

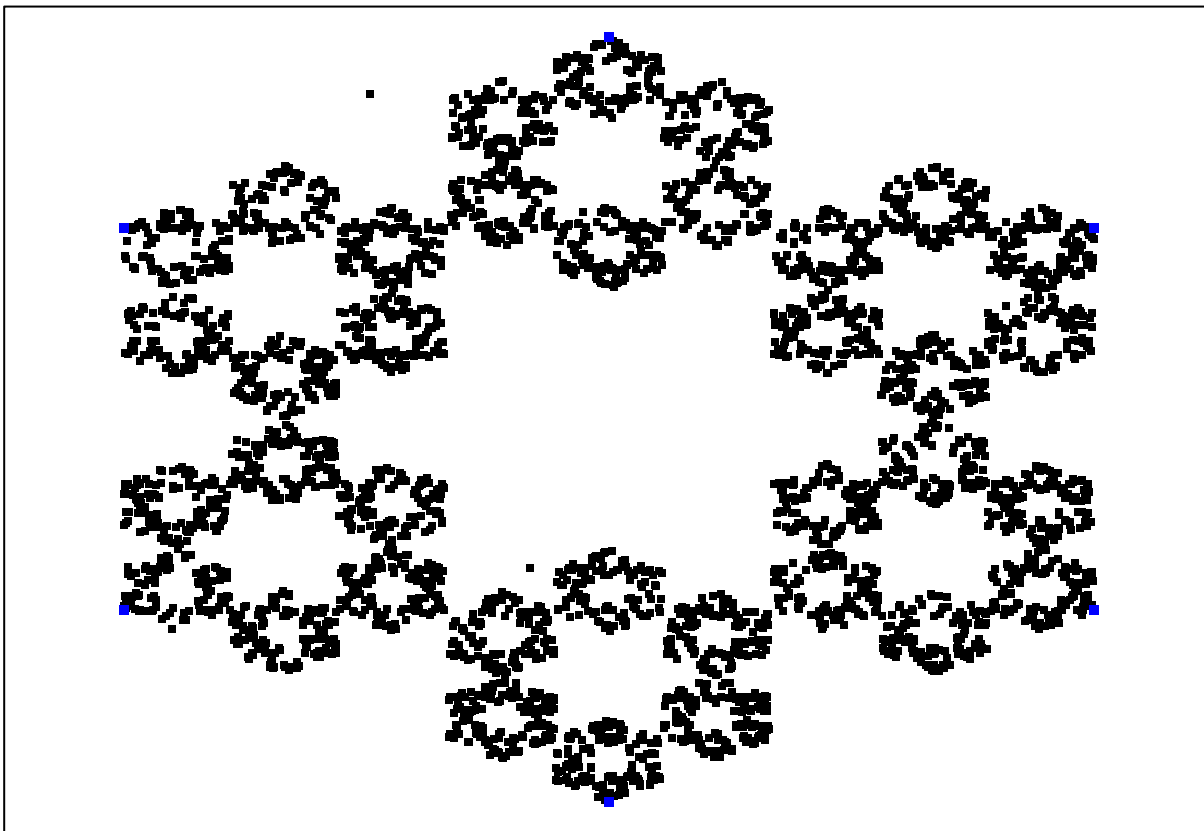
Chaos Game: hexagon

Neagu Marian - 244 Soare Cristian - 244

Studiu de caz

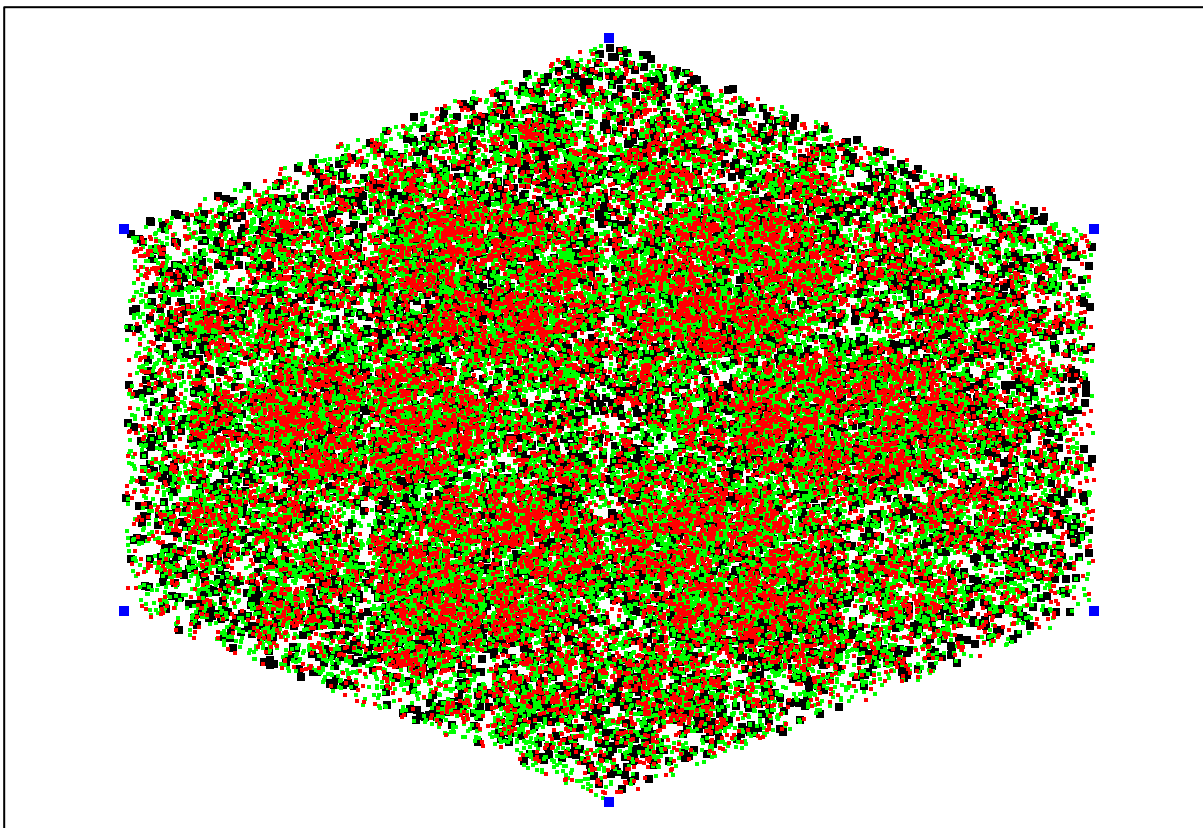
Chaos game este cunoscut in matematica drept o modalitate de a crea un fractal, folosind un poligon (in cazul de fata un hexagon), si un punct initial ales in mod aleator in interiorul poligonului. Astfel, de la punctul initial se alege o distanta constanta care se aplica pentru a crea un nou punct intre punctul initial si unul dintre varfuri ales aleator. Se repeta procesul si astfel se obtine fractalul⁽¹⁾.

Pentru hexagon, folosind fractia $1/3$, se obtine hexagonul Sierpinski (Sierpinski's n-gons):



Imaginea de mai sus a fost generata folosind 5000 de puncte si fractia $1/3$

Totusi, daca modificam fractia, spre exemplu in $1/2$, tiparul se intercaleaza si formeaza un tipar nou, mai putin vizibil (un fulg/stea cu 6 varfuri), dar care poate fi accentuat folosind puncte cu dimensiuni si culori diferite:



Imaginea de mai sus a fost generata folosind 55000 de puncte si fractia 1/2, cu puncte negre, verzi si rosi, de dimensiuni diferite

O alta caracteristica a acestor fractali este si faptul ca, daca ar fi generate un numar infinit de puncte, oricat de mult ai mari imaginea obtinuta, se observa acelasi tipar. Acest lucru poate fi vizualizat mai usor daca se ruleaza aplicatia shiny in browser si se da zoom.

Explicatii cod

```
library(shiny)
library(shinyBS)

shinyServer (function(input, output, session){

  coord_lists <- reactive({
```

Setam valoarea ratio in functie de valoarea de input primita. Valoarea default pentru hexagonul Sierpinski este 0.33.

```
    ratio <- input$fraction*(input$gen>-1)
```

Numarul maxim de puncte ce pot fi desenate.

```
len <- 55000
```

Matrice care retine coordonatele celor 6 varfuri ale heagonului.

```
vertices_coord <- matrix(NA, ncol=3, nrow=6)
```

Constante ce ajuta la definirea coordonatelor folosite des.

```
y_const <- 0.5  
x_const <- 0.75
```

Retinerea coordonatelor varfurilor pe fiecare rand al matricei.

```
vertices_coord[1,] <- c(1,0,2*y_const) #up  
vertices_coord[2,] <- c(2,x_const,0.5) #up-right  
vertices_coord[3,] <- c(3,x_const,-0.5) #down-right  
vertices_coord[4,] <- c(4,0,-2*y_const) #down  
vertices_coord[5,] <- c(5,-x_const,-0.5) #down-left  
vertices_coord[6,] <- c(6,-x_const,0.5) #up-left
```

Matrice care retine toate coordonatele punctelor ce pot fi desenate.

```
coords <- matrix(NA,ncol=2,nrow=(len+1))
```

Primul punct este la coordonate random.

```
coords[1,] <- c(runif(1,-x_const,x_const),runif(1,-2*y_const,2*y_const))
```

Retinerea unor coordonate random pentru fiecare punct in matrice.

```
for (i in 1:len){  
  row <- i+1  
  a <- sample(1:6, 1)  
  #random generated number for the direction of the new point  
  x <- vertices_coord[a,2]  
  y <- vertices_coord[a,3]  
  x.new <- (1-ratio)*x + (ratio)*coords[i,1]  
  y.new <- (1-ratio)*y + (ratio)*coords[i,2]  
  # new coords for the next point  
  coords[row,]<-c(x.new,y.new)  
}  
  
return(list(vertices_coord,coords))  
})  
  
output$Plot <- renderPlot({
```

Retinem in vertices_coord matricea varfurilor hexagonului, iar in coords matricea de coordonate a tuturor punctelor.

```
vertices_coord <- coord_lists()[[1]]
coords <- coord_lists()[[2]]
```

```
y_const <- 0.5
x_const <- sqrt(3)/2
```

Definirea plotului.

```
par(mar = c(0.5,0.5,0.5,0.5))
plot(0,
      0,
      xlim = c(-x_const, x_const),
      ylim = c(-2*y_const, 2*y_const),
      col = 0,
      yaxt = "none",
      xaxt = "none"
    )
```

Putem seta ca simularea sa deseneze puncte care isi alterneaza culoarea dupa un anumit numar de puncte deja desenate.

```
if(input$colorsCheckBox == TRUE){
```

```
  if (input$points != 0){
    for(i in 1:input$points){
      if (i > 40000)
        points(coords[i,1], coords[i,2], pch = ".", cex = 0.1, col = "red")
      else if (i > 10000)
        points(coords[i,1], coords[i,2], pch = ".", cex = 1, col = "green")
      else
        points(coords[i,1], coords[i,2], pch = ".", cex = 4, col = "black")
    }
  }
}
# non-color simulation
else
  points(coords[1:input$points,1], coords[1:input$points,2], pch=".", cex= 0.2, col = "black")
```

Desenarea varfurilor hexagonului.

```
points(vertices_coord[,2], vertices_coord[,3], pch = ".", cex = 5, col = "blue")
```

```
})# renderPlot

})# shinyServer

shinyUI(fluidPage(

  titlePanel("Chaos Game: hexagon"),
```

```

sidebarLayout(

  sidebarPanel(

    sliderInput("fraction", label = "Fraction to hexagon vertex:",
               min = 0.01,
               max = .99,
               value = .33,
               step = .01),
    "Simularea default este cu fractia 1/3 (Hexagonul Sierpinski)",

    sliderInput("points",
               "Number of points (n):",
               min = 1,
               max = 55000,
               step = 30,
               value = 1000,
               animate = animationOptions(interval = 10)),

    checkboxInput("colorsCheckBox", "Color mode (slow)", value = FALSE, width = NULL),

    bsButton("gen", style = "secondary", label="Randomize")

  ),#sidebarPanel

  mainPanel(

    div(
      div(plotOutput("Plot"),
          style="width:500px",
          inline="TRUE"
        )
      ,align="center"
    )

  )#mainPanel
)#sidebarLayout
)# fluidPage
)# shinyUI

```