Carlos Soliz

Documentación Actividad n°16

Año: 2024
Profesores: York Elias MANSILLA MUÑOZ y Jorge Fabián SILES GUZMÁN

Materia: Proyecto de Implementación de Sitios Web Dinámicos
Grupo: Carlos Soliz

Carlos Soliz

Descripción General

Este proyecto tiene como objetivo implementar visualizaciones interactivas de dos fractales clásicos: **el Triángulo de Sierpinski** y **el Conjunto de Mandelbrot**. Ambas visualizaciones permiten personalización y ajustes en tiempo real por parte del usuario, así como efectos visuales animados. El Triángulo de Sierpinski se construye usando recursividad y tiene animación rotatoria, mientras que el Conjunto de Mandelbrot permite la exploración mediante zoom interactivo y ajuste de la calidad de renderizado.

Características Principales

1. **Triángulo de Sierpinski**:
    - Construcción recursiva del triángulo.
    - Interactividad con el usuario para ajustar la profundidad de la recursión.
    - Animación de rotación del fractal.
    - Colores que cambian dinámicamente según la profundidad.
2. **Conjunto de Mandelbrot**:
    - Representación precisa del conjunto mediante iteración.
    - Control interactivo del nivel de detalle mediante ajustes en el número de iteraciones.
    - Función de zoom para explorar diferentes regiones del fractal.
    - Colores vibrantes en el espectro HSB que resaltan las áreas del fractal.

Documentación del Código

Triángulo de Sierpinski

Variables Globales

- `int depth`: Define la profundidad máxima de recursión.
- `float triangleHeight`: Altura del triángulo principal, calculada en función del ancho del lienzo.
- `PVector p1, p2, p3`: Vértices del triángulo principal.
- `float angle`: Controla el ángulo de rotación del triángulo.

Funciones

1. `setup()`

- Inicializa el entorno de Processing, define el tamaño del lienzo, calcula la altura del triángulo y los vértices principales.

```
void setup() {
  size(800, 800);
  triangleHeight = sqrt(3) / 2 * width;
  p1 = new PVector(width / 2, 0);
  p2 = new PVector(0, triangleHeight);
  p3 = new PVector(width, triangleHeight);
  frameRate(30);  // Animación suave
}
```

2. `draw()`

- Dibuja el fractal y aplica una rotación continua.
- Llama a la función recursiva `drawTriangle()` para generar los triángulos.

```
void draw() {
  background(255);
  fill(0);
  noStroke();

  translate(width / 2, height / 2);  // Centramos el origen
  rotate(angle);  // Aplicamos la rotación
  translate(-width / 2, -height / 2);

  drawTriangle(depth, p1, p2, p3);

  angle += 0.01;  // Incremento del ángulo para rotación continua
}
```

3. `drawTriangle()`

- Función recursiva que dibuja los triángulos de Sierpinski.
- Calcula los puntos medios de los lados de cada triángulo y continúa dividiendo hasta alcanzar la profundidad cero.

```
void drawTriangle(int depth, PVector p1, PVector p2, PVector p3) {
  if (depth == 0) {
    fill(map(depth, 0, 5, 0, 255), 100, 150);
    triangle(p1.x, p1.y, p2.x, p2.y, p3.x, p3.y);
  } else {
```

```
    PVector mid1 = new PVector((p1.x + p2.x) / 2, (p1.y + p2.y) /
2);
    PVector mid2 = new PVector((p2.x + p3.x) / 2, (p2.y + p3.y) /
2);
    PVector mid3 = new PVector((p1.x + p3.x) / 2, (p1.y + p3.y) /
2);

    drawTriangle(depth - 1, p1, mid1, mid3);
    drawTriangle(depth - 1, mid1, p2, mid2);
    drawTriangle(depth - 1, mid3, mid2, p3);
  }
}
```

4. `keyPressed()`

- Controla la profundidad de recursión en tiempo real usando las teclas de flechas. La profundidad aumenta con la tecla `UP` y disminuye con la tecla `DOWN`.

```
void keyPressed() {
  if (keyCode == UP && depth < 8) {
    depth++;
  }
  if (keyCode == DOWN && depth > 1) {
    depth--;
  }
}
```

Conjunto de Mandelbrot

Variables Globales

- `float xmin, ymin, xmax, ymax`: Define los límites del área del plano complejo que se visualiza.
- `int maxIterations`: Número máximo de iteraciones para determinar si un punto pertenece al conjunto.

Funciones

1. `setup()`

- Configura el entorno y llama a `renderMandelbrot()` para dibujar el fractal por primera vez.

```
void setup() {
  size(800, 800);
  noLoop();
  colorMode(HSB, 255);  // Colores vivos usando HSB
  renderMandelbrot();
}
```

2. `renderMandelbrot()`

- Recorre cada píxel de la pantalla, lo transforma en coordenadas del plano complejo, y calcula si pertenece o no al conjunto de Mandelbrot.
- Colorea los píxeles en función del número de iteraciones necesarias para escapar del conjunto.

```
void renderMandelbrot() {
  loadPixels();
  for (int x = 0; x < width; x++) {
    for (int y = 0; y < height; y++) {
      float a = map(x, 0, width, xmin, xmax);
      float b = map(y, 0, height, ymin, ymax);
      float ca = a;
      float cb = b;
      int n = 0;
      while (n < maxIterations) {
        float aa = a * a - b * b;
        float bb = 2 * a * b;
        a = aa + ca;
        b = bb + cb;
        if (abs(a + b) > 16) break;
        n++;
      }
      int bright = int(map(n, 0, maxIterations, 0, 255));
      if (n == maxIterations) bright = 0;
      int pix = int(x + y * width);
      pixels[pix] = color(bright % 255, 255, bright > 0 ? 255 : 0);
    }
  }
  updatePixels();
```

```
}
```

3. `keyPressed()`

- Permite ajustar el número de iteraciones con las teclas `+` y `-` para cambiar el nivel de detalle. Las teclas `w` y `s` permiten acercar y alejar el zoom.

```
void keyPressed() {
  if (key == '+') {
    maxIterations += 50;
    renderMandelbrot();
  }
  if (key == '-') {
    maxIterations = max(50, maxIterations - 50);
    renderMandelbrot();
  }
  if (key == 'w') {  // Acercarse
    zoom(0.5);
  }
  if (key == 's') {  // Alejarse
    zoom(2);
  }
}
```

4. `zoom()`

- Ajusta los límites del área visible para permitir hacer zoom sobre el fractal.

```
void zoom(float factor) {
  float newWidth = (xmax - xmin) * factor;
  float newHeight = (ymax - ymin) * factor;
  float centerX = (xmin + xmax) / 2;
  float centerY = (ymin + ymax) / 2;
  xmin = centerX - newWidth / 2;
  xmax = centerX + newWidth / 2;
  ymin = centerY - newHeight / 2;
  ymax = centerY + newHeight / 2;
  renderMandelbrot();
}
```

Controles Interactivos

Triángulo de Sierpinski:

- **Flecha hacia arriba (`UP`)**: Aumenta la profundidad de recursión.
- **Flecha hacia abajo (`DOWN`)**: Disminuye la profundidad de recursión.
- **Animación automática**: El fractal rota continuamente.

Conjunto de Mandelbrot:

- **Tecla `+`**: Aumenta el número de iteraciones, añadiendo más detalle al fractal.
- **Tecla `-`**: Disminuye el número de iteraciones.
- **Tecla `w`**: Acerca el zoom sobre el fractal.
- **Tecla `s`**: Aleja el zoom.