

Recursive Images

MCS 275 L-7

30 January 2008

A Simple GUI

- a regular n -gon
- recursive images

A Simple GUI

- a regular n -gon
- recursive images

The Cantor Set

- recursive definition
- recursive drawing algorithm

The Cantor Set

- recursive definition
- recursive drawing algorithm

The Koch Curve and Flake

- recursive definition
- recursive drawing algorithm
- GUI for a Koch flake

The Koch Curve and Flake

- recursive definition
- recursive drawing algorithm
- GUI for a Koch flake

MCS 275 Lecture 7
Programming Tools and File Management
Jan Verschelde, 30 January 2008

Recursive Images

MCS 275 L-7

30 January 2008

A Simple GUI

a regular n -gon

recursive images

A Simple GUI

a regular n -gon

recursive images

The Cantor Set

recursive definition

recursive drawing algorithm

The Cantor Set

recursive definition

recursive drawing algorithm

The Koch Curve and Flake

recursive definition

recursive drawing algorithm

GUI for a Koch flake

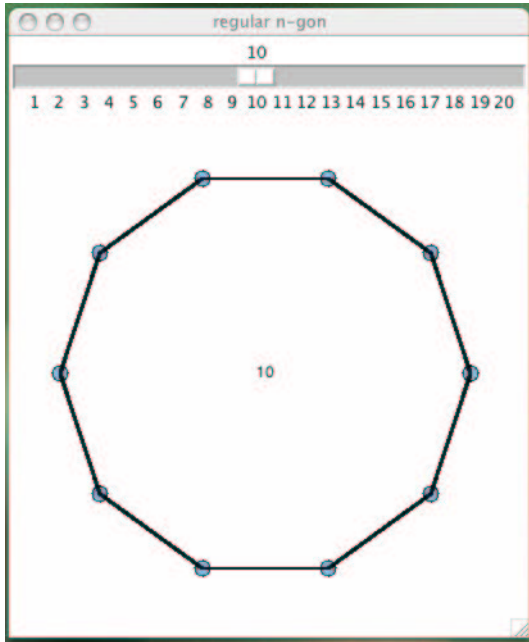
The Koch Curve and Flake

recursive definition

recursive drawing algorithm

GUI for a Koch flake

A Regular n-Gon



MCS 275 L-7

30 January 2008

A Simple GUI

a regular n-gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

The Class MultiGon

MCS 275 L-7

30 January 2008

```
from Tkinter import *
import math

class MultiGon():
    """
    GUI to draw a regular n-gon on canvas.
    """
    def __init__(self, wdw):
        "determines the layout of the GUI"
    def DrawGon(self, v):
        "Draws a regular n-gon"

def main():
    top = Tk()
    show = MultiGon(top)
    top.mainloop()

if __name__ == "__main__": main()
```

[A Simple GUI](#)

a regular n-gon
recursive images

[The Cantor Set](#)

recursive definition
recursive drawing algorithm

[The Koch Curve
and Flake](#)

recursive definition
recursive drawing algorithm
GUI for a Koch flake

The Class MultiGon

MCS 275 L-7

30 January 2008

```
from Tkinter import *
import math

class MultiGon():
    """
    GUI to draw a regular n-gon on canvas.
    """
    def __init__(self, wdw):
        "determines the layout of the GUI"
    def DrawGon(self, v):
        "Draws a regular n-gon"

def main():
    top = Tk()
    show = MultiGon(top)
    top.mainloop()

if __name__ == "__main__": main()
```

[A Simple GUI](#)

a regular n-gon
recursive images

[The Cantor Set](#)

recursive definition
recursive drawing algorithm

[The Koch Curve
and Flake](#)

recursive definition
recursive drawing algorithm
GUI for a Koch flake

The Layout of the GUI

the constructor `__init__`

```
def __init__(self,wdw):  
    "determines the layout of the GUI"  
    wdw.title('regular n-gon')  
    self.d = 400          # dimension of canvas  
    self.n = IntVar()     # number of points  
    self.sn = Scale(wdw,orient='horizontal',\  
        from_=1,to=20,tickinterval=1,\  
        length=self.d,variable=self.n,\  
        command=self.DrawGon)  
    self.sn.grid(row=0,column=0)  
    self.sn.set(10)  
    self.c = Canvas(wdw,width=self.d,\  
        height=self.d,bg='white')  
    self.c.grid(row=1,column=0)
```

MCS 275 L-7

30 January 2008

[A Simple GUI](#)

a regular n-gon
recursive images

[The Cantor Set](#)

recursive definition
recursive drawing algorithm

[The Koch Curve
and Flake](#)

recursive definition
recursive drawing algorithm
GUI for a Koch flake

The Layout of the GUI

the constructor `__init__`

```
def __init__(self,wdw):  
    "determines the layout of the GUI"  
    wdw.title('regular n-gon')  
    self.d = 400          # dimension of canvas  
    self.n = IntVar()    # number of points  
    self.sn = Scale(wdw,orient='horizontal',\  
        from_=1,to=20,tickinterval=1,\  
        length=self.d,variable=self.n,\  
        command=self.DrawGon)  
    self.sn.grid(row=0,column=0)  
    self.sn.set(10)  
    self.c = Canvas(wdw,width=self.d,\  
        height=self.d,bg='white')  
    self.c.grid(row=1,column=0)
```

MCS 275 L-7

30 January 2008

A Simple GUI

a regular n-gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve
and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

The Layout of the GUI

the constructor `__init__`

```
def __init__(self,wdw):  
    "determines the layout of the GUI"  
    wdw.title('regular n-gon')  
    self.d = 400          # dimension of canvas  
    self.n = IntVar()    # number of points  
    self.sn = Scale(wdw,orient='horizontal',\  
        from_=1,to=20,tickinterval=1,\  
        length=self.d,variable=self.n,\  
        command=self.DrawGon)  
    self.sn.grid(row=0,column=0)  
    self.sn.set(10)  
    self.c = Canvas(wdw,width=self.d,\  
        height=self.d,bg='white')  
    self.c.grid(row=1,column=0)
```

A Simple GUI

a regular n-gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

The Function DrawGon()

MCS 275 L-7

30 January 2008

The Scale triggers the command DrawGon () and determines the data attribute n of a MultiGon object.

The second argument v is the value for n passed to DrawGon() via the Scale.

```
def DrawGon(self,v):  
    "Draws a regular n-gon"  
    cx = self.d/2  
    cy = self.d/2  
    radius = 0.4*self.d  
    self.c.delete(ALL)  
    self.c.create_text(cx,cy,text=v,tags="text")  
    n = int(v)
```

The value v is passed as a string, to use it as an integer n we convert it with int ().

A Simple GUI

a regular n-gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve
and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

The Function DrawGon()

The Scale triggers the command DrawGon() and determines the data attribute n of a MultiGon object.

The second argument v is the value for n passed to DrawGon() via the Scale.

```
def DrawGon(self,v):  
    "Draws a regular n-gon"  
    cx = self.d/2  
    cy = self.d/2  
    radius = 0.4*self.d  
    self.c.delete(ALL)  
    self.c.create_text(cx,cy,text=v,tags="text")  
    n = int(v)
```

The value v is passed as a string,
to use it as an integer n we convert it with int().

[A Simple GUI](#)[a regular n-gon](#)[recursive images](#)[The Cantor Set](#)[recursive definition](#)[recursive drawing algorithm](#)[The Koch Curve
and Flake](#)[recursive definition](#)[recursive drawing algorithm](#)[GUI for a Koch flake](#)

The Function DrawGon()

The Scale triggers the command DrawGon () and determines the data attribute n of a MultiGon object.

The second argument v is the value for n passed to DrawGon() via the Scale.

```
def DrawGon(self,v):  
    "Draws a regular n-gon"  
    cx = self.d/2  
    cy = self.d/2  
    radius = 0.4*self.d  
    self.c.delete(ALL)  
    self.c.create_text(cx,cy,text=v,tags="text")  
    n = int(v)
```

The value v is passed as a string, to use it as an integer n we convert it with int ().

[A Simple GUI](#)

a regular n-gon
recursive images

[The Cantor Set](#)

recursive definition
recursive drawing algorithm

[The Koch Curve and Flake](#)

recursive definition
recursive drawing algorithm
GUI for a Koch flake

All n points lie on a circle,
equispaced using angle $2\pi/n$.

a regular n-gon

- recursive definition
- recursive drawing algorithm
- GUI for a Koch flake

All n points lie on a circle,
equispaced using angle $2\pi/n$.

```
L = []
for i in range(0,n):
    vx = cx + radius*math.cos(2*i*math.pi/n)
    vy = cy + radius*math.sin(2*i*math.pi/n)
    self.c.create_oval(vx-6,vy-6,vx+6,vy+6,width=1,\
        outline='black',fill='SkyBlue2',tags="dot")
    L.append((vx,vy))
for i in range(0,n-1):
    self.c.create_line(L[i][0],L[i][1],\
        L[i+1][0],L[i+1][1],width=2)
self.c.create_line(L[n-1][0],L[n-1][1],\
    L[0][0],L[0][1],width=2)
```

a regular n-gon
recursive images

- recursive definition
- recursive drawing algorithm
- GUI for a Koch flake

The Function DrawGon() continued

MCS 275 L-7

30 January 2008

All n points lie on a circle,
equispaced using angle $2\pi/n$.

```
L = []
for i in range(0,n):
    vx = cx + radius*math.cos(2*i*math.pi/n)
    vy = cy + radius*math.sin(2*i*math.pi/n)
    self.c.create_oval(vx-6,vy-6,vx+6,vy+6,width=1,\
        outline='black',fill='SkyBlue2',tags="dot")
    L.append((vx,vy))
for i in range(0,n-1):
    self.c.create_line(L[i][0],L[i][1],\
        L[i+1][0],L[i+1][1],width=2)
self.c.create_line(L[n-1][0],L[n-1][1],\
    L[0][0],L[0][1],width=2)
```

A Simple GUI

a regular n -gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve
and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

Recursive Images

MCS 275 L-7

30 January 2008

A Simple GUI

a regular n -gon
recursive images

A Simple GUI

a regular n -gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

The Koch Curve and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

A Cantor Set

MCS 275 L-7

30 January 2008

A Simple GUI

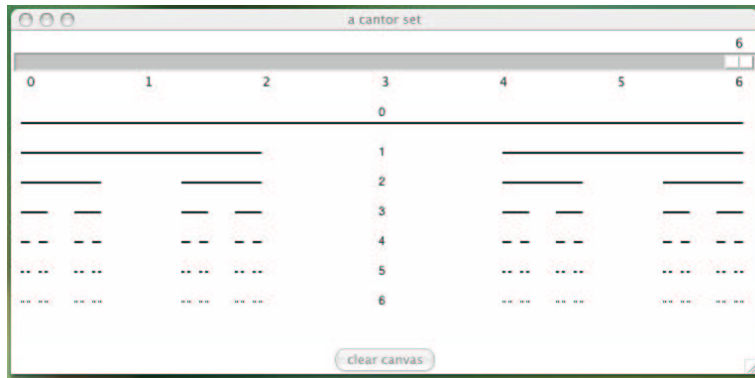
a regular n-gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake



A Koch Curve

MCS 275 L-7

30 January 2008

A Simple GUI

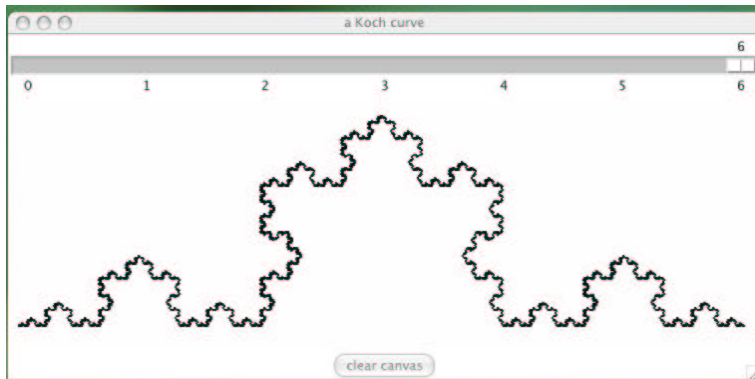
a regular n -gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve and Flake

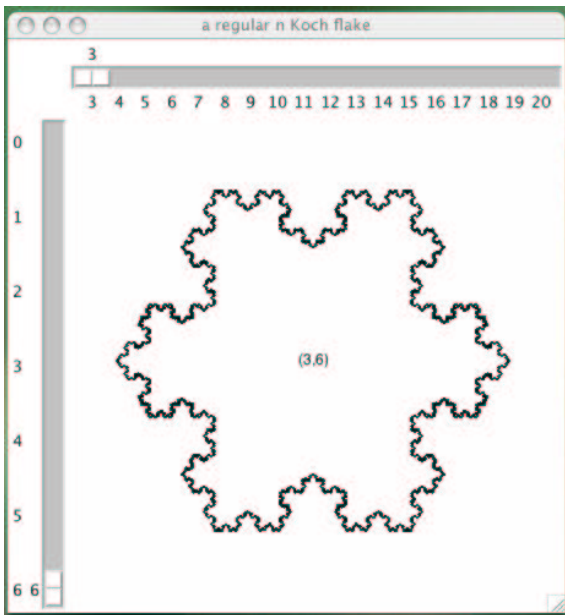
recursive definition
recursive drawing algorithm
GUI for a Koch flake



A Koch Flake

MCS 275 L-7

30 January 2008



A Simple GUI

a regular n -gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

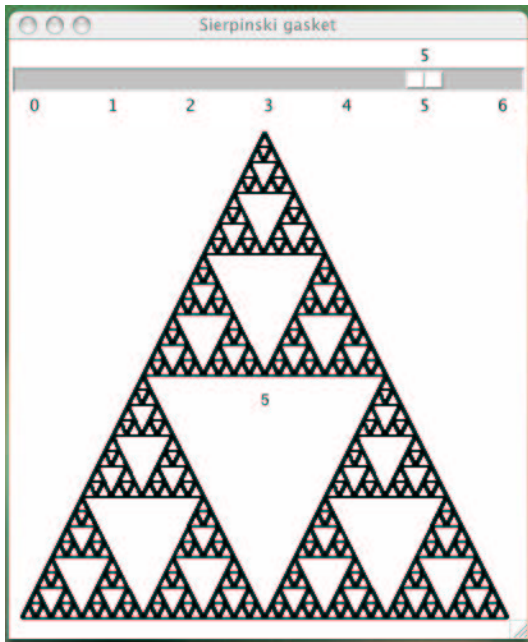
The Koch Curve and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

A Sierpinski Gasket

MCS 275 L-7

30 January 2008



A Simple GUI

a regular n -gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

Recursive Images

MCS 275 L-7

30 January 2008

A Simple GUI

a regular n -gon
recursive images

A Simple GUI

a regular n -gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

The Koch Curve and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

A Simple GUI

a regular n -gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

The Cantor Set is defined by three rules

1. take the interval $[0, 1]$
2. remove the middle part third of the interval
3. repeat rule 2 on the first and third part

The Cantor set is infinite, to visualize at level n :

- ▶ $n = 0$: start at $[0, 1]$
- ▶ $n > 0$: apply rule 2 n times

A Simple GUI

a regular n -gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

The Cantor Set is defined by three rules

1. take the interval $[0, 1]$
2. remove the middle part third of the interval
3. repeat rule 2 on the first and third part

The Cantor set is infinite, to visualize at level n :

- ▶ $n = 0$: start at $[0, 1]$
- ▶ $n > 0$: apply rule 2 n times

A Simple GUI

a regular n -gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

The Cantor Set is defined by three rules

1. take the interval $[0, 1]$
2. remove the middle part third of the interval
3. repeat rule 2 on the first and third part

The Cantor set is infinite, to visualize at level n :

- ▶ $n = 0$: start at $[0, 1]$
- ▶ $n > 0$: apply rule 2 n times

The Cantor Set is defined by three rules

1. take the interval $[0, 1]$
2. remove the middle part third of the interval
3. repeat rule 2 on the first and third part

The Cantor set is infinite, to visualize at level n :

- ▶ $n = 0$: start at $[0, 1]$
- ▶ $n > 0$: apply rule 2 n times

The Cantor Set is defined by three rules

1. take the interval $[0, 1]$
2. remove the middle part third of the interval
3. repeat rule 2 on the first and third part

The Cantor set is infinite, to visualize at level n :

- ▶ $n = 0$: start at $[0, 1]$
- ▶ $n > 0$: apply rule 2 n times

GUI for a Cantor Set

MCS 275 L-7

30 January 2008

A Simple GUI

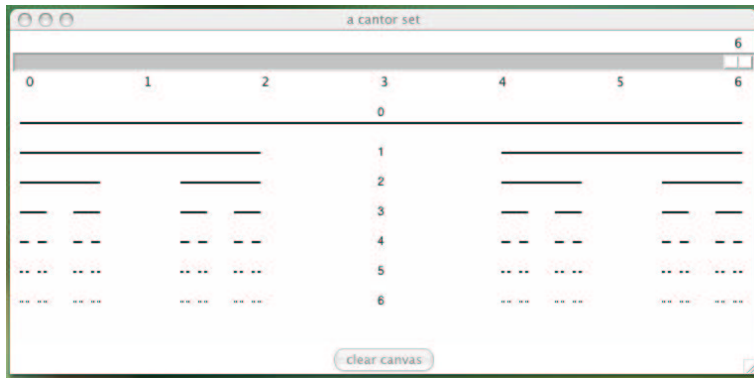
a regular n-gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake



the Class CantorSet

MCS 275 L-7

30 January 2008

```
from Tkinter import *
import math

class CantorSet():
    """
    GUI to draw a Cantor set on canvas.
    """
    def __init__(self, wdw, N):
        "a Cantor set with N levels"

    def DrawSet(self, v):
        "Draws a Cantor set"

    def ClearCanvas(self):
        "Clears the entire canvas"
```

A Simple GUI

a regular n-gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve
and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

```
def __init__(self,wdw,N):
    "a Cantor set with N levels"
    wdw.title('a cantor set')
    self.d = 3**N+20
    self.n = IntVar()
    self.sn = Scale(wdw,orient='horizontal',\
        from_=0,to=N,tickinterval=1,\
        length=self.d,variable=self.n,\
        command=self.DrawSet)
    self.sn.grid(row=0,column=0)
    self.sn.set(0)
    self.c = Canvas(wdw,width=self.d,\
        height=self.d/3,bg='white')
    self.c.grid(row=1,column=0)
    self.b = Button(wdw,text="clear canvas",\
        command=self.ClearCanvas)
    self.b.grid(row=2,column=0)
```

The Layout of the GUI

MCS 275 L-7

30 January 2008

```
def __init__(self,wdw,N):
    "a Cantor set with N levels"
    wdw.title('a cantor set')
    self.d = 3**N+20
    self.n = IntVar()
    self.sn = Scale(wdw,orient='horizontal',\
        from_=0,to=N,tickinterval=1,\
        length=self.d,variable=self.n,\
        command=self.DrawSet)
    self.sn.grid(row=0,column=0)
    self.sn.set(0)
    self.c = Canvas(wdw,width=self.d,\
        height=self.d/3,bg='white')
    self.c.grid(row=1,column=0)
    self.b = Button(wdw,text="clear canvas",\
        command=self.ClearCanvas)
    self.b.grid(row=2,column=0)
```

A Simple GUI

a regular n-gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

The Layout of the GUI

MCS 275 L-7

30 January 2008

```
def __init__(self,wdw,N):
    "a Cantor set with N levels"
    wdw.title('a cantor set')
    self.d = 3**N+20
    self.n = IntVar()
    self.sn = Scale(wdw,orient='horizontal',\
        from_=0,to=N,tickinterval=1,\
        length=self.d,variable=self.n,\
        command=self.DrawSet)
    self.sn.grid(row=0,column=0)
    self.sn.set(0)
    self.c = Canvas(wdw,width=self.d,\
        height=self.d/3,bg='white')
    self.c.grid(row=1,column=0)
    self.b = Button(wdw,text="clear canvas",\
        command=self.ClearCanvas)
    self.b.grid(row=2,column=0)
```

A Simple GUI

a regular n-gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

The Layout of the GUI

MCS 275 L-7

30 January 2008

```
def __init__(self,wdw,N):
    "a Cantor set with N levels"
    wdw.title('a cantor set')
    self.d = 3**N+20
    self.n = IntVar()
    self.sn = Scale(wdw,orient='horizontal',\
        from_=0,to=N,tickinterval=1,\
        length=self.d,variable=self.n,\
        command=self.DrawSet)
    self.sn.grid(row=0,column=0)
    self.sn.set(0)
    self.c = Canvas(wdw,width=self.d,\
        height=self.d/3,bg='white')
    self.c.grid(row=1,column=0)
    self.b = Button(wdw,text="clear canvas",\
        command=self.ClearCanvas)
    self.b.grid(row=2,column=0)
```

A Simple GUI

a regular n-gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve
and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

Methods ClearCanvas and DrawSet

MCS 275 L-7

30 January 2008

The method `ClearCanvas()` is triggered by a Button.

```
def ClearCanvas(self):  
    "Clears the entire canvas"  
    self.c.delete(ALL)
```

The method `DrawSet()` is triggered by a Scale.

```
def DrawSet(self,v):  
    "Draws a Cantor set"  
    n = int(v)  
    self.cantor(10,self.d-10,30,v,n)
```

The method `cantor` is recursive.

A Simple GUI

a regular n-gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve
and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

Methods ClearCanvas and DrawSet

MCS 275 L-7

30 January 2008

The method `ClearCanvas()` is triggered by a Button.

```
def ClearCanvas(self):  
    "Clears the entire canvas"  
    self.c.delete(ALL)
```

The method `DrawSet()` is triggered by a Scale.

```
def DrawSet(self,v):  
    "Draws a Cantor set"  
    n = int(v)  
    self.cantor(10, self.d-10, 30, v, n)
```

The method `cantor` is recursive.

A Simple GUI

a regular n-gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve
and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

Recursive Images

MCS 275 L-7

30 January 2008

A Simple GUI

a regular n -gon
recursive images

A Simple GUI

a regular n -gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

The Koch Curve and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

Parameters of a Recursive Function

MCS 275 L-7

30 January 2008

```
def cantor(self,A,B,z,t,k):  
    """  
    draws a line from A to B, at height z  
    t is a string, int(t) equals the number  
    of times the middle third must be removed  
    k is level of recursion, start at k = int(t)  
    """
```

The parameters A, B, and z define
the line segment from (A,z) to (B,z).

The parameter t is the value passed via the Scale,
as text string, t is also put on Canvas.

Initially: `k = int(t)`.

With every recursive call, k is decremented by 1.

A Simple GUI

a regular n-gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve
and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

Parameters of a Recursive Function

MCS 275 L-7

30 January 2008

```
def cantor(self,A,B,z,t,k):  
    """  
    draws a line from A to B, at height z  
    t is a string, int(t) equals the number  
    of times the middle third must be removed  
    k is level of recursion, start at k = int(t)  
    """
```

The parameters A, B, and z define
the line segment from (A,z) to (B,z).

The parameter t is the value passed via the Scale,
as text string, t is also put on Canvas.

Initially: `k = int(t)`.

With every recursive call, k is decremented by 1.

A Simple GUI

a regular n-gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve
and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

Parameters of a Recursive Function

MCS 275 L-7

30 January 2008

```
def cantor(self,A,B,z,t,k):
```

```
    """
```

```
    draws a line from A to B, at height z
    t is a string, int(t) equals the number
    of times the middle third must be removed
    k is level of recursion, start at k = int(t)
    """
```

The parameters A, B, and z define
the line segment from (A,z) to (B,z).

The parameter t is the value passed via the Scale,
as text string, t is also put on Canvas.

Initially: `k = int(t)`.

With every recursive call, k is decremented by 1.

A Simple GUI

a regular n-gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve
and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

Parameters of a Recursive Function

MCS 275 L-7

30 January 2008

```
def cantor(self,A,B,z,t,k):  
    """  
    draws a line from A to B, at height z  
    t is a string, int(t) equals the number  
    of times the middle third must be removed  
    k is level of recursion, start at k = int(t)  
    """
```

The parameters A, B, and z define
the line segment from (A,z) to (B,z).

The parameter t is the value passed via the Scale,
as text string, t is also put on Canvas.

Initially: `k = int(t)`.

With every recursive call, k is decremented by 1.

A Simple GUI

a regular n-gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve
and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

Recursive Drawing Algorithm

MCS 275 L-7

30 January 2008

The k in `cantor(self, A, B, z, t, k)` controls the recursion.

At $k = 0$, the line segment from (A, z) to (B, z) is drawn.

For $k > 0$, we compute left and right limit of the middle third of $[A, B]$, respectively denoted by L and R as

$$L = A + (B - A)/3 = (2A + B)/3$$

$$R = B - (B - A)/3 = (A + 2B)/3$$

Then we make two recursive calls:

```
self.cantor(A, L, z+30, t, k-1)
self.cantor(R, B, z+30, t, k-1)
```

A Simple GUI

a regular n -gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve
and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

Recursive Drawing Algorithm

MCS 275 L-7

30 January 2008

The k in `cantor(self, A, B, z, t, k)` controls the recursion.

At $k = 0$, the line segment from (A, z) to (B, z) is drawn.

For $k > 0$, we compute left and right limit of the middle third of $[A, B]$, respectively denoted by L and R as

$$L = A + (B - A)/3 = (2A + B)/3$$

$$R = B - (B - A)/3 = (A + 2B)/3$$

Then we make two recursive calls:

```
self.cantor(A, L, z+30, t, k-1)
self.cantor(R, B, z+30, t, k-1)
```

A Simple GUI

a regular n -gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve
and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

Recursive Drawing Algorithm

MCS 275 L-7

30 January 2008

The k in `cantor(self, A, B, z, t, k)` controls the recursion.

At $k = 0$, the line segment from (A, z) to (B, z) is drawn.

For $k > 0$, we compute left and right limit of the middle third of $[A, B]$, respectively denoted by L and R as

$$L = A + (B - A)/3 = (2A + B)/3$$

$$R = B - (B - A)/3 = (A + 2B)/3$$

Then we make two recursive calls:

```
self.cantor(A, L, z+30, t, k-1)
self.cantor(R, B, z+30, t, k-1)
```

A Simple GUI

a regular n -gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve
and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

Recursive Drawing Algorithm

MCS 275 L-7

30 January 2008

The k in `cantor(self, A, B, z, t, k)` controls the recursion.

At $k = 0$, the line segment from (A, z) to (B, z) is drawn.

For $k > 0$, we compute left and right limit of the middle third of $[A, B]$, respectively denoted by L and R as

$$\begin{aligned}L &= A + (B - A)/3 = (2A + B)/3 \\ R &= B - (B - A)/3 = (A + 2B)/3\end{aligned}$$

Then we make two recursive calls:

```
self.cantor(A, L, z+30, t, k-1)
self.cantor(R, B, z+30, t, k-1)
```

A Simple GUI

a regular n -gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve
and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

- recursive definition
- recursive drawing algorithm
- GUI for a Koch flake

A set of small navigation icons typically found in Beamer presentations, including symbols for back, forward, search, and other slide controls.

```
def cantor(self,A,B,z,t,k):
    " ... "
    if(k==0):                # draw line segment
        self.c.create_line(A,z,B,z,width=2)
    else:
        L = (2*A+B)/3
        R = (A+2*B)/3
        self.cantor(A,L,z+30,t,k-1)
        self.cantor(R,B,z+30,t,k-1)
    if(k==int(t)):           # put text string t
        cx = self.d/2;
        if(t == '0'):
            self.c.create_text(cx,z-10,text=t)
        else:
            self.c.create_text(cx,z+k*30,text=t)
```

A set of small navigation icons typically found in Beamer presentations, including symbols for back, forward, search, and other slide controls.

Recursive Images

MCS 275 L-7

30 January 2008

A Simple GUI

a regular n -gon
recursive images

A Simple GUI

a regular n -gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

The Koch Curve and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

The Koch Curve

MCS 275 L-7

30 January 2008

The Koch Curve is a Cantor Set where the removed middle third of the interval is replaced by a wedge.

The top of the wedge is above the midpoint of the removed middle interval.

The slopes of the wedge make an angle of 60 degrees with respect to the rest of the line segment.

To visualize a Koch curve at level n :

- ▶ $n = 0$: start at $[0, 1]$
- ▶ $n > 0$: make wedges n times.

A Simple GUI

a regular n -gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve
and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

The Koch Curve

MCS 275 L-7

30 January 2008

The Koch Curve is a Cantor Set where the removed middle third of the interval is replaced by a wedge.

The top of the wedge is above the midpoint of the removed middle interval.

The slopes of the wedge make an angle of 60 degrees with respect to the rest of the line segment.

To visualize a Koch curve at level n :

- ▶ $n = 0$: start at $[0, 1]$
- ▶ $n > 0$: make wedges n times.

A Simple GUI

a regular n -gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve
and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

The Koch Curve

MCS 275 L-7

30 January 2008

The Koch Curve is a Cantor Set where the removed middle third of the interval is replaced by a wedge.

The top of the wedge is above the midpoint of the removed middle interval.

The slopes of the wedge make an angle of 60 degrees with respect to the rest of the line segment.

To visualize a Koch curve at level n :

- ▶ $n = 0$: start at $[0, 1]$
- ▶ $n > 0$: make wedges n times.

A Simple GUI

a regular n -gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve
and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

The Koch Curve is a Cantor Set where the removed middle third of the interval is replaced by a wedge.

The top of the wedge is above the midpoint of the removed middle interval.

The slopes of the wedge make an angle of 60 degrees with respect to the rest of the line segment.

To visualize a Koch curve at level n :

- ▶ $n = 0$: start at $[0, 1]$
- ▶ $n > 0$: make wedges n times.

A Simple GUI

a regular n -gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve
and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

The Koch Curve is a Cantor Set where the removed middle third of the interval is replaced by a wedge.

The top of the wedge is above the midpoint of the removed middle interval.

The slopes of the wedge make an angle of 60 degrees with respect to the rest of the line segment.

To visualize a Koch curve at level n :

- ▶ $n = 0$: start at $[0, 1]$
- ▶ $n > 0$: make wedges n times.

A Simple GUI

a regular n -gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve
and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

GUI for a Koch Curve

MCS 275 L-7

30 January 2008

A Simple GUI

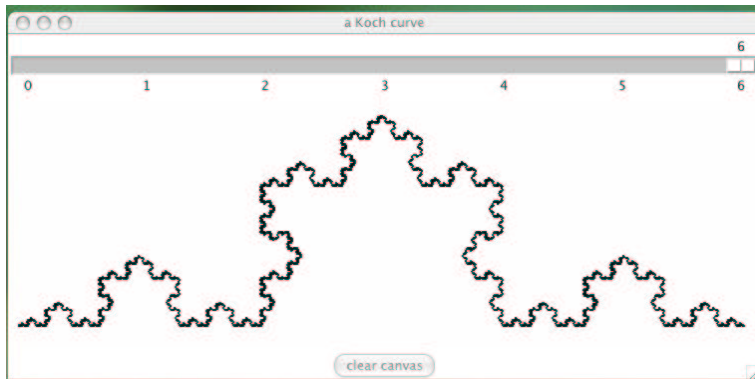
a regular n -gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake



The Class KochCurve

MCS 275 L-7

30 January 2008

```
from Tkinter import *
import math

class KochCurve():
    """
    GUI to draw a Koch curve on canvas.
    """
    def __init__(self, wdw, N):
        "a Koch curve with N levels"

    def DrawCurve(self, v):
        "Draws a regular n-gon"

    def ClearCanvas(self):
        "Clears the entire canvas"
```

A Simple GUI

a regular n-gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve
and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

The Layout of the GUI

MCS 275 L-7

30 January 2008

```
def __init__(self,wdw,N):
    "a Koch curve with N levels"
    wdw.title('a Koch curve')
    self.d = 3**N+20
    self.n = IntVar()
    self.sn = Scale(wdw,orient='horizontal',\
        from_=0,to=N,tickinterval=1,\
        length=self.d,variable=self.n,\
        command=self.DrawCurve)
    self.sn.grid(row=0,column=0)
    self.sn.set(0)
    self.c = Canvas(wdw,width=self.d,\
        height=self.d/3,bg='white')
    self.c.grid(row=1,column=0)
    self.b = Button(wdw,text="clear canvas",\
        command=self.ClearCanvas)
    self.b.grid(row=2,column=0)
```

A Simple GUI

a regular n-gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve
and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

Method ClearCanvas and DrawCurve

MCS 275 L-7

30 January 2008

The method `ClearCanvas()` is triggered by a Button.

```
def ClearCanvas(self):  
    "Clears the entire canvas"  
    self.c.delete(ALL)
```

The method `DrawCurve()` is activated by a Scale.

```
def DrawCurve(self,v):  
    "Draws a regular n-gon"  
    n = int(v)  
    A = (10,self.d/3-20)  
    B = (self.d-10,self.d/3-20)  
    self.koch(A,B,n)
```

The method `koch()` is recursive,
to draw a Koch curve of n levels from A to B .

A Simple GUI

a regular n -gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve
and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

Method ClearCanvas and DrawCurve

The method `ClearCanvas()` is triggered by a Button.

```
def ClearCanvas(self):  
    "Clears the entire canvas"  
    self.c.delete(ALL)
```

The method `DrawCurve()` is activated by a Scale.

```
def DrawCurve(self,v):  
    "Draws a regular n-gon"  
    n = int(v)  
    A = (10,self.d/3-20)  
    B = (self.d-10,self.d/3-20)  
    self.koch(A,B,n)
```

The method `koch()` is recursive,
to draw a Koch curve of n levels from A to B .

A Simple GUI

a regular n -gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve
and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

Recursive Images

MCS 275 L-7

30 January 2008

A Simple GUI

a regular n -gon
recursive images

A Simple GUI

a regular n -gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

The Koch Curve and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

Recursive Drawing Algorithm

MCS 275 L-7

30 January 2008

The n in $\text{koch}(A, B, n)$ controls the recursion.

For $n = 0$, the line segment from A to B is drawn.

For $n > 0$, we compute left L , right R ,
and midpoint M of the middle third of line segment.

Angle of 60 degrees: $\sin(\frac{\pi}{3}) = \frac{\sqrt{3}}{2}$ is height,
multiplied with $\frac{1}{3}$ th of length of segment.

The the peak P of the wedge is relative to the size of the
interval and the position of the midpoint M :

$$T = (\frac{\sqrt{3}}{6}(A[1] - B[1]), \frac{\sqrt{3}}{6}(B[0] - A[0])),$$

$$P = (M[0] - T[0], M[1] - T[1])$$

Recall that on Canvas: $(0,0)$ is at topmost left corner.

A Simple GUI

a regular n -gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve
and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

Recursive Drawing Algorithm

MCS 275 L-7

30 January 2008

The n in $\text{koch}(A, B, n)$ controls the recursion.

For $n = 0$, the line segment from A to B is drawn.

For $n > 0$, we compute left L , right R ,
and midpoint M of the middle third of line segment.

Angle of 60 degrees: $\sin(\frac{\pi}{3}) = \frac{\sqrt{3}}{2}$ is height,
multiplied with $\frac{1}{3}$ th of length of segment.

The the peak P of the wedge is relative to the size of the
interval and the position of the midpoint M :

$$T = (\frac{\sqrt{3}}{6}(A[1] - B[1]), \frac{\sqrt{3}}{6}(B[0] - A[0])),$$

$$P = (M[0] - T[0], M[1] - T[1])$$

Recall that on Canvas: $(0,0)$ is at topmost left corner.

A Simple GUI

a regular n -gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve
and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

Recursive Drawing Algorithm

MCS 275 L-7

30 January 2008

The n in $\text{koch}(A, B, n)$ controls the recursion.

For $n = 0$, the line segment from A to B is drawn.

For $n > 0$, we compute left L , right R ,
and midpoint M of the middle third of line segment.

Angle of 60 degrees: $\sin(\frac{\pi}{3}) = \frac{\sqrt{3}}{2}$ is height,
multiplied with $\frac{1}{3}$ th of length of segment.

The the peak P of the wedge is relative to the size of the
interval and the position of the midpoint M :

$$T = (\frac{\sqrt{3}}{6}(A[1] - B[1]), \frac{\sqrt{3}}{6}(B[0] - A[0])),$$

$$P = (M[0] - T[0], M[1] - T[1])$$

Recall that on Canvas: $(0,0)$ is at topmost left corner.

A Simple GUI

a regular n -gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve
and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

Recursive Drawing Algorithm

MCS 275 L-7

30 January 2008

The n in $\text{koch}(A, B, n)$ controls the recursion.

For $n = 0$, the line segment from A to B is drawn.

For $n > 0$, we compute left L , right R ,
and midpoint M of the middle third of line segment.

Angle of 60 degrees: $\sin(\frac{\pi}{3}) = \frac{\sqrt{3}}{2}$ is height,
multiplied with $\frac{1}{3}$ th of length of segment.

The the peak P of the wedge is relative to the size of the
interval and the position of the midpoint M :

$$T = (\frac{\sqrt{3}}{6}(A[1] - B[1]), \frac{\sqrt{3}}{6}(B[0] - A[0])),$$

$$P = (M[0] - T[0], M[1] - T[1])$$

Recall that on Canvas: $(0,0)$ is at topmost left corner.

A Simple GUI

a regular n -gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve
and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

Recursive Drawing Algorithm

MCS 275 L-7

30 January 2008

The n in $\text{koch}(A, B, n)$ controls the recursion.

For $n = 0$, the line segment from A to B is drawn.

For $n > 0$, we compute left L , right R ,
and midpoint M of the middle third of line segment.

Angle of 60 degrees: $\sin(\frac{\pi}{3}) = \frac{\sqrt{3}}{2}$ is height,
multiplied with $\frac{1}{3}$ th of length of segment.

The the peak P of the wedge is relative to the size of the
interval and the position of the midpoint M :

$$T = (\frac{\sqrt{3}}{6}(A[1] - B[1]), \frac{\sqrt{3}}{6}(B[0] - A[0])),$$

$$P = (M[0] - T[0], M[1] - T[1])$$

Recall that on Canvas: $(0,0)$ is at topmost left corner.

A Simple GUI

a regular n -gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve
and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

Code for koch

MCS 275 L-7

30 January 2008

```
def koch(self,A,B,k):
    """
    a Koch curve from A to B with k levels
    """
    if(k==0):
        self.c.create_line(A[0],A[1],B[0],B[1],\
                           width=2)
    else:
        L = ((2*A[0]+B[0])/3.0,(2*A[1]+B[1])/3.0)
        R = ((A[0]+2*B[0])/3.0,(A[1]+2*B[1])/3.0)
        M = ((A[0]+B[0])/2.0,(A[1]+B[1])/2.0)
        s = math.sqrt(3)/6
        T = (s*(A[1]-B[1]),s*(B[0]-A[0]))
        P = (M[0]-T[0],M[1]-T[1])
        self.koch(A,L,k-1)
        self.koch(L,P,k-1)
        self.koch(P,R,k-1)
        self.koch(R,B,k-1)
```

A Simple GUI

a regular n-gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve
and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

Code for koch

MCS 275 L-7

30 January 2008

```
def koch(self,A,B,k):
    """
    a Koch curve from A to B with k levels
    """
    if(k==0):
        self.c.create_line(A[0],A[1],B[0],B[1],\
                           width=2)
    else:
        L = ((2*A[0]+B[0])/3.0,(2*A[1]+B[1])/3.0)
        R = ((A[0]+2*B[0])/3.0,(A[1]+2*B[1])/3.0)
        M = ((A[0]+B[0])/2.0,(A[1]+B[1])/2.0)
        s = math.sqrt(3)/6
        T = (s*(A[1]-B[1]),s*(B[0]-A[0]))
        P = (M[0]-T[0],M[1]-T[1])
        self.koch(A,L,k-1)
        self.koch(L,P,k-1)
        self.koch(P,R,k-1)
        self.koch(R,B,k-1)
```

A Simple GUI

a regular n-gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve
and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

Code for koch

MCS 275 L-7

30 January 2008

```
def koch(self,A,B,k):
    """
    a Koch curve from A to B with k levels
    """
    if(k==0):
        self.c.create_line(A[0],A[1],B[0],B[1],\
                           width=2)
    else:
        L = ((2*A[0]+B[0])/3.0,(2*A[1]+B[1])/3.0)
        R = ((A[0]+2*B[0])/3.0,(A[1]+2*B[1])/3.0)
        M = ((A[0]+B[0])/2.0,(A[1]+B[1])/2.0)
        s = math.sqrt(3)/6
        T = (s*(A[1]-B[1]),s*(B[0]-A[0]))
        P = (M[0]-T[0],M[1]-T[1])
        self.koch(A,L,k-1)
        self.koch(L,P,k-1)
        self.koch(P,R,k-1)
        self.koch(R,B,k-1)
```

A Simple GUI

a regular n-gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve
and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

Code for koch

MCS 275 L-7

30 January 2008

```
def koch(self,A,B,k):
    """
    a Koch curve from A to B with k levels
    """
    if(k==0):
        self.c.create_line(A[0],A[1],B[0],B[1],\
                           width=2)
    else:
        L = ((2*A[0]+B[0])/3.0,(2*A[1]+B[1])/3.0)
        R = ((A[0]+2*B[0])/3.0,(A[1]+2*B[1])/3.0)
        M = ((A[0]+B[0])/2.0,(A[1]+B[1])/2.0)
        s = math.sqrt(3)/6
        T = (s*(A[1]-B[1]),s*(B[0]-A[0]))
        P = (M[0]-T[0],M[1]-T[1])
        self.koch(A,L,k-1)
        self.koch(L,P,k-1)
        self.koch(P,R,k-1)
        self.koch(R,B,k-1)
```

A Simple GUI

a regular n-gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve
and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

Recursive Images

MCS 275 L-7

30 January 2008

A Simple GUI

a regular n -gon
recursive images

A Simple GUI

a regular n -gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

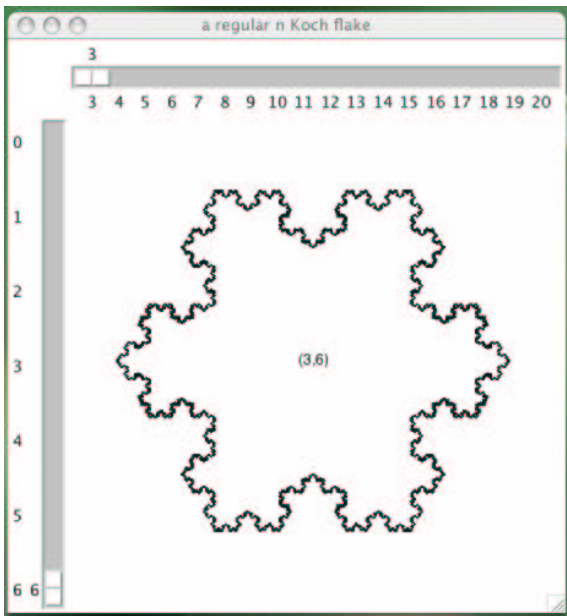
The Koch Curve and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

GUI for a Koch flake

MCS 275 L-7

30 January 2008



A Simple GUI

a regular n-gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

The Class KochFlake

MCS 275 L-7

30 January 2008

```
from Tkinter import *  
import math
```

```
class KochFlake():  
    """  
    GUI to draw a Koch flake canvas.  
    """  
    def __init__(self,wdw):  
        "determines the layout of the GUI"  
  
    def DrawFlake(self,v):  
        "Draws a regular Koch flake"
```

A Simple GUI

a regular n-gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve
and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

a regular n-gon
recursive images

- recursive definition
- recursive drawing algorithm
- GUI for a Koch flake

```
def __init__(self,wdw):
    ...
    self.n = IntVar()
    self.k = IntVar()
    self.sn = Scale(wdw,orient='horizontal',\
        from_=3,to=20,tickinterval=1,\
        length=self.d,variable=self.n,\
        command=self.DrawFlake)
    self.sn.grid(row=0,column=1)
    self.sn.set(10)
    self.sk = Scale(wdw,orient='vertical',\
        from_=0,to=6,tickinterval=1,\
        length=self.d,variable=self.k,\
        command=self.DrawFlake)
    self.sk.grid(row=1,column=0)
    self.sk.set(0)
```

Code for DrawFlake()

MCS 275 L-7

30 January 2008

```
def DrawFlake(self,v):
    "Draws a regular Koch flake"
    cx = self.d/2
    cy = self.d/2
    radius = 0.4*self.d
    self.c.delete(ALL)
    n = self.n.get()
    k = self.k.get()
    t = '(' + str(n) + ', ' + str(k) + ')'
    self.c.create_text(cx,cy,text=t,tags="text")
    L = []
    for i in range(0,n):
        vx = cx + radius*math.cos(2*i*math.pi/n)
        vy = cy + radius*math.sin(2*i*math.pi/n)
        L.append((vx,vy))
    for i in range(0,n-1):
        self.koch(L[i],L[i+1],k)
    self.koch(L[n-1],L[0],k)
```

A Simple GUI

a regular n-gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve
and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

Exercises

1. Make a GUI to visualize the Sierpinski gasket.
2. A variant of the Sierpinski gasket starts with a square and removes a smaller square at the center of the first square. This removal is repeated to the 8 remaining squares. Design a GUI for this Sierpinski carpet. Define the recursive drawing algorithm.
3. Make a GUI to visualize a Brownian bridge between two points. The rule is to replace a line segment from A to B by the segments $[A, M]$ and $[M, B]$, where M is calculated as $(A + B)/2$ with random noise added to it. Repeat the rule to the new segments, etc.
4. Design a recursive algorithm to draw a tree, starting with a trunk and 3 branches. Put at the top of each branch again a trunk and 3 branches, and then again... What are the parameters of the recursive function to define the rule to generate the tree?

A Simple GUI

a regular n -gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

Summary and Assignments

MCS 275 L-7

30 January 2008

A Simple GUI

a regular n -gon
recursive images

The Cantor Set

recursive definition
recursive drawing algorithm

The Koch Curve and Flake

recursive definition
recursive drawing algorithm
GUI for a Koch flake

We recalled elements of GUIs, see

- ▶ Chapter 15 of *Making Use of Python*
- ▶ `www.python.org` has plenty of documentation

Homework will be collected on Friday 1 February
(bring your answers to class):

- ▶ exercise 5 of Lecture 5;
- ▶ exercises 1 and 4 of Lecture 6;
- ▶ exercises 1 and 3 of Lecture 7.