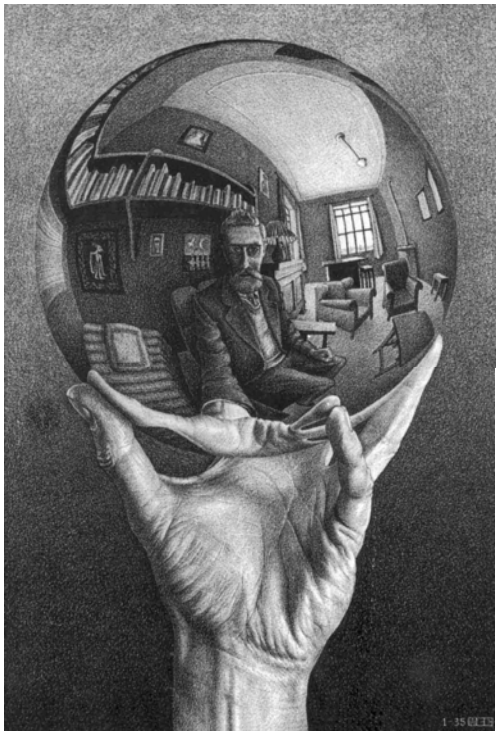
“An Intersection of Art, Biology, Ethics, and Computer Science,"

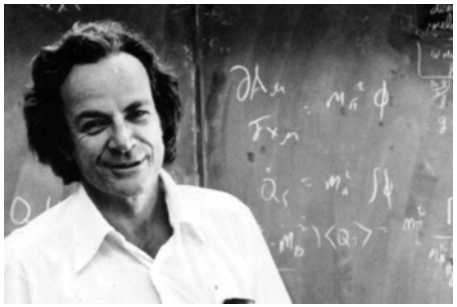
Gustavo Rodriguez-Rivera
Computer Science
Purdue University



Computer as a Mirror

- Computers are not only used to
  - Execute things fast.
  - Send e-mail
  - Browse the web
  - Listen to music
  - Watch Youtube
  - ….
- They also can be used as a tool to understand ourselves and the world around us.

Using the Computer as a Tool to Understand Ourselves



"What I cannot create, I do not understand. " - Richard Feynman

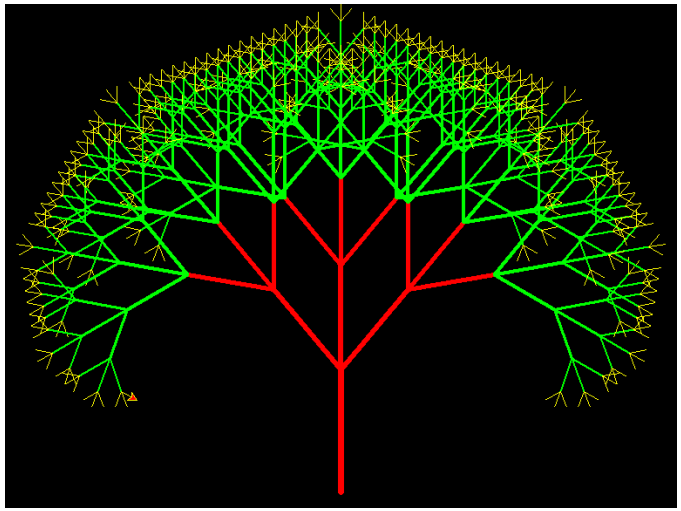
- If we think we know how something works, we can write a program to simulate it and to predict its behavior.
- If the program accurately predicts the behavior, then our model works.
- The computer can be used as a tool to understand our world and ourselves.
- That is why computer programming is a important skill to have for all science majors.

Three examples of how Computers can Help Understand Science

- ART and CS
  - Fractals
- Biology and CS
  - Life and Cellular Automatons
- Ethics and CS
  - Cooperation

ART and CS: Fractals

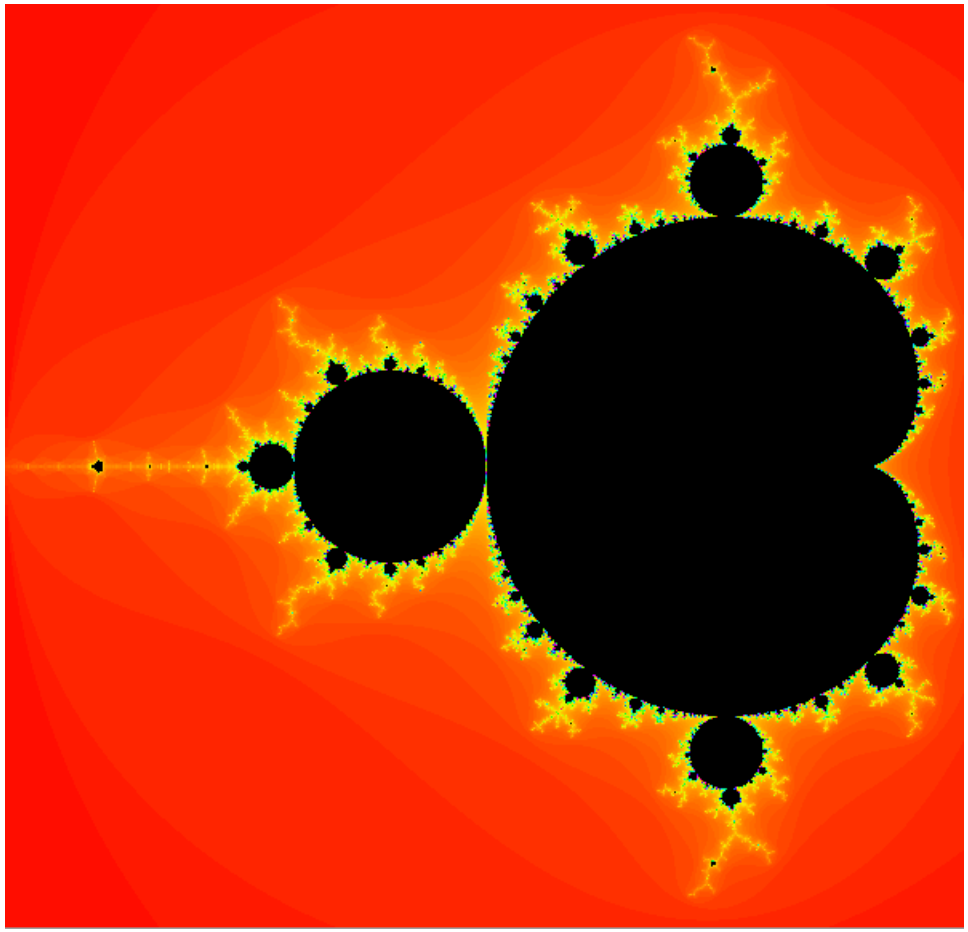
- They are patterns that are formed by repeating a simple process over and over again.
- Fractals have a long history but the term war created by Benoit Mandelbrot.



We can find fractals all over the world:
plants, seashells, trees, coastlines etc.

Adapted from Fractal Foundation
http://fractalfoundation.org/OFC/OFC-index.htm

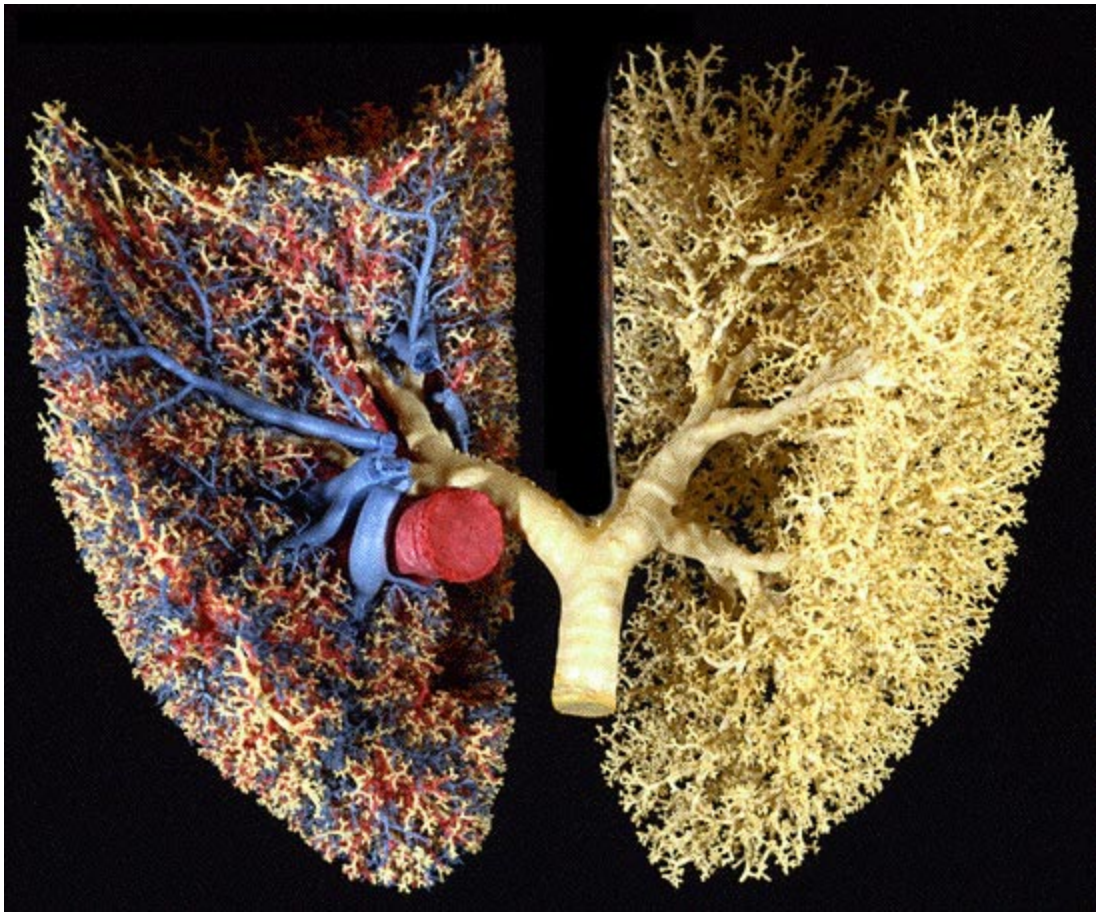
Also fractals can be generated by the computer



Branching Fractals

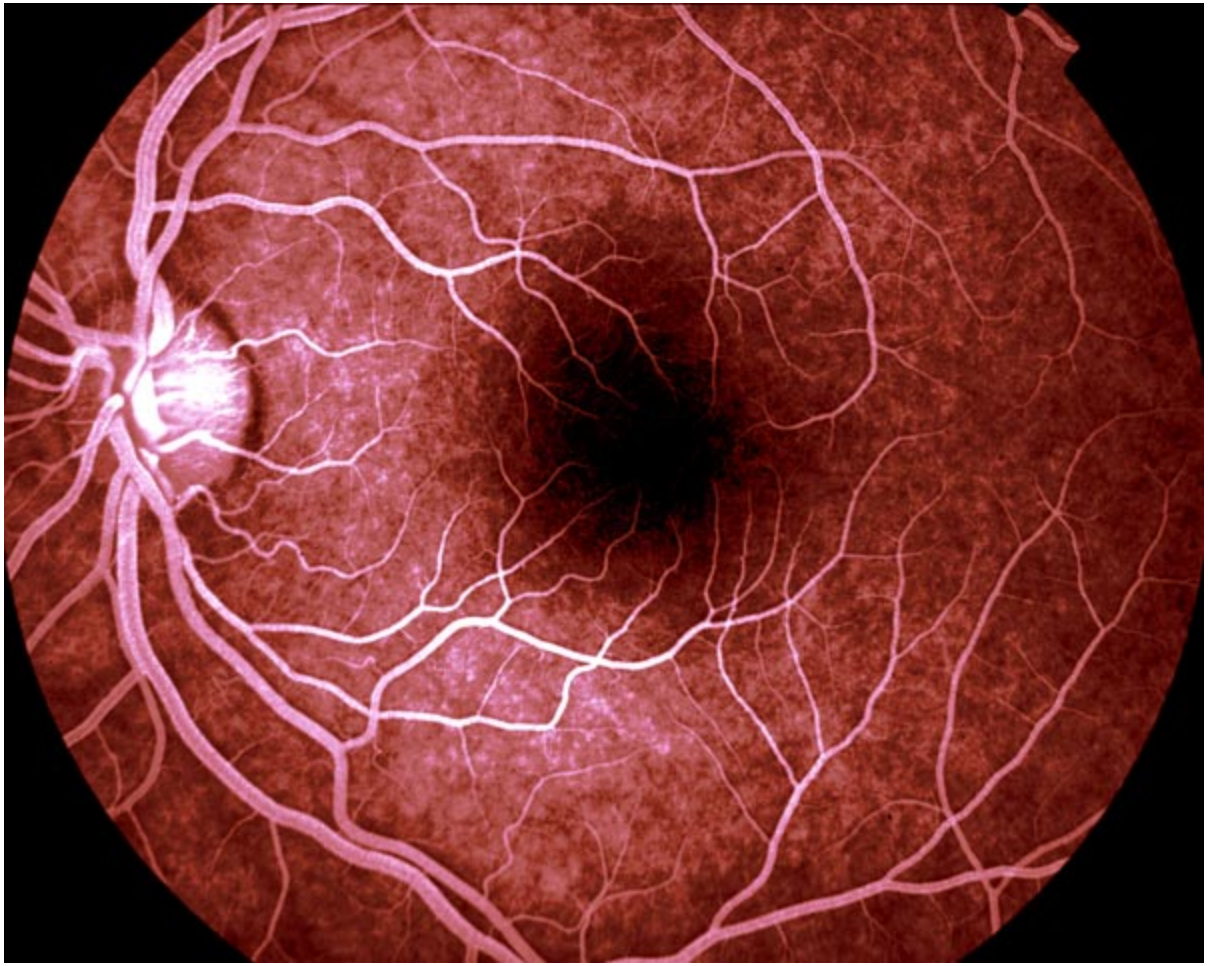
Fractals in the Body

Human lungs, showing blood vessels



Fractals in the Body

- Human retina



Fractals in Rivers

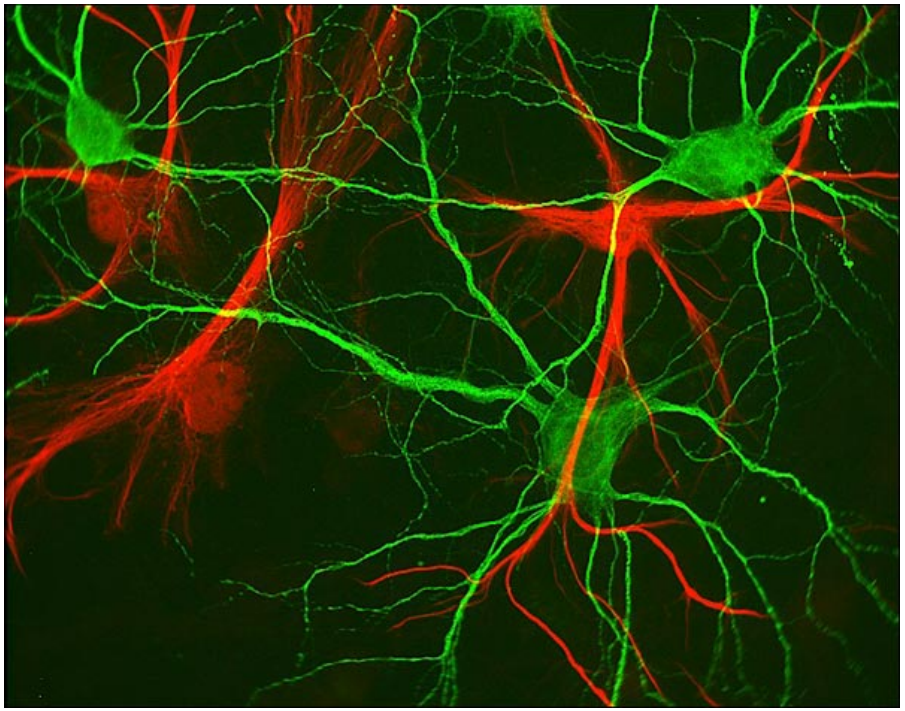
- Grand Canyon

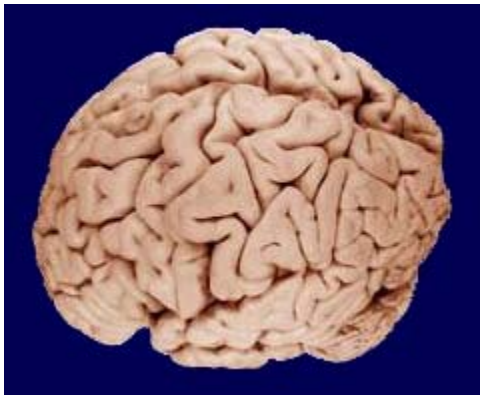
Fractals in Lightning

Lightning over Albuquerque, New Mexico

Fractals Neurons

- Hippocampal neurons (green) and glial cells (red).





Spirals in Nature

- M51 the "Whirlpool Galaxy". Scale approximately 100,000 light years.

Spirals in Nature

- Hurricane Katrina, 2005.

Spirals in Nature

- Agave Cactus.

Spirals in Nature

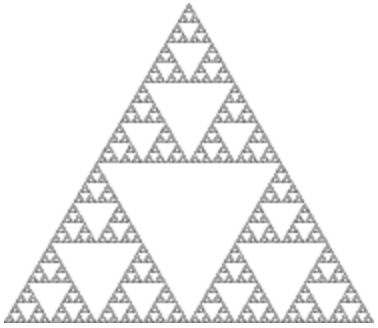
- Sunflower (Helianthus annuus).

Spirals in Nature

- Nautilus shell.

Geometric Fractals

- Sierpinski Fractals

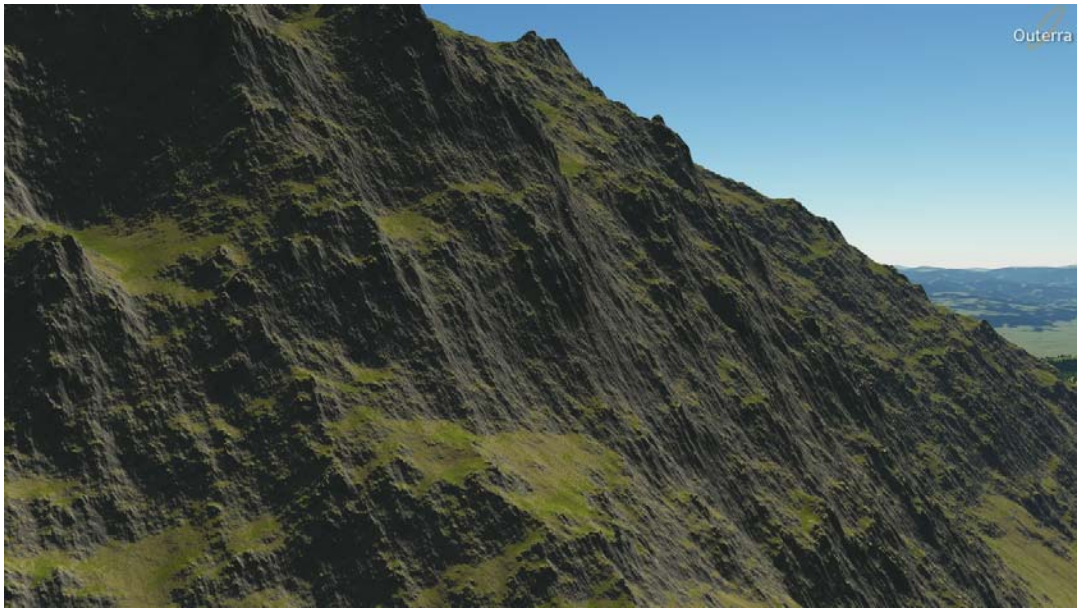


Fractals in Minecraft

- Minecraft uses fractals extensively to generate worlds

Fractals in Computer Graphics

- Programmers use Fractals in Computer Graphics because with few lines of code they can generate images that look very real.



Fractals in Computer Graphics

Another fractal landscape

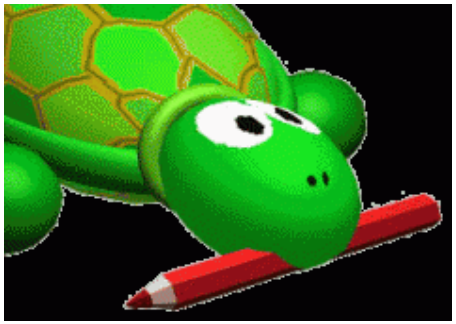
Program Your own Fractal

- We will use Python and Turtle Graphics
- Go to http://python.org to get python for free

- The 2.9 billion base pairs of the haploid human genome
- Each base pair can be represented with 2 bits.
- That corresponds to a maximum of about 725MB
- Many of the tissues in our body follow a fractal pattern. That allows having a short encoding.

Turtle Graphics

- There is a turtle with a pen



- You have the following instructions:

  color(“green”) – Set pen color to green

  forward(200) – Move forward 200 units

  backward(200) – Move backward 200 units

Turtle Graphics

- Other instructions:

  left(90) – Turn left 90 degrees

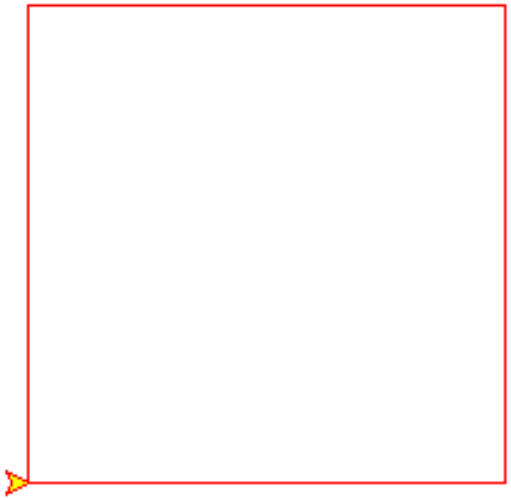
  right(90) – Turn right 90 degrees

  up() – Pen up

  down() – Pen down

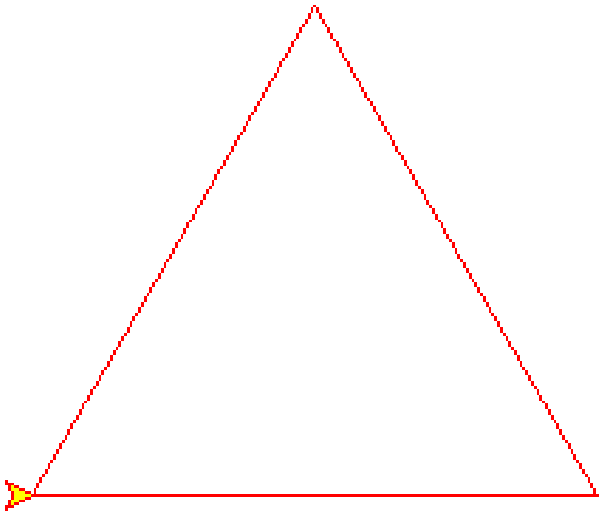
Draw a square

```
from turtle import *

color('red', 'yellow')

forward(200)
left(90)

forward(200)
left(90)

forward(200)
left(90)

forward(200)
left(90)

done()
```



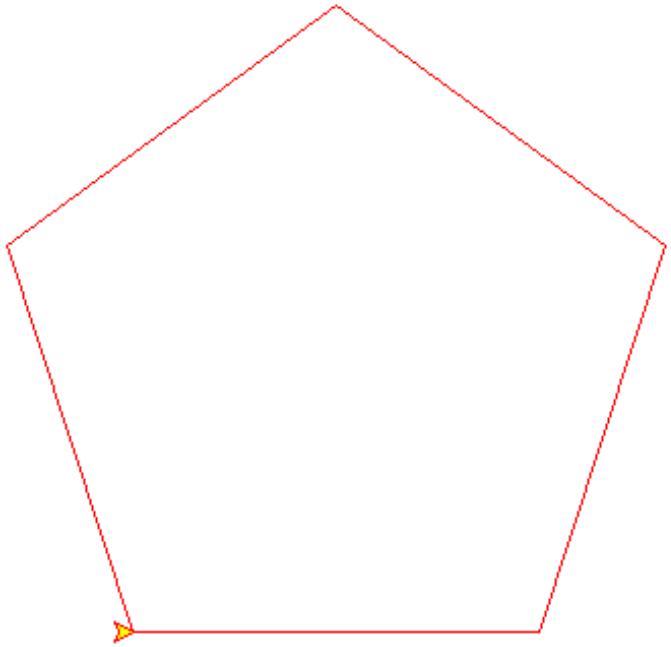
Draw a triangle

```
from turtle import *

color('red', 'yellow')

forward(200)
left(120)

forward(200)
left(120)

forward(200)
left(120)

done()
```



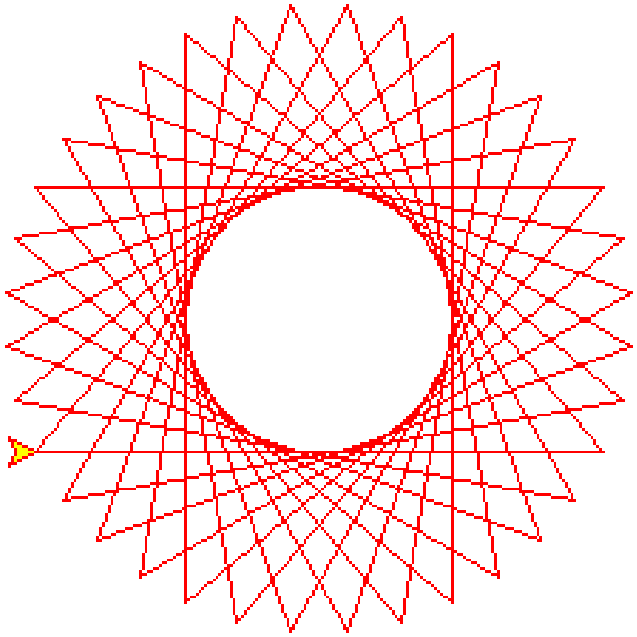
Draw a polygon with n sides

```
from turtle import

def polygon(n):
  color('red', 'yello
  for i in range(n):
      forward(200)
      left(360/n)

polygon(5)

done()
```



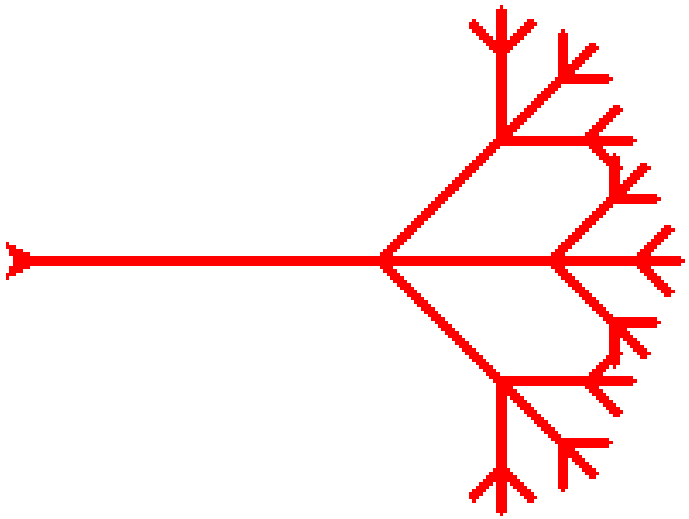
Draw a star

```
from turtle imp

color('red', 'yell
while True:
    forward(200
    left(130)
    if abs(pos())
        break

done()
```



Draw a Tree

```
from turtle import

def mytree(n):

    if (n < 10): retu

    forward(n)

    left(45)
    mytree(n/2)

    right(45)
    mytree(n/2)

    right(45)
    mytree(n/2)

    left(45)
    backward(n)

color(“green")
pensize(3)
mytree(100)

done()
```



Laplace Prediction and Determinism

"An intelligence knowing all the forces acting in nature at a given instant, as well as the momentary positions of all things in the universe, would be able to comprehend in one single formula the motions of the largest bodies as well as the lightest atoms in the world, …. " (Laplace 1820)

The Universe, Life, and Cellular Automatons

- We can think of the universe as a computer program.
- The current state of the universe at time ***t*** is stored as variables in the computer.
- The program includes the laws of physics that can predict the universe at time ***t+delta***.
- *Problems*:
  - The universe is analog. But we can approximately digitally with some error.
  - The universe is too complex and large.
  - We will need a ***very*** large and ***powerful*** computer.
  - Lots of simultaneous differential equations solved numerically.

Simplifying the Universe to One Dimension

- Like everything in Science

  "All models are wrong; some models are useful."

  (George Box 1987)

- We will try to simplify our universe to a One-Dimension binary universe and see if we get something ***useful***.

One-Dimension Cellular Automatons

- Stephen Wolfram in his book “A New Kind of Science” proposes the simplest class of one-dimensional cellular automaton.
  - ***http://mathworld.wolfram.com/ElementaryCellularAutomaton.html***

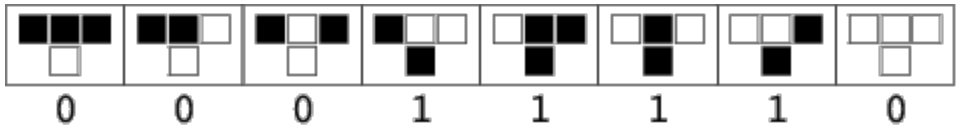
- The initial “Universe” (The One-Dimensional Big Bang ☺) is represented as a string of 0’s with only a 1 in the middle.
  - **‘ ‘ = 0**
  - **# = 1**
- Example:

00000000000000010000000000000000

“ # “

Rules

- There are rules that depend only on nearest neighbor to produce the next string.

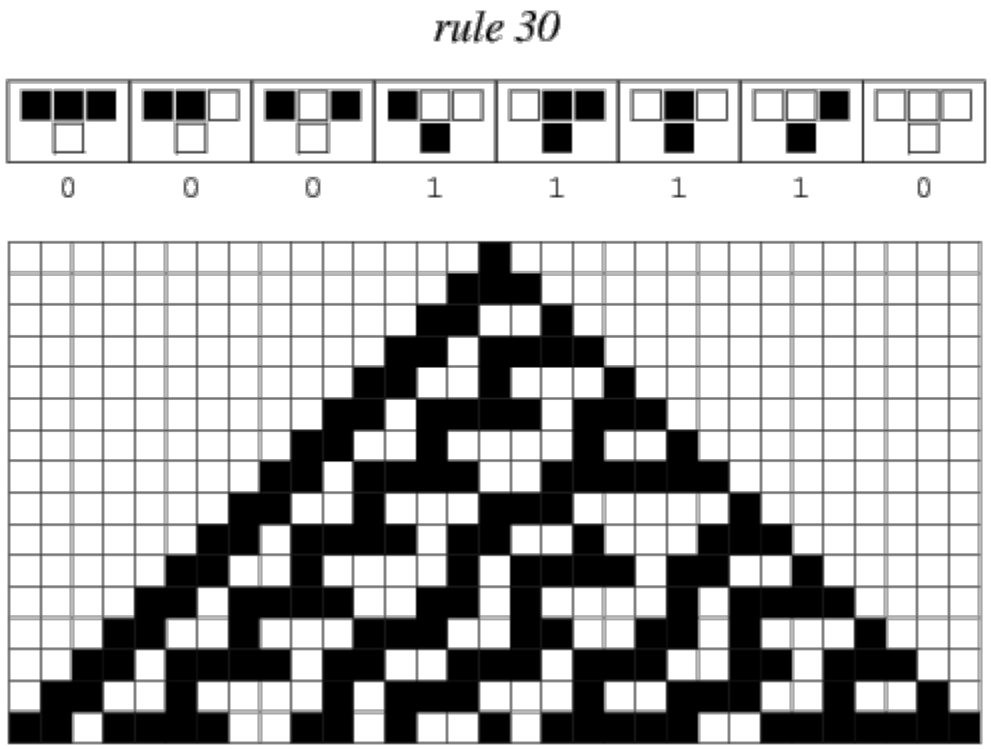


- Example:

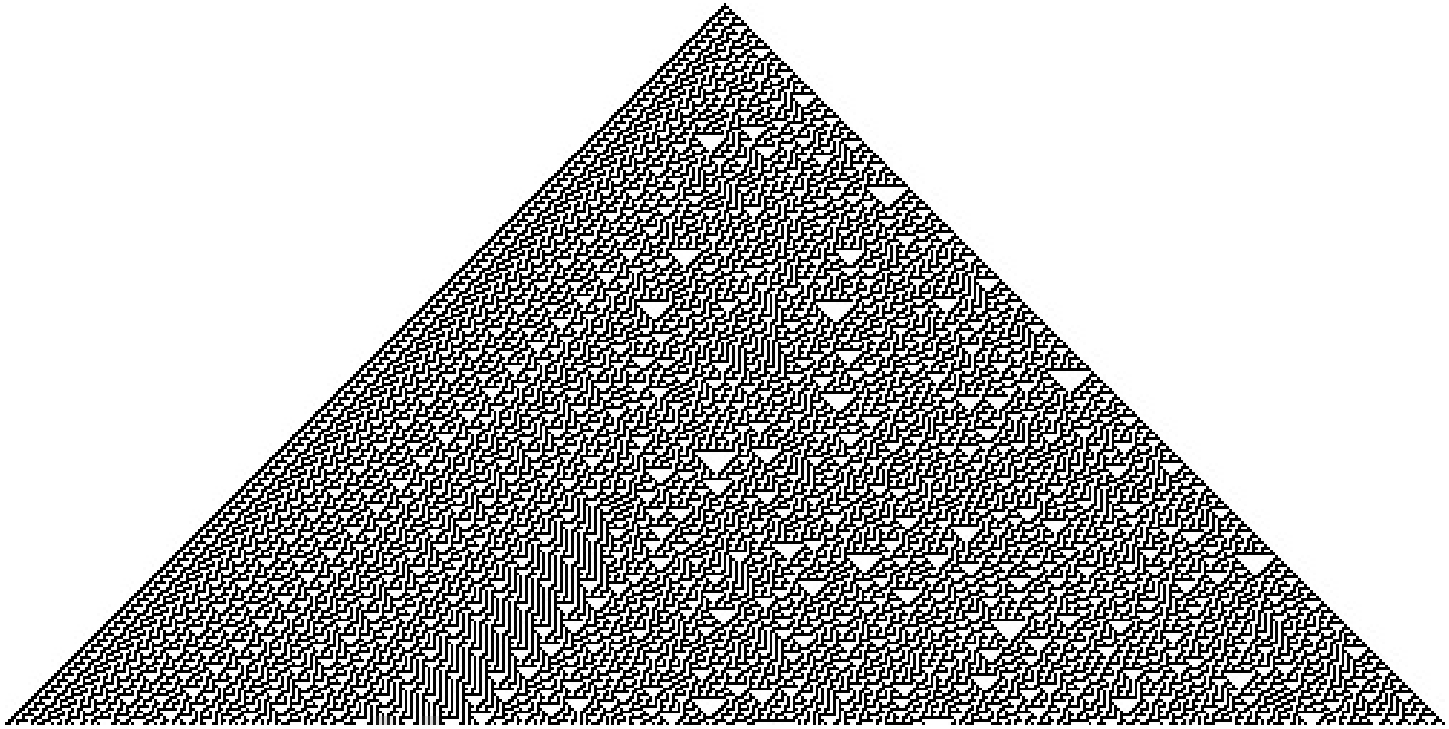
```
Time 0: “                  #                  “
Time 1: “                 ###                 “
Time 3: “                ##  #                “
...
```

- **Rules can be encoded in 8 bits : 1 byte.**
- **There are 2^8 = 256 different rules.**

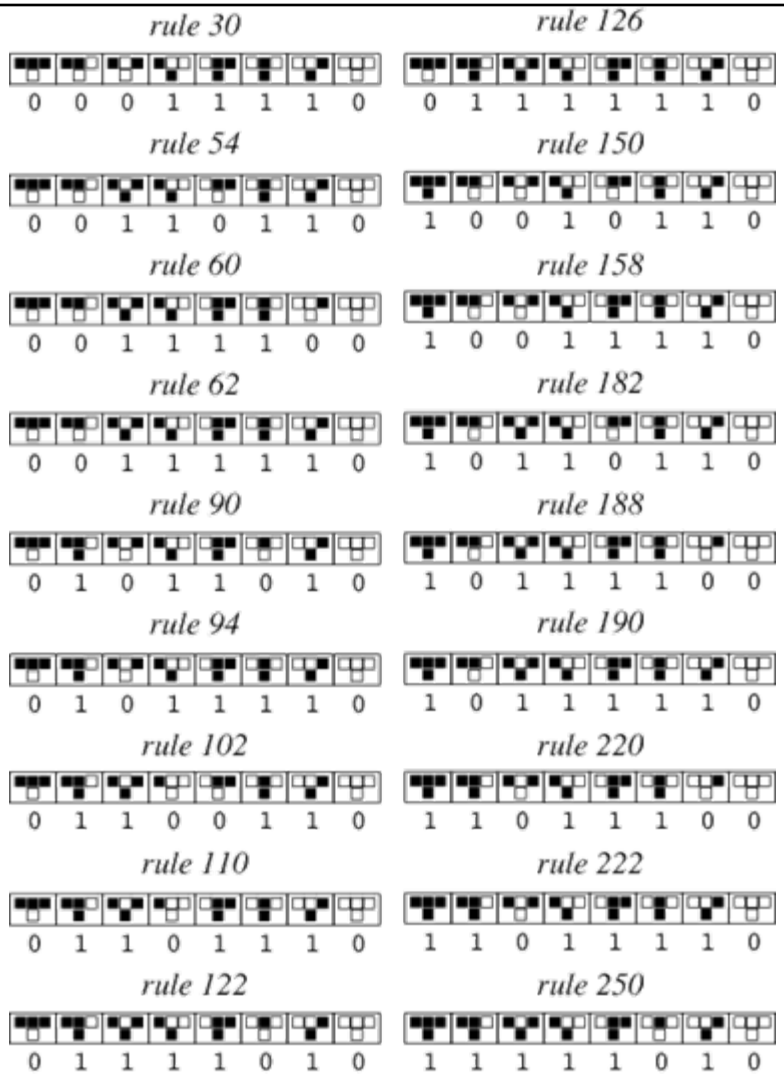
Applying the rules further



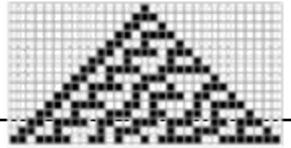
And further…



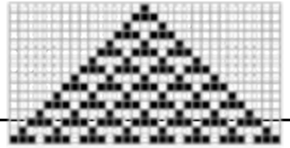
Interesting Rules



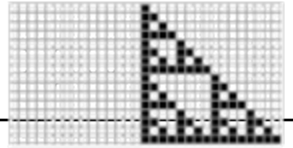
rule 30



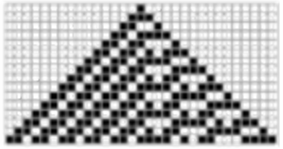
rule 54



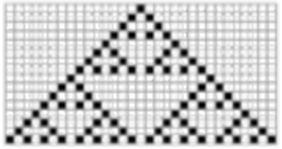
rule 60



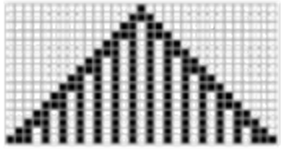
rule 62



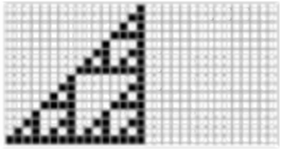
rule 90



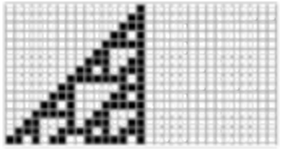
rule 94



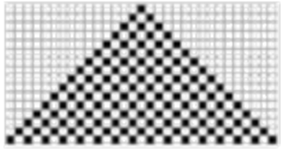
rule 102



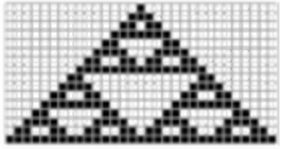
rule 110



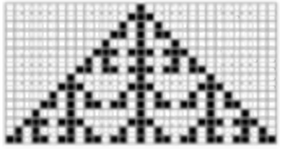
rule 122



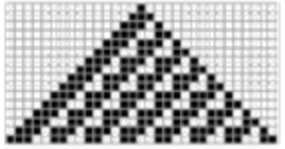
rule 126



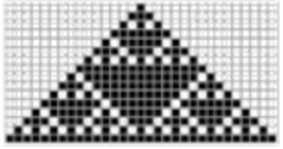
rule 150



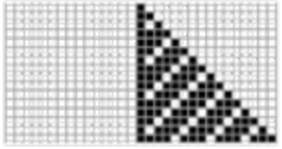
rule 158



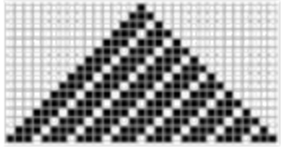
rule 182



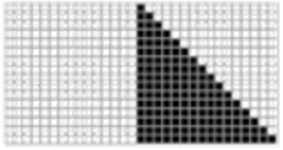
rule 188



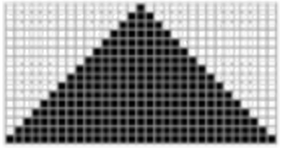
rule 190



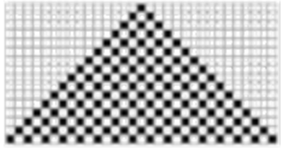
rule 220



rule 222



rule 250



Life and Cellular Automatons

- What about Life?
- We can extend the 1D Cellular Automatons to 2D.
- Conway's Game of Life:
  - Conway's game of life in JavaScript
- Life is a complicated system but we can model it with simple models and get interesting results.

Ethics and Computers- The Prisoner's Dilemma

- The Prisoners Dilemma is an example of game theory that demonstrates when cooperation may be good.
- It was originally framed by Merrill Flood and Melvin Dresher
- We used here a variation of the original dilemma use something more familiar like roommates washing dishes (As described in: Minotauromachia Journal).

Game Theory of Washing Dishes

- Ben and Jack are roommates
- They have two choices: washing dishes or not.
- The possible outcomes are:

1. Ben will wash the dishes, Jack will do nothing (Y,N)

2. Ben will do nothing, Jack will wash the dishes (N,Y)

3. Both will wash the dishes together (Y,Y)

4. Nobody will wash the dishes (N,N)

Outcome Payoff Matrix

Table of Points based on the decision of each player

Points indicate the “benefits” of the decision

BEN / JACK	WASH	NO WASH
WASH	2,2	0,3
NO WASH	3,0	1,1

What is the Best Choice?

- If I am Ben what is the best choice? (Assuming that Jack has the same probability choosing cooperate and no cooperate.)
  - Average Points for Ben if cooperates =
    (2 + 0) / 2 = 1
  - Average Points for Ben if no cooperate =
    (3+1) / 2 = 2
- Therefore, assuming that Jack has the same probability to cooperate and not to cooperate it is better not to cooperate.
- Most likely Jack will make a similar choice not to cooperate and both will get 1,1.
- This could be a typical game if both were roommates only for one day.

Iterated Prisoner's Dilemma

- Roommates will see each other today, tomorrow, many weeks, months and maybe years.
- The move of the other player yesterday may affect the move of the player today.
- Cooperation starts.
- If both cooperate they will get 2 points each instead of only 1.

Strategies:

- Slacker: Never wash dishes
- Naive: Wash the dishes no matter what the other roomate does.
- Random: Make a random move
- Tit for Tat: Start washing dishes and then do what the other player did before.
- Unforgiving: Wash dishes first and do not wash dishes after the first time the other roomate does not wash them.
- Many more…

The Evolution of Cooperation

- Robert Axelrod in his book “The Evolution of Cooperation” organized a tournament with a N steps prisoner’s dilemma.
- Participants submitted programs implementing their own strategy.
- Tournament
- Winning Strategy was “Tit-For-Tat”.

Characteristics of Tit-For-Tat

- **Be nice**: cooperate, never be the first to defect.
- **Be provocable**: return defection for defection, cooperation for cooperation.
- **Don't be envious**: be fair with your partner.
- **Don't be too clever**: or, don't try to be tricky. Be predictable.

Cooperation is good for you and your roomate.

Conclusions

- A computer is a great tool to help us understand ourselves.
- Simple models may not be complete but may be helpful to understand big systems.
- Learning how to program is a skill useful for everybody and not only for computer scientists.