Example: Fractals

Object Oriented Programming
2024 First Semester
Shin-chi Tadaki (Saga University)

Purposes of this example

- Creating instances of the same class
- Instances have slightly different parameters
- Creating an instance when it is necessary

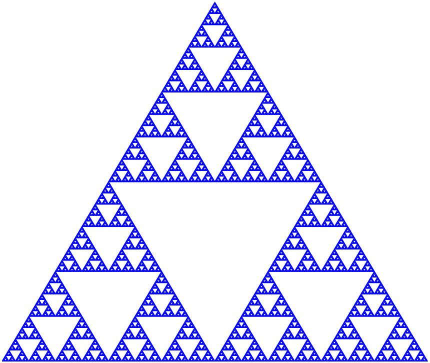
`https://github.com/oop-mc-saga/Fractal`

Fractals

- Complex shapes or structure with self similarity
- Self similarity: any parts are similar to the whole
  - exactly: *scale invariant*
  - statistically
- *Affine fractals* are mathematical objects defined by a set of affine transformations
- Examples in the nature

Sierpinski gasket

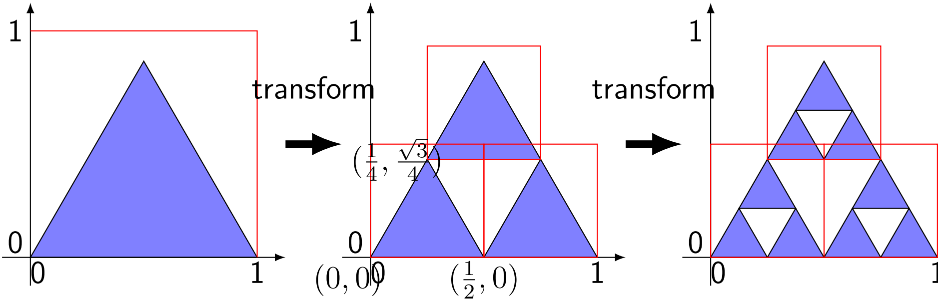
- Triangles similar to the whole repeatedly appear



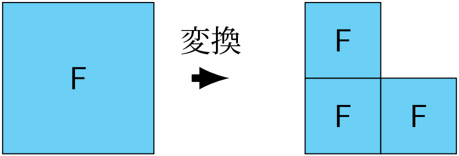
Trasformation for Sierpinski gasket

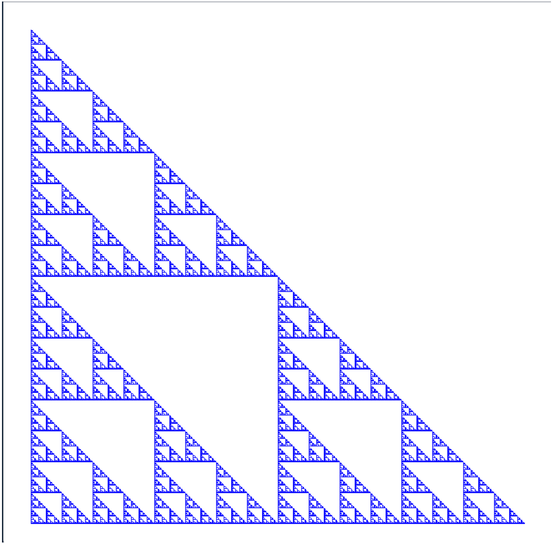
- Repeating to hollow the center from the triangle

Another Transformation for Sierpinski gasket

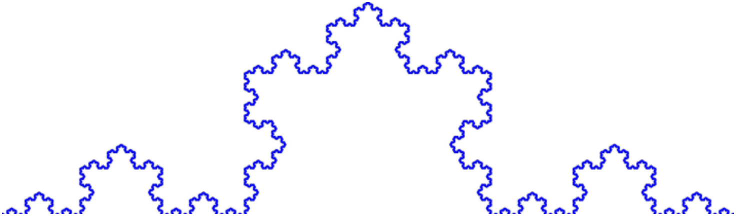


Sierpinski-like fractal

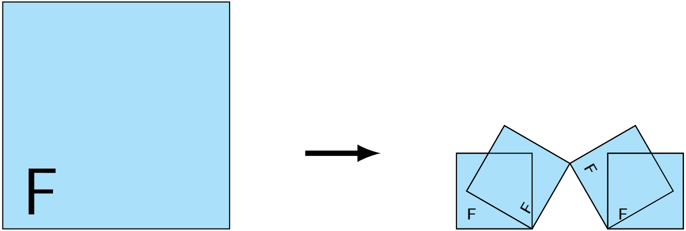




Koch Curve



Transformation for Koch curve

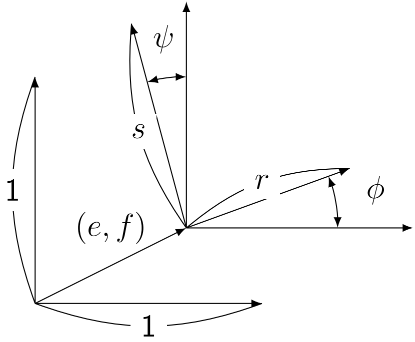


Affine transformations

- Representing geometrical operations such as
  - translations
  - rotations
  - shearing

Two dimensional case

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} r\cos\phi & -s\sin\psi \\ r\sin\phi & s\cos\psi \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} \tag{3.1}$$



Another expression

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} r\cos\phi & -s\sin\psi & e \\ r\sin\phi & s\cos\psi & f \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{3.2}$$

Affine trasformations for Sierpinski gasket

$$w_0 = \begin{pmatrix} 1/2 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.3}$$

$$w_1 = \begin{pmatrix} 1/2 & 0 & 1/4 \\ 0 & 1/2 & \sqrt{3}/4 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.4}$$

$$w_2 = \begin{pmatrix} 1/2 & 0 & 1/2 \\ 0 & 1/2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.5}$$

Sierpinski gasket defined by affine transformations

- Consider a set P_0 of points $x \in [0, 1]$ and $y \in [0, 1]$.
- All points in P_0 are transformed by w_0, w_1, and w_2.

$$P_1 = w_0(P_0) \cup w_1(P_0) \cup w_2(P_0) \tag{3.6}$$

- The transformations are applied repeatedly.

$$P_{n+1} = w_0(P_n) \cup w_1(P_n) \cup w_2(P_n) \tag{3.7}$$

- Sierpinski gasket is the limit of P_n as $n \to \infty$.

Affine transformations for Koch curve

$$w_0 = \begin{pmatrix} r & 0 & 0 \\ 0 & r & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.8}$$

$$w_1 = \begin{pmatrix} r\cos\phi & -r\sin\phi & r \\ r\sin\phi & r\cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.9}$$

$$w_2 = \begin{pmatrix} r\cos\phi & r\sin\phi & 1/2 \\ -r\sin\phi & r\cos\phi & r\sin\phi \\ 0 & 0 & 1 \end{pmatrix} \tag{3.10}$$

$$w_3 = \begin{pmatrix} r & 0 & 2r \\ 0 & r & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.11}$$

$$\phi = \pi/3 \qquad r = 1/3 \tag{3.12}$$

Affine trasformation in Java

- `java.awt.geom.AffineTransform` class

```
AffineTransform(double m00, double m10,
    double m01, double m11,
    double m02, double m12)
```

- Corresponding mathematical expression

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} m_{00} & m_{01} & m_{02} \\ m_{10} & m_{11} & m_{12} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{3.13}$$

Polygons in Java

- `java.awt.Shape` interface represents a geometric shape
- `Path2D.Double` class represents a geometric path
    - `moveTo()`: adds a point to the path by moving
    - `lineTo()`: adds a point to the path by drawing a straight line
    - `closePath()`: closes the current subpath
- `Path2D.Double` class implements `Shape` interface
    - `Path2D.Double` class instance can be used as a shape

Transforming shapes

- `AffineTransform.createTransformedShape(Shape s)` returns transformed shape

Class Outline

- model package
  - Fractal class
  - FractalFactory class
- gui package
  - FractalMain class
  - DrawPanel class

Fractal class

- Initialized by specifying a set of affine transformations
- The initial shape is a square of unit length
- Applies transformations on the current set of shapes
- Shows the set of affine transformations

```
public Set<Shape> transform(Set<Shape> shapeList) {
    Set<Shape> newShapes = new HashSet<>();
    for (AffineTransform af : transformations) {
        for (Shape s : shapeList) {
            Shape t = af.createTransformedShape(s);
            newShapes.add(t);
        }
    }
    return newShapes;
}
```

`FractalFactory` class

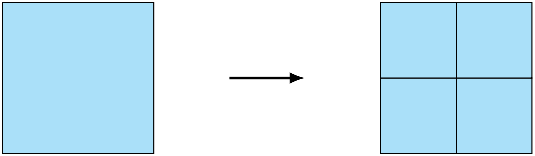
- It is not easy to define fractals by giving a set of transformations
- This class returns `Fractal` instances for some predefined cases.
  - Fractals are defined by the set of transformations

```java
public static Fractal createInstance(FractalName fractalName) {
    Set<AffineTransform> affineList = new HashSet<>();

    switch (fractalName) {
        case Sierpinski -> {
            double r = 1. / 2.;
            affineList.add(createTransformation(r, r, 0, 0, 0, 0));
            affineList.add(createTransformation(r, r, 0, 0, r, 0));
            affineList.add(createTransformation(r, r, 0, 0, 1. / 4,
                    Math.sqrt(4) / 4));
        }

    ....

        default -> {
            affineList.add(new AffineTransform());
        }
    }
    return new Fractal(affineList);
}
```

Topological Dimensions

- *Topological dimension* is a concept to measure the size of a set
- The dimension of a line is 1. The size is its length.
- The dimension of a square is 2. The size is its area.
- The dimension of a cube is 3. The size is its volume.

- The *volume* of D dimensional set:
  - If the scale for measurement is changed from 1 to r, the volume changes by a factor of r^{-D}.



$$r = 1/2 \Rightarrow N = 4$$

Strange features of fractals

- The area of Sierpinski gasket.
  - Every step reduces the area to $3/4$.
  - It goes to zero at ∞ steps.
- The length of Koch curve.
  - Every step increases the length to $4/3$.
  - It goes to ∞ at ∞ steps.

Fractal Dimensions

- Topological dimension: By changing scale from 1 to r, the volume changes $N = r^{-D}$

$$D = -\frac{\ln N}{\ln r} \tag{6.1}$$

- Sierpinski Gasket: $N = 3$, $r = 1/2$

$$D = \ln 3/\ln 2 = 1.585... \tag{6.2}$$

- Koch curve: $N = 4$, $r = 1/3$

$$D = \ln 4/\ln 3 = 1.261... \tag{6.3}$$