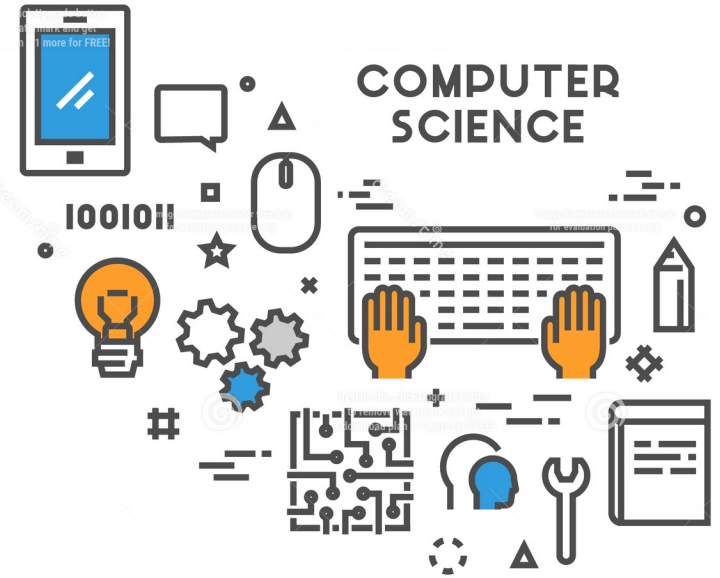
https://bit.ly/SCmusic2021

Music for todays challenges!



Animal Kingdom

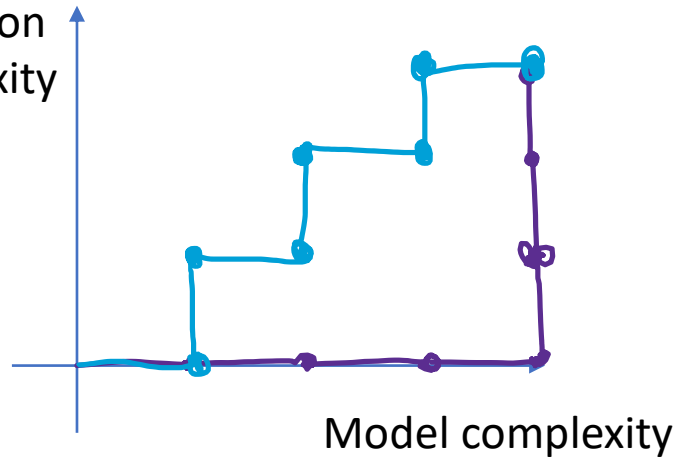
Cellular Automata

Seminar 2

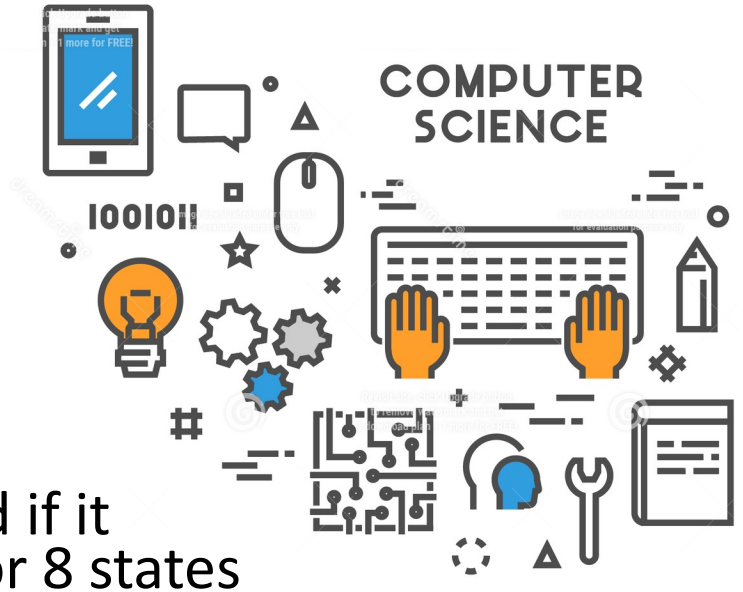
Breaking down a problem

My method for breaking down a problem

- What information/concepts do we need?
  - Go through the definitions and mark/write down the information that is needed.
  - Other questions: initial state, visualization, correctness?
  - Are there any large groups of logically dependent information/concepts that we can separate those into?
- A first draft of data structures
- Functionality:
  - What information do we need to calculate and what is given?
  - What functionality do we need?
- A first draft of a flow-chart.



Which animal to start with



Rules for Fish

A fish can breed, and die or move

- it tries to breed if it has been alive for 12 states
- to breed, it creates a new fish in a neighbouring empty cell (if it exists)
- it dies (overcrowding) if there are now 2 or more neighbouring fish
- if it does not die it tries to move to a neighbouring empty cell

Rules for Bears

- A bear can breed if it has been alive for 8 states
- Bear need to eat.
- If there is one or more fish in neighbouring cells it picks a random one and eats it
- If the bear cannot eat for 10 states, it dies
- If it does not die, it tries to move to a neighbouring empty cell.

Rules for Fish

A fish can breed, and die or move

- it tries to breed if it has been alive for 12 states
- to breed, it creates a new fish in a neighboring empty cell (if it exists)
- it dies (overcrowding) if there are now 2 or more neighboring fish
- if it does not die it tries to move to a neighboring empty cell

Fish information & functionality:

- "Be" in a position in grid
- Decide if a fish has been alive in 12 states
- Decide if there is an empty neighbour cell
- Decide where the fish is positioned in the grid
- Create a new fish
- Insert a "new fish" into a specified cell
- Decide whether there are at least two neighbour-fish-cells
- Move a fish to a *random* empty neighbour cell

5-minutes challenge

- How to represent the grid and fishes?

Think – Pair – Share:

1. Think for yourself
2. Pair up with your neighbour and come up with an aligned answer
3. Share your pair-answer with the class

Representing grid & fishes?

- A list of fishes with their positions
- Using the fish's position, get Neighbour positions: Requires knowledge about the grid dimensions
- Go through the list of fishes and list of neighbour-positions to determine empty and fish-neighbours

Requirements:

- Visualize the grid

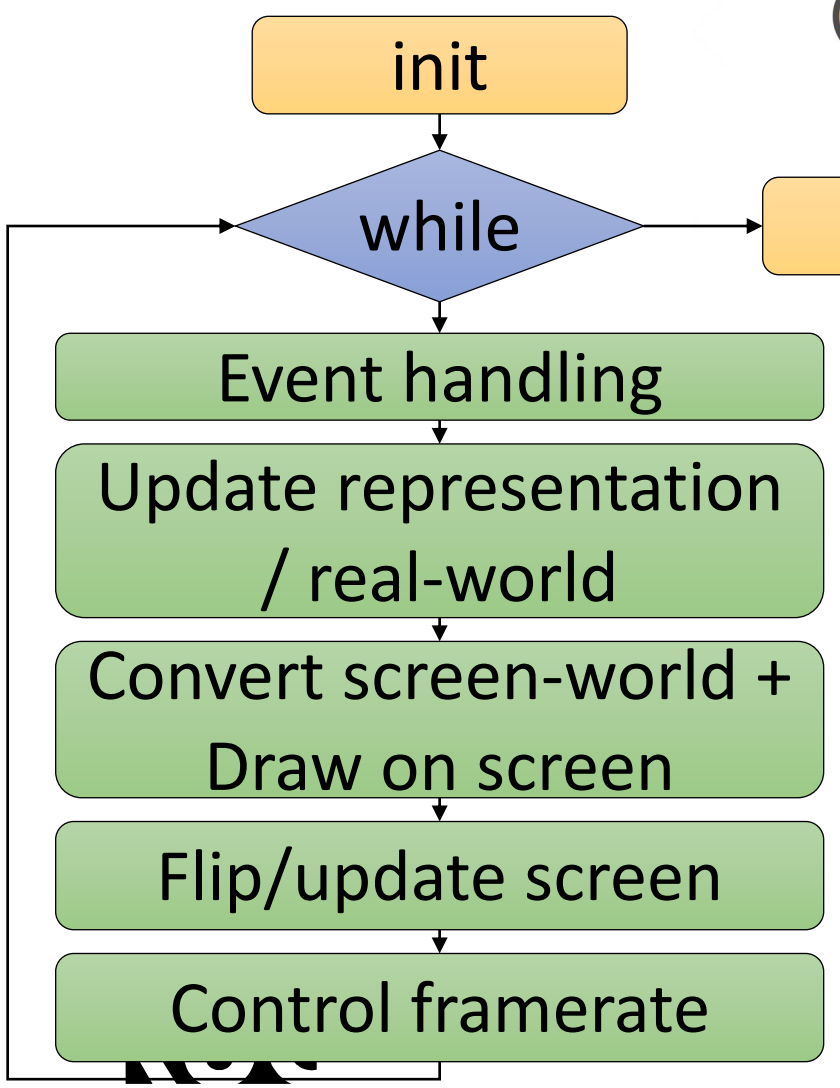
- A 2D array with fish/empty-representations in the cells.
- The position of a fish is its grid-position.
- The array's dimensions determines the world
- Determine whether neighbours are fish or empty by looking into the cells

Requirements:

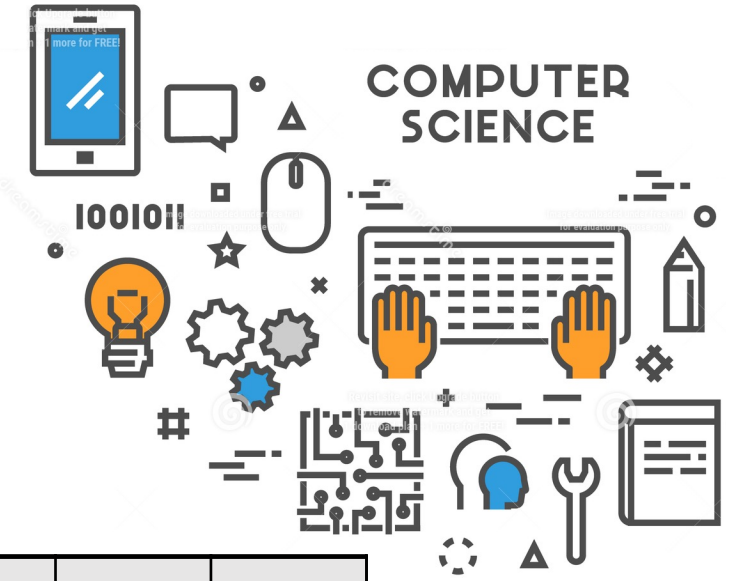
- Get a fish's position
- Get its empty-neighbour cells,
- Get its fish-neighbour cells
- Requires information of grid-size and positions of other fishes

AK skeleton w neighbours

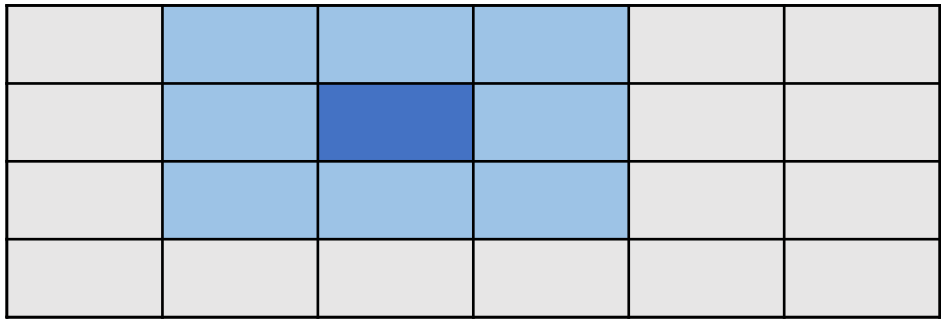
1. Download Animal Kingdom skeleton w neighbours
2. Make the Pygame skeleton run
3. Make comments, during the tour.

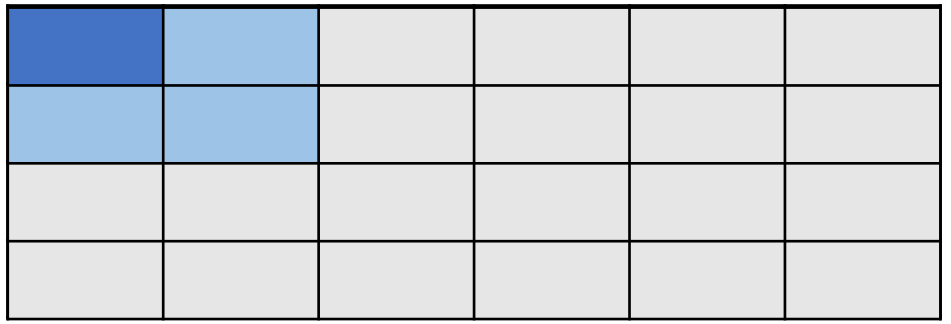


Neighbours



- The function get_neighbors
- It takes as input an array and a cell position





- It is a function that returns a list of all existing neighbour positions

Rules for Fish

A fish can breed, and die or move

- it tries to breed if it has been alive for 12 states
- to breed, it creates a new fish in a neighboring empty cell (if it exists)
- it dies (overcrowding) if there are now 2 or more neighboring fish
- if it does not die it tries to move to a neighboring empty cell

Fish information & functionality:

- "Be" in a position in grid
- Decide if a fish has been alive in 12 states
- Decide if there is an empty neighbour cell
- Decide where the fish is positioned in the grid
- Create a new fish
- Insert a "new fish" into a specified cell
- Decide whether there are at least two neighbour-fish-cells
- Move a fish to a *random* empty neighbour cell

Challenge

- Insert fish die of overcrowding

- Hint: The number of fish neighbours is equal to the length of the list with fish-neighbour positions.

How do we ensure correctness?

- Construct the program in steps that we can check
- Complex “world” -> it can help to visualize the grid
- Keep track of time and slow down time without slowing down the framerate
- Give the animals IDs such that we can track them.

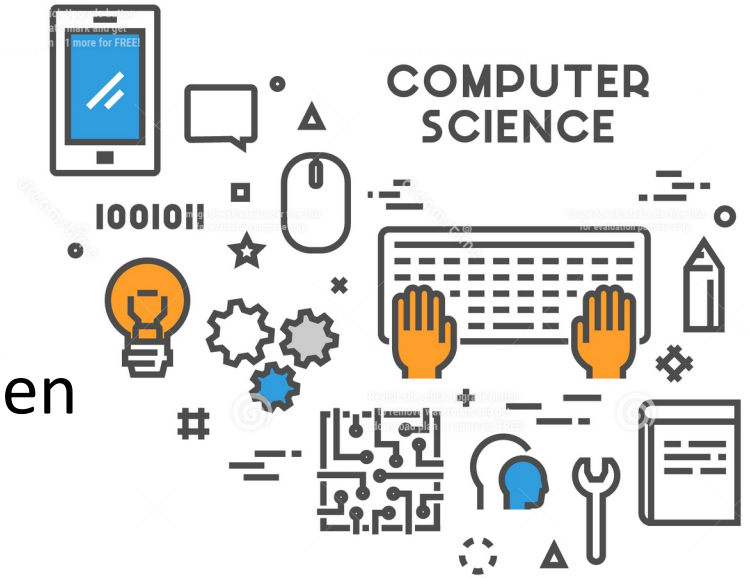
5 minutes challenge

- How does the speed_count and SPEED works?

Think – Pair – Share:

1. Think for yourself
2. Pair up with your neighbour and come up with an aligned answer
3. Share your pair-answer with the class

5 minutes challenge



- Insert unique IDs into the animal data structure when they are created –use the function new_ID()

Think – Pair – Share:

1. Think for yourself
2. Pair up with your neighbour and come up with an aligned answer
3. Share your pair-answer with the class

Rules for Fish

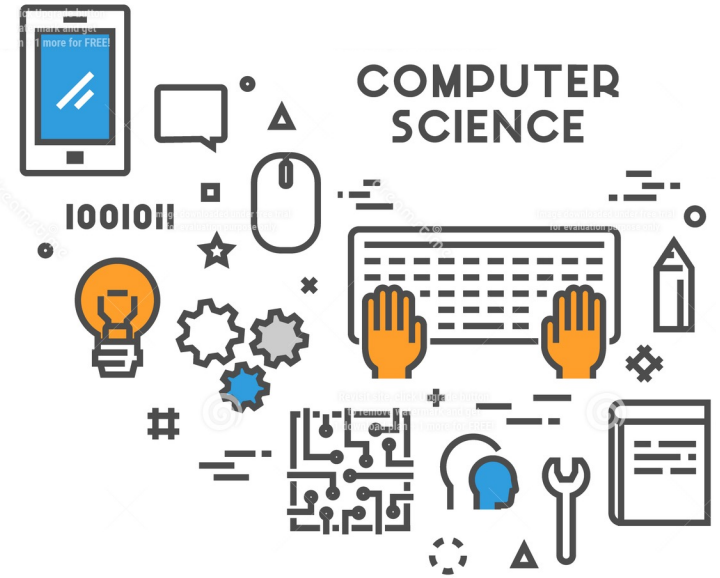
A fish can breed, and die or move

- it tries to breed if it has been alive for 12 states
- to breed, it creates a new fish in a neighboring empty cell (if it exists)
- it dies (overcrowding) if there are now 2 or more neighboring fish
- if it does not die it tries to move to a neighboring empty cell

Fish information & functionality:

- "Be" in a position in grid
- Decide if a fish has been alive in 12 states
- Decide if there is an empty neighbour cell
- Decide where the fish is positioned in the grid
- Create a new fish
- Insert a "new fish" into a specified cell
- Decide whether there are at least two neighbour-fish-cells
- Move a fish to a *random* empty neighbour cell

15-minutes Challenge



Implement:

- if it does not die it tries to move to a neighboring empty cell

Rules for Fish

A fish can breed, and die or move

- it tries to breed if it has been alive for 12 states
- to breed, it creates a new fish in a neighboring empty cell (if it exists)

- it dies (overcrowding) if there are now 2 or more neighboring fish

- if it does not die it tries to move to a neighboring empty cell

Fish information & functionality:

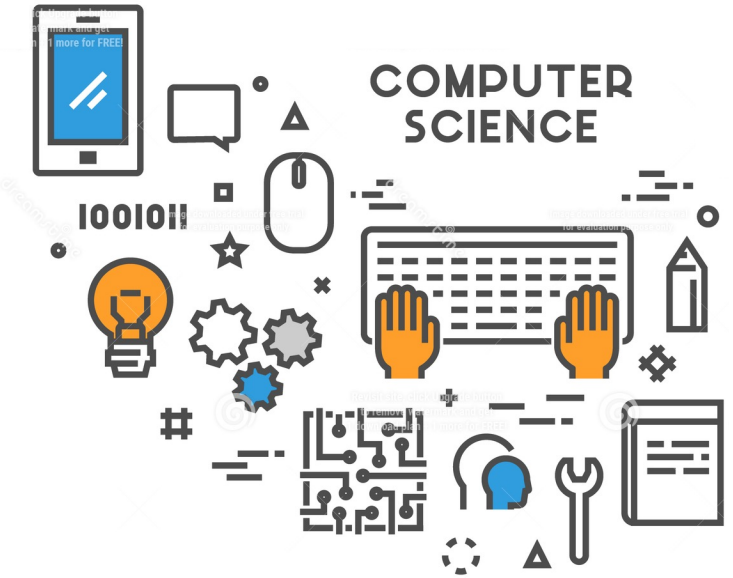
- "Be" in a position in grid
- Decide if a fish has been alive in 12 states
- Decide if there is an empty neighbour cell
- Decide where the fish is positioned in the grid
- Create a new fish
- Insert a "new fish" into a specified cell
- Decide whether there are at least two neighbour-fish-cells
- Move a fish to a *random* empty neighbour cell

Pair-Challenge

Implement:

- it tries to breed if it has been alive for 12 states
- to breed, it creates a new fish in a neighboring empty cell (if it exists)
- Colours of Fish changing according to its life-states (just created, young, breeding age) to have a visual confirmation

Rules for Bears



A Bear can breed and eat and then die or move

- A bear can breed if it has been alive for 8 states
- Bear needs to eat.
- If there is one or more fish in neighbouring cells it picks a random one and eats it
- If the bear cannot eat for 10 states, it dies
- If it does not die, it tries to move to a neighbouring empty cell.