Tree(depth, startX, startY, angle)

if depth >= 1

Calculate length and endpoint of branch

Draw line startX, startY to endpoint, at angle

Tree(depth-1, endpoint.X, endpoint.Y, angle)

Tree(depth-1, endpoint.X, endpoint.Y, -angle)

KochCurve(depth, startX, startY, angle, length)

if depth == 0

Find endpoint based on angle and length

Draw line startX, startY to endpoint.X, endpoint.Y

else

Calculate the new four points of the line by dividing the line

into three segments and replacing the middle segment with an

equilateral triangle

newLength = length / 3

DrawKoch(depth - 1, point1, angle, newLength);

DrawKoch(depth - 1, point2, angle - 60, newLength);

DrawKoch(depth - 1, point3, angle + 60, newLength);

DrawKoch(depth - 1, point4, angle, newLength);

SierpinkskiGasket(depth, startX, startY, length)

if depth == 0

Draw a square at startX,startY with sides length

else

Divide current square into four equal squares

Don't recurse into bottom left square

newLength = length / 2

SierpinkskiGasket(depth - 1, upperRightX, upperRightY, newLength)

SierpinkskiGasket(depth - 1, lowerRightX, lowerRightY, newLength)

SierpinkskiGasket(depth - 1, upperLeftX, upperLeftY, newLength)

Vicsek(depth, startX, startY, length)

if depth == 0

Draw a square at startX,startY with sides length

else

Divide current square into nine equal squares

Don't recurse into corner squares

newLength = length / 3

Vicsek(depth - 1, upperX, upperY, newLength)

Vicsek(depth - 1, leftX, leftY, newLength)

Vicsek(depth - 1, midX, midY, newLength)

Vicsek(depth - 1, rightX, rightY, newLength)

Vicsek(depth - 1, bottomX, bottomY, newLength)

Dragon(depth, startX, startY, endX, endY, length)

if depth == 0

Draw line startX, startY to endX, endY

else

Calculate new corner point - create right angle triangle with current

start and end as the hypotenuse

The length of the sides of this triangle becomes newLength

Dragon(depth - 1, startX, startY, corner.X, corner.Y, newLength)

Dragon(depth - 1, endX, endY, corner.X, corner.Y, newLength)