

FRACTAL TREES

A. Martín Dinnbier

<https://ggbm.at/xnyzgybn>

```
1
2 var ap=ggbApplet;
3 //valor de margen con respecto a la ventana visible en el que queremos
4 //situar nuestros nuevo árboles fractales
5 var marco=100;
6
7 //obtenemos el ancho de la ventan
8 var winWidth=ap.evalCommandGetLabels("winWidth=x(Corner(5))");
9 var winWidth=ap.getValue(winWidth);
10
11 //obtenemos el alto de la ventana
12 var winHeight=ap.evalCommandGetLabels("winHeight=y(Corner(5))");
13 var winHeight=ap.getValue(winHeight);
14
15
16 //valor arbitrario para las coordenadas del punto A. El marco evita elementos en los
    bordes
17 var x=getRandomArbitrary(marco, winWidth-marco);
18 var y=getRandomArbitrary(marco, winHeight-marco);
19
20 //convertimos los valores de las coordenadas a coordenadas de Geogebra
21 var puntoPartida=new Array();
22 puntoPartida=pointToAxis(x,y);
23
24 generarFractal(puntoPartida[0], puntoPartida[1]);
25
26
27 //////////////////////////////////////////////////
28 //FUNCIÓN: generarFractal
29 //Parámetros de entrada: Recibe las coordenadas del punto de partida
30 //Devuelve: un array con todos los elementos nuevos creados
31 //PuntoA, PuntoB, PuntoC, SegmentoA, SegmentoB
32 //////////////////////////////////////////////////
33
34 function generarFractal(x, y){
35
36 var ap=ggbApplet;
37
38
39 //conseguir las coordenadas de las esquinas
40 var distX=ap.evalCommandGetLabels("distX=Distance(Corner(1), Corner(2))");
41 var distX=ap.getValue(distX);
42 //conseguir el ancho y alto de la venta
43 var winWidth=ap.evalCommandGetLabels("winWidth=x(Corner(5))");
44 var winWidth=ap.getValue(winWidth);
45 //obtenemos el alto de la ventana
46 var winHeight=ap.evalCommandGetLabels("winHeight=y(Corner(5))");
47 var winHeight=ap.getValue(winHeight);
48
```

```

49 //obtengo el factor de conversión calculado en Geogebra (pixels/unidad real)
50 var factorXY=distX/winWidth;
51
52 //valores para situar los puntos B y C cerca del punto A con respecto a la ventana
53 var mod=getRandomArbitrary(1, 3);
54 var angulo=getRandomArbitrary(15, 65);
55 //convertimos el ángulo a radianes
56 angulo=angulo*Math.PI/180;
57
58 var distanciaAX=x+(winWidth/20)*(1/factorXY);
59 var distanciaAY=y+(winHeight/20)*(1/factorXY);
60
61 var radio=mod*(winWidth/30)*(1/factorXY);
62 var distanciaBX=distanciaAX+radio*Math.cos(angulo);
63 var distanciaBY=distanciaAY+radio*Math.sin(angulo);
64
65 //situar los puntos
66 //var PuntoA=ap.evalCommandGetLabels("("+"x+", "+y+")");
67 var PuntoA=ap.evalCommandGetLabels("("+"x+", "+y+")");
68 var PuntoB=ap.evalCommandGetLabels("("+"distanciaAX+", "+distanciaAY+")");
69 var PuntoC=ap.evalCommandGetLabels("("+"distanciaBX+", "+distanciaBY+")");
70 //crear los segmentos
71 var segmentoA=ap.evalCommandGetLabels("Segment("+PuntoA+", "+PuntoB+")");
72 var segmentoB=ap.evalCommandGetLabels("Segment("+PuntoB+", "+PuntoC+")");
73 //ocultar etiquetas de los elementos de partida
74 ap.evalCommand("ShowLabel("+PuntoA+", "+false+")");
75 ap.evalCommand("ShowLabel("+PuntoB+", "+false+")");
76 ap.evalCommand("ShowLabel("+PuntoC+", "+false+")");
77 ap.evalCommand("ShowLabel("+segmentoA+", "+false+")");
78 ap.evalCommand("ShowLabel("+segmentoB+", "+false+")");
79 //cambiamos el grosor de los segmentos originales
80 ap.evalCommand("SetLineThickness("+segmentoA+", 4)");
81 ap.evalCommand("SetLineThickness("+segmentoB+", 4)");
82
83 var levels=ap.getValue("levels");
84
85 arbol(PuntoA, PuntoB, PuntoC, segmentoB, levels);
86
87 }
88
89
90 //////////////////////////////////////////////////
91 //FUNCIÓN: getRandomArbitrary
92 //Parámetros de entrada: Recibe dos valores (min y max)
93 //Devuelve: Un valor aleatorio entre min y max
94 //////////////////////////////////////////////////
95
96 function getRandomArbitrary(min, max) {
97     return Math.random() * (max - min) + min;
98 }
99
100 //////////////////////////////////////////////////
101 //FUNCIÓN: valueToAxis
102 //Parámetros de entrada: Recibe un valor en píxeles
103 //Devuelve: El valor en coordenadas de los ejes de Geogebra
104 //Desc: Transforma un valor en píxeles en coordenadas de los ejes
105 //////////////////////////////////////////////////
106
107 function valueToAxis(x){
108     //definimos el applet
109     var ap=ggbApplet;
110     //conseguir las coordenadas de las esquinas

```

```

111 var distX=ap.evalCommandGetLabels("distX=Distance(Corner(1), Corner(2))");
112 var distX=ap.getValue(distX);
113 //conseguir el ancho y alto de la venta
114 var winWidth=ap.evalCommandGetLabels("winWidth=y(Corner(5))");
115 var winWidth=ap.getValue(winWidth);
116 //obtengo el factor de conversión calculado en Geogebra (pixels/unidad real)
117 var factorXY=distX/winWidth;
118 //devolvemos el valor calculado
119 return(x*(1/factorXY))
120 }
121
122 //////////////////////////////////////////////////
123 //FUNCIÓN: pointToAxis
124 //Parámetros de entrada: Recibe las coordenadas de un punto en pixeles
125 //Devuelve: Las coordenadas de un punto en coordenadas de los ejes de Geogebra
126 //Desc: Transforma un par de coordenadas (x,y) en en píxeles en coordenadas de los
      ejes
127 //////////////////////////////////////////////////
128
129 function pointToAxis(x, y){
130     var point= new Array();
131     //definimos el applet
132     var ap=ggbApplet;
133     //conseguir las coordenadas de las esquinas
134     var corner1=ap.evalCommandGetLabels("corner1=x(Corner(1))");
135     var valuelxcorner1=ap.getValue(corner1);
136     var corner4=ap.evalCommandGetLabels("corner4=y(Corner(4))");
137     var valueycorner4=ap.getValue(corner4);
138     //distancia entre las esquinas (ancho en coordenadas de los ejes)
139     var distX=ap.evalCommandGetLabels("distX=Distance(Corner(1), Corner(2))");
140     var distX=ap.getValue(distX);
141     //conseguir el ancho y alto de la venta (en pixeles)
142     var winWidth=ap.evalCommandGetLabels("winWidth=x(Corner(5))");
143     var winWidth=ap.getValue(winWidth);
144     //obtengo el factor de conversión calculado en Geogebra (pixels/unidad real)
145     var factorXY=distX/winWidth;
146     //Dados dos valores x,y en píxeles, los convertimos a coordenadas de los ejes de
      Geogebra haciendo la siguiente conversión.
147     point[0]=valuelxcorner1+x*factorXY;
148     point[1]=valueycorner4-y*factorXY;
149
150     //devolvemos el valor calculado
151     return(point)
152 }
153
154
155 //FRACTALES
156 //Crea un esquema de fractales a partir de la herramienta definida como "fractal" en
      la interfaz de Geogebra
157
158
159 //////////////////////////////////////////////////
160 //FUNCIÓN: arbol (recursiva)
161 //Parámetros de entrada: recibe tres puntos, un segmento y la profundidad del árbol (
      n)
162 //Devuelve: 0 cuando llega al final
163 //Desc: Construye un arbol fractal basado en la herramienta "Fractal" creada en el
      entorno Geogebra
164 //////////////////////////////////////////////////
165
166
167 function arbol(p1, p2, p3, segmento, n){

```

```

168
169 var ap=ggbApplet;
170 var t, seg_reflex, pun_reflex;
171
172
173
174 t=ap.evalCommandGetLabels("Fractal("+p1+", "+p2+", "+ p3+"");
175 var arrayDeNuevos=t.split(",");
176
177 //ocultamos las etiquetas de los puntos nuevos
178 ap.evalCommand("ShowLabel("+arrayDeNuevos[0]+", "+false+"");
179 ap.evalCommand("ShowLabel("+arrayDeNuevos[1]+", "+false+"");
180 //cambiamos el grosor de la línea creada
181 ap.evalCommand("SetLineThickness("+arrayDeNuevos[1]+", 2)");
182
183 //Reflejar segmento
184     seg_reflex=ap.evalCommandGetLabels("Reflect("+arrayDeNuevos[1]+", "+segmento+"");
185 //Reflejar punto
186 pun_reflex=ap.evalCommandGetLabels("Reflect("+arrayDeNuevos[0]+", "+segmento+"");
187 //Cambio de colores de los nuevos elementos
188 //puntos nuevos
189     ap.evalCommand("SetColor("+arrayDeNuevos[0]+", 1, 0.2, 0.5)");
190
191
192 n=n-1;
193 if(n==0) return 0;
194 else{
195     arbol(p2, p3, arrayDeNuevos[0], arrayDeNuevos[1], n);
196     arbol(p2, p3, pun_reflex, seg_reflex, n);
197     return 0;
198 }
199
200
201 }

```

Listing 1: Borrar elementos en Geogebra