POLITECNICO
MILANO 1863

IMAGE AND SOUND
ISPG
PROCESSING GROUP

CREATIVE PROGRAMMING AND COMPUTING

Multi-agent creative systems

MULTI-AGENT SYSTEMS

- Several behaviours in the animal kingdom exhibits similarities with reactive-agents
- Specifically swarm behaviour is highly fascinating

- Swarm of ants

MULTI-AGENT SYSTEMS

- Each ant can be seen as a reactive agent
- They do not exabits a proper knowledge of the world
- They do not build a symbolic map of the world but they are react with actions to specific input

- Rather than a **single "intelligence",** a **swarm "intelligence"** emerges which is the behaviour resulting by the sum of the behaviours of the single ants and the interaction between them
- Single agents are very simple systems; multi-agent systems can be very complex

- Single behaviour can be modelled using

  input -> rule selection -> action

- Interaction between agents

  social behaviour design

SOCIAL BEHAVIOUR DESIGN

- Design the behaviour of single agents in the context (environment) where multiple agents coexists

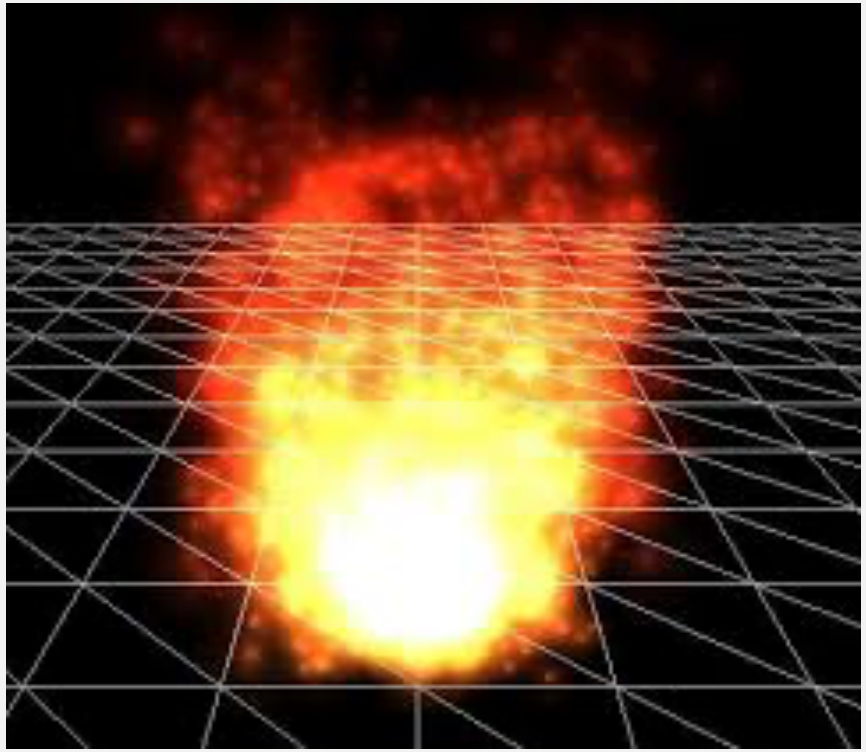
- Different types of interaction
  - **No interaction**: agents do not interact, they do not have the knowledge of the existence of the others
  - **Personal goal**: each agent has a goal, it has the knowledge of the existence of the others and interacts with other in order to accomplish its goal
  - **Collective goal**: the complex systems of agents has a goal and single agents give the personal contribute to reach the goal
    - In this scenario agents may communicate

PARTICLE SYSTEMS

DEFINITION

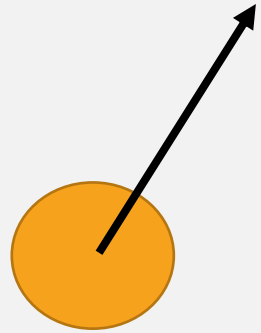
- *"A particle system is a collection of many many minute particles that together represent a fuzzy object. Over a period of time, particles are generated into a system, move and change from within the system, and die from the system."*
- 1982, William T. Reeves, a researcher at Lucasfilm Ltd., was working on the film *Star Trek II: The Wrath of Khan*

They are interesting example of complex systems



MULTI-AGENTS – NO INTERACTION

- In many cases particles in the particle system do not interact
- Let's consider a single particle
  - It is characterized by 3 PVectors: **Location**, **Velocity**, **Acceleration**
  - and a value: **LifeSpan**



  - **LifeSpan** represents the time the Particle will be alive

- It is possible to add many particles and the global behaviour is given by the Location, Velocity, Acceleration and LifeSpan of the single particles

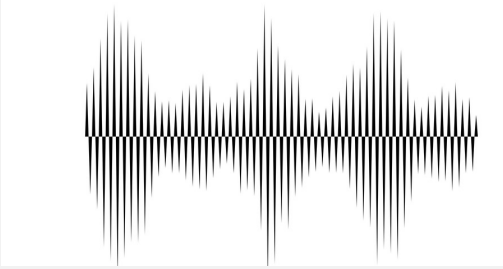
SingleParticle

SimpleParticleSystem

http://refikanadol.com/works/wdch-dreams/

MULTI-AGENTS – NO INTERACTION

- A use case: wind affect the particle system based on audio



Wind audio

Compute energy

Convert energy into force

Particle_System_wind

MULTI-AGENTS - PERSONAL GOAL

- Each agents has a personal system of rules and a personal goal
- Agents interact with the other to reach the goal
- Some examples:
  - **Avoid collision (Repel)**
  - **Attraction**
  - **Connection: particles are connected through spring**

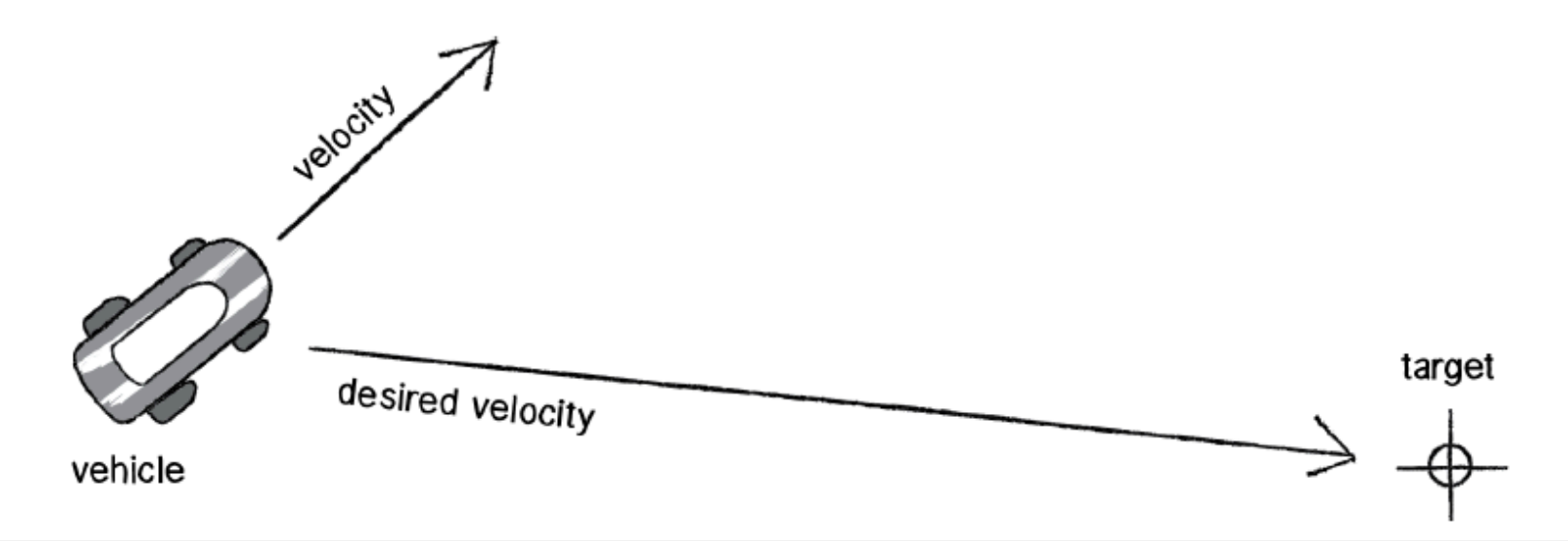
- All of these can me modelled using Location, Velocity, Acceleration and LifeSpan for each single particle

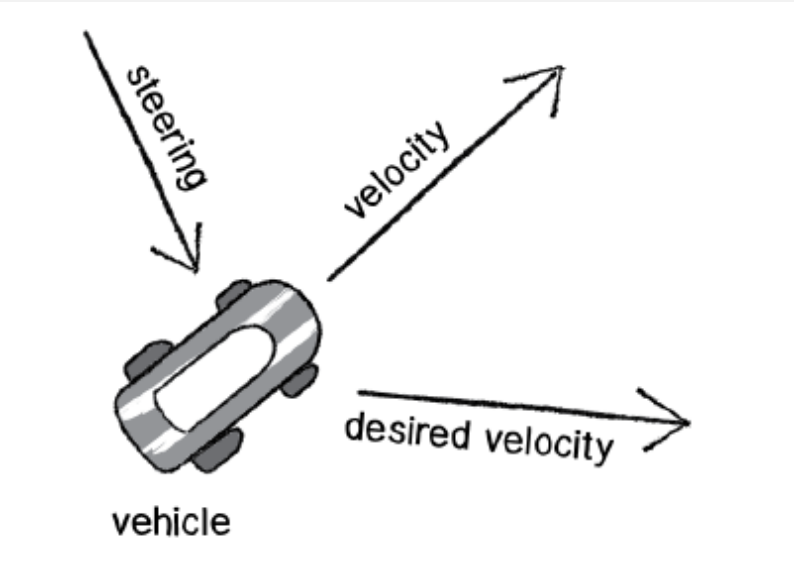
- In **Processing:**
  - **Box2D** and **toxiclibs** are two physics engines

MULTI-AGENTS - PERSONAL GOAL

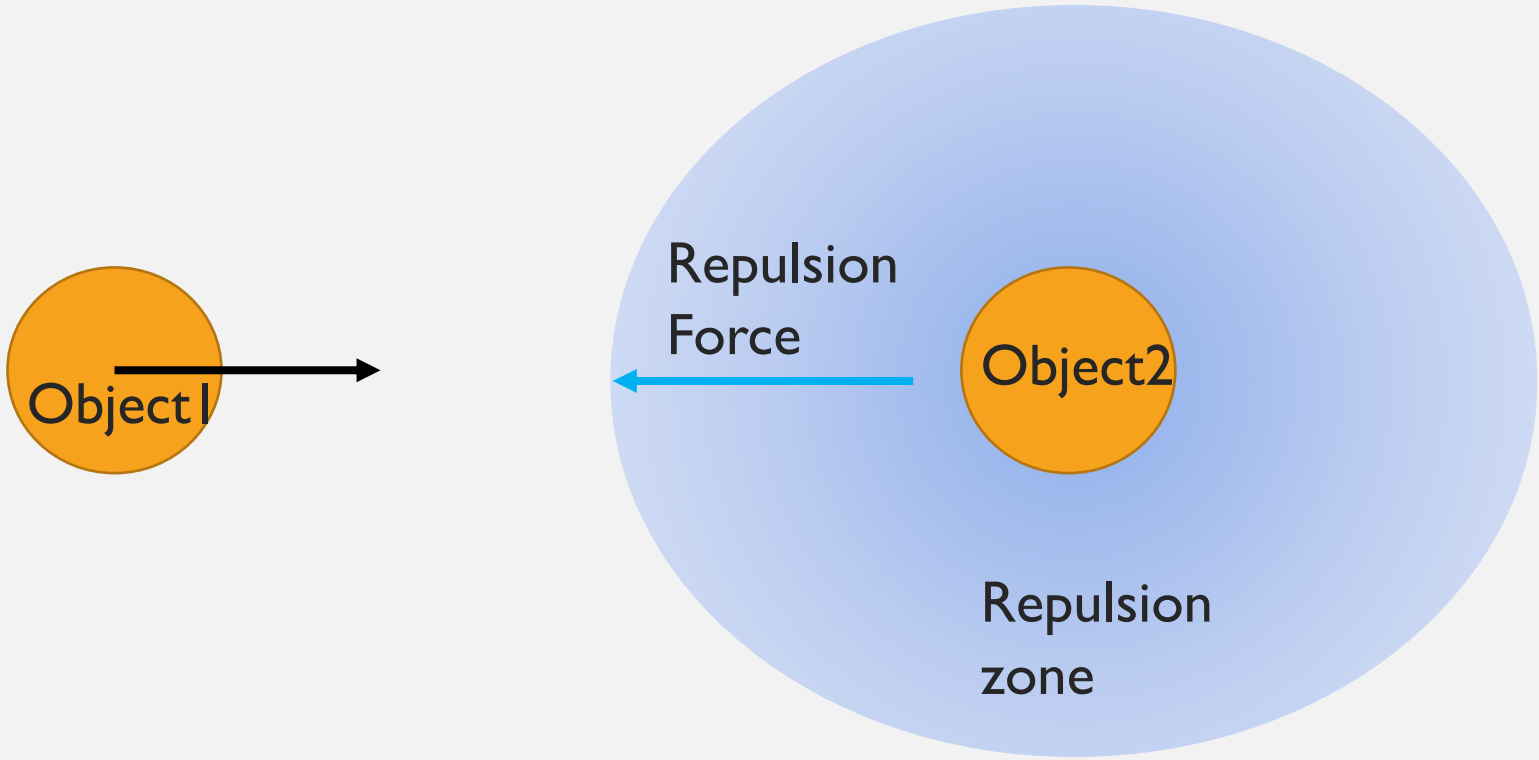
- To model attraction, repulsion and connection we use the steering force model

steering force = desired velocity - current velocity



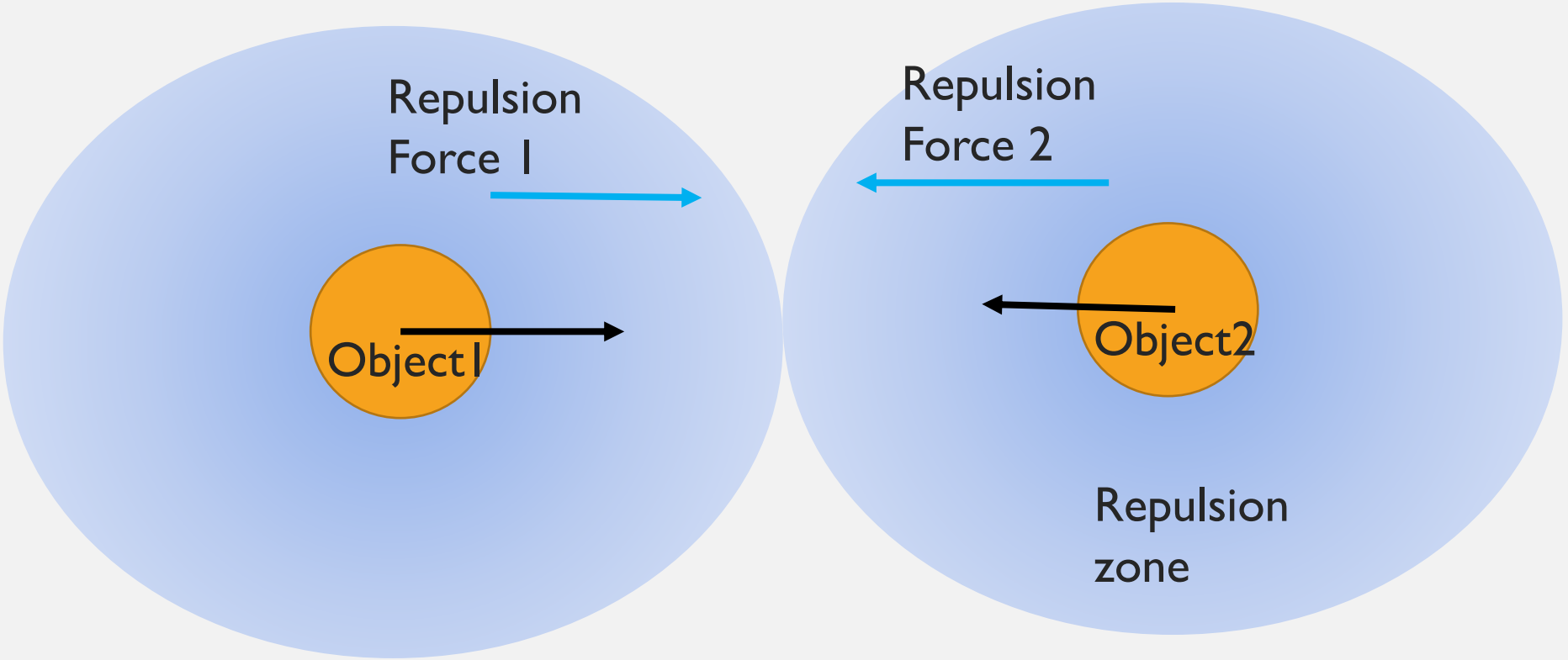


MULTI-AGENTS - REPULSION



- Repulsion Force increases getting closer to the object 2
- Repulsion Force is applied inversely to the velocity of the object 1

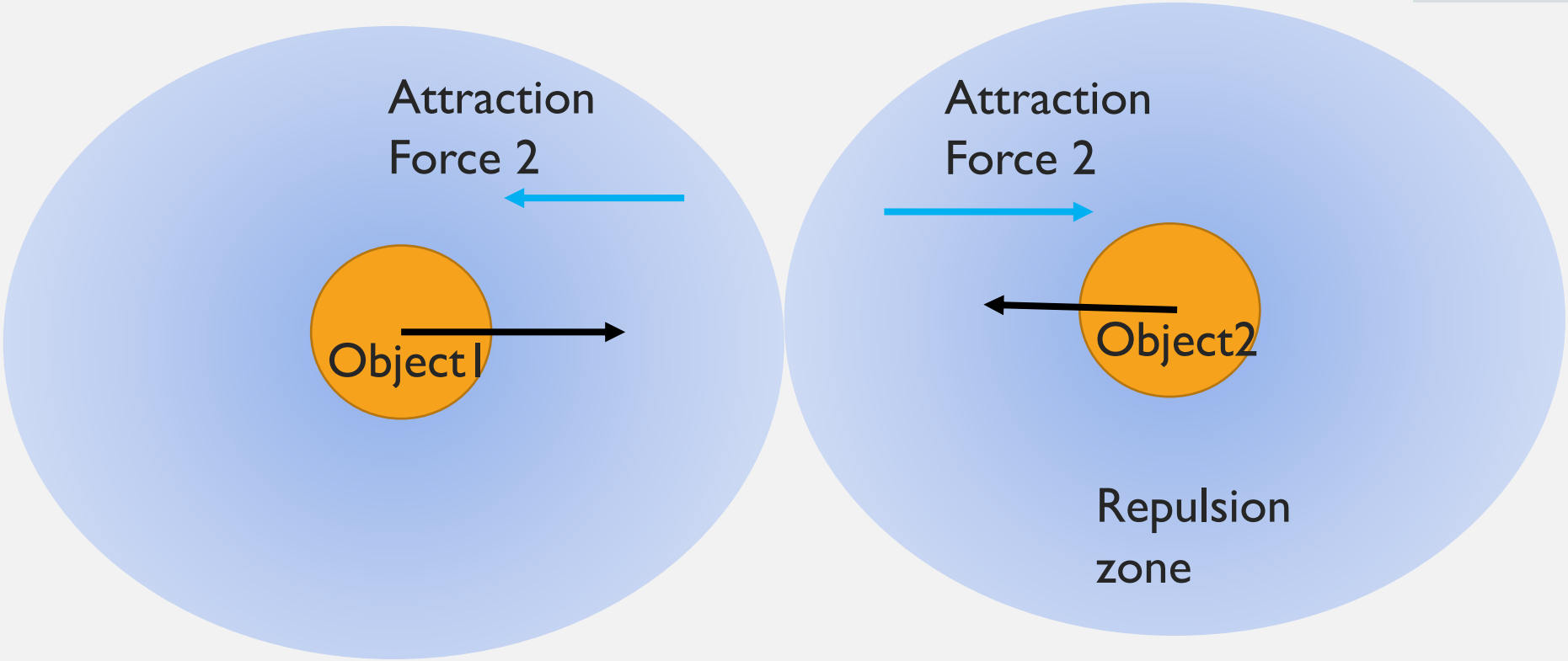
MULTI-AGENTS - REPULSION



- Repulsion Force 1 is applied inversely to the velocity of the object 2
- Repulsion Force 2 is applied inversely to the velocity of the object 1

MULTI-AGENTS - ATTRACTION

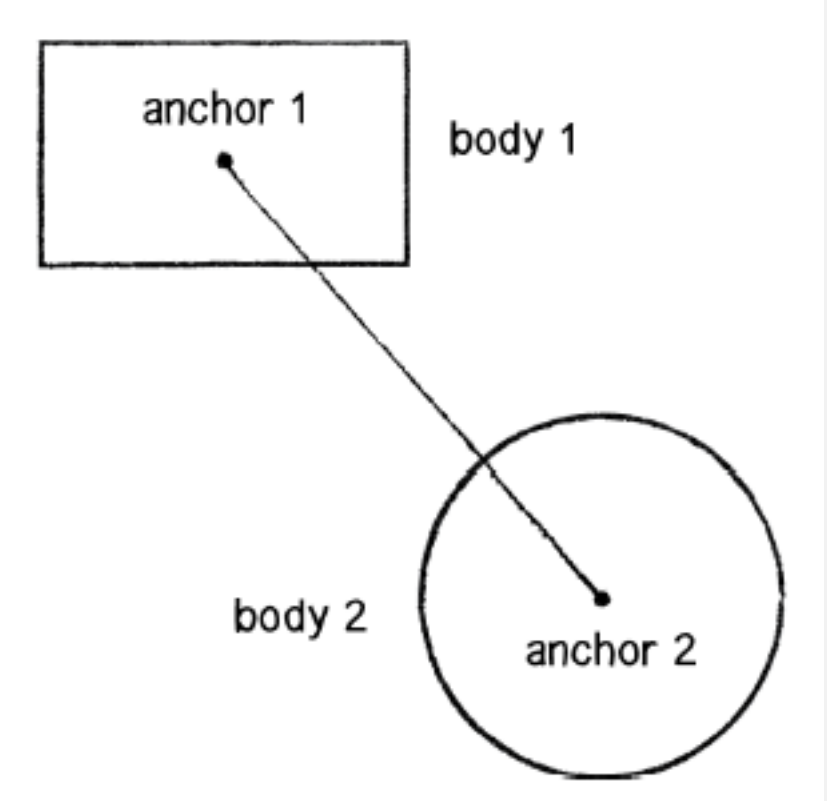
toxiclibs_AttractRepel



- Attraction Force 1 is applied accordingly to the velocity of the object 2
- Attraction Force 2 is applied accordingly to the velocity of the object 1

MULTI-AGENT - JOINTS

- Objects can be joints together
- We need to model the joints e its rigidity

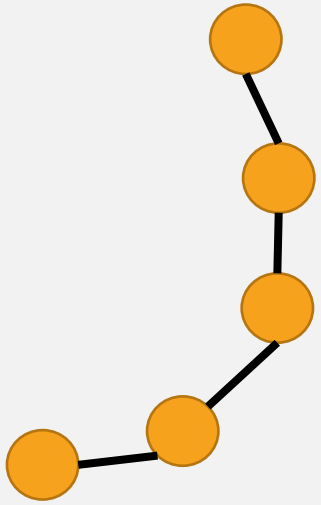


toxiclibs_SimpleSpring

toxiclibs_SoftBodySquareAdapted

MULTI-AGENT – FLEX JOINTS

- Flex joints can be modelled as a chain of simple joints



toxiclibs_SoftStringPendulum

PARTICLE SYSTEMS - EXAMPLE

https://www.youtube.com/watch?time_continue=78&v=N8Sed-c1sJI

Prophet in the wind by O3LAB

BOIDS

SWARM INTELLIGENCE - THEORY

- *The whole is greater than the sum of its parts"* is a well-known quote that according to the **Gestalt psychology** sums up the idea that a system
- The whole is something more complex and different from the aggregation of its basic elements

- The attempt to understand a complex system trying to dissect it into its basic parts would inevitably fail

- Swarm intelligence mimics complex swarm behaviour in animal kingdom

SWARM INTELLIGENCE - THEORY

- Self-organisation (SO), the science of emergence, allude to the pre-conditions for the emergence of large scale forms from local influences.
- Bonabeau et al (1999) propose that SO relies on multiple interactions between component parts of a system, an ability to amplify **fluctuations**, and **positive** and **negative** feedback between components.

- Positive feedback allows reinforcement of new forms.
- Negative feedback stabilises the system and prevents runaway.
- Random fluctuations play a crucial role in SO, enabling the system to find novel situations, which are exploitable through positive feedback.

SWARM INTELLIGENCE - THEORY

- Let's observe ant colonies
- when an ant leaves the nest in search of food, she would lay down a pheromone trail that the other ants are able to follow
- if she reaches some food, she will return to the nest — using the same path — leaving even more pheromone on her way back
- If she does not find any food, she will not lay down any pheromone in the way back to the nest and the previous trail will evaporate away
- Some ant can leave trail to find other routes -> fluctuation

- This indirect communication mechanism based on pheromone is at the basis of the knowledge sharing within the colony

SWARM INTELLIGENCE - THEORY

- Real-life swarms organise themselves into remarkable, beautiful spatio-temporal structures in a process known as **self-organisation**.
- This organisation is thought to arise from the instantaneous dynamics of the swarming creatures, and not by any central leadership.
- Swarming animals **communicate** with each other over long time scales through the modification of the environment in a biological process known as **stigmergy**.

- Evidence that flocks and swarms are self-organising is provided by the "**boid**" animations of Reynolds (Reynolds 1987).
- **The collective behaviour of the group is emergent because the rules concerning the parts of the swarm do not contain any notion of the whole.**

SWARM INTELLIGENCE - THEORY

- Real-life swarms organise themselves into remarkable, beautiful spatio-temporal structures in a process known as **self-organisation**.

- This organisation is thought to arise from the instantaneous dynamics of the swarming creatures, and not by any central leadership.

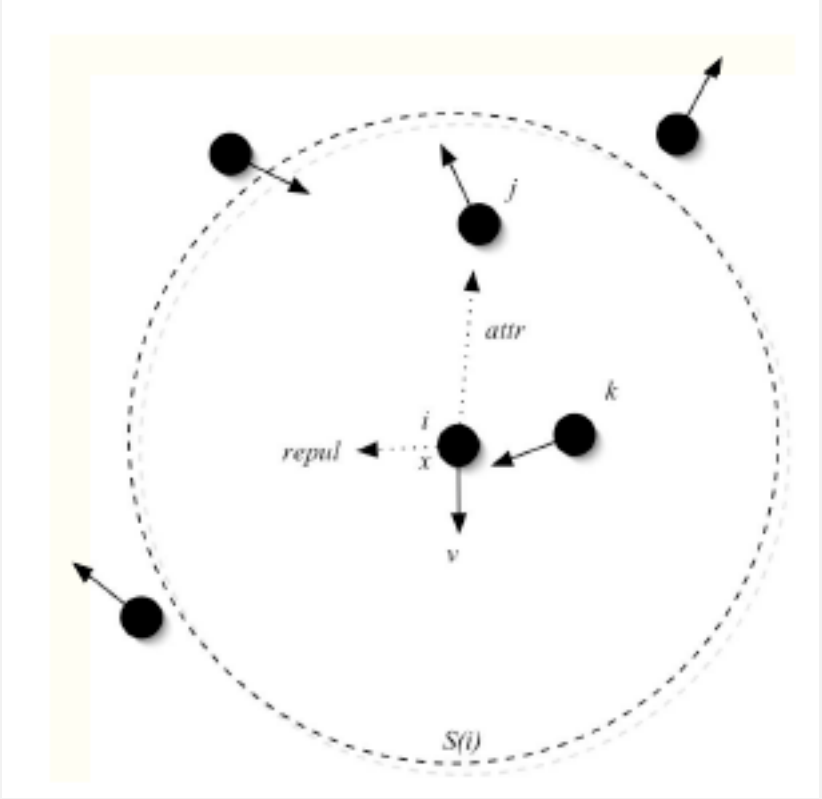
- Swarming animals **communicate** with each other over long time scales through the modification of the environment in a biological process known as **stigmergy**.

BOIDS - THEORY

- Evidence that flocks and swarms are self-organising is provided by the "**boid**" animations of Reynolds (Reynolds 1987).

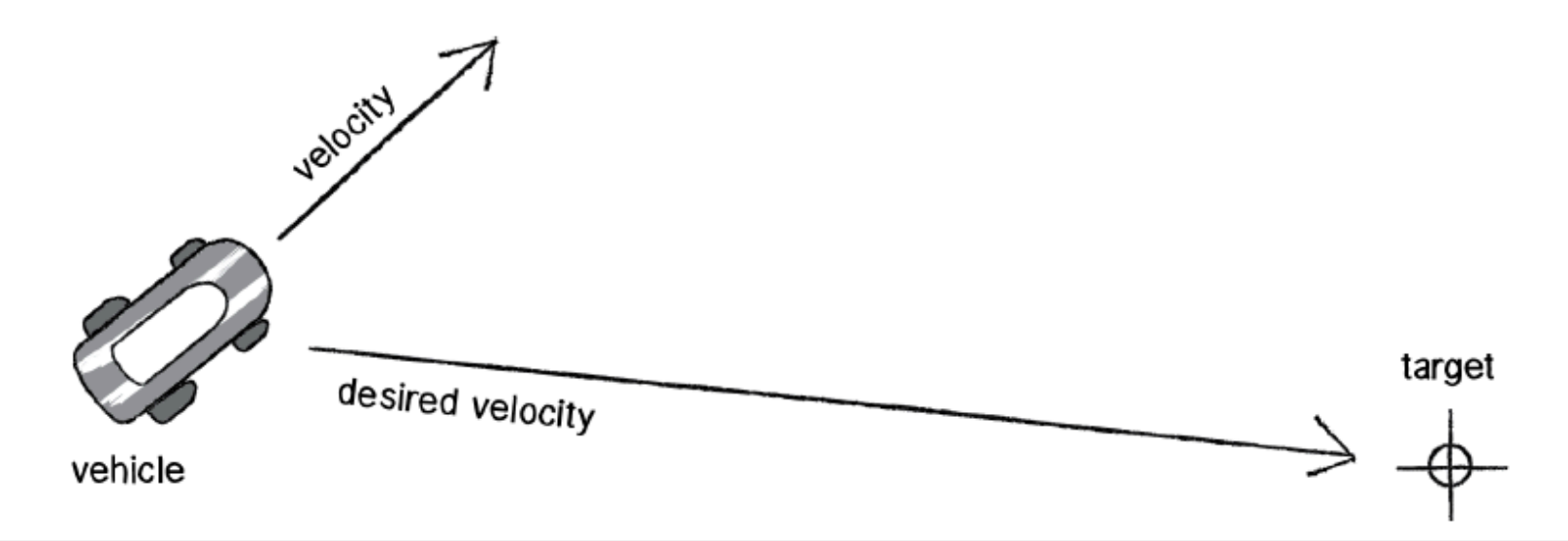
- **The collective behaviour of the group is emergent because the rules concerning the parts of the swarm do not contain any notion of the whole.**

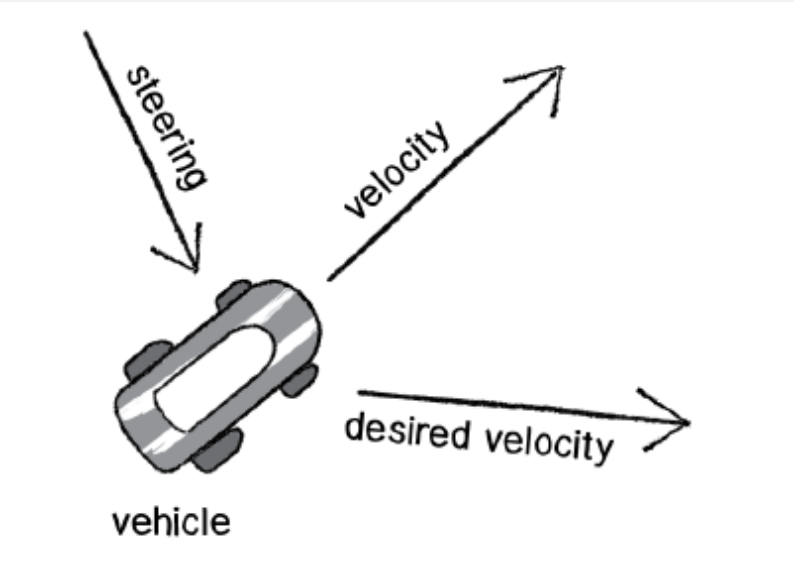
- Swarms assume that the individuals have a finite range of perception in which a given individual feels the influence of neighbours.
- Individuals **repel** each other at close range, **attract** each other at medium range and are **oblivious** to each other at long range



BOIDS – COMPUTER GRAPHICS

- The attractions provide coherence, maintaining a shared neighbourhood (colony) and the repulsions prevent collisions
- The attractive and repulsive accelerations are the analogues of positive and negative feedback
- Let's consider the steering rules used in agents





steering force = desired velocity - current velocity

BOIDS – COMPUTER GRAPHICS

- Set of rule to update each boid:

1. We will use the steering force formula (steer = desired - velocity) to implement the rules of flocking.

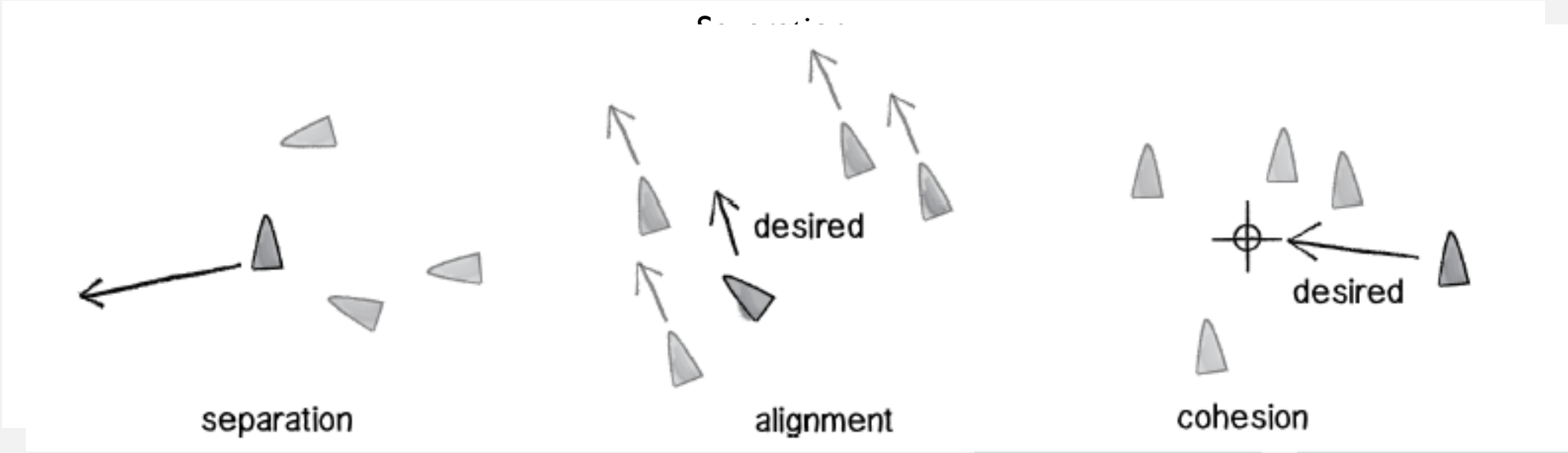
2. These steering forces will be group behaviors and require each vehicle to look at all the other vehicles.

3. We will combine and weight multiple forces.

4. The result will be a complex system—intelligent group behavior will emerge from the simple rules of flocking without the presence of a centralized system or leader.

BOIDS – COMPUTER GRAPHICS

- The basic rules governing the interactions between neighbouring particles in boids are:
- 1. **Separation** (also known as "**avoidance**"): Steer to avoid colliding with your neighbors.
- 2. **Alignment** (also known as "**copy**"): Steer in the same direction as your neighbors.
- 3. **Cohesion** (also known as "**center**"): Steer towards the center of your neighbors (stay with the group).

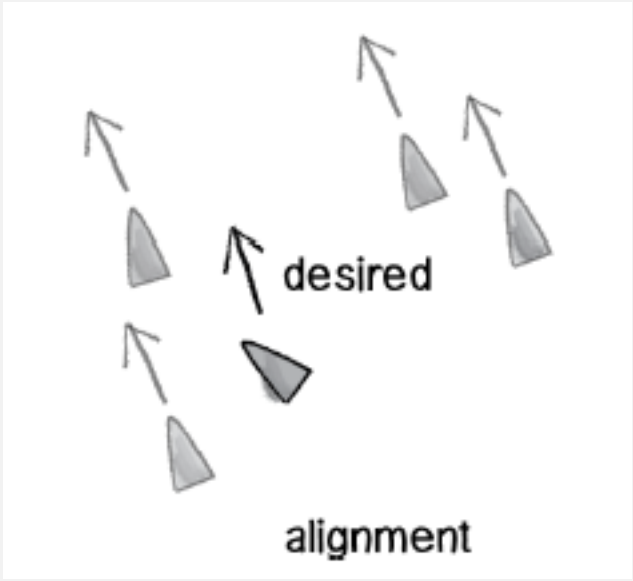


Separation **Cohesion**

BOIDS - ALIGNEMENT

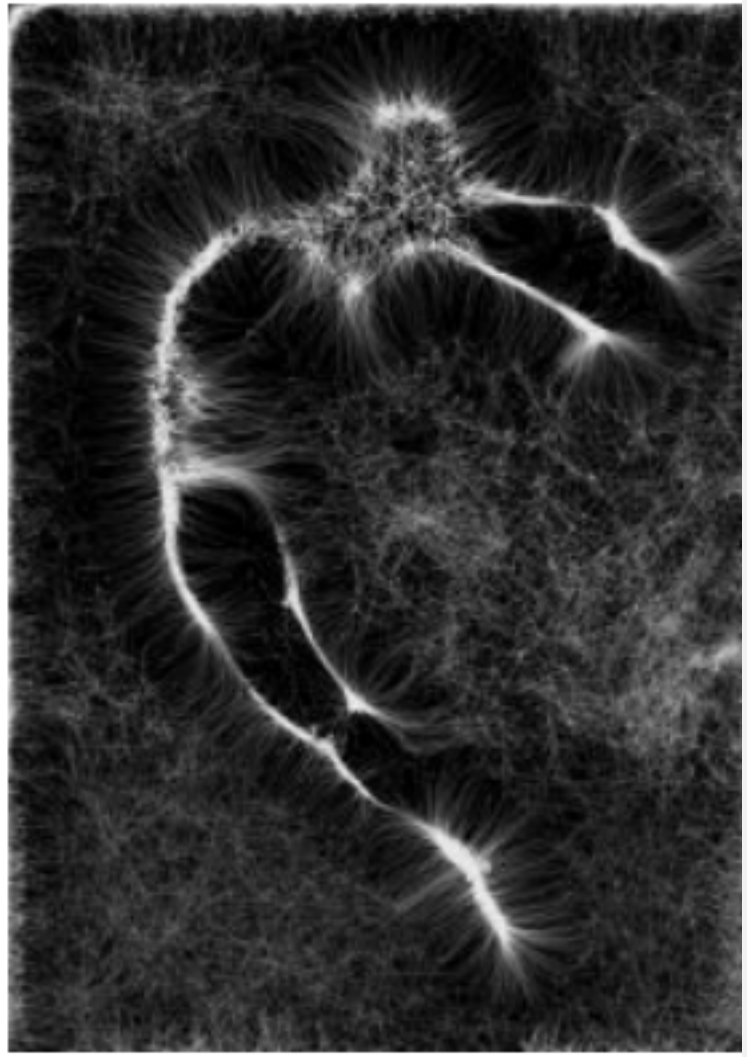
Alignement

Flocking



- Given a boid
- Compute the average velocity vector Vavg of the neighbours in a certain range
- Set the desired velocity of the Boid to Vavg
- Apply steering force = desired velocity - current velocity

SWARM PAINTING



https://vimeo.com/139853937

Doxia studio

Virtual agents explore the world painting small trails to great paintings.

These agents have Artificial Intelligence and move by themselves. each bright pixel of the source image works as food for them so they try to catch them before they die from hunger.

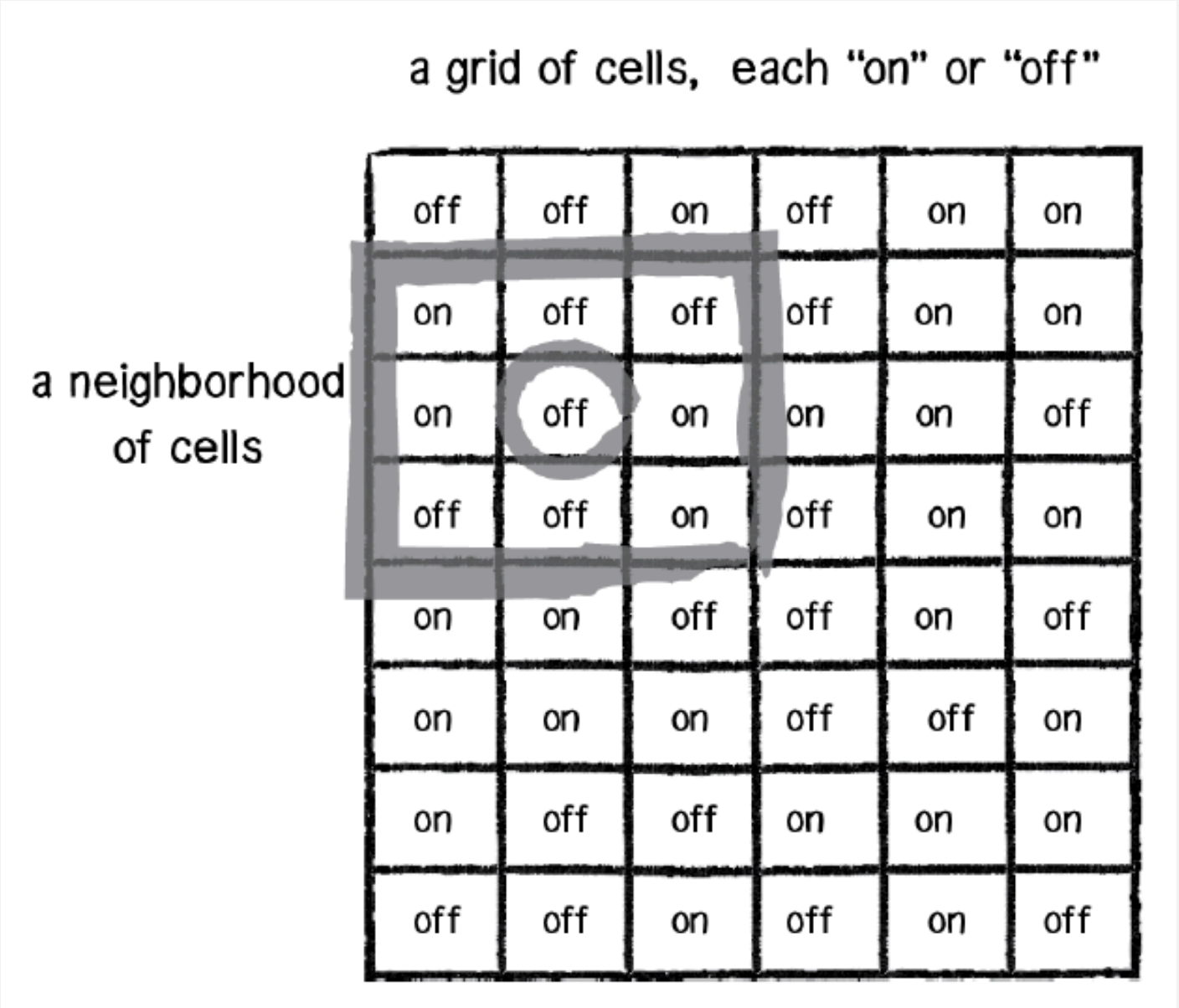
Flocking_painter

CELLULAR AUTOMATA

CELLULAR AUTOMATA - THEORY

- Developed in 1950 by Konrad Zuse, Stanislav Ulam, John Vin Neuman
- A CA consists in:
  - A **universe:** A n-dimensional grid of cells
  - **States**: Each cell has a state. The number of state possibilities is typically finite. The simplest example has the two possibilities of 1 and 0 (otherwise referred to as "on" and "off" or "alive" and "dead").
  - Each cell has a **neighbourhood**. This can be defined in any number of ways, but it is typically a list of adjacent cells.
  - A **transition rule:** indicates for a given cell what state it should be at in the next time step given: (a) its current state and (b) the states of the cells in the neighbourhood
  - Starting from an initial state at time t0 the transition rule is **applied to all the cell in parallel** to produce the next time step states
  - Time step is also call **iteration** of **generation**

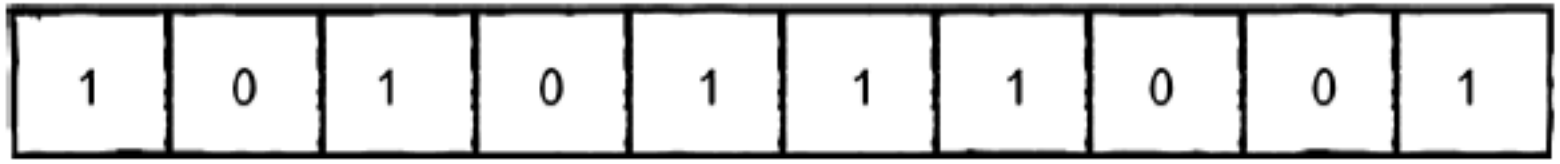
CELLULAR AUTOMATA - THEORY



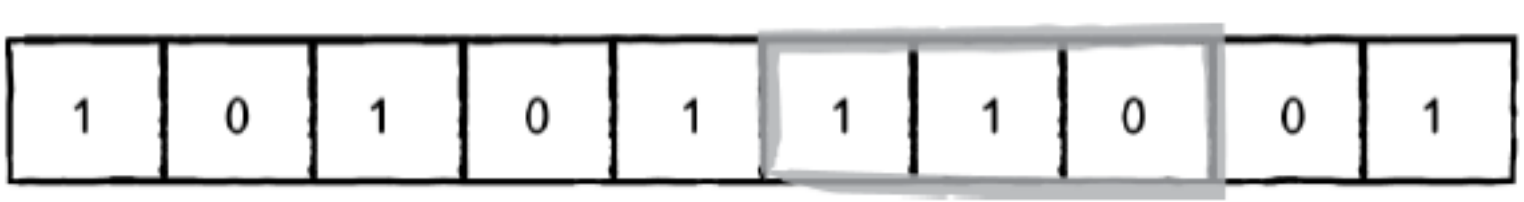
CA – 1D GRID

- The simplest case
- **Grid**. The simplest grid would be one-dimensional: a line of cells

- **States**. The simplest set of states (beyond having only one state) would be two states: 0 or 1

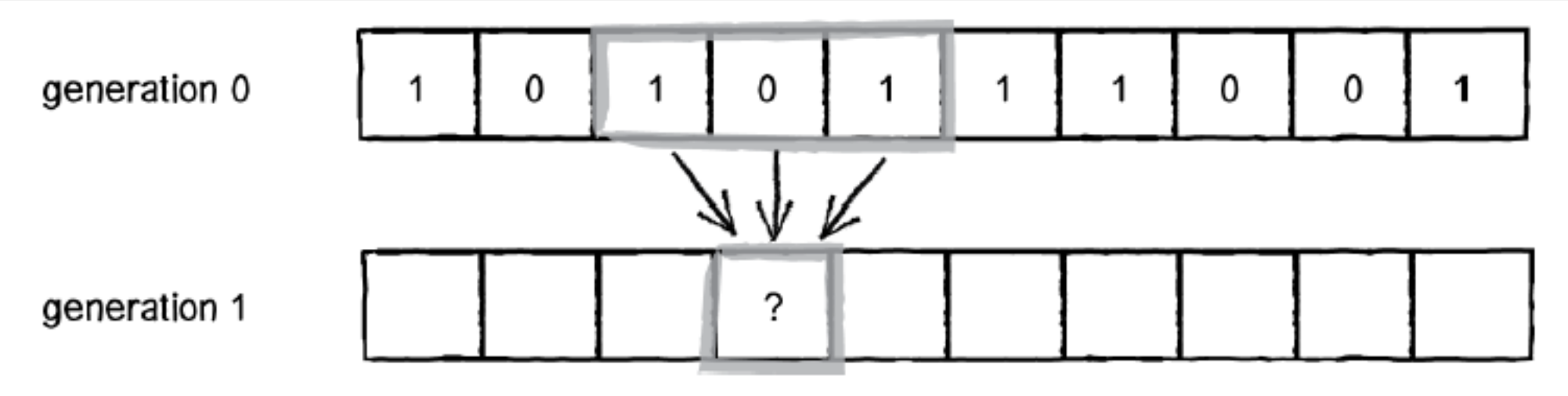


- **Neighbourhood**. The simplest neighbourhood in one dimension for any given cell would be the cell itself and its two adjacent neighbours: one to the left and one to the right.



CA – 1D GRID

- Transitional rules



- We can look at all the possible configurations of a cell and its neighbour and define the state outcome for every possible configuration

CA – 1D GRID

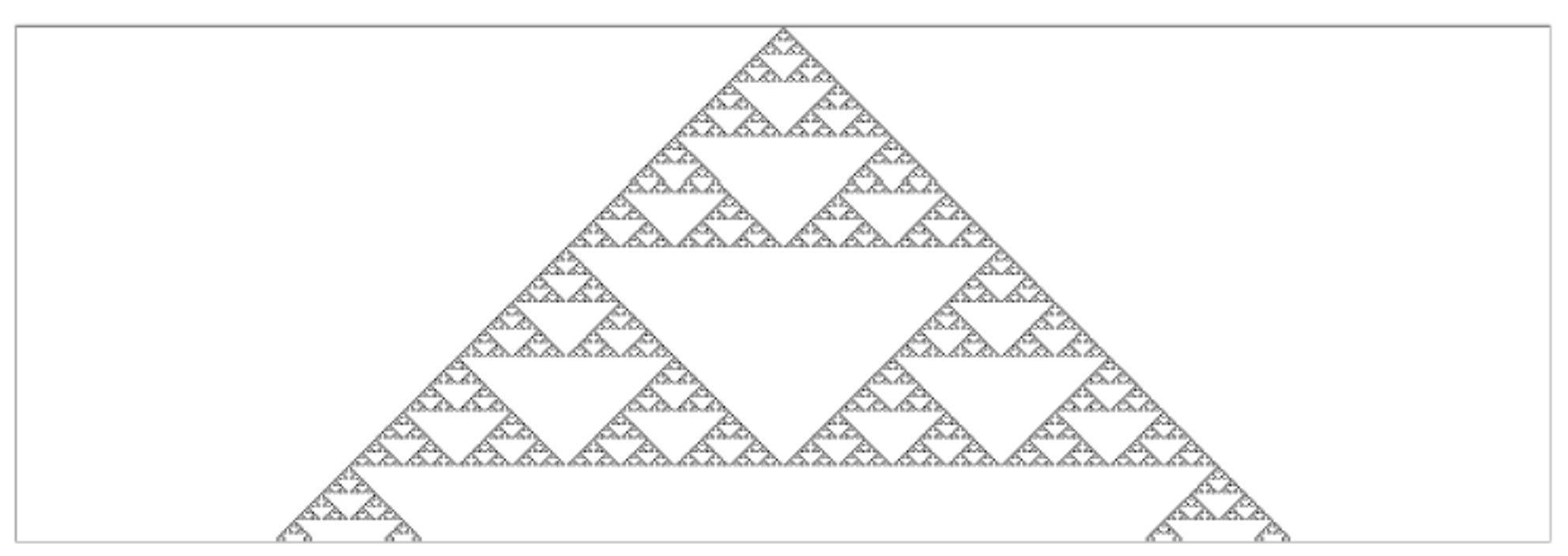
- The standard Wolfram model

Neighborhood	111	110	101	100	011	010	001	000
Resulting state	0	0	0	1	1	1	1	0

Initial State							X							t_0
Generation 1						X	X	X						t_1
Gen 2					X	X			X					t_2
Gen 3				X	X		X	X	X	X				t_3
Gen 4			X	X			X				X			t_4
Gen 5		X	X		X	X	X	X		X	X	X		t_5
Gen 6	X	X			X					X			X	t_6

CA – 1D GRID

- The standard Wolfram model



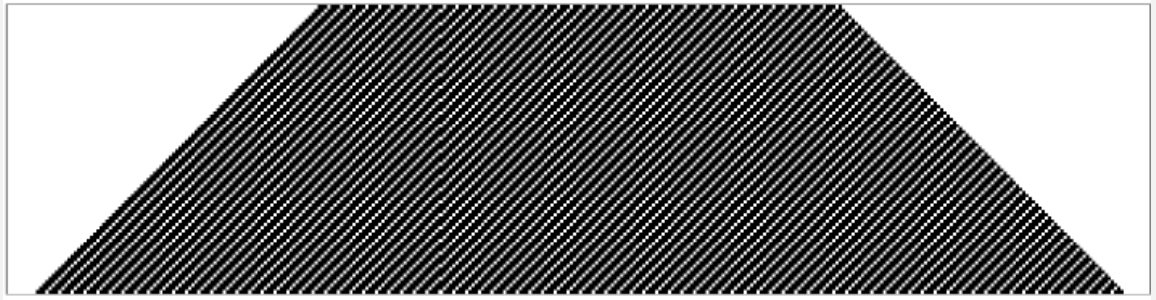
- The above ruleset is commonly referred to as "Rule 90" because if you convert the binary sequence—01011010—to a decimal number, you'll get the integer 90

WolframCA_simple

CA – 1D GRID

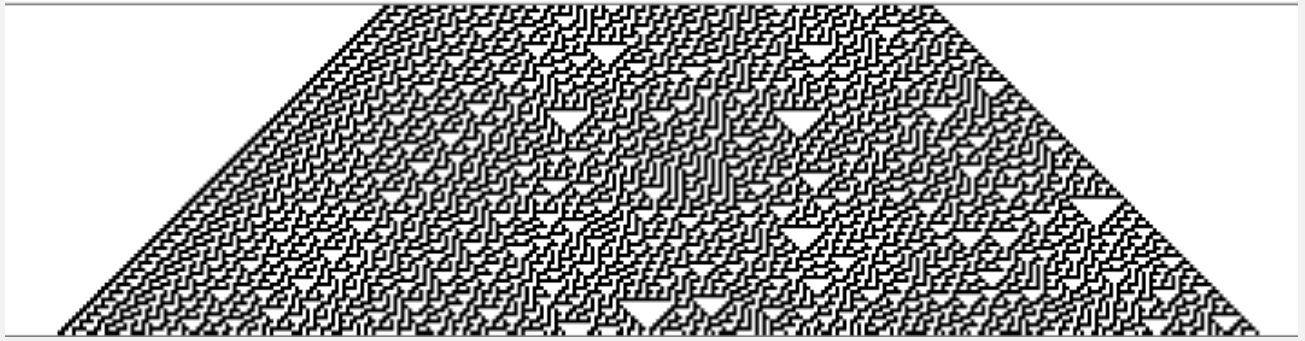
- Wolfram classified the 1d cellular automata into four classes
- Class1: simple behaviour and evolves to homogenous, constant final state

- Class 2: different final states consisting of simple stable or periodic repetitive structures

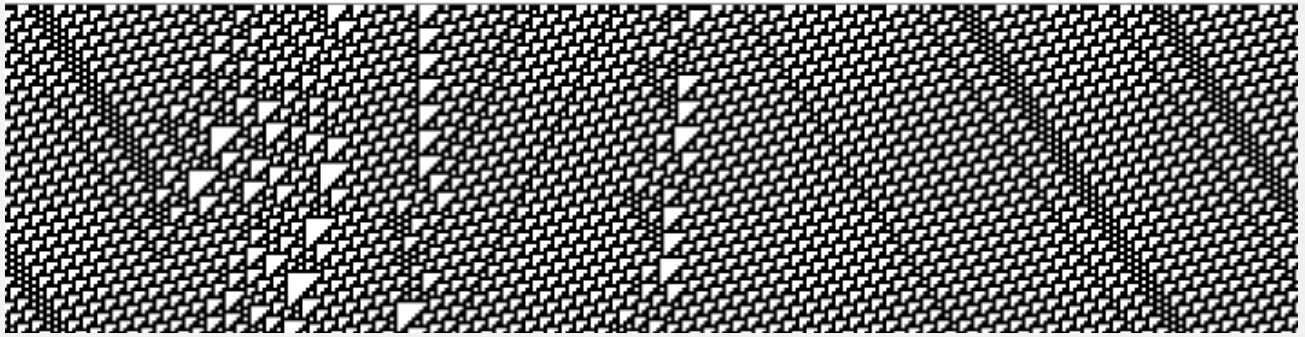


CA – 1D GRID

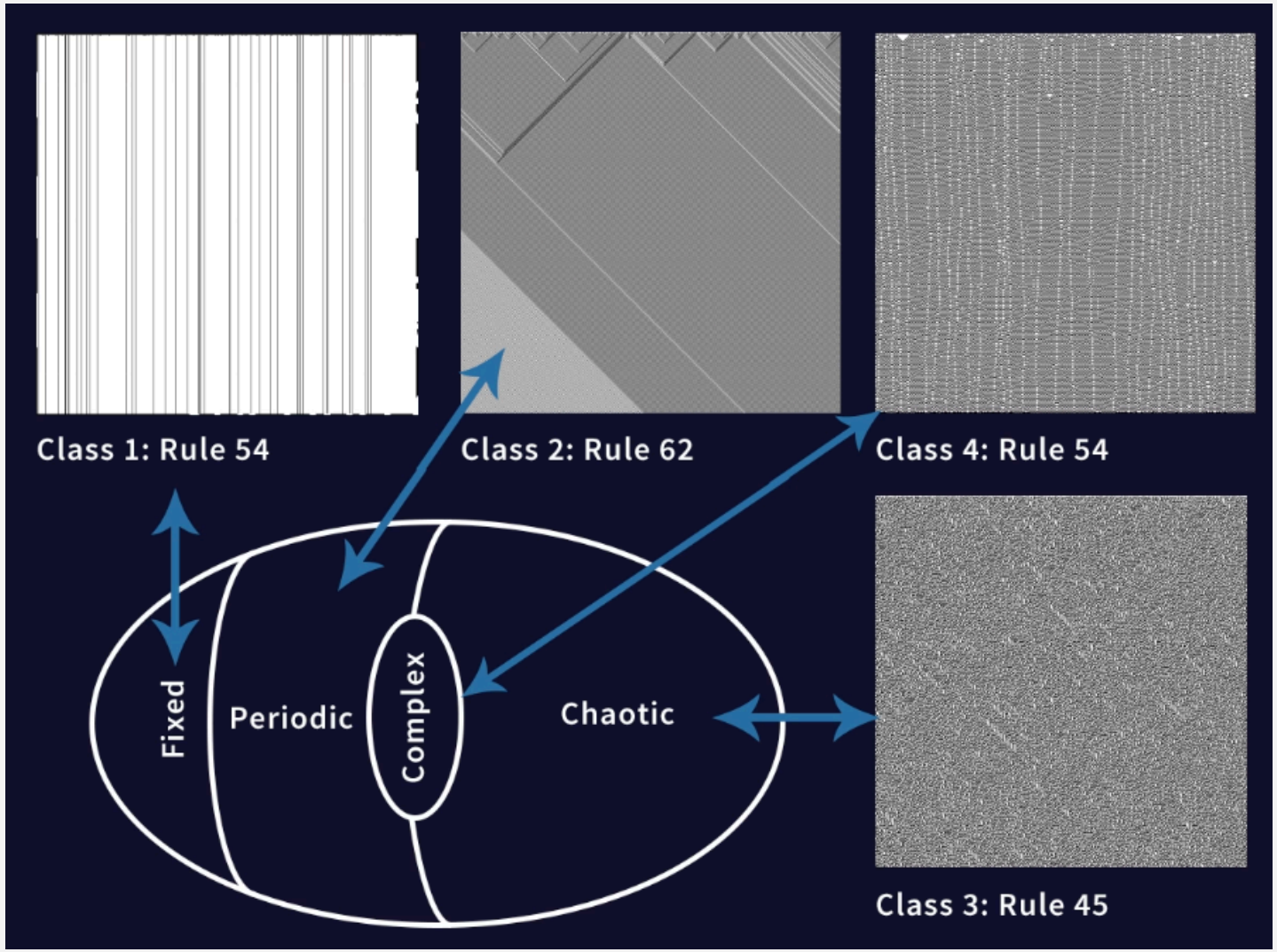
- Wolfram classified the 1d cellular automata into four classes
- Class3: chaotic behaviour



- Class 2: produces complex patterns which may also repeat self-similarity



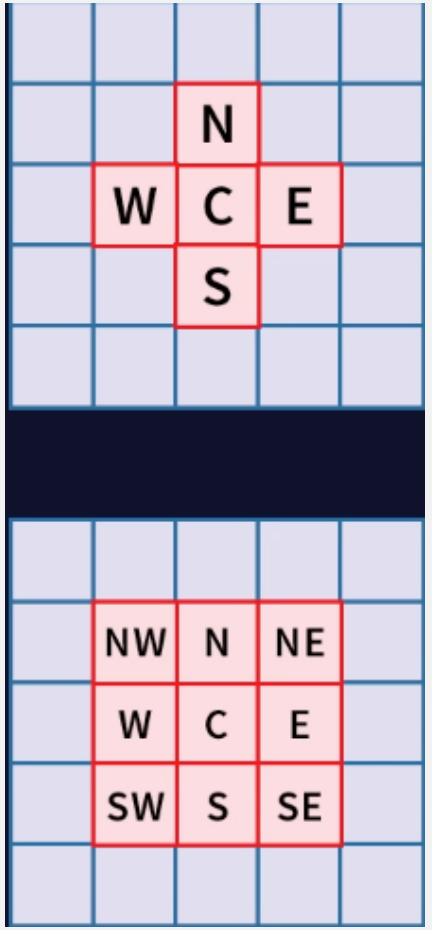
CA – 1D GRID



https://www.youtube.com/watch?v=dQJ5aEsP6Fs

CA – 2D GRID

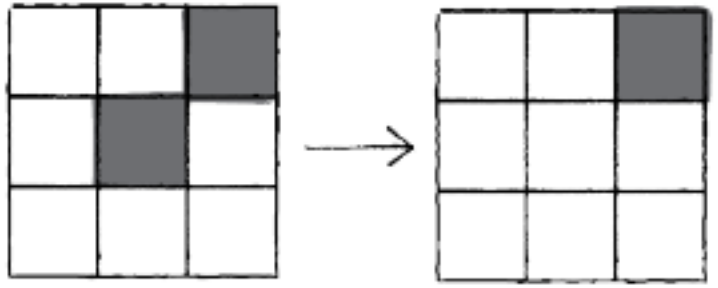
- 2D grid cellular automata
- Each cell will have a bigger neighbourhood, but that will open up the door to a range of possible applications.
- Two types of neighbourhood:
  - Von Neumann neighbourhood
  - Moore neighbourhood



CA – 2D GRID

- In 1970 John Conway's proposed the "**Game of Life**"
- It uses the Moore behaviour
- With nine cells, we have 9 bits, or 512 possible neighbourhoods.

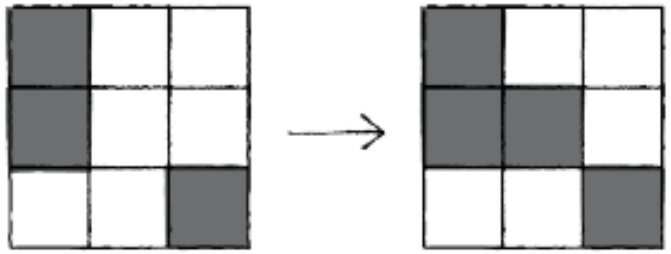
- The Game of Life gets defines a set of rules according to general characteristics of the neighbourhood.
- **Death**. If a cell is alive (state = 1) it will die (state becomes 0) under the following circumstances:
  - **Overpopulation**: If the cell has three or more alive neighbours, it dies.
  - **Loneliness**: If the cell has one alive neighbours, it dies.



Death
(only one live neighbor)

CA – 2D GRID

- **Birth**. If a cell is dead (state = 0) it will come to life (state becomes 1) if it has exactly three alive neighbours (no more, no less).

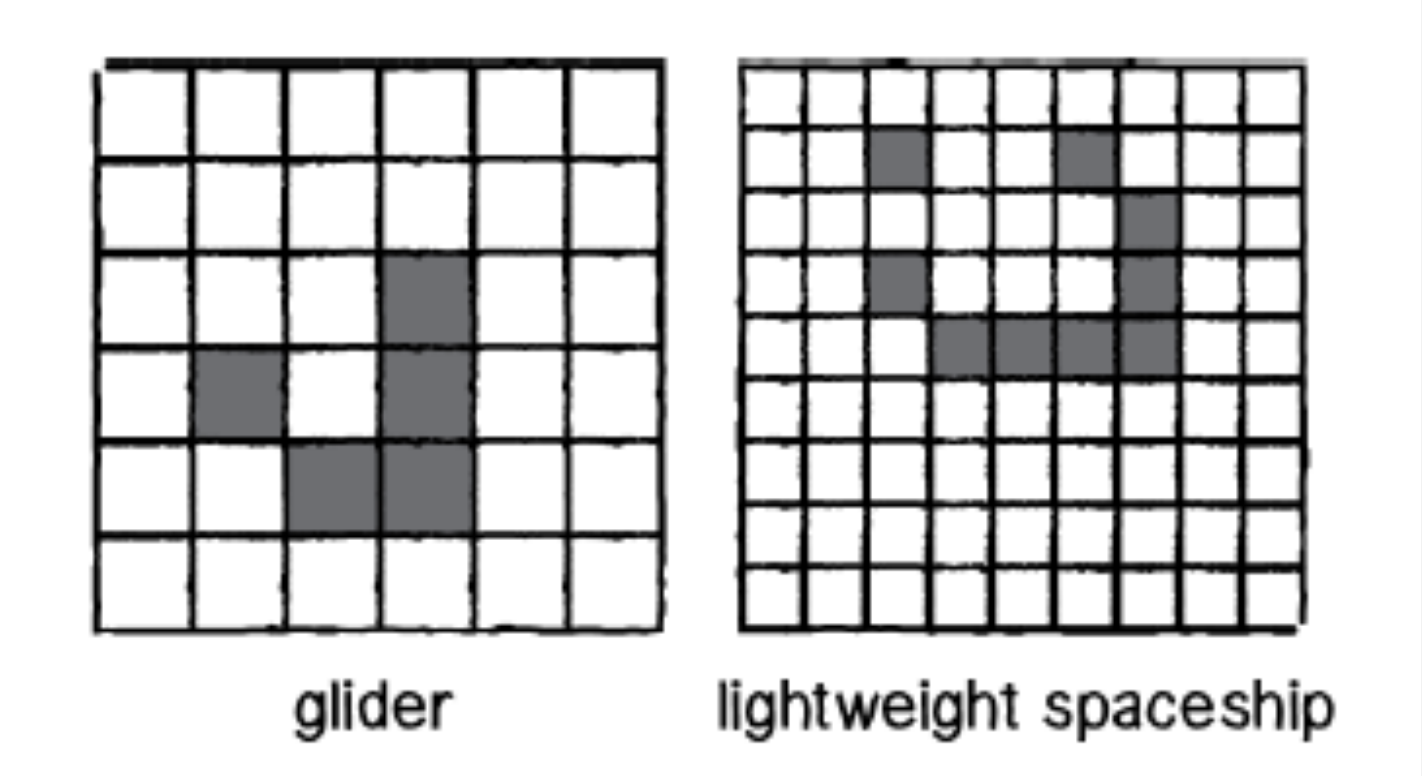


Birth
(three live neighbors)

- **Stasis**. In all other cases, the cell state does not change. To be thorough, let's describe those scenarios.
  - **Staying Alive**: If a cell is alive and has exactly two or three live neighbours, it stays alive.
  - **Staying Dead**: If a cell is dead and has anything other than three live neighbours, it stays dead.

CA – 2D GRID

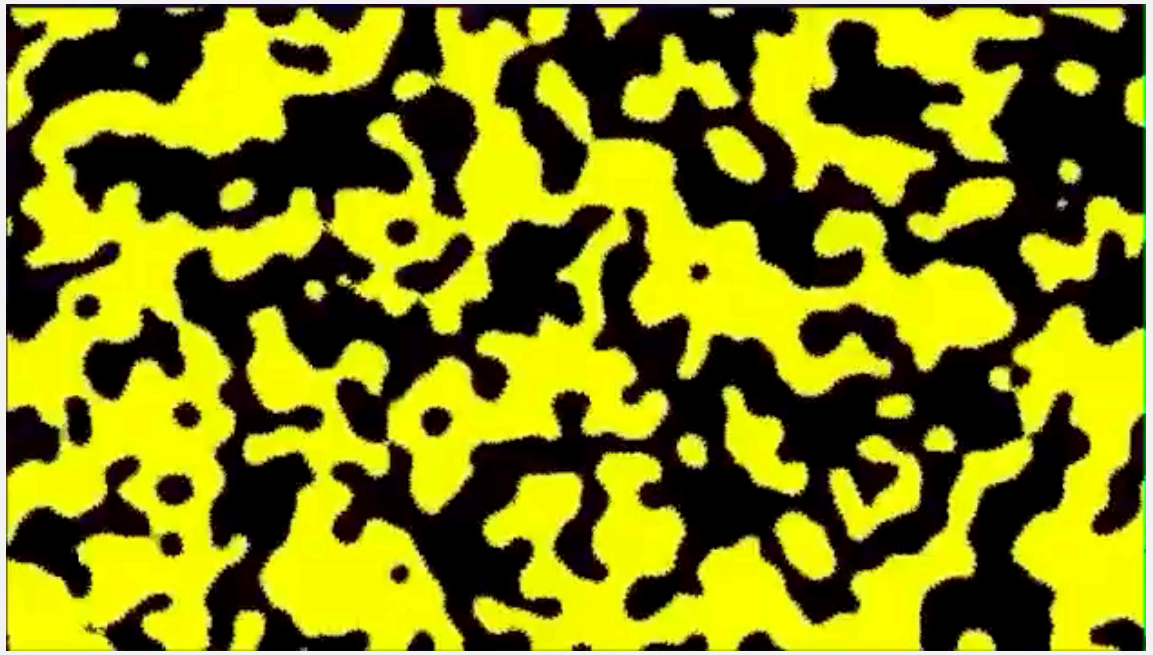
- The initial pattern influences much the evolution
- Some patterns from generation to generation move about the grid.



GameOfLifeSimple

CA – CYCLE 2D GRID

- N possible states per cell, numbered from 0 to n-1
- The first generation starts out with random states in each of the cells
- In each subsequent generation, if a cell has a neighbouring cell whose value is the successor of the cell's value, the cell is "consumed" and takes on the succeeding value
- Modular arithmetic is used so that 0 is the successor of n-1



https://www.youtube.com/watch?v=dQJ5aEsP6Fs

MATERIALS

- **References**
  - Gary Greenfield and Penousal, Smart art, Leonardo Vol. 47 no. 1, 2014
- **Further readings**
  - Daniel Shiffman – Nature of code, 2012 – Chapter 4, 5, 6, 7
  - Russell - Artificial Intelligence a modern approach, 2012 – Chapter 2,11
  - McCormak – Computers and Creativity, 2012 – Chapter 10
  - M. Delgado, W. Fajardo, M.M. Solana - Inmamusys: Intelligent multiagent music system, Expert Systems with Applications, Volume 36, Issue 3, Part 1, April 2009
  - BLACKWELL T. (2007) Swarming and Music. In: Miranda E.R., Biles J.A. (eds) Evolutionary Computer Music. Springer, London.