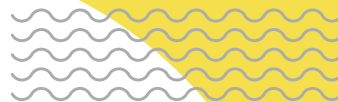




Facultatea de
Matematică și Informatică
Universitatea din București



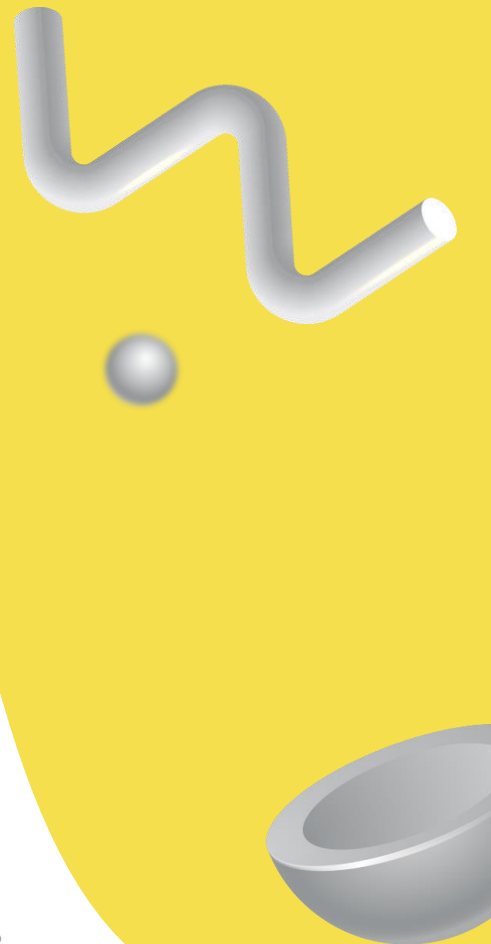
Programarea Aplicațiilor de Simulare

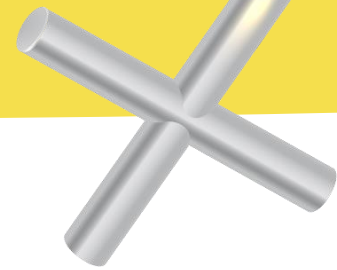
Algoritmi de Generare

Procedurală

Curs 6

Pătrânjel David-George





Agenda Cursului

01

Hărți de tip grid

02

Graph Rewriting

03

Wave Function
Collapse (WFC)

04

Implementarea
WFC



01

Hărți de tip grid



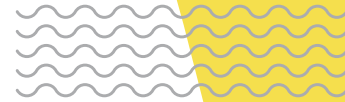
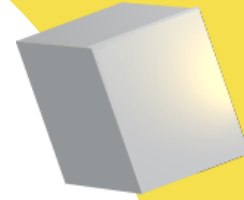


Metode de PCG de teren

- După abordare:
 - Teleologice
 - Ontologice
- După timp:
 - În procesul de design
 - La runtime
- După reprezentare:
 - Hărți de înălțime
 - Voxeli
 - Hărți tip mesh
 - Hărți de tip grid (Tilemaps)

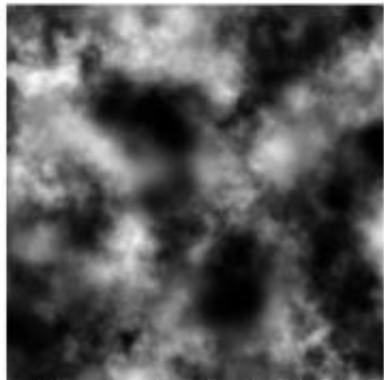


**Să ne amintim
din cursul
anterior...**

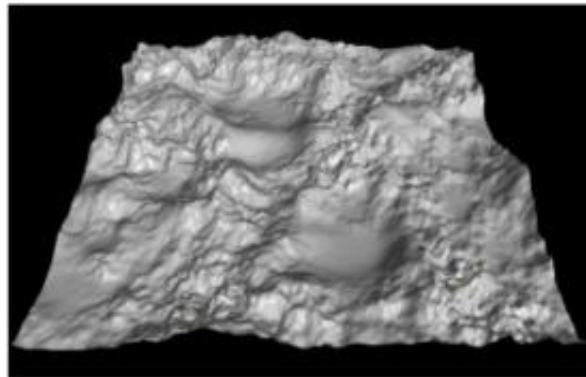


Hărți de înălțime

- Folosind o hartă de înălțime (en. heightmap) putem asocia valorii fiecărui pixel dintr-o imagine o valoare a înălțimii terenului.
- Putem genera un relief diversificat, similar cu cel din lumea reală.
- Nu putem genera peșteri sau prăpastii.

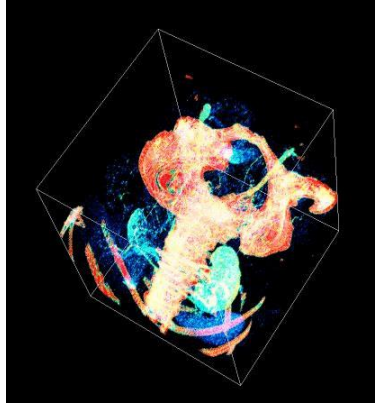
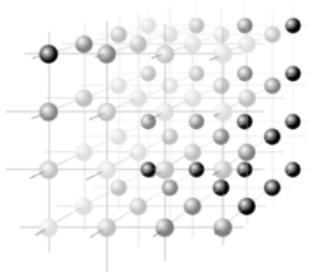


height visualized as brightness



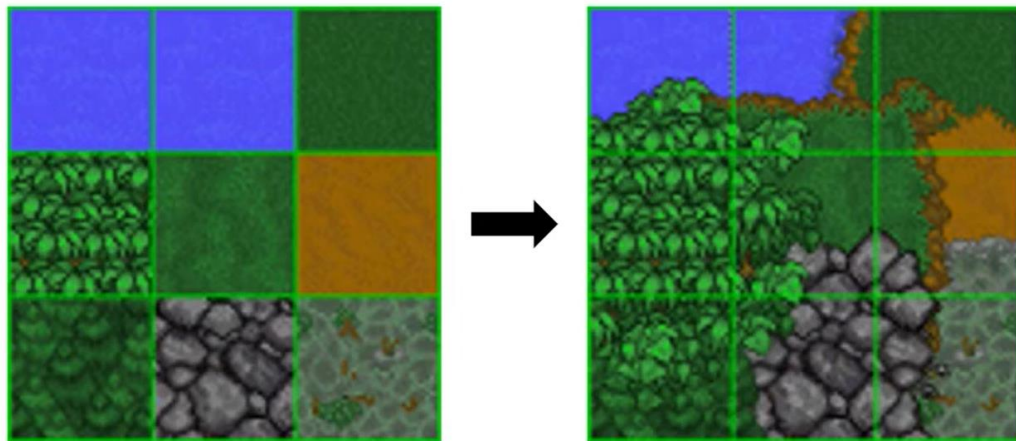
Voxeli

- Voxels = Volume elements
- Putem reprezenta orice forme 3D.
- Necesită un volum mare de date și algoritmi complecși de vizualizare.



Hărți de tip grid

- Caz discretizat al hărților de înălțime, în care fiecărei poziție din grid i se atribuie o placă de joc dintr-un set prestabilit.
- Specifice jocurilor 2D (topdown, isometrice) și des întâlnite în cazul jocurilor de societate.



Griduri Pătrate

- Cea mai întâlnită formă de hartă de tip grid.
- Poate fi extinsă la jocuri 3D.
- Transformare ușoară între coordonatele de grid, coordonatele lumii și coordonatele ecranului.



**The Legend of Zelda:
A Link to the Past**

Griduri Pătrate

- Cea mai întâlnită formă de hartă de tip grid.
- Poate fi extinsă la jocuri 3D.
- Transformare ușoară între coordonatele de grid, coordonatele lumii și coordonatele ecranului.



Stardew Valley

Griduri Pătrate

- Cea mai întâlnită formă de hartă de tip grid.
- Poate fi extinsă la jocuri 3D.
- Transformare ușoară între coordonatele de grid, coordonatele lumii și coordonatele ecranului.



Carcassonne

Griduri Hexagonale

- Pot avea două forme: vârf-sus sau latură-sus
- Se poate porni de la o structură pătratică pentru a ajunge la forma hexagonală.



Civilization VII

Griduri Hexagonale

- Pot avea două forme: vârf-sus sau latură-sus
- Se poate porni de la o structură pătratică pentru a ajunge la forma hexagonală.



For The King

Griduri Hexagonale

- Pot avea două forme: vârf-sus sau latură-sus
- Se poate porni de la o structură pătratică pentru a ajunge la forma hexagonală.



Catan

Griduri Triunghiulare

- Utile în generarea de meshuri pentru grafica obiectelor, dar rar folosite pentru generarea de teren.
- Prezintă toate dezavantajele tipurilor de grid pătratic și hexagonal.

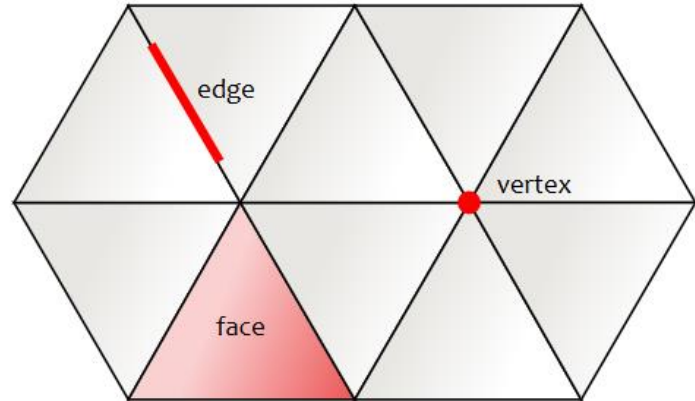


TransAmerica

Reprezentarea hărților de tip grid

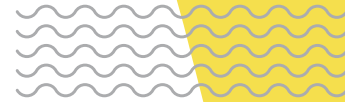
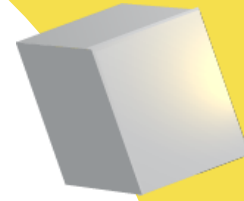
Pentru o vizualizare a reprezentării matematice a hărților de tip grid și a transformărilor între tipuri diferite de hărți, vom folosi de platforma de [aici](#).

(Red Blob Games)



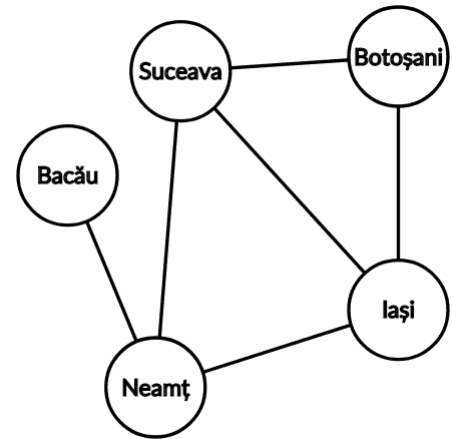
02

Graph Rewriting



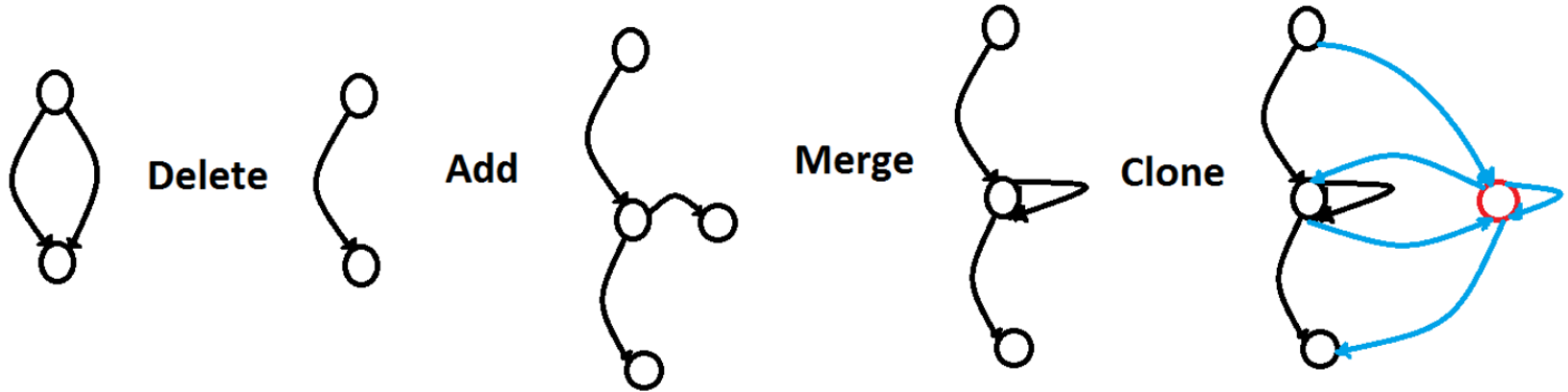
Graph Rewriting

- **Transformarea grafurilor** reprezintă o metodă computațională de modificare a grafurilor conform unor reguli.
- Un graf este o pereche ordonată de mulțimi, $G=(V,E)$ (vârfuri și muchii).
- Putem avea grafuri neorientate, grafuri orientate, multigrafuri, hipergrafuri, grafuri adnotate etc.
- Aplicații în: Compilatoare și limbaje formale, Inteligență artificială, Grafică și PCG, Sisteme Biologice etc.



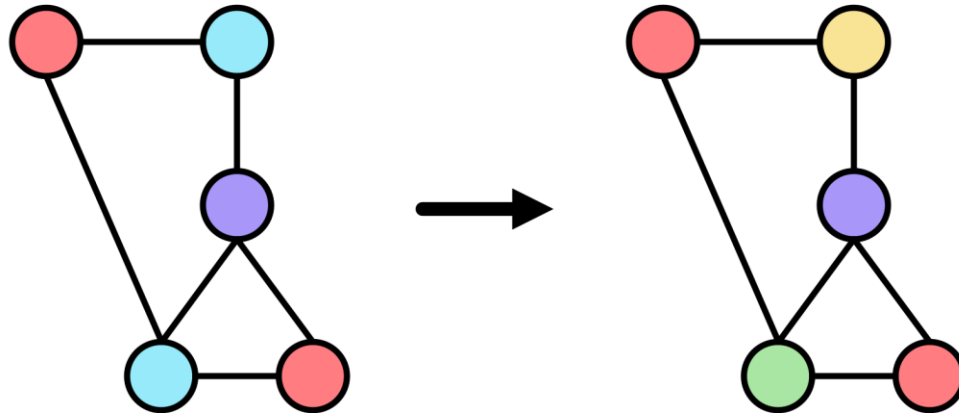
Reguli de transformare

O regulă de transformare reprezintă aplicarea unei modificări asupra unui subgraf. Putem avea mai multe tipuri de transformări ce pot afecta nodurile și/sau muchiile.



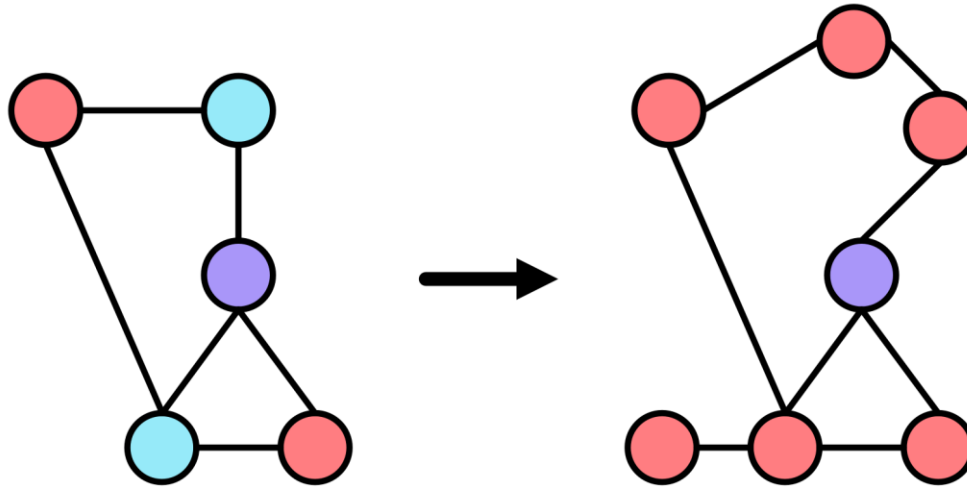
Exemplu 1

Înlocuim toate nodurile albastre cu noduri verzi sau galbene.



Exemplu 2

Înlocuim toate nodurile albastre cu două noduri roșii conectate aleatoriu în graficul original.



Exemplu 3

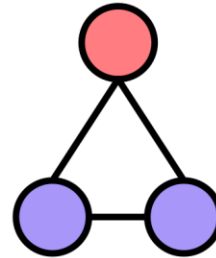
Folosim succesiv următoarele reguli

Rule 1

Match

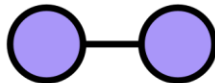


Replace with

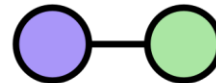


Rule 2

Match

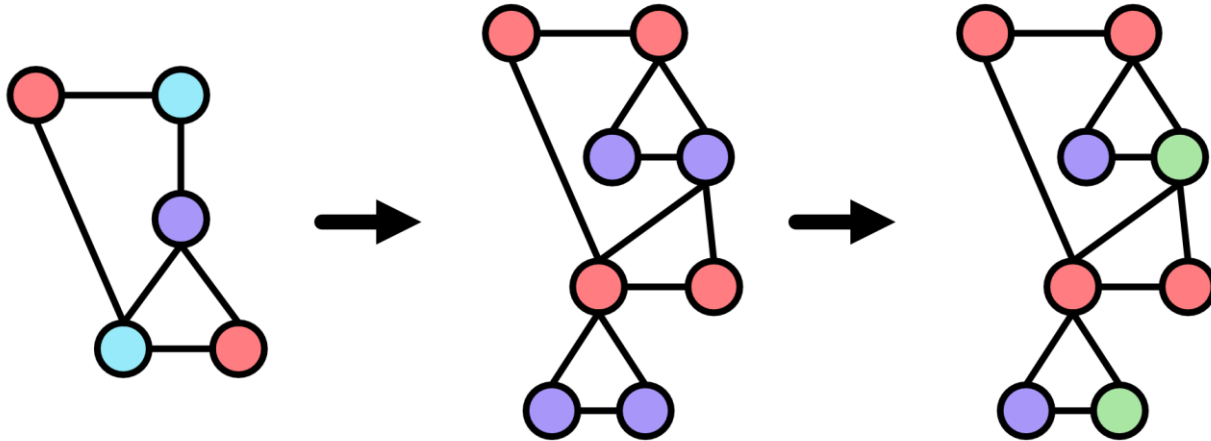


Replace with



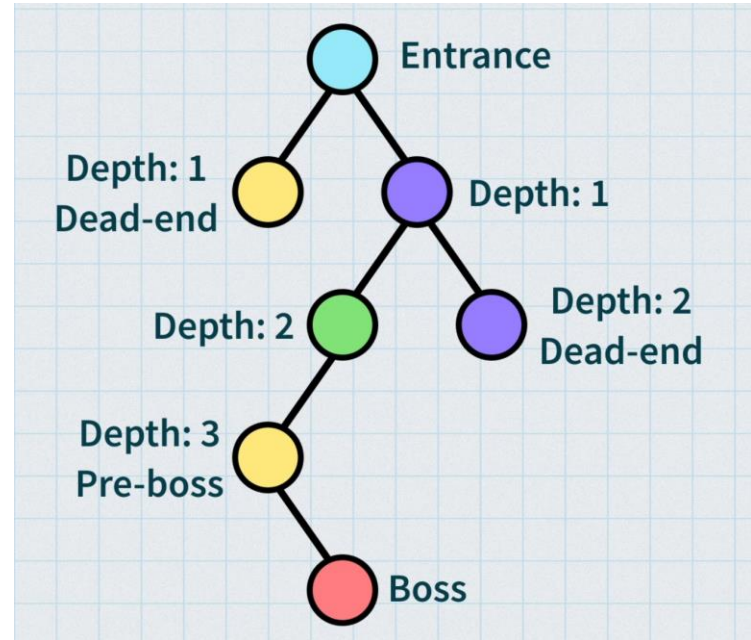
Exemplu 3

Obținem rezultatul următor



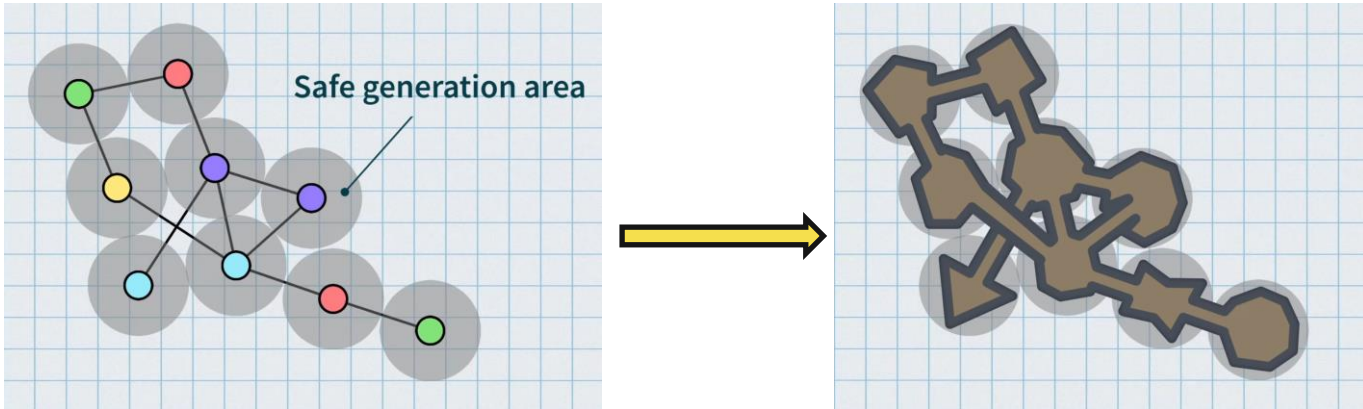
PCG folosind grafuri

- Avem nevoie de un arbore inițial.
- Putem să hardcodăm niște arbori inițiali sau să folosim un algoritm de plimbare aleatorie.
- Altă abordare: generarea unui arbore de meta-date.
- Conectăm în mod aleator noduri cu anumite condiții (dificultate similară, adâncime similară, puzzle-uri, inamici, obiecte etc.)



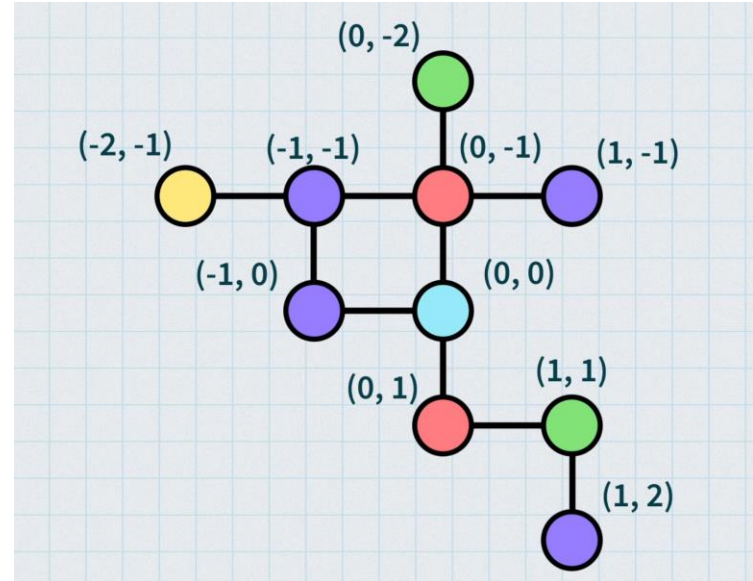
PCG folosind grafuri

- Putem adăuga nivele/arene/temnițe prin procesul de transformare a grafului și utilizând niște reguli de transformare.
- Adicional, se pot utiliza mecanici de fizică precum forțe adiționale pentru a apropia nodurile grafului și a crea o hartă de joc.



PCG folosind grafuri

- Putem aplica acest process și pentru o hartă de tip grid.
- Nodurile sunt fixate pe un sistem cartezian de axe.

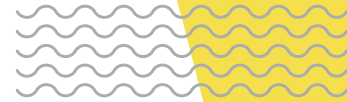




►►► Putem genera hărți topologice cu grafuri

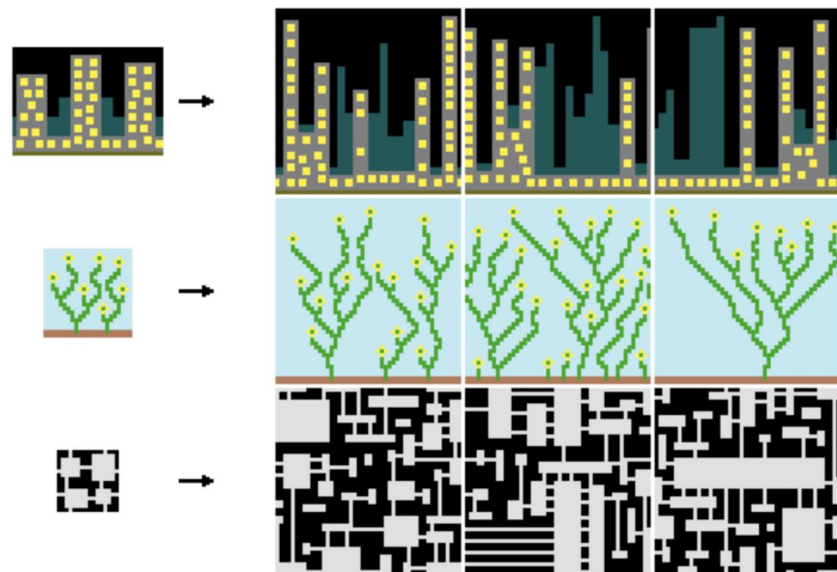
03

Wave Function Collapse (WFC)



Wave Function Collapse (WFC)

- Wave Function Collapse este un algoritm de generare procedurală de imagini, text, sunet, hărți ș.a.m.d.
- Învăță un set inițial de reguli pornind de la un model și le aplică în generarea de conținut.
- Se bazează pe conceptul de entropie din Teoria Informației.



Teoria Informației

- **Informația** reprezintă datele ce afectează incertitudinea pe care o avem despre un eveniment.
- Evenimentele rare produc multă informație!
- **Entropia** reprezintă media informației primite de o variabilă X .
- O entropie mică reprezintă o predictibilitate/certitudine a sistemului.

$$I(x_i) = \log_2 \left(\frac{1}{p_i} \right)$$

$$H(X) = E(I(X)) = \sum_{i=1}^N p_i \log_2 \frac{1}{p_i} = \sum_{i=1}^N -p_i \log_2 p_i$$

Mini-Sudoku

Numerele de la 1 la 4 trebuie să apară:

- exact o singură dată în fiecare rând.
- exact o singură dată în fiecare coloană.
- exact o singură dată în fiecare dintre cele patru colțuri formate din blocuri de 2×2 .

	4		
	2	3	
			3
4	3		2

Mini-Sudoku

12 34	4	12 34	12 34
12 34	2	3	12 34
12 34	12 34	12 34	3
4	3	12 34	2

12 3	4	12 3	12 3
12 34	2	3	12 34
12 34	12 34	12 34	3
4	3	12 34	2

Mini-Sudoku

1 3	4	1 2	1
1	2	3	1 4
1 2	1	1 4	3
4	3	1	2

Identificăm celula cu **entropia** minimă.

Entropie minimă = Certitudine mare
= **Cele mai puține opțiuni.**

Repetăm pașii.

Mini-Sudoku

¹ ₃	4	^{1 2}	1
1	2	3	¹ ₄
^{1 2}	1	¹ ₄	3
4	3	1	2

3	4	2	1
1	2	3	4
2	1	4	3
4	3	1	2

Sudoku

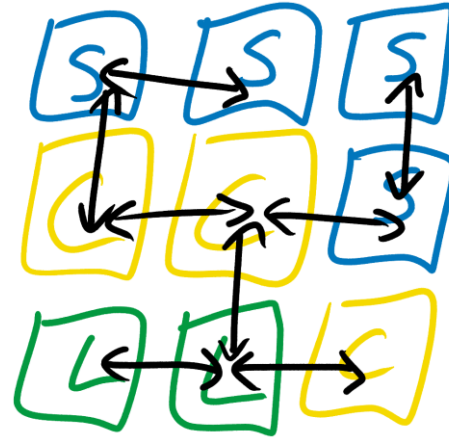
Ajungem rapid la o
problemă complexă!

			6					3
8					5			
			4			5	2	
				7	2			
	7	6			4			
5	4	2	3			8		
	3	8	1	4			9	5
7				3				
	2			6	8	3		7

Algoritmul WFC

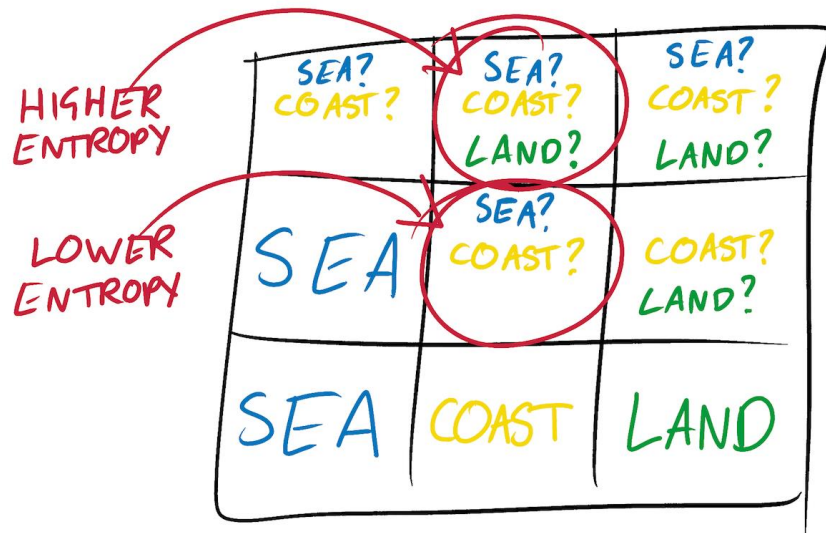
1. Definim un **grid** cu valori posibile de plăci de teren (tiles) și regulile de adiacență.

SEA COAST LAND	SEA COAST LAND	SEA COAST LAND	SEA COAST LAND
SEA COAST LAND	SEA COAST LAND	SEA COAST LAND	SEA COAST LAND
SEA COAST LAND	SEA COAST LAND	SEA COAST LAND	SEA COAST LAND



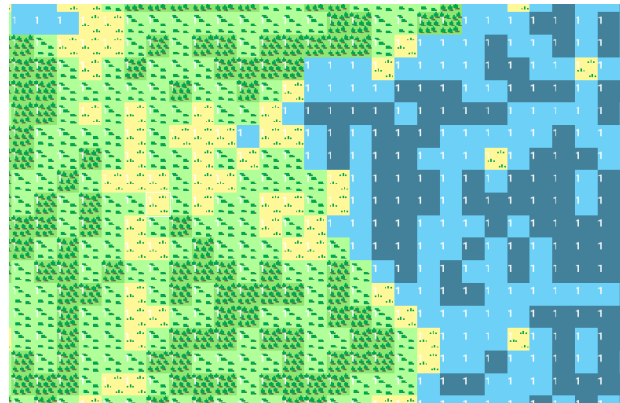
Algoritmul WFC

2. Alegem celula cu **entropia minimă** și se realizează **pasul de colapsare**. Se atribuie aleator celulei o placă de teren din setul de plăci posibile.



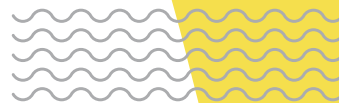
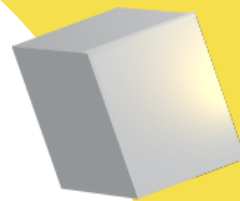
Algoritmul WFC

3. În **pasul de propagare** actualizăm celulele vecine prin eliminarea opțiunilor de plăci de teren care încalcă regulile de adiacență.
4. Pașii se repetă până când:
 - 4a. Gridul este complet și avem un teren final.
 - 4b. Nu mai sunt opțiuni valide, caz în care algoritmul revine asupra grilei (reia de la 0 pașii/urmează un pas de backtracking).



04

Implementarea WFC



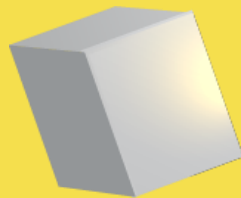


Implementarea WFC

Cerință

Pornind de la codul sursă de pe MS Teams, se vor completa cele trei TODOs pentru a genera hărți folosind WFC:

1. Cautăm celula cu entropia minimă și realizăm pasul de colapsare.
2. Se parcurge gridul și se caută celula cu cea mai mică entropie.
3. Afișarea texturilor de teren.



Resurse

B. Courcelle, “Graph rewriting: An algebraic and logic approach,” in *Elsevier eBooks*, 1990, pp. 193–242. doi: 10.1016/b978-0-444-88074-1.50010-x.

“Grid parts and relationships,” 2025. <https://www.redblobgames.com/grids/parts/>

“mxgmn/WaveFunctionCollapse: Bitmap & tilemap generation from a single example with the help of ideas from quantum mechanics,” *GitHub*. <https://github.com/mxgmn/WaveFunctionCollapse>

Computerphile, “Information Theory & Coding - Professor Brailsford - Computerphile,” YouTube Playlist. [Online]. Available: https://www.youtube.com/playlist?list=PLzH6n4zXuckpKAj1_88VS-8Z6yn9zX_P6

Cursuri - „Arhitectura Sistemelor de Calcul” – UB, FMI, Informatică An 1

The Coding Train, “Coding Challenge 171: Wave Function Collapse,” YouTube. Jul 3, 2022. [Online]. Available: https://www.youtube.com/watch?v=rl_y2GAIQFM&t=1065s

J. McCollum, “An introduction to graph rewriting for procedural content generation,” The Shaggy Dev, Nov. 20, 2022. <https://shaggydev.com/2022/11/20/graph-rewriting/>

