

## Laboratorul 03. Sisteme bazate pe Gramatici și Reguli

### Lindenmayer Systems

Sistemele Lindenmayer (L-Systems) sunt un model matematic introdus de biologul Aristid Lindenmayer în 1968 pentru a modela procesele de creștere ale plantelor. Acestea sunt deosebit de eficiente pentru simularea structurilor și fractalilor asemănătoare plantelor.

Componentele ale sistemelor L:

- **Alfabet:** Un set de simboluri care reprezintă diverse componente ale unei plante
- **Axiomă:** Șirul inițial de simboluri de la care sistemul își începe construcția
- **Reguli de producție:** O colecție de reguli de substituție care definesc modul în care fiecare simbol poate fi înlocuit sau extins într-un șir mai complex de simboluri.

Exemple de structuri care imită plantele se pot vedea în imaginea de mai jos:



**a**  
 $n=5, \delta=25.7^\circ$   
 $F$   
 $F \rightarrow F[+F]F[-F]F$



**b**  
 $n=5, \delta=20^\circ$   
 $F$   
 $F \rightarrow F[+F]F[-F][F]$



**c**  
 $n=4, \delta=22.5^\circ$   
 $F$   
 $F \rightarrow FF[-F+F+F] + [+F-F-F]$



**d**  
 $n=7, \delta=20^\circ$



**e**  
 $n=7, \delta=25.7^\circ$



**f**  
 $n=5, \delta=22.5^\circ$

$$\begin{array}{l} X \\ X \rightarrow F[+X]F[-X]+X \\ F \rightarrow FF \end{array}$$

$$\begin{array}{l} X \\ X \rightarrow F[+X][-X]FX \\ F \rightarrow FF \end{array}$$

$$\begin{array}{l} X \\ X \rightarrow F-[[X]+X]+F[+FX]-X \\ F \rightarrow FF \end{array}$$

În exemplul de mai sus, se folosește interpretarea “turtle” a stringurilor. Astfel, se întâlnesc următoarele simboluri:

- **F** - Mers înainte cu distanța d
- **+** - Rotație la stânga cu unghiul  $\delta$
- **-** - Rotație la dreapta cu unghiul  $\delta$
- **[** - Adăugarea într-o stivă a stării actuale. Informațiile salvate pe stivă conțin poziția, orientarea și, eventual, alte atribute, cum ar fi culoarea și lățimea liniilor desenate
- **]** - Scoaterea unei stări din stivă și transformarea acesteia în starea curentă. Nu se desenează nimic, deși, în general, poziția se schimbă
- **n** - Numărul de iterații

O variantă în care se poate implementa ușor un L-System este folosind biblioteca **turtle** din Python. Astfel, pentru a inițializa biblioteca, se folosește următoarea secvență de cod:

```
# Initialize turtle
t = turtle.Turtle()
wn = turtle.Screen()
t.speed(0)
t.left(90)
t.penup()
t.goto(0, -wn.window_height() // 2 + 20)
t.pendown()

# TODO Your implementation goes here

# Wait for user to close window
wn.mainloop()
```

Funcții utile în implementarea unui sistem L, folosind **turtle**:

```
t.forward(distance) # Move forward
t.goto(position) # Move to position, without drawing anything
t.left(angle) # Turn left
t.right(angle) # Turn right
t.setheading(heading) # Set the rotation angle
t.penup() # Stop drawing
t.pendown() # Start drawing
```

## Cellular Automata

Pentru a genera un dungeon 2D de bază folosind cellular automata, se urmează următorii pași:

- **Inițializarea unui grid:** Se crează un grid 2D în care fiecare celulă este atribuită aleatoriu ca perete sau podea pe baza unei probabilități specificate
- **Aplicarea regulilor stabilite prin cellular automata:** Actualizarea în mod iterativ a gridului, aplicând reguli care determină starea fiecărei celule pe baza vecinilor săi
- **Rafinare rezultat:** După mai multe iterații, gridul va forma spații interconectate, asemănătoare unui dungeon. Procesarea suplimentară poate asigura conectivitate și poate adăuga funcții precum camere și coridoare

## Conway's Game of Life

Conway's Game of Life este un cellular automata care simulează evoluția celulelor pe un grid bazat pe reguli simple. Deși este folosit în principal pentru studiul modelelor complexe care apar din reguli simple, aplicarea sa directă în generarea unui dungeon este limitată. Cu toate acestea, alte tehnici cellular automata au fost utilizate în mod eficient pentru a crea modele de dungeons 2D.

În Conway's Game of Life fiecare celulă din grid este inițializată cu una din 2 stări posibile: vie sau moartă. Fiecare celulă interacționează cu cei opt vecini ai săi. La fiecare iterație, au loc următoarele tranziții:

- Orice celulă vie cu mai puțin de doi vecini vii moare
- Orice celulă vie cu doi sau trei vecini vii trăiește
- Orice celulă vie cu mai mult de trei vecini vii moare
- Orice celulă moartă cu exact trei vecini vii devine o celulă vie

Starea inițială constituie seed-ul sistemului. Prima generație este creată prin aplicarea regulilor de mai sus simultan la nivelul fiecărei celule din seed, vie sau moartă. Fiecare generație este determinată în funcție de generația precedentă. Regulile continuă să fie aplicate în mod repetat pentru a crea generații viitoare.

Pentru a implementa un dungeon folosind Conway's Game of Life, se poate considera o celulă vie echivalentul unui zid, iar o celulă moartă drept podea.

## Tasks

1. Generați planta F din imaginea de mai sus, folosind L-systems.
2. Implementați un dungeon 2D, folosind a cincea generație din Conway's Game of Life.
3. **Bonus 1.** Generați plantele D și E din imaginea de mai sus, folosind L-systems.
4. **Bonus 2.** Creați o animație în care să ilustrați fiecare generație din Conway's Game of Life.

gp/laboratoare/03.txt · Last modified: 2025/03/14 15:46 by maria\_anca.balutoiu