Instituto Politécnico Nacional

Escuela Superior de Cómputo

Evolutionary Computing

3CM1

7 Fractals

Santiago Nieves Edgar Augusto

Teacher: Jorge Luis Rosas Trigueros

Date: 18/11/2016

Theoretical framework:

A fractal is a mathematical set that exhibits a repeating pattern that displays at every scale. It is also known as expanding symmetry or evolving symmetry. Fractals are different from other geometric figures because of the way in which they scale. Doubling the edge lengths of a polygon multiplies its area by four, which is two raised to the power of two. As mathematical equations, fractals are usually nowhere differentiable. An infinite fractal curve can be conceived of as winding through space differently from an ordinary line, still being a 1-dimensional line yet having a fractal dimension indicating it also resembles a surface.

The word "fractal" often has different connotations for laypeople than for mathematicians, where the layperson is more likely to be familiar with fractal art than a mathematical conception. The mathematical concept is difficult to define formally even for mathematicians, but key features can be understood with little mathematical background.

One often cited description that Mandelbrot published to describe geometric fractals is "a rough or fragmented geometric shape that can be split into parts, each of which is a reduced-size copy of the whole", this is generally helpful but limited.

Fractals are not just complex shapes and pretty pictures generated by computers. Anything that appears random and irregular can be a fractal. Fractals permeate our lives, appearing in places as tiny as the membrane of a cell and as majestic as the solar system. Fractals are the unique, irregular patterns left behind by the unpredictable movements of the chaotic world at work.

In theory, one can argue that everything existent on this world is a fractal:

- The branching of tracheal tubes.
- The leaves in trees.
- The veins in a hand.
- Water swirling and twisting out of a tap.
- A puffy cumulus cloud.
- Tiny oxygene molecule, or the DNA molecule.
- The stock market.

The most useful use of fractals in computer science is the fractal image compression. This kind of compression uses the fact that the real world is well described by fractal geometry. Images are compressed much more than by usual ways and an other advantage of fractal compression is that when the picture is enlarged, there is no pixelisation. The picture seems very often better when its size is increased.

Material and equipment:

Software:

Sublime Text.

Python 2.7.6.

Ubuntu 16.04 LTS.

Hardware:

Processor: Intel® Core™ i3 CPU M 380 @ 2.1GHz × 4.

Body:

In this practice we will see 5 type of fractals that we saw in class, but we need to check with different size or different iterations and in the end we need to create a landscape with this figures or with others figures but it's necessary to use the figures with fractals.

All the figures need this code to create the image, but each figure need a little modify and it is the parameters and the recursive function, this is the principal function to create all the figures.

```
def ec(w,x,y,l,a,sl,da,n):
    if n==0:
        return

    color='black'

    if n>5:
        color='brown'
    elif n>2:
        color='green'
    elif n>0:
        color='white'

    x2=x+l
    y2=y
    ar=math.radians(a)

    coseno=math.cos(ar)
    seno=math.sin(ar)
    xrot= (x2-x)*seno + (y2-y)*coseno
    yrot= (x2-x)*coseno - (y2-y)*seno

    x2=xrot + x
    y2=yrot + y

    w.create_line(x,y,x2,y2,fill=color)
```

Figure 1. The principal code to create all the fractals.

1. Tree:

This is the recursive function to create this figure:

```
ec(w,x2,y2,l*sl,a-da,sl,da,n-1)
ec(w,x2,y2,l*sl,a+da,sl,da,n-1)
ec(w,x2,y2,l*sl,a,sl,da,n-1)
```

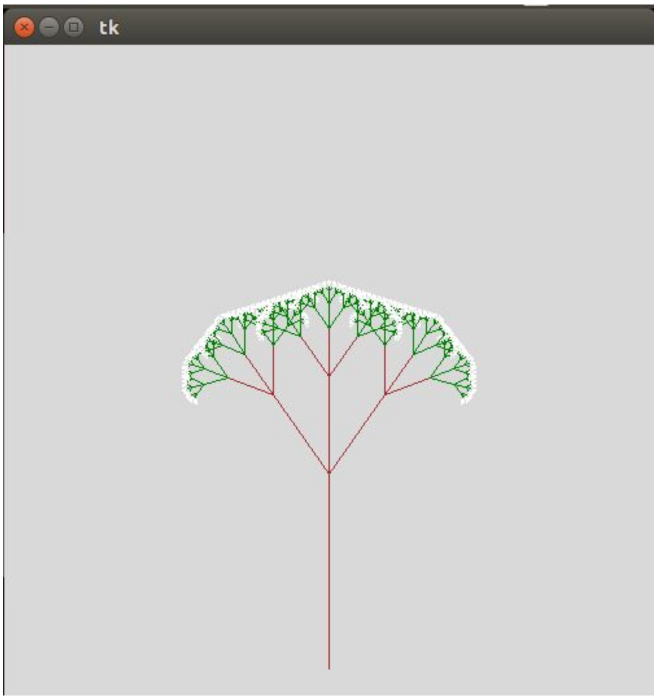


Figure 2. The fractals to make a tree

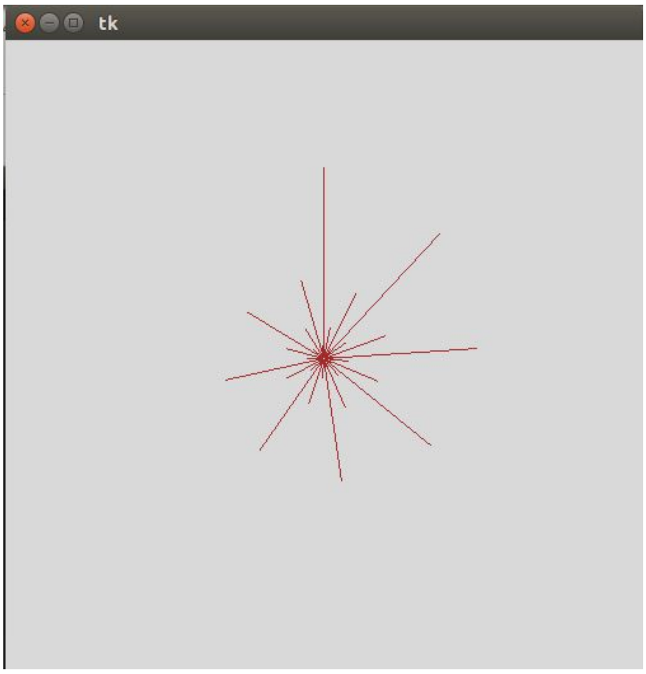
```
w = Canvas(master, width=xmax, height=y
w.pack()

##1
ec(w,250,480,150,180,0.5,358)

##2
```

Figure 3. The parameters to create figure 2

2. Star:

This is the recursive function to create this figure:

```
ec(w,x,y,l*sl,a-da,sl,da,n-1)
```




```
ec(w,250,250,150,180,0.90,43,100)
```

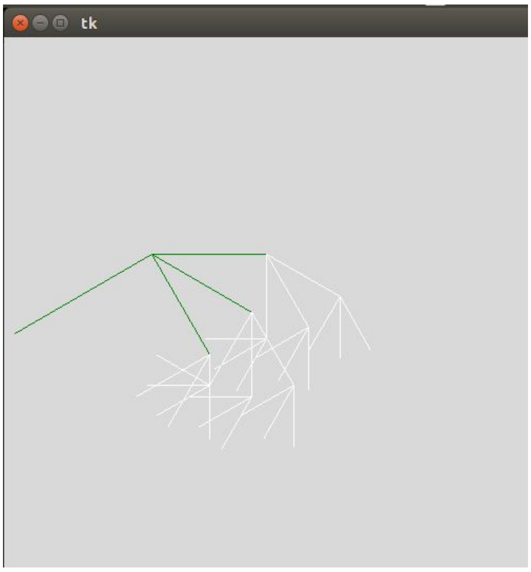
Figure 4. The fractals to make a star.

Figure 5. The parameters to create figure 4

3. Mountain:

This is the recursive function to create this figure:

```
if(y>280):return
ec(w,x2,y2,l*sl,a-da,sl,da,n-1)
ec(w,x2,y2,l*sl,a-2*da,sl,da,n-1)
ec(w,x2,y2,l*sl,a-3*da,sl,da,n-1)
```




```
ec(w,10,280,150,120,0.73,30,4)
```

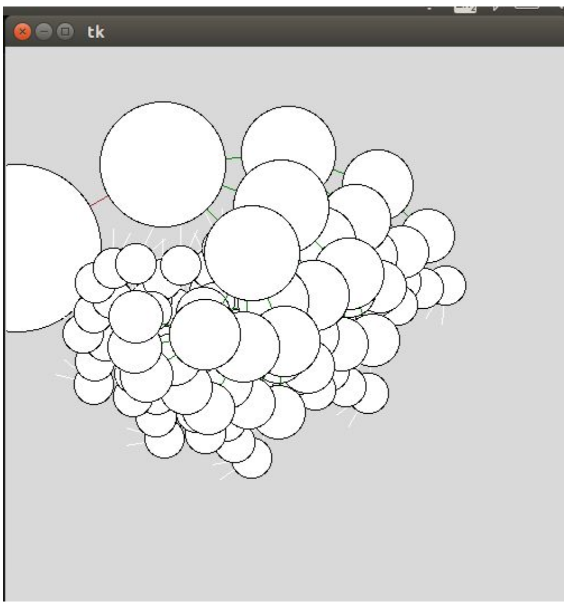
Figure 6. The fractals to make a mountain

Figure 7. The parameters to create figure 6

4. Cloud:

This is the recursive function to create this figure:

```
w.create_oval((x-l/2),(y-l/2),(x+l/2),(y+l/2),fill='white')
ec(w,x2,y2,l*sl,a-da,sl,da,n-1)
ec(w,x2,y2,l*sl,a-2*da,sl,da,n-1)
ec(w,x2,y2,l*sl,a-3*da,sl,da,n-1)
```




```
ec(w,10,180,150,120,0.75,25,6)
```

Figure 8. The fractals to make a cloud.

Figure 9. The parameters to create figure 8

5. Pentagonal spiral:

This is the recursive function to create this figure:

```
ec(w,x2,y2,l+sl,a+da,sl,da,n-1)
```

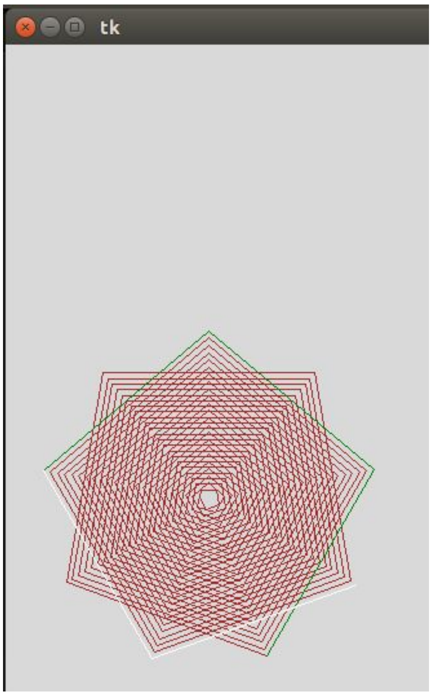


Figure 10. The fractals to make a pentagonal spiral.

```
ec(w,150,350,10,30,.8,80,200)
```

Figure 11. The parameters to create figure 10

The final landscape I use fractals to draw clouds, mountains, trees and a sun, I use the code of the reference.

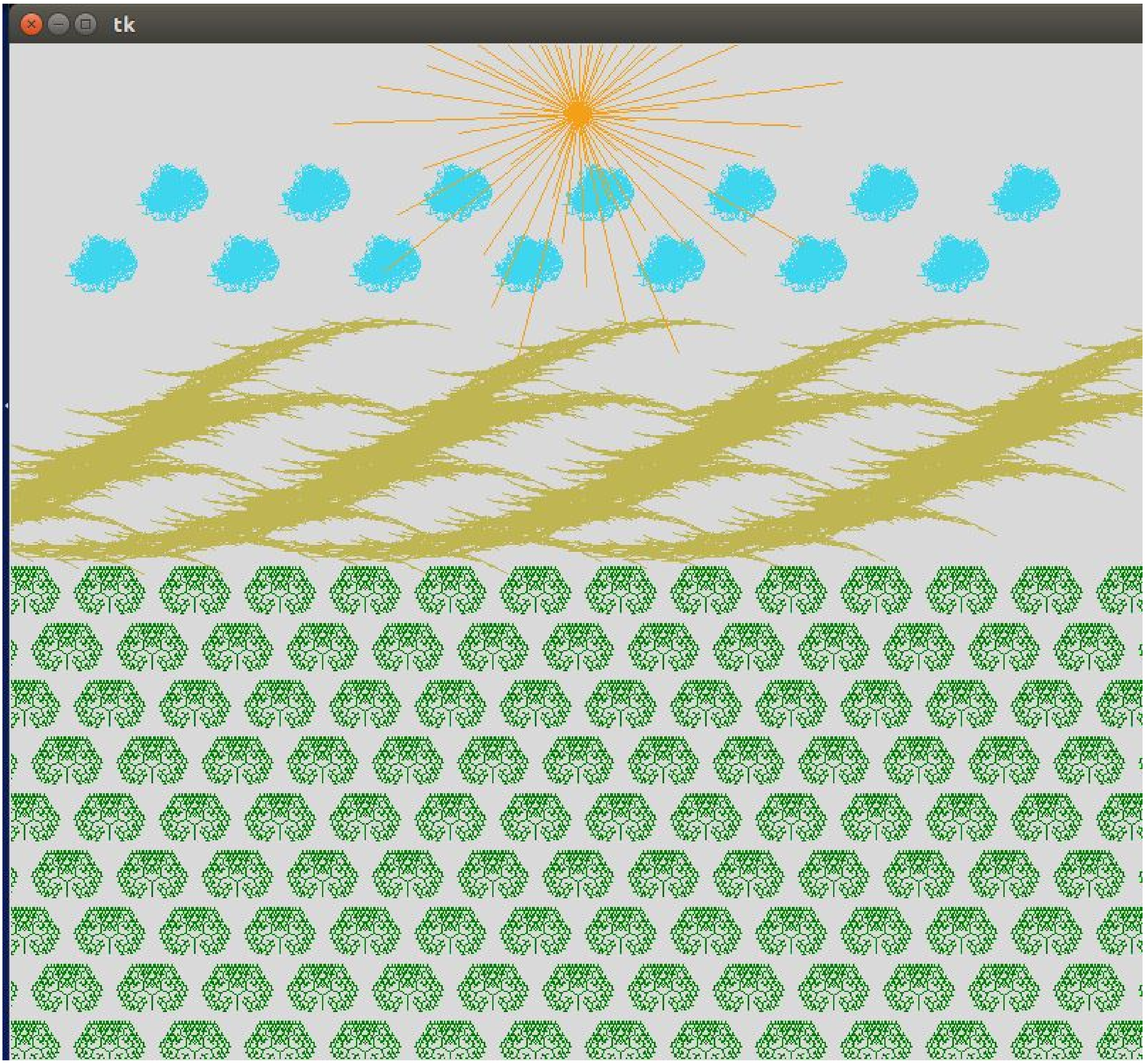


Figure 12. The landscape and I use 4 different type of fractals.

```
def Tree(w, x0, y0, l, an):
    ind = 1;
    if (l > ind ):
        x1 = x0 - (l * math.cos(an / 57.29578))
        y1 = y0 - (l* math.sin(an / 57.29578))
        w.create_line(x0, y0, x1, y1, fill = 'green')
        Tree(w, x1, y1, l / 1.3, an - 57)
        Tree(w, x1, y1, l / 1.3, an + 57)
    return

def Sun(w, x0, y0, l, an):
    if(l > 0):
        coseno = math.cos(an / 57.29578)
        seno = math.sin(an/57.29578)
        x1 = x0 - (l * coseno)
        y1 = y0 - (l * seno)
        w.create_line(x0, y0, x1, y1, fill='#f39f18')
        l -= 3
        an += 37
        Sun(w, x0, y0, l, an)
    return

def Mountain(w, x0, y0, l, an, ind):
    X = [0, 0]
    if ind > 0:
        Straight(w, x0, y0, l, an, X)
        Mountain(w, X[0], X[1], l / 1.2, an + 6, ind - 1)
        Mountain(w, X[0], X[1], l / 1.55, an + 172, ind - 1)
        Mountain(w, X[0], X[1], l / 1.8, an + 186, ind - 1)

def Straight(w, x0, y0, l, an, X):
    X[0] = x0 - (l * math.cos(an / 57.29578))
    X[1] = y0 - (l * math.sin(an / 57.29578))
    w.create_line(x0, y0, X[0], X[1], fill = '#bfb552')

    return

def Cloud(w, x0, y0, l, an, ind):
    if (ind > 0):
        coseno = math.cos(an / 57.29578)
        seno = math.sin(an / 57.29578)
        x1 = x0- (l * coseno)
        y1 = y0 - (l * seno)
        w.create_line(x0, y0, x1, y1, fill = '#3ed6ef')
        Cloud(w, x0, y0, l / 1.25, an + 30, ind - 1)
        Cloud(w, x1, y1, l / 1.3, an + 50, ind - 1)
        Cloud(w, x1, y1, l / 1.4, an + 100, ind - 1)
    return
```

Figure 13. The code to draw all the figures in the landscape.

Conclusions:

In this practice I can learn about Fractals, I didn’t have idea about these, but I can see that it is import the environment of geometry because you can draw some figures and they can be similar to the reality also you can programmer these figures with few lines of code because the most import thing to make them it’s the recursivity and this is very import in our environment as programmers, and I research some things about this topic a lot of company of videogames or simulates to make images in 3D some people use them to draw very nice in their programs, I can notice that it’s very slow to run the program because it needs to search in all the ways to draw the image, finally this practice was very interested for me, because I didn’t know about graphics and this topic it helps me to have knowledge about this environment and the different ways to draw a thing.

Bibliography

Chapter XXVIII Fractal Geometry and Computer Science Nicoletta Sala Università della Svizzera Italiana, Switzerland Università dell'Insubria, Italy

http://www.irma-international.org/viewtitle/28736/

Fractals in nature and applications, Tino Kluge

http://kluge.in-chemnitz.de/documents/fractal/node2.html