Diamond Catcher

```
import random
from OpenGL.GL import *
from OpenGL.GLUT import *
from OpenGL.GLU import *

w_height,w_width=900,600
game_pause=False
game_over=False
game_restart=False

diamond_storer=[]
is_diamond_falling=False

def point_generator(x,y):
    glPointSize(3)
    glBegin(GL_POINTS)
    glVertex2f(x,y)
    glEnd()

def zone_finder(a,b,x,y): #a,b is the initial and x,y are the final points
    dx=x-a
    dy=y-b
    zone=0

    if abs(dx)>=abs(dy):
        if dx>0 and dy>0:
            zone=0
        elif dx>0 and dy<0:
            zone=7
```

```
        elif dx<0 and dy>0:
            zone=3
        elif dx<0 and dy<0:
            zone=4
    else:
        if dx>0 and dy>0:
            zone=1
        elif dx<0 and dy>0:
            zone=2
        elif dx<0 and dy<0:
            zone=5
        else:
            zone=6
    return zone

def original_to_convert(o_z,x,y):
    if o_z==0:
        return x,y
    elif o_z==1:
        return y,x
    elif o_z==2:
        return y,-x
    elif o_z==3:
        return -x,y
    elif o_z==4:
        return -x,-y
    elif o_z==5:
        return -y,-x
    elif o_z==6:
        return -y,x
    elif o_z==7:
```

```
        return x,-y


def from_convert_to_original(o_z,x,y):
    if o_z==0:
        return x,y
    elif o_z==1:
        return y,x
    elif o_z==2:
        return -y,x
    elif o_z==3:
        return -x,y
    elif o_z==4:
        return -x,-y
    elif o_z==5:
        return -y,-x
    elif o_z==6:
        return y,-x
    elif o_z==7:
        return x,-y


def midpoint_line(zone,a,b,x,y):
    dx=x-a
    dy=y-b
    d=(2*dy)-dx


    east=2*dy
    n_east=2*(dy-dx)


    while a<x:
        temp_x,temp_y=from_convert_to_original(zone,a,b)
        point_generator(temp_x,temp_y)
```

```
        if d<0:
            d+=east
            a+=1
        else:
            d+=n_east
            a+=1
            b+=1

def eight_way(a,b,x,y):
    zone=zone_finder(a,b,x,y)

    temp_a,temp_b=original_to_convert(zone,a,b)
    temp_x,temp_y=original_to_convert(zone,x,y)
    midpoint_line(zone,temp_a,temp_b,temp_x,temp_y)

def diamond():
    global diamond_storer
    d_yax=810
    d_xax=random.randint(30,530)
    color=None

    while True:
        r=random.uniform(0,1)
        g=random.uniform(0,1)
        b=random.uniform(0,1)
        if r==0 and g==0 and b==0:
            pass
        else:
            color=(r,g,b)
            break
```

```
    if diamond_storer==[]:
        diamond_storer=[(d_xax,d_yax,color)]
    else:
        diamond_storer.append((d_xax,d_yax,color))

def dimond_draw(x,y,color):
    glColor3fv(color)
    eight_way(x,y,x+15,y+15)
    eight_way(x-15,y+15,x,y)
    eight_way(x,y+25,x+15,y+15)
    eight_way(x,y+25,x-15,y+15)

x0,x1,x2,x3 = 25,55,135,165
y0,y1 = 30,60
plate_width=x3-x1
plate_height=y1-y0
initial_color=(1.0,1.0,1.0)
plate_speed=10
points=0

def plate():
    global x0,x1,x2,x3,y0,y1
    glColor3fv(initial_color)
    eight_way(x1,y0,x2,y0)
    eight_way(x0,y1,x1,y0)
    eight_way(x2,y0,x3,y1)
    eight_way(x0,y1,x3,y1)
###########Buttons#######################################

def back_arrow():
    glColor3f(0.0,1.0,1.0)
```

```
    eight_way(25,855,105,855)
    eight_way(45,860,25,855)
    eight_way(45,850,25,855)

def cross():
    glColor3f(1.0,0.0,0.0)
    eight_way(500,845,580,885)
    eight_way(500,885,580,845)

def pause():
    glColor3f(1.0,1.0,0.0)
    eight_way(297,870,297,820)
    eight_way(297,870,310,845)
    eight_way(297,820,310,845)

def play():
    glColor3f(1.0,1.0,0.0)
    eight_way(297,870,297,820)
    eight_way(303,870,303,820)



##########################Drivers####################################

def specialKeyListener(key, x, y):
    global x0,x1,x2,x3,plate_speed,game_pause
    if game_pause==False:
        if key==GLUT_KEY_RIGHT:
            if all(x < 600 for x in (x0,x1, x2, x3)):
                x1 += plate_speed
                x2 += plate_speed
                x3 += plate_speed
```

```
            x0 += plate_speed
        if key==GLUT_KEY_LEFT:
            if all(x > 0 for x in (x0,x1, x2, x3)):
                x0 -= plate_speed
                x1 -= plate_speed
                x2 -= plate_speed
                x3 -= plate_speed
    glutPostRedisplay()

def mouseListener(button,state,x,y):
    global game_pause,game_over,game_restart,x0,x1,x2,x3,points
    win_lenght=w_height-y
    pause_state=0

    if button==(GLUT_LEFT_BUTTON) and state==GLUT_DOWN:

        if(298<=x<=311) and (821<=win_lenght<=871):
            pause_state+=1

        if(pause_state%2 !=0):
            game_pause = True
            print(f"Game Paused Current Points: {points}")

        if(pause_state%2 ==0):
            game_pause = False

        if (21<=x<=100) and (846<=win_lenght<=856):
            game_restart=True
            print("A new Game has started")

        if (500<=x<=575) and (840<=win_lenght<=880):
```

```
            print(f"Game has ended and your final points were: {points}")
            glutLeaveMainLoop()



def animate():
    global game_pause,game_over
    if game_pause==False and game_over==False:
        glutPostRedisplay()


def display():
    global is_diamond_falling
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
    glLoadIdentity()
    init()
    plate()
    back_arrow()
    cross()

    if is_diamond_falling:
        xd,yd,color=is_diamond_falling
        dimond_draw(xd,yd,color)

    if game_pause==True:
        pause()
    if game_pause==False:
        play()
    glutSwapBuffers()

gamespeed=3
```

```
def main_game_engine(self):
    global
points,initial_color,x0,x1,x2,x3,y0,y1,plate_speed,game_pause,game_over,game_restart,is_diamond_falling,games
peed,diamond_storer


    if game_restart==True:
        initial_color=(1,1,1)
        points=0
        x0,x1,x2,x3 = 25,55,135,165
        game_over=False
        game_restart=False
        diamond_storer=[]
        diamond_storer.clear()
        for i in range(100):
            diamond()
        glutDisplayFunc(display)


    elif game_over==False and game_pause==False:
        if is_diamond_falling==False and diamond_storer!=[]:
            is_diamond_falling=diamond_storer.pop()
        elif is_diamond_falling is None and diamond_storer != []:
            is_diamond_falling = diamond_storer.pop()


        if is_diamond_falling!=None:
            dx,dy,d_color=is_diamond_falling
            dy-=gamespeed
            is_diamond_falling=[dx,dy,d_color]


            if dy<=plate_height and abs(dx-((x0+x3)/2))< (plate_width/2):
                points+=1
                print(f"Current Game Score: {points}")
```

```
            is_diamond_falling=None

            if points >=5:
                gamespeed+=0.2
                diamond()
        elif dy<=0:
            game_over=True
            is_diamond_falling=None
            initial_color=(1,0,0)
            if game_over==True:
                game_pause=True
                print(f"The Game is over and your Final score is: {points}")
    glutPostRedisplay()
    glutTimerFunc(10, main_game_engine, 0)

for i in range(15):
    diamond()

glutInit()

glutInitDisplayMode(GLUT_RGBA | GLUT_DOUBLE | GLUT_DEPTH)
glutInitWindowSize(w_width,w_height)
glutInitWindowPosition(700,0)
glutCreateWindow(b"423 Lab-2 Diamond Game!")

def init():
    glViewport(0,0,w_width,w_height)
    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()
    glOrtho(0.0,w_width,0.0,w_height,0.0,1.0)
    glMatrixMode(GL_MODELVIEW)
```

```
    glLoadIdentity()
init()


glutDisplayFunc(display)
glutSpecialFunc(specialKeyListener)
glutMouseFunc(mouseListener)
glutIdleFunc(animate)
glutTimerFunc(10, main_game_engine, 0)
glutMainLoop()
```