```
import pygame
import random
from pygame.locals import *

from OpenGL.GL import *
from OpenGL.GLU import *

vertices = (
  # ( x,  y,  z)
    ( 1, -1, -1),  # A
    ( 1,  1, -1),  # B
    (-1,  1, -1),  # C
    (-1, -1, -1),  # D
    ( 1, -1,  1),  # E
    ( 1,  1,  1),  # F
    (-1, -1,  1),  # G
    (-1,  1,  1)   # H
)

edges = (
    (0, 1),
    (0, 3),
    (0, 4),
    (2, 1),
    (2, 3),
    (2, 7),
    (6, 3),
    (6, 4),
    (6, 7),
    (5, 1),
    (5, 4),
    (5, 7)
)

surfaces  = (
    (0, 1, 2, 3),
    (3, 2, 7, 6),
    (6, 7, 5, 4),
    (4, 5, 1, 0),
    (1, 5, 7, 2),
    (4, 0, 3, 6)
)

color = (
    (1, 0, 0),
    (0, 1, 0),
```

```
    (0, 0, 0),
    (0, 0, 1),
    (1, 1, 1),
    (0, 1, 1),
    (1, 0, 0),
    (0, 1, 0),
    (0, 0, 1),
    (0, 1, 0),
    (0, 0, 1),
    (0, 0, 0)
)

def Cube(vertices):
    glBegin(GL_QUADS)
    for surface in surfaces:
        x = 0
        glColor3fv((1, 0, 0))
        for vertex in surface:
            x += 1
            glColor3fv(color[x])
            glVertex3fv(vertices[vertex])
    glEnd()

    glBegin(GL_LINES)
    glColor3fv((0, 1, 0))
    for edge in edges:
        for vertex in edge:
            glVertex3fv(vertices[vertex])
    glEnd()


def set_vertices(vertices):
    x_change = random.randrange(-10, 10)
    y_change = random.randrange(-10, 10)
    z_change = random.randrange(-60, -40)

    new_vertices = []

    for vert in vertices:
        new_vert = []

        new_x = vert[0] + x_change
        new_y = vert[1] + y_change
        new_z = vert[2] + z_change

        new_vert.append(new_x)
        new_vert.append(new_y)
```

```
        new_vert.append(new_z)

        new_vertices.append(new_vert)

    return(new_vertices)


def main():
    pygame.init()
    screen = pygame.display.set_mode((800, 600), DOUBLEBUF | OPENGL)

    gluPerspective(45, (800 / 600), 0.1, 50)
    glTranslatef(random.randrange(-5, 5), random.randrange(-5, 5), -40)
    glRotatef(0, 0, 0, 0)

    move_x = 0
    move_y = 0

    cube_dict = {}

    for i in range(20):
        cube_dict[i] = set_vertices(vertices)

    while True:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()

            if event.type == pygame.KEYDOWN:
                if event.key == pygame.K_LEFT:
                    move_x = 0.5
                if event.key == pygame.K_RIGHT:
                    move_x = -0.5
                if event.key == pygame.K_UP:
                    move_y = -0.5
                if event.key == pygame.K_DOWN:
                    move_y = 0.5

            if event.type == pygame.KEYUP:
                if event.key == pygame.K_LEFT or event.key == pygame.K_RIGHT:
                    move_x = 0
                if event.key == pygame.K_UP or event.key == pygame.K_DOWN:
                    move_y = 0.5

            if event.type == pygame.MOUSEBUTTONDOWN:
                if event.button == 4:
                    glTranslatef(0, 0, -5)
```

```
                if event.button == 5:
                    glTranslatef(0, 0, 5)

        x = glGetDoublev(GL_MODELVIEW_MATRIX)
        coord = [[c for c in r] for r in x]

        camera_x = coord[3][0]
        camera_y = coord[3][1]
        camera_z = coord[3][2]

        glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
        glTranslatef(move_x, move_y, 0.5)

        for each_cube in cube_dict:
            Cube(cube_dict[each_cube])

        pygame.display.flip()
        pygame.time.wait(10)

main()
```