Kathmandu University

Department of Computer Science and Engineering

Dhulikhel, Kavre

Mini Project Report: 2D Hilbert Curve

COMP 342

(For partial fulfillment of 3rd Year/ 2nd Semester in Computer Engineering)

Submitted to:

Mr. Dhiraj Shrestha

Department of Computer Science and Engineering

Submitted by:

Neha Malla

C.E.

Roll No. 27

Title: 2D Hilbert Curve

Introduction:

The Hilbert curve is a continuous fractal space-filling curve described by the German mathematician David Hilbert in 1891. Higher ordered Hilbert curve tends to fill every pixel or space in the defined area, that's why the curve is called a space filling curve.

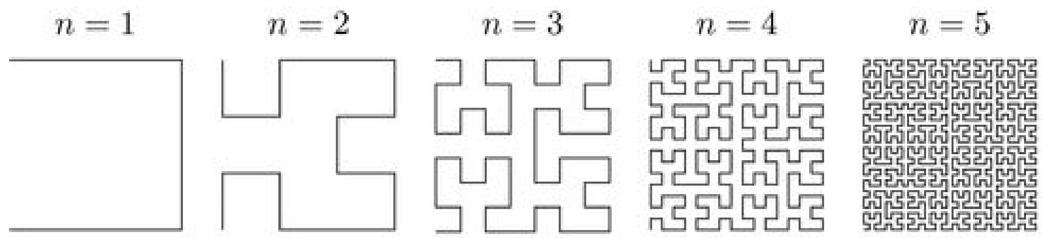

Figure: Hilbert curve in increasing order

The recursive function for obtaining the 2D Hilbert curve is as follows:

```
Hilbert(x0, y0, xi, xj, yi, yj, n):
    if n <= 0:
        X = x0 + (xi + yi)/2
        Y = y0 + (xj + yj)/2
        Points <-- (X, Y)
    else:
        Hilbert(x0,                 y0,                 yi/2, yj/2,  xi/2,  xj/2,   n - 1)
        Hilbert(x0 + xi/2,          y0 + xj/2,          xi/2, xj/2,  yi/2,  yj/2,   n - 1)
        Hilbert(x0 + xi/2 + yi/2,   y0 + xj/2 + yj/2,   xi/2, xj/2,  yi/2,  yj/2,   n - 1)
        Hilbert(x0 + xi/2 + yi,     y0 + xj/2 + yj,    -yi/2,-yj/2, -xi/2, -xj/2,   n - 1)
```

(Source: http://www.fundza.com/algorithmic/space_filling/hilbert/basics/)

Source code:

```
import pygame
import pygame.gfxdraw


# Initializing the game engine
pygame.init()

# Colors in RGB format
white = (255, 255, 255)
black = (0, 0, 0)
red = (255, 0, 0)
green = (0, 255, 0)
blue = (0, 0, 255)
cyan = (0, 255, 255)
magenta = (255, 0, 255)
yellow = (255, 255, 0)
orange = (253, 106, 2)

colours = [white, red, magenta, green, cyan, blue, orange, yellow]

# height and width of the screen
length = 1000
screen = pygame.display.set_mode([length, length], pygame.RESIZABLE)
pygame.display.set_caption("2D Hilbert Curve")

points = []


# Function to find the points of the curve
def Hilbert(x0, y0, xi, xj, yi, yj, n):
    """ x and y are the coordinates of the bottom left corner
        xi & xj are the i & j components of the unit x vector of the frame
        similarly yi and yj """

    if n <= 0:
        X = x0 + (xi + yi)/2
        Y = y0 + (xj + yj)/2
        points.append((int(X), int(Y)))
```

```
    else:
        Hilbert(x0,                  y0,                  yi/2, yj/2, xi/2,
xj/2, n - 1)
        Hilbert(x0 + xi/2,          y0 + xj/2,          xi/2, xj/2, yi/2,
yj/2, n - 1)
        Hilbert(x0 + xi/2 + yi/2, y0 + xj/2 + yj/2, xi/2, xj/2, yi/2,
yj/2, n - 1)
        Hilbert(x0  +  xi/2  +  yi,      y0  +  xj/2  +  yj,
-yi/2,-yj/2,-xi/2,-xj/2, n - 1)


# Function to connect the points of the curve
def draw(colour):
    for i in range(len(points) - 1):
        pygame.gfxdraw.line(screen,  points[i][0],  points[i][1],
points[i+1][0], points[i+1][1], colour)
    points.clear()


done = False

while not done:
    # If user clicks close
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            done = True

    # The screen background as black
    screen.fill(black)

    # Drawing all the Hilbert curve from order 1 to 9
    for i in range(1, 10):
        Hilbert(0.0, 0.0, length, 0.0, 0.0, length, i)
        draw(colours[i%len(colours)])

    # Drawing Hilbert curve of specific order
    # Hilbert(0.0, 0.0, length, 0.0, 0.0, length, 10)
    # draw(colours[-1])

    # Update the contents of the entire display
```

```
    pygame.display.flip()

pygame.quit()
```

Output:

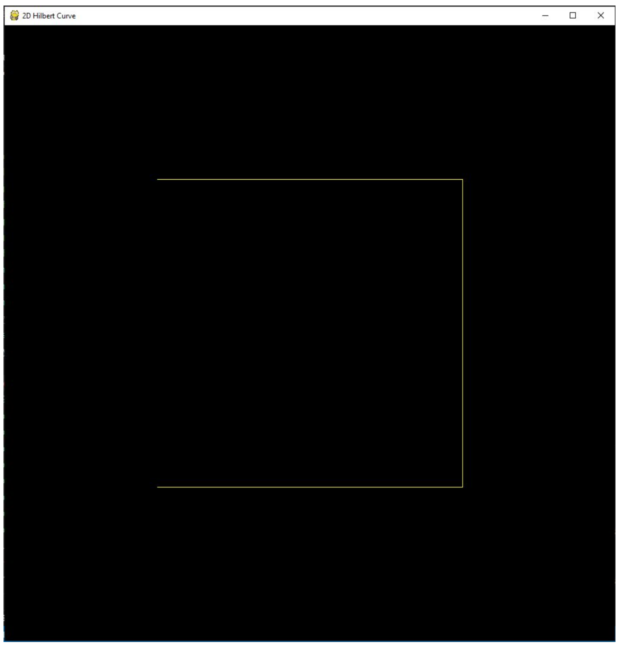

Figure: Order 1

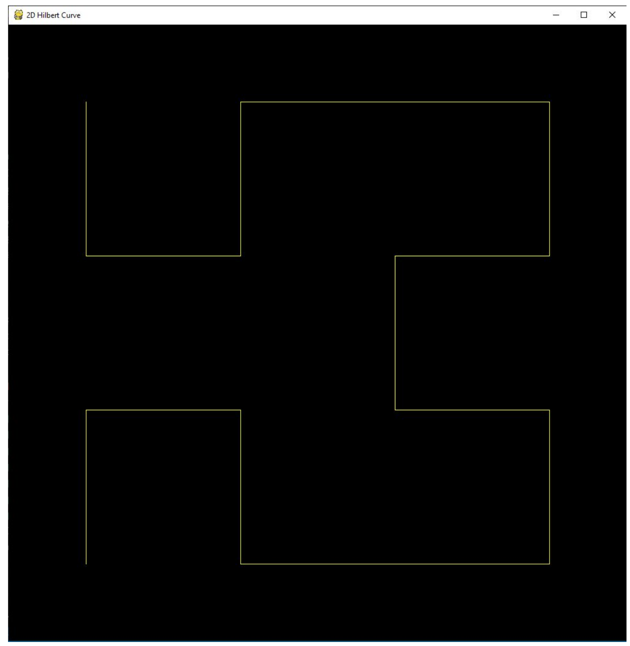

Figure: Order 2

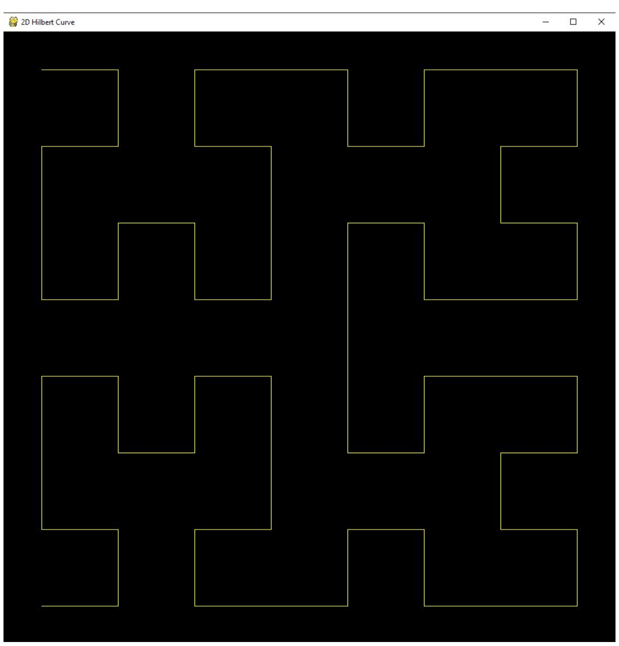

Figure: Order 3

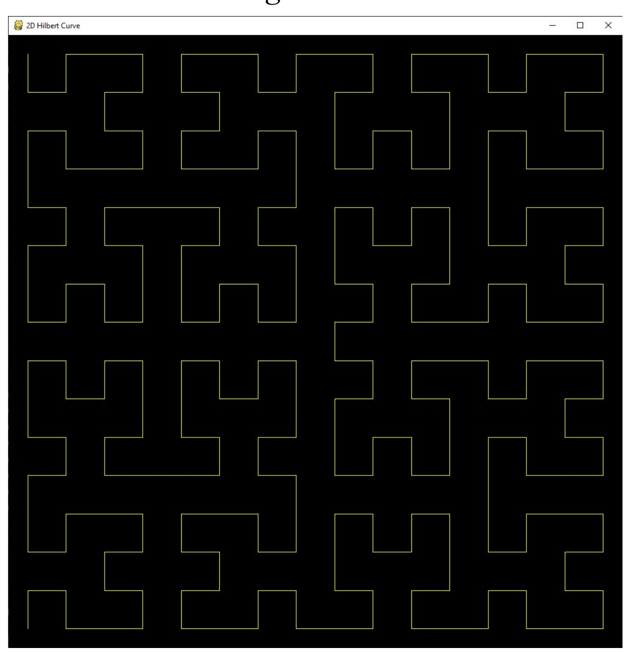

Figure: Order 4

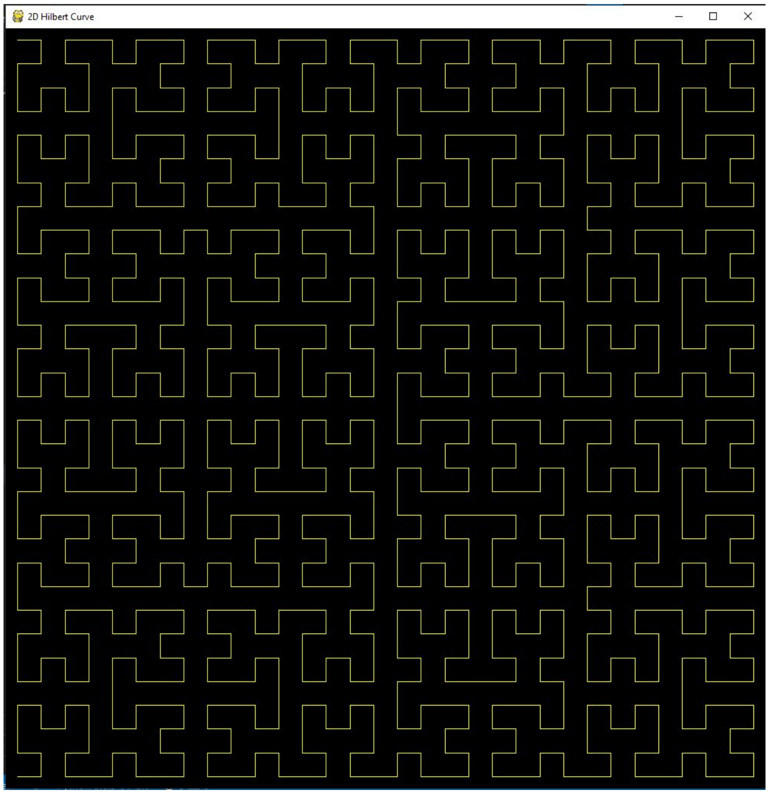

Figure: Order 5

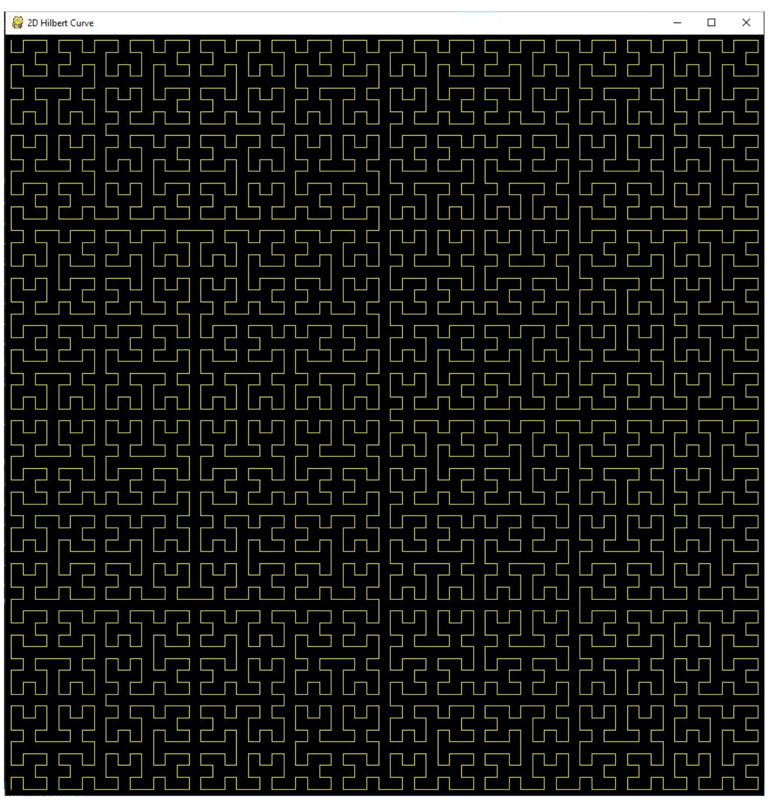

Figure: Order 6

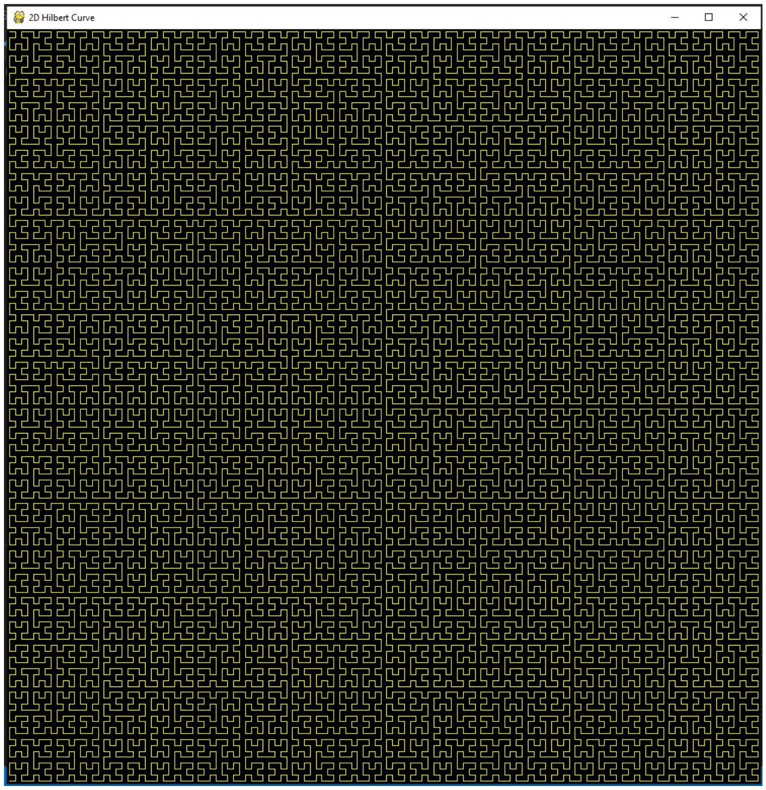

Figure: Order7

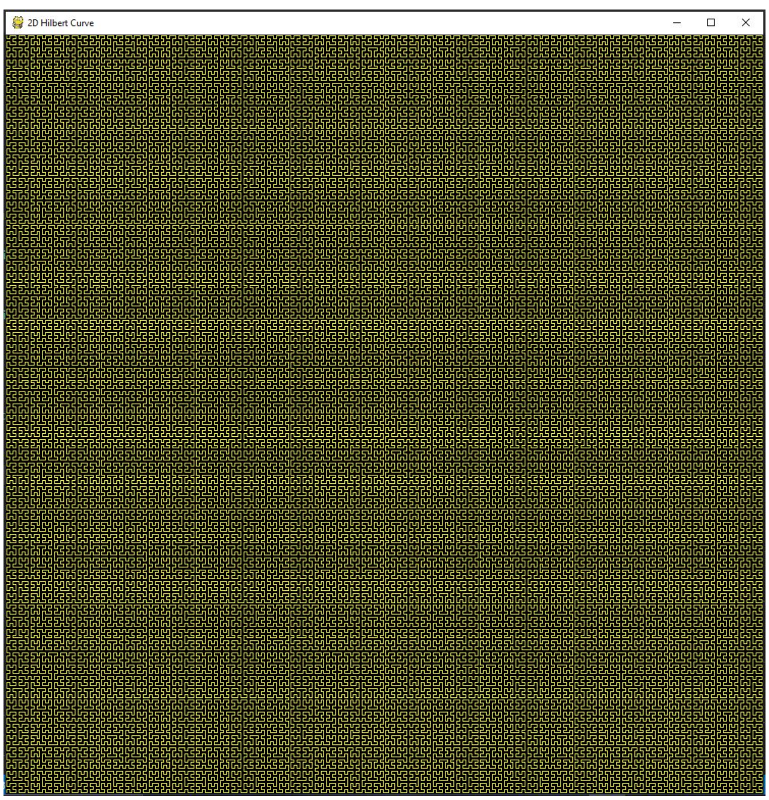

Figure: Order 8

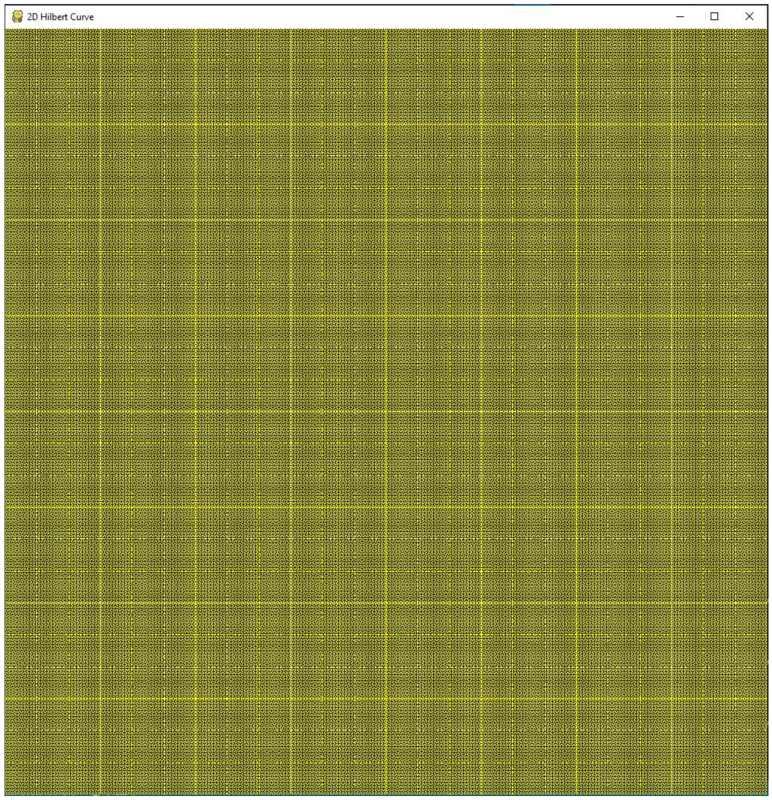

Figure: Order 9

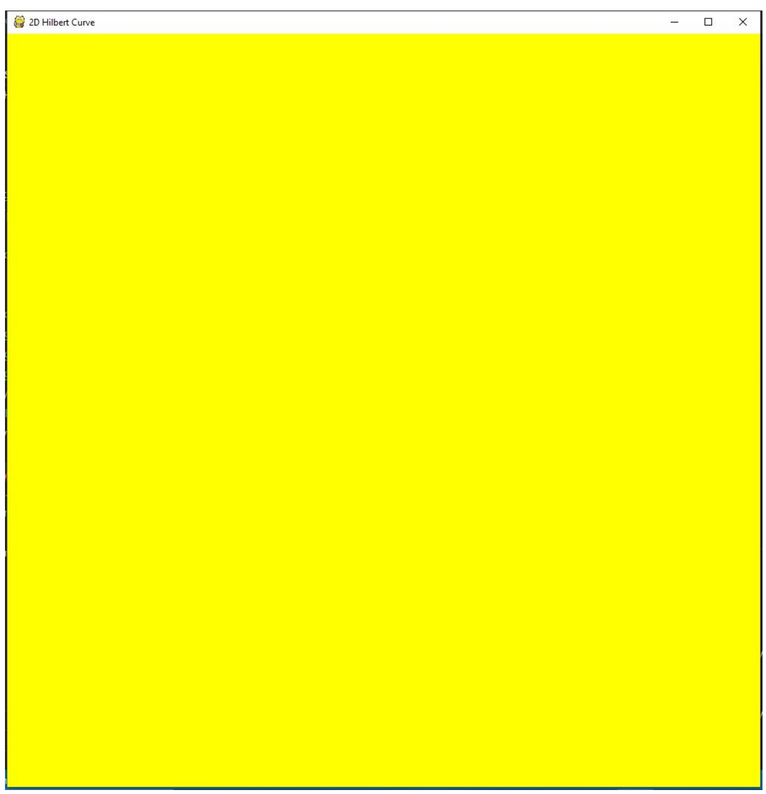

Figure: Order 10

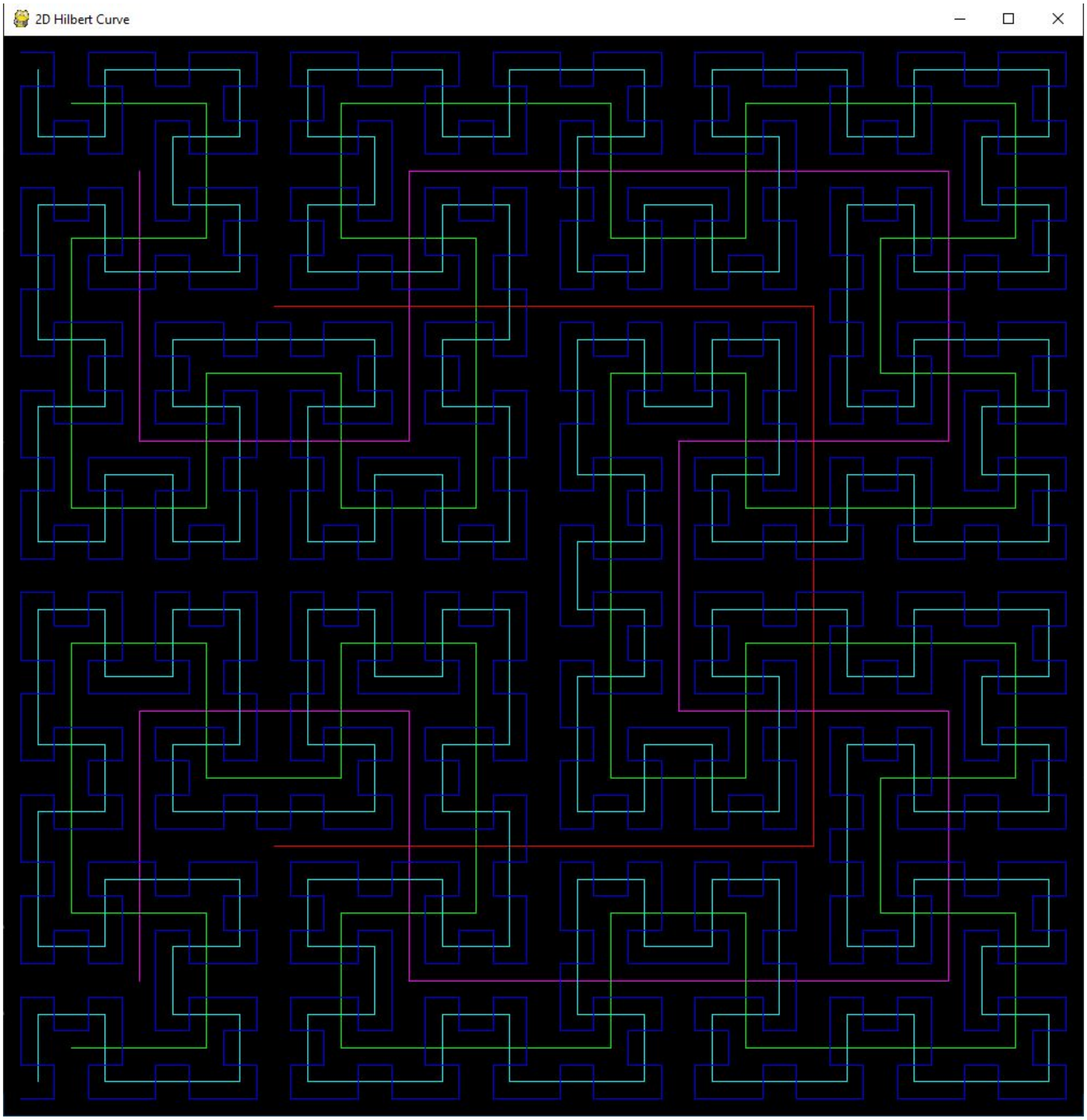

Figure: Curves from Order 1 to 5

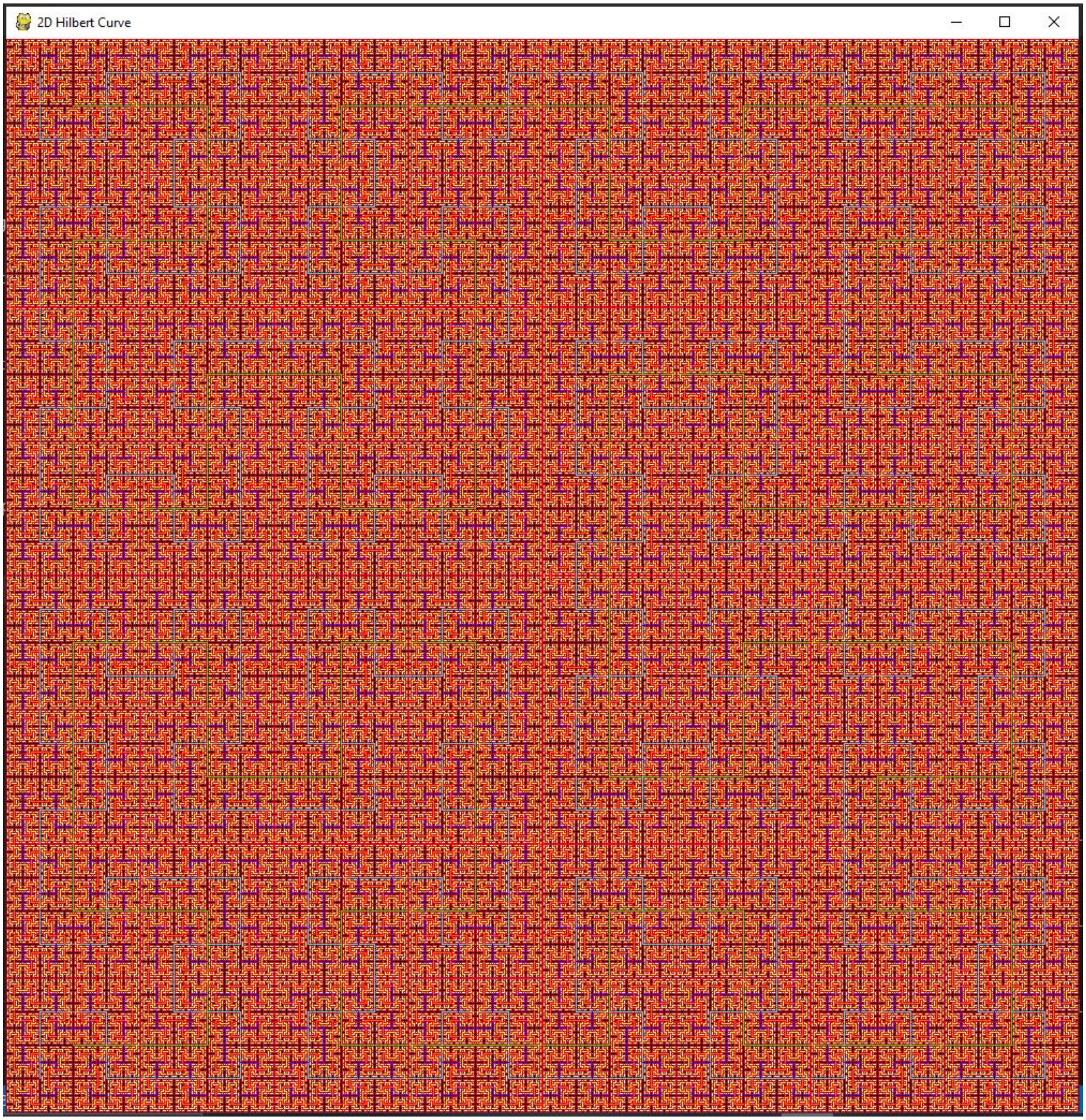

Figure: Curves from Order 1 to 9

Conclusion:

Pygame was used to draw and display the 2D graphics. The window is of 1000X1000 which could clearly display Hilbert Curve till 9th order. After that (from order 10 and upward) the window was filled with a solid colour (space completely filled).

From the figure: *Curves from Order 1 to 5*, we can see the relationship of consecutive ordered Hilbert curves.

The source code and output images can be found through the link below:

https://github.com/mallaneha/RasterGraphics/tree/master/mini