

# SIMULATIONS

## Systems Analysis

Author: Eng. Carlos Andrés Sierra, M.Sc.  
cavirguezs@udistrital.edu.co

Lecturer  
Computer Engineer  
School of Engineering  
Universidad Distrital Francisco José de Caldas

2024-III



UNIVERSIDAD DISTRITAL  
FRANCISCO JOSÉ DE CALDAS

# Outline

- 1 Basic Concepts
- 2 Cellular Automata



# Outline

## 1 Basic Concepts

## 2 Cellular Automata



# What is a simulation?

- **Simulations** are a sort of real-world representation, but rarely exactly the real world.
- Sometimes you need to **test** or **experiment** with expensive use cases. Simulations let **play** with different inputs, conditions, hyperparameter optimizations.
- Also, there are dangerous or hard to reach **stages** where **simulations** become the best option.
- You should define the **detail level** good enough to represent the expected behavior, without fail in both **high complexity** or **lazy simplicity**.



# What is a simulation?

- **Simulations** are a sort of real-world representation, but rarely exactly the real world.
- Sometimes you need to **test** or **experiment** with expensive use cases. Simulations let play with different inputs, conditions, hyper parameter optimizations.
- Also, there are dangerous or hard to reach stages where simulations become the best option.
- You should define the detail level good enough to represent the expected behavior, without fail in both high complexity or lazy simplicity.



# What is a simulation?

- **Simulations** are a sort of real-world representation, but rarely exactly the real world.
- Sometimes you need to **test** or **experiment** with expensive use cases. Simulations let play with different inputs, conditions, hyper parameter optimizations.
- Also, there are dangerous or hard to reach stages where **simulations** become the best option.
- You should define the **detail level** good enough to represent the expected behavior, without fail in both high complexity or lazy simplicity.



# What is a simulation?

- **Simulations** are a sort of real-world representation, but rarely exactly the real world.
- Sometimes you need to **test** or **experiment** with expensive use cases. Simulations let play with different inputs, conditions, hyper parameter optimizations.
- Also, there are dangerous or hard to reach stages where **simulations** become the best option.
- You should define the **detail level** good enough to represent the expected behavior, without fail in both high complexity or lazy simplicity.



# Events and Stochastic Processes

- It is typical to play with **events probability**, creating **stochastic behaviors**. That is how *reality works*.
- One way to **simulate** a lot of **systems** is to use **event-based models**. Embrace the chaos and Murphy's Law.





# Events and Stochastic Processes

- It is typical to play with **events probability**, creating **stochastic behaviors**. That is how *reality works*.
- One way to **simulate** a lot of **systems** is to use **event-based models**. Embrace the chaos and Murphy's Law.



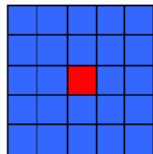
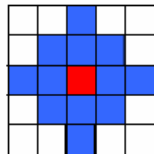
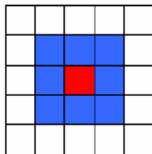
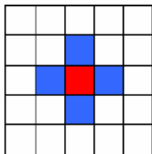
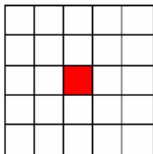
# Outline

- 1 Basic Concepts
- 2 Cellular Automata



# Cellular Automatas

- **Cellular Automata** are a **discrete model** defined by a **grid** of cells, each one with a **state**.
- The **state** of a cell is updated based on the **state of its neighbors**.



Cellular Automata Typical Neighborhoods



# Game of Life

- **Game of Life** is a **cellular automaton** devised by the British mathematician **John Horton Conway** in 1970.
- It is a **zero-player game**, meaning that its **evolution** is determined by its **initial state**, requiring no further input.
- Rules:
  - Any **live** cell with **fewer than two live neighbors** **dies**, as if by underpopulation.
  - Any **live** cell with **two or three live neighbors** **lives** on to the next generation.
  - Any **live** cell with **more than three live neighbors** **dies**, as if by overpopulation.
  - Any **dead** cell with **exactly three live neighbors** becomes a **live** cell, as if by reproduction.



# Game of Life

- **Game of Life** is a **cellular automaton** devised by the British mathematician **John Horton Conway** in 1970.
- It is a **zero-player game**, meaning that its **evolution** is determined by its **initial state**, requiring no further input.
- Rules:
  - Any **live** cell with **fewer than two live neighbors** **dies**, as if by underpopulation.
  - Any **live** cell with **two or three live neighbors** **lives** on to the next generation.
  - Any **live** cell with **more than three live neighbors** **dies**, as if by overpopulation.
  - Any **dead** cell with **exactly three live neighbors** becomes a **live** cell, as if by reproduction.



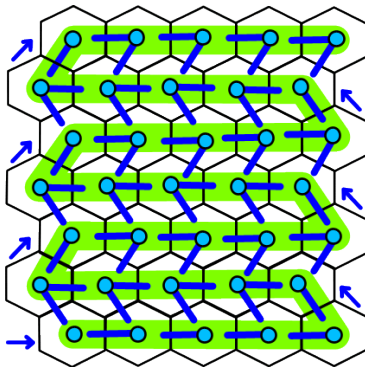
# Game of Life

- **Game of Life** is a **cellular automaton** devised by the British mathematician **John Horton Conway** in 1970.
- It is a **zero-player game**, meaning that its **evolution** is determined by its **initial state**, requiring no further input.
- Rules:
  - Any **live** cell with **fewer than two live** neighbors **dies**, as if by underpopulation.
  - Any **live** cell with **two or three live** neighbors **lives** on to the next generation.
  - Any **live** cell with **more than three live** neighbors **dies**, as if by overpopulation.
  - Any **dead** cell with **exactly three live** neighbors becomes a **live** cell, as if by reproduction.



# HoneyComb Cellular Automata

- **HoneyComb Cellular Automata** is a different topology where a cell has **six neighbors**.
- This representation has different **dispersion properties**, sometimes, more interesting.



# SIR Model

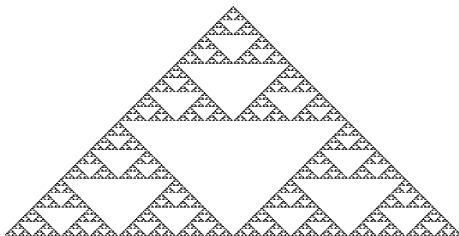
- **SIR Model** is a **compartmental model** used to represent the **transmission** of a **contagious disease**.
- The **model** divides the population into three compartments: **S** for the number of **susceptible**, **I** for the number of **infected**, and **R** for the number of **recovered**.
- The model is defined by the following **differential equations** where  $\beta$  is the **transmission rate** and  $\gamma$  is the **recovery rate**:
  - $\frac{dS}{dt} = -\beta \cdot S \cdot I$
  - $\frac{dI}{dt} = \beta \cdot S \cdot I - \gamma \cdot I$
  - $\frac{dR}{dt} = \gamma \cdot I$





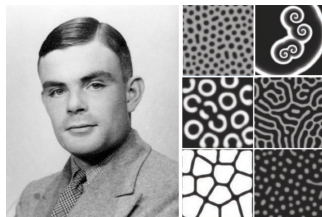
# Chaotic Systems

- **Chaotic Systems** are a class of **dynamical systems** that exhibit **sensitive dependence** on initial conditions.
- This means that the **future behavior** of the system is **highly dependent** on the **initial conditions**.
- The **Lorenz System** is a well-known example of a **chaotic system**.
- Using **cellular automata** to simulate **chaotic systems** is a common practice. A lot of **fractals** can be created using something called **chaotic rules**.



# Turing Morphogenesis

- **Turing Morphogenesis** is a theory of **biological development** that explains how **patterns** form in **living organisms**.
- The theory is based on the idea that **chemical signals** can interact to create **patterns** in a **cellular automaton**.
- The **reaction-diffusion** model is a common way to simulate **Turing morphogenesis**.
- The model is defined by a set of **reaction** and **diffusion** equations that describe how the **chemical signals** interact.



# Outline

- 1 Basic Concepts
- 2 Cellular Automata



# Thanks!

## Questions?



Repo: <https://github.com/EngAndres/ud-public/tree/main/courses/systems-analysis>

