1 = vertex2[i][0]+tx
point.append(x1)
y1 = vertex2[i][1]+ty
point.append(y1)
z1 = vertex2[i][2]+tz
point.append(z1)
vertex4.append(point)
```

```
drawSolid(vertex3,vertex4,clr1)
```

```
vertex1=vertex3
```

```
vertex2=vertex4
```

```
print("Enter the principle plane 1 for xy or 2 for yz or 3 for xz:")
```

```
k1=int(input("Enter digit"))
```

```
vertex3=[]
```

```
vertex4=[]
```

```
if(k1==1):
```

```
    for i in range(c):
```

```
        vertex1[i][2]=0
```

```
        vertex2[i][2]=0
```

```
    drawSolid(vertex1,vertex2,clr2)
```

```
elif (k1==2):
```

```
    for i in range(c):
```

```
        vertex1[i][0]=0
```

```
        vertex2[i][0]=0
```

```
    drawSolid(vertex1,vertex2,clr2)
```

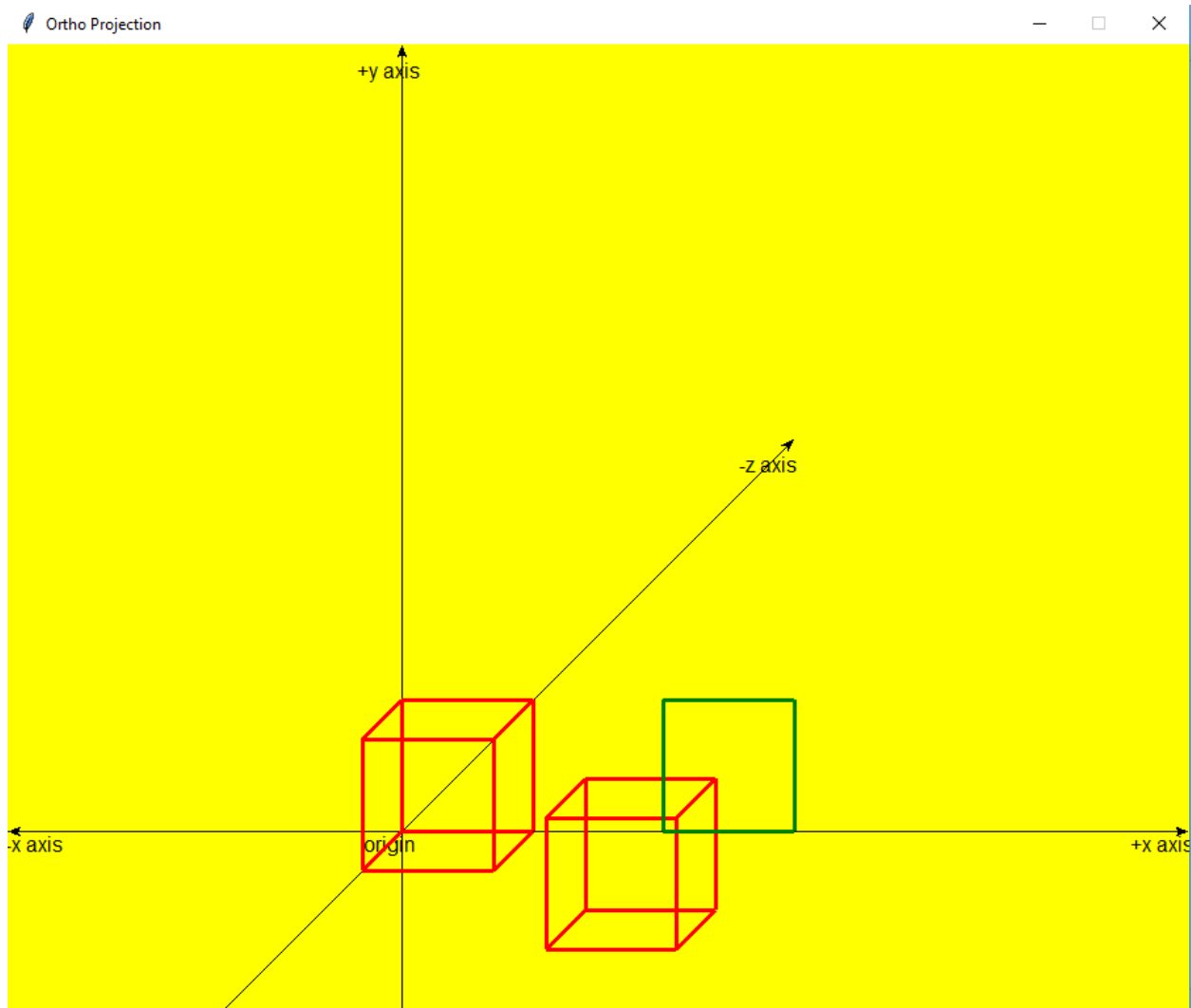
```
else:
```

```
for i in range(c):  
    vertex1[i][1]=0  
    vertex2[i][1]=0  
drawSolid(vertex1,vertex2,clr2)
```

```
win_obj.getMouse()  
win_obj.close()
```

Example1:

```
===== RESTART: C:\Users\Ashish\Desktop\py\proj1.py =  
Enter Tx:200  
Enter Ty:0  
Enter Tz:200  
Enter the principle plane 1 for xy or 2 for yz or 3 for xz:  
Enter digit1  
|
```



Ques 2: How does the image change if you define the plane of projection with a reference point and normal to the plane; and the center of projection is at infinity in the direction perpendicular to the plane of projection?

Code:

```
from graphics import *
import math
clr1="red"
clr2="green"
f=open("file.txt","r")
```

```

c=int(f.readline())
vertex1=[]
vertex2=[]
for i in range(c):
    number_string=f.readline().split(' ')
    number_Array=[int(i) for i in number_string]
    point=[]
    point.append(number_Array[0])
    point.append(number_Array[1])
    point.append(number_Array[2])
    vertex1.append(point)
    point=[]
    point.append(number_Array[3])
    point.append(number_Array[4])
    point.append(number_Array[5])
    vertex2.append(point)


win_obj=GraphWin("General parallel Projection",900,900)
win_obj.setBackground("Yellow")
win_obj.setCoords(-300,-300,600,600)
x_axis=Line(Point(-300,0),Point(600,0))
y_axis=Line(Point(0,-300),Point(0,600))
z_axis=Line(Point(300,300),Point(-300,-300))


x_axis.setOutline("Black")
y_axis.setOutline("Black")

```



```
z_axis.setOutline("Black")
```

```
x_axis.setArrow('both')
```

```
y_axis.setArrow('both')
```

```
z_axis.setArrow('both')
```

```
x_axis.draw(win_obj)
```

```
y_axis.draw(win_obj)
```

```
z_axis.draw(win_obj)
```

```
info_x=Text(Point(580,-10),"+x axis")
```

```
info_x.draw(win_obj)
```

```
info_x=Text(Point(-280,-10),"-x axis")
```

```
info_x.draw(win_obj)
```

```
info_y=Text(Point(-10,580),"+y axis")
```

```
info_y.draw(win_obj)
```

```
info_y=Text(Point(-10,-280),"-y axis")
```

```
info_y.draw(win_obj)
```

```
info_ny=Text(Point(-280,-280),"+z axis")
```

```
info_ny.draw(win_obj)
```

```
info_ny=Text(Point(280,280),"-z axis")
```

```
info_ny.draw(win_obj)
```

```
origin=Text(Point(-10,-10),"origin")
```

```
origin.draw(win_obj)
```

```
##### Projection #####
```

```
def drawLine(x0,y0,z0,x1,y1,z1,color):
```

```

ax, ay = x0-(z0*0.3), y0-(z0*0.3)
bx, by = x1-(z1*0.3), y1-(z1*0.3)
#print(ax," ",ay)
#print(bx," ",by)
line=Line(Point(ax,ay),Point(bx,by));
line.setFill(color)
line.setWidth(3)
line.draw(win_obj)

```

```

def drawSolid(vertex1,vertex2,clr):
    for i in range(c):
        x0 = vertex1[i][0]
        y0 = vertex1[i][1]
        z0 = vertex1[i][2]
        x1 = vertex2[i][0]
        y1 = vertex2[i][1]
        z1 = vertex2[i][2]
        drawLine(x0,y0,z0,x1,y1,z1,clr)

```

```

drawSolid(vertex1,vertex2,clr1)
tx=int(input("Enter Tx:"))
ty=int(input("Enter Ty:"))
tz=int(input("Enter Tz:"))
vertex3=[]
vertex4=[]

```

```

for i in range(c):
    point=[]
    x0 = vertex1[i][0]+tx
    point.append(x0)
    y0 = vertex1[i][1]+ty
    point.append(y0)
    z0 = vertex1[i][2]+tz
    point.append(z0)
    vertex3.append(point)
    point=[]
    x1 = vertex2[i][0]+tx
    point.append(x1)
    y1 = vertex2[i][1]+ty
    point.append(y1)
    z1 = vertex2[i][2]+tz
    point.append(z1)
    vertex4.append(point)

drawSolid(vertex3,vertex4,clr1)

vertex1=vertex3
vertex2=vertex4
vertex3=[]
vertex4=[]
print("enter reference point")

r1=int(input("x:"))

```

```

r2=int(input("y:"))
r3=int(input("z:"))
print("enter normal")
n1=int(input("x:"))
n2=int(input("y:"))
n3=int(input("z:"))
a1,b1,c1=n1,n2,n3
#print(a1,b1,c1,n1,n2,n3)
d0=r1*n1+r2*n2+r3*n3
d1=a1*n1+b1*n2+c1*n3
if (d1==0):
    print("Grazing Case (div by zero)")
    exit(0)
for i in range(c):
    point=[]
    x=vertex1[i][0]
    y=vertex1[i][1]
    z=vertex1[i][2]
    
$$x0 = (x*(d1-a1*n1)-y*a1*n2-z*a1*n3+a1*d0)/d1$$

    x0=int(x0)
    point.append(x0)
    
$$y0 = (y*(d1-b1*n2)-x*b1*n1-z*b1*n3+b1*d0)/d1$$

    y0=int(y0)
    point.append(y0)
    
$$z0 = (z*(d1-c1*n3) - x*c1*n1 - y*c1*n2 + c1*d0)/d1$$

    z0=int(z0)
    point.append(z0)

```

```
vertex3.append(point)
```

```
x=vertex2[i][0]
```

```
y=vertex2[i][1]
```

```
z=vertex2[i][2]
```

```
point=[]
```

```
x1 = (x*(d1-a1*n1)-y*a1*n2-z*a1*n3+a1*d0)/d1
```

```
x1=int(x1)
```

```
point.append(x1)
```

```
y1 = (y*(d1-b1*n2)-x*b1*n1-z*b1*n3+b1*d0)/d1
```

```
y1=int(y1)
```

```
point.append(y1)
```

```
z1 = (z*(d1-c1*n3) - x*c1*n1 -y*c1*n2 + c1*d0)/d1
```

```
z1=int(z1)
```

```
point.append(z1)
```

```
vertex4.append(point)
```

```
drawSolid(vertex3,vertex4,clr2)
```

```
win_obj.getMouse()
```

```
win_obj.close()
```

Example2:

```
===== RESTART: C:\Users\Ashish\Desktop\py\proj2.py
```

```
Enter Tx:200
```

```
Enter Ty:0
```

```
Enter Tz:200
```

```
enter reference point
```

```
x:300
```

```
y:0
```

```
z:100
```

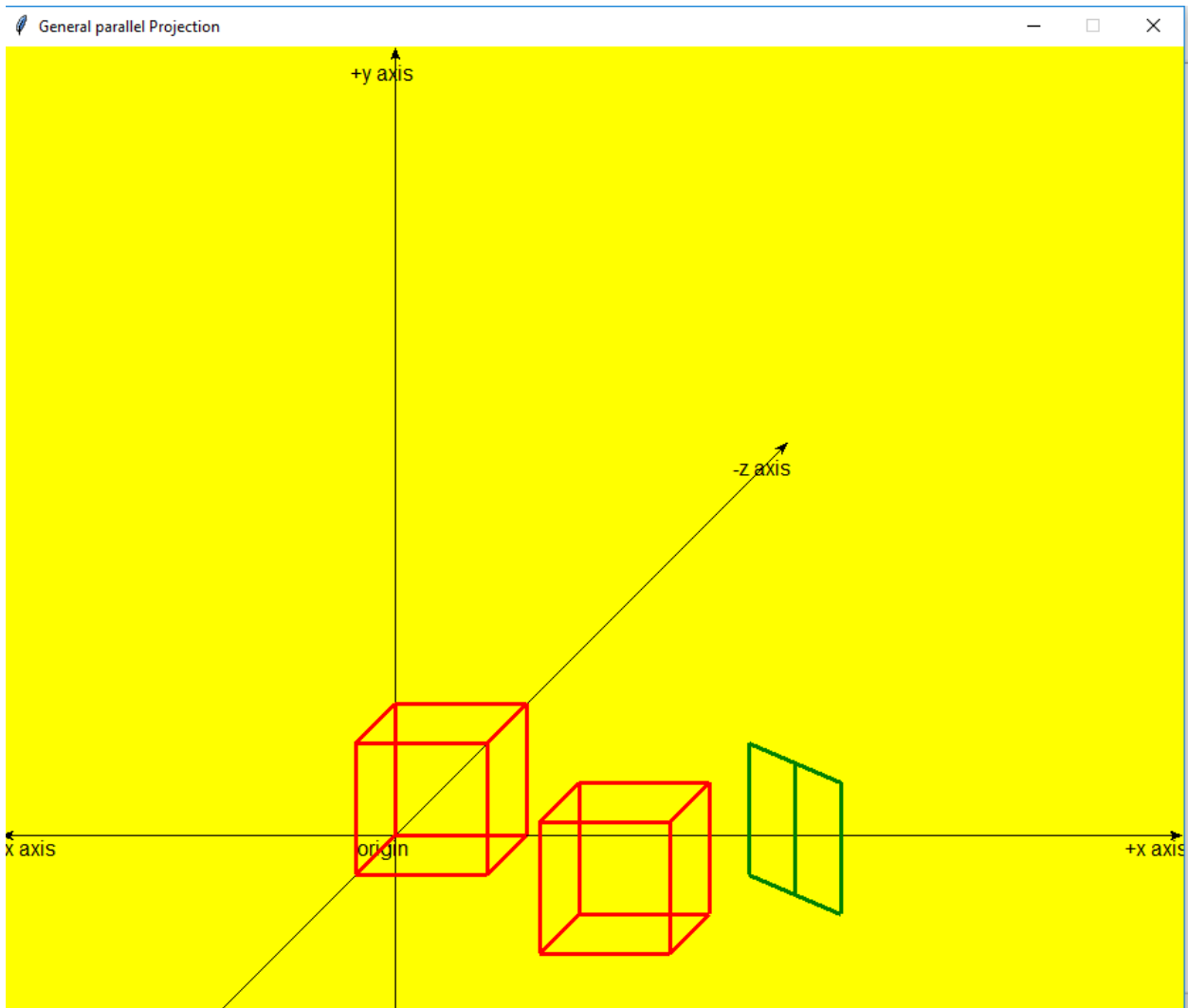
```
enter normal
```

```
x:1
```

```
y:0
```

```
z:-1
```

```
|
```



Ques 3: How does the image change if you define the plane of projection with a reference point and normal to the plane; and the center of projection is at a point

Code:

```
from graphics import *
import math
clr1="red"
clr2="green"
f=open("file.txt","r")
c=int(f.readline())
vertex1=[]
vertex2=[]
for i in range(c):
    number_string=f.readline().split(' ')
    number_Array=[int(i) for i in number_string]
    point=[]
    point.append(number_Array[0])
    point.append(number_Array[1])
    point.append(number_Array[2])
    vertex1.append(point)
    point=[]
    point.append(number_Array[3])
    point.append(number_Array[4])
    point.append(number_Array[5])
    vertex2.append(point)

win_obj=GraphWin("Perspective Projection",900,900)
win_obj.setBackground("Yellow")
win_obj.setCoords(-300,-300,600,600)
x_axis=Line(Point(-300,0),Point(600,0))
```

```
y_axis=Line(Point(0,-300),Point(0,600))
z_axis=Line(Point(300,300),Point(-300,-300))
```

```
x_axis.setOutline("Black")
y_axis.setOutline("Black")
z_axis.setOutline("Black")
x_axis.setArrow('both')
y_axis.setArrow('both')
z_axis.setArrow('both')
x_axis.draw(win_obj)
y_axis.draw(win_obj)
z_axis.draw(win_obj)
```

```
info_x=Text(Point(580,-10),"+x axis")
info_x.draw(win_obj)
info_x=Text(Point(-280,-10),"-x axis")
info_x.draw(win_obj)
info_y=Text(Point(-10,580),"+y axis")
info_y.draw(win_obj)
info_y=Text(Point(-10,-280),"-y axis")
info_y.draw(win_obj)
info_ny=Text(Point(-280,-280),"+z axis")
info_ny.draw(win_obj)
info_ny=Text(Point(280,280),"-z axis")
info_ny.draw(win_obj)
origin=Text(Point(-10,-10),"origin")
origin.draw(win_obj)
```

```
##### Projection #####
```



```
def drawLine(x0,y0,z0,x1,y1,z1,color):
```

```
    ax, ay = x0-(z0*0.3), y0-(z0*0.3)
```

```
    bx, by = x1-(z1*0.3), y1-(z1*0.3)
```

```
    #print(ax,"",ay)
```

```
    #print(bx,"",by)
```

```
    line=Line(Point(ax,ay),Point(bx,by));
```

```
    line.setFill(color)
```

```
    line.setWidth(3)
```

```
    line.draw(win_obj)
```

```
def drawSolid(vertex1,vertex2,clr):
```

```
    for i in range(c):
```

```
        x0 = vertex1[i][0]
```

```
        y0 = vertex1[i][1]
```

```
        z0 = vertex1[i][2]
```

```
        x1 = vertex2[i][0]
```

```
        y1 = vertex2[i][1]
```

```
        z1 = vertex2[i][2]
```

```
        drawLine(x0,y0,z0,x1,y1,z1,clr)
```

```
drawSolid(vertex1,vertex2,clr1)
```

```
tx=int(input("Enter Tx:"))
```

```
ty=int(input("Enter Ty:"))
```

```
tz=int(input("Enter Tz:"))
```

```
vertex3=[]
```

```
vertex4=[]
```

```
for i in range(c):
```

```
    point=[]
```

```
    x0 = vertex1[i][0]+tx
```

```
point.append(x0)
y0 = vertex1[i][1]+ty
point.append(y0)
z0 = vertex1[i][2]+tz
point.append(z0)
vertex3.append(point)
point=[]
x1 = vertex2[i][0]+tx
point.append(x1)
y1 = vertex2[i][1]+ty
point.append(y1)
z1 = vertex2[i][2]+tz
point.append(z1)
vertex4.append(point)
```

```
drawSolid(vertex3,vertex4,clr1)
```

```
vertex1=vertex3
vertex2=vertex4
vertex3=[]
vertex4=[]
print("enter reference point")
r1=int(input("x:"))
r2=int(input("y:"))
r3=int(input("z:"))
print("enter normal")
n1=int(input("x:"))
n2=int(input("y:"))
n3=int(input("z:"))
print("enter COP")
a1=int(input("x:"))
```

```
b1=int(input("y:"))
```

```
c1=int(input("z:"))
```

```
#print(a1,b1,c1,n1,n2,n3)
```

```
d0=r1*n1+r2*n2+r3*n3
```

```
d1=a1*n1+b1*n2+c1*n3
```

```
d=d0-d1
```

```
for i in range(c):
```

```
    point=[]
```

```
    x=vertex1[i][0]
```

```
    y=vertex1[i][1]
```

```
    z=vertex1[i][2]
```

```
    if((n1*x+n2*y+n3*z-d1)==0):
```

```
        print("Grazing Case (div byzero)")
```

```
        exit(0)
```

```
    x0 = (x*(a1*n1+d)+y*a1*n2+z*a1*n3-a1*d0)/(n1*x+n2*y+n3*z-d1)
```

```
    x0=int(x0)
```

```
    point.append(x0)
```

```
    y0 = (x*b1*n1+y*(b1*n2+d)+z*b1*n3-b1*d0)/(n1*x+n2*y+n3*z-d1)
```

```
    y0=int(y0)
```

```
    point.append(y0)
```

```
    z0 = (x*c1*n1 +y*c1*n2 +z*(c1*n3+d)- c1*d0)/(n1*x+n2*y+n3*z-d1)
```

```
    z0=int(z0)
```

```
    point.append(z0)
```

```
    vertex3.append(point)
```

```
x=vertex2[i][0]
```

```
y=vertex2[i][1]
```

```
z=vertex2[i][2]
```

```
point=[]
```

```
x1 = (x*(a1*n1+d)+y*a1*n2+z*a1*n3-a1*d0)/(n1*x+n2*y+n3*z-d1)
```

```
x1=int(x1)
```

```
point.append(x1)
```

```
y1 = (x*b1*n1+y*(b1*n2+d)+z*b1*n3-b1*d0)/(n1*x+n2*y+n3*z-d1)
```

```
y1=int(y1)
```

```
point.append(y1)
```

```
z1 = (x*c1*n1 +y*c1*n2 +z*(c1*n3+d)- c1*d0)/(n1*x+n2*y+n3*z-d1)
```

```
z1=int(z1)
```

```
point.append(z1)
```

```
vertex4.append(point)
```

```
drawSolid(vertex3,vertex4,clr2)
```

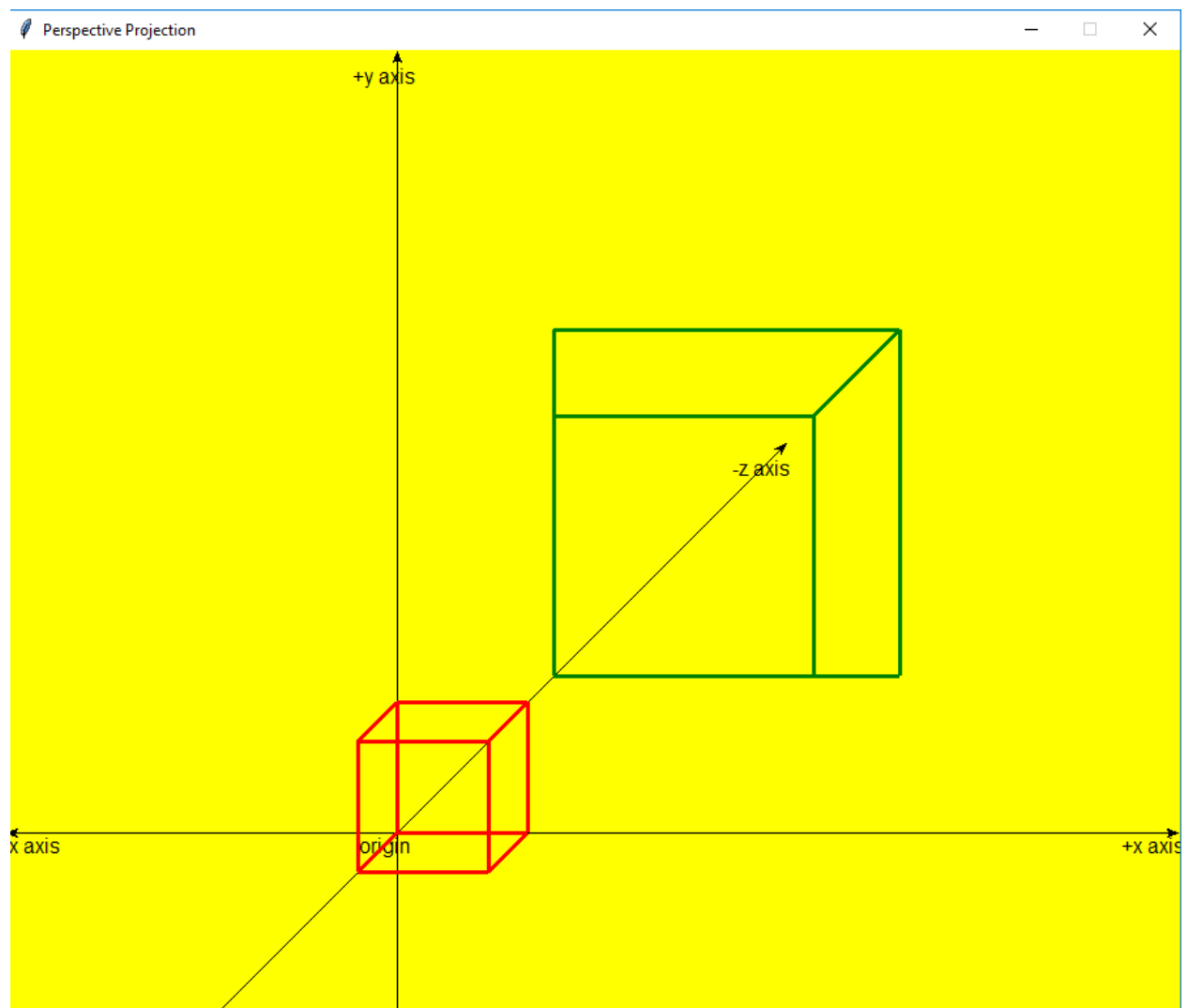
```
win_obj.getMouse()
```

```
win_obj.close()
```

i) On any of the axis

Example3:

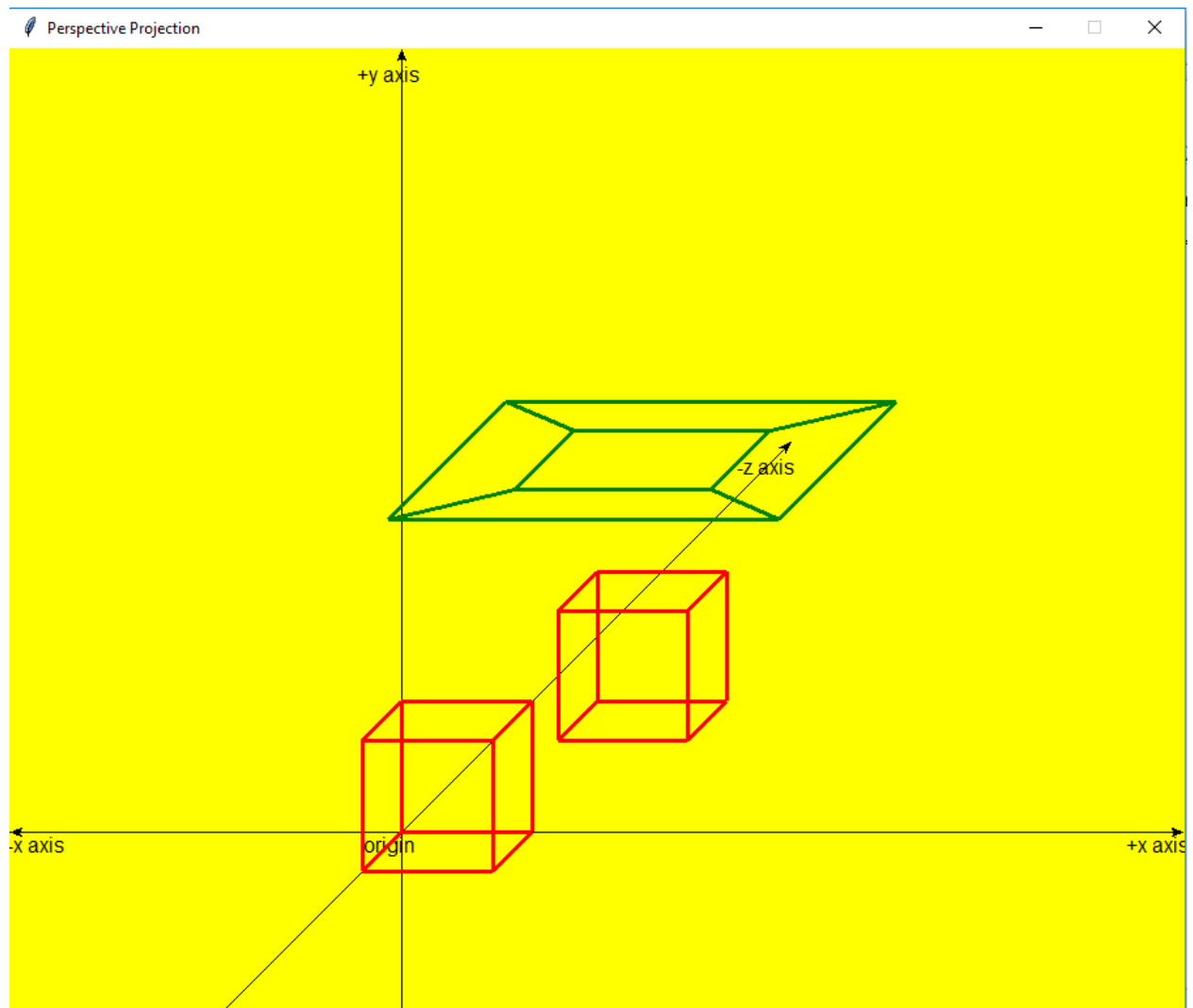
```
===== RESTART: C:\Users\Ashish\Desktop\py\proj3.py :
Enter Tx:0
Enter Ty:0
Enter Tz:0
enter reference point
x:0
y:0
z:-400
enter normal
x:0
y:0
z:-1
enter COP
x:0
y:0
z:400
|
```



ii) It is in a plane

Example4:

```
===== RESTART: C:\Users\Ashish\Desktop\py\proj3.py
Enter Tx:150
Enter Ty:100
Enter Tz:0
enter reference point
x:0
y:300
z:0
enter normal
x:0
y:1
z:0
enter COP
x:200
y:0
z:50
|
```



iii) It is in space.

Example5:

```
===== RESTART: C:\Users\Ashish\Desktop\py\proj3.py :
Enter Tx:100
Enter Ty:200
Enter Tz:0
enter reference point
x:300
y:0
z:0
enter normal
x:1
y:0
z:0
enter COP
x:-200
y:250
z:50
|
```

