

SCHEDA TECNICA

FRATTALI

Un frattale è un oggetto geometrico dotato di omotetia interna: si ripete nella sua forma allo stesso modo su scale diverse, quindi ingrandendo una qualunque sua parte si ottiene una figura simile all'originale.

La geometria frattale, è quindi la geometria che studia queste strutture, ricorrenti ad esempio nella progettazione ingegneristica di reti, nel moto browniano e nelle galassie.

Triangolo di Sierpinski



In questo programma andiamo a disegnare il triangolo di Sierpinski in base al numero di interazioni, ossia i “livello” dato dai click del mouse.

Andiamo a creare la funzione frattale, che si servirà di coordinate per andare a creare successivamente gli altri triangoli.

```
frattale(new PVector(0, 600), new PVector(800, 600), new PVector(400, 0), click, 0);
```

a questo punto ogni volta che il mouse viene cliccato il frattale inizia a generarsi, in un procedimento basato su livelli di numero crescente:

- Livello 0** Si parte da un triangolo equilatero, con i tre vertici tangenti ai bordi della finestra, e che ha il lato uguale alla lunghezza della finestra, che per comodità indicheremo con A.
- Livello 1** Si congiungono i punti medi di ciascun lato individuando tre triangoli simili al primo di lato $A/2$.
- Livello 2** Si ripete l'operazione di scomposizione precedente su ciascuno dei tre triangoli ottenendo 9 triangolini di lato $a/4$.
- Livello 3** Si ripete la stessa operazione sui 9 triangoli ottenendone 27 di lato $a/8$.
- Livello 4** Si ripete la stessa operazione sui 27 triangoli ottenendone 81 di lato $a/16$.
- Livello 5** Si ripete la stessa operazione su 81 triangoli ottenendone 243 di lato $a/32$.
- Livello 6** Si ripete la stessa operazione su 243 triangoli ottenendone 729 di lato $a/64$.
- Livello 7** Si ripete la stessa operazione su 729 triangoli ottenendone 2187 di lato $a/128$.

```

void frattale(PVector lowerLeft, PVector lowerRight, PVector topCenter, int maxClick, int curClick) {
  // curClick=0 maxClick=1 --> solo per la prima volta

  if (curClick < maxClick) { // curClick e maxClick sono stati dichiarati alla riga precedente

    // Il punto centrale del lato sinistro del triangolo
    PVector middleLeft = new PVector(lowerLeft.x + (topCenter.x - lowerLeft.x) / 2, lowerLeft.y + (topCenter.y - lowerLeft.y) / 2); // x = 200, y = 300

    // Il punto centrale del lato destro del triangolo
    PVector middleRight = new PVector(lowerRight.x + (topCenter.x - lowerRight.x) / 2, lowerRight.y + (topCenter.y - lowerRight.y) / 2); // x = 600, y = 500

    // Il punto centrale del lato basso del triangolo
    PVector bottomCenter = new PVector(lowerLeft.x + (lowerRight.x - lowerLeft.x) / 2, lowerLeft.y); // x = 400, y = 600

    int newClick = curClick+1;

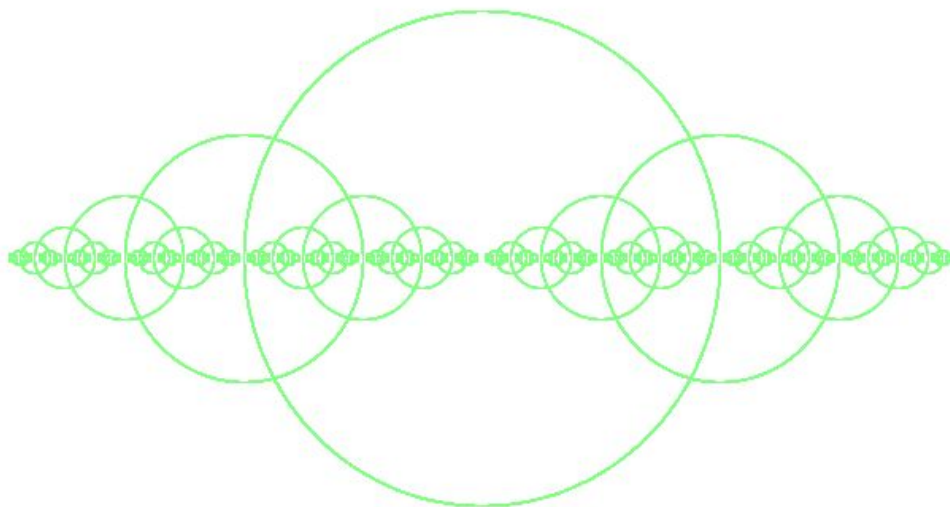
    // chiamata ricorsiva della funzione frattale per ognuno dei tre nuovi triangoli
    frattale(middleLeft, middleRight, topCenter, maxClick, newClick); // tringolo in alto centrale
    frattale(lowerLeft, bottomCenter, middleLeft, maxClick, newClick); // triangolo in basso a sinistra
    frattale(bottomCenter, lowerRight, middleRight, maxClick, newClick); // triangolo in basso a destra
  } else {
    // quando il numero massimo di click viene raggiunto si ritorna alla situazione iniziale (un triangolo verde)
    triangle(lowerLeft.x, lowerLeft.y, lowerRight.x, lowerRight.y, topCenter.x, topCenter.y); //funzione triangolo di processing
  }
}

```

Nel nostro programma (attraverso *PVector middleLeft*, *PVector middleRight* e *PVector middleCenter*) andiamo a trovare i punti medi dei lati per ogni livello e poi creiamo i rispettivi frattali richiamando la funzione *frattale*.

Quando si dovrebbe arrivare al livello 8, il contatore del livello ritorna a 0 e viene ridisegnato il triangolo iniziale del livello 0 attraverso la funzione *triangle*.

Cerchio



In questo programma andiamo a disegnare un frattale composto da cerchi, prima creiamo la funzione *frattale* che ci servirà successivamente per andare a creare i cerchi dei livelli successivi.

```
frattale(width/2, height/2, 300, click, 0);
```

Ogni nuova circonferenza che si andrà a creare, avrà due caratteristiche principali, ovvero: tutti i centri delle circonferenze giacciono sulla stessa retta e intersecano la circonferenza del livello precedente.

```
void frattale(float x, float y, float d, int maxClick, int curClick) {  
    // curClick=0 maxClick=1 --> solo per la prima volta  
  
    if (curClick < maxClick) { // curClick e maxClick sono stati dichiarati alla riga precedente  
  
        int newClick = curClick+1;  
  
        // chiamata ricorsiva della funzione frattale per ognuno dei nuovi cerchi  
        frattale(x + d/2, y, d/2, maxClick, newClick); // cerchio a destra  
        frattale(x - d/2, y, d/2, maxClick, newClick); // cerchio a sinistra  
    } else {  
        // quando il numero massimo di click viene raggiunto si ritorna alla situazione iniziale (un cerchio con bordo verde e senza riempimento)  
        ellipse(x, y, d, d);  
    }  
}
```

Richiamando la funzione frattale andiamo a disegnare le circonferenze, di cui il centro di una sarà spostato verso sinistra e quello dell'altra verso destra. Entrambi le circonferenze avranno il proprio centro ad una distanza dal centro della circonferenza del livello precedente pari al raggio di quest'ultima.

Quando il contatore dei click arriva a 7 il disegno viene cancellato e cliccando un'altra volta si può ripartire dal cerchio iniziale.