Conway's Game of Life

Game of Life

- October 1970 issue of Scientific American, in Martin Gardner's "Mathematical Games" column.
- Begin of a research area: cellular automaton
- Simple rules yield non-trial, complex patterns, with the formation of local structures that are able to survive for long periods of time

Rules

- The universe of the Game of Life is an infinite two-dimensional orthogonal grid of square cells.
- Each cell is in one of two possible states, alive or dead,
- .Every cell interacts with its eight neighbours.
- At each step in time, the following transitions occur:
  1. Any live cell withfewer than two live neighbours dies, as if caused by under-population.

Rules

- The universe of the Game of Life is an infinite two-dimensional orthogonal grid of square cells.
- Each cell is in one of two possible states, alive or dead,
- .Every cell interacts with its eight neighbours.
- At each step in time, the following transitions occur:
  1. Any live cell withfewer than two live neighbours dies, as if caused by under-population.
  2. Any live cell with two or three live neighbours lives on to the next generation.

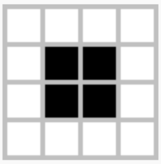
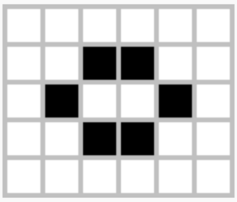
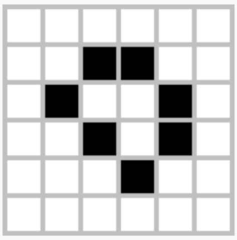
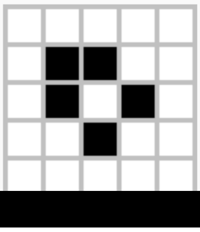
Rules

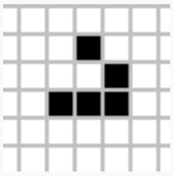
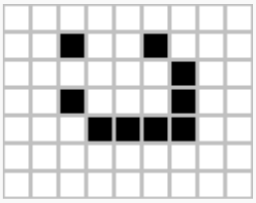
- The universe of the Game of Life is an infinite two-dimensional orthogonal grid of square cells.
- Each cell is in one of two possible states, alive or dead,
- .Every cell interacts with its eight neighbours.
- At each step in time, the following transitions occur:
  1. Any live cell withfewer than two live neighbours dies, as if caused by under-population.
  2. Any live cell with two or three live neighbours lives on to the next generation.
  3. Any live cell with more than three live neighbours dies, as if by over-population.

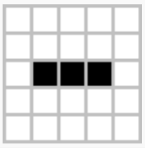
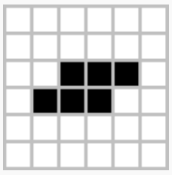
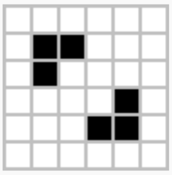
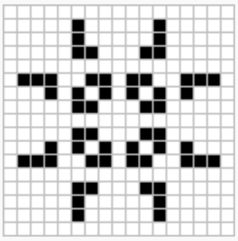
Rules

- The universe of the Game of Life is an infinite two-dimensional orthogonal grid of square cells.
- Each cell is in one of two possible states, alive or dead,
- .Every cell interacts with its eight neighbours.
- At each step in time, the following transitions occur:
  1. Any live cell withfewer than two live neighbours dies, as if caused by under-population.
  2. Any live cell with two or three live neighbours lives on to the next generation.
  3. Any live cell with more than three live neighbours dies, as if by over-population.
  4. Any dead cell with exactly three live neighbours becomes a live cell, as if by reproduction.

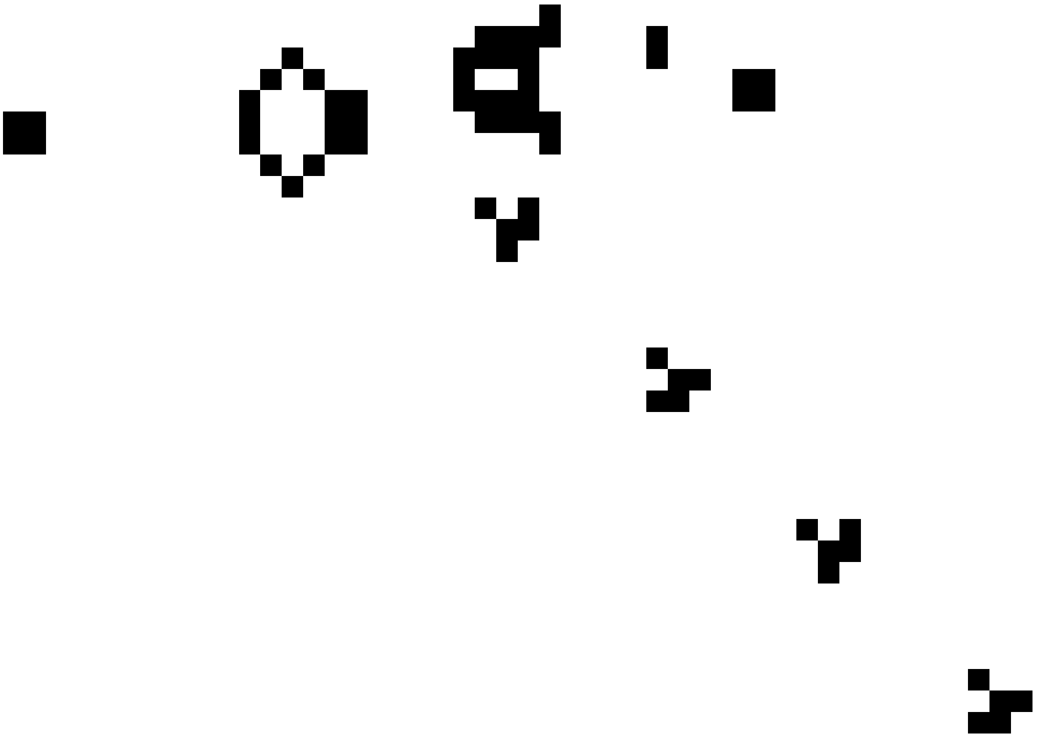
Examples

Still lifes	
Block	
Beehive	
Loaf	
Boat	

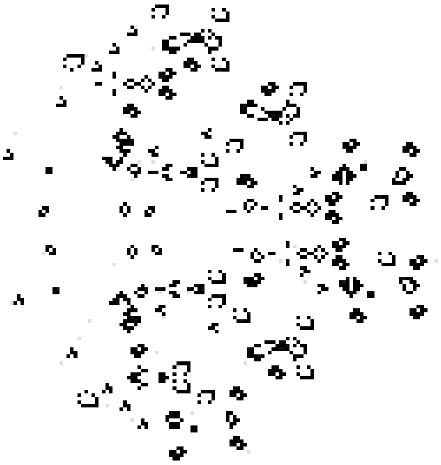
Spaceships	
Glider	
Lightweight spaceship (LWSS)	

Oscillators	
Blinker (period 2)	
Toad (period 2)	
Beacon (period 2)	
Pulsar (period 3)	

Conway's Game of Life



Conway's Game of Life



Challenge

Challenge

1. Code a well-structured code in C that implements the Game of Life.

Challenge

Challenge

1. Code a well-structured code in C that implements the Game of Life.
2. Use the ncurses library for visualization

Challenge

Challenge

1. Code a well-structured code in C that implements the Game of Life.
2. Use the ncurses library for visualization
3. Simulate the following structures:

Challenge

Challenge

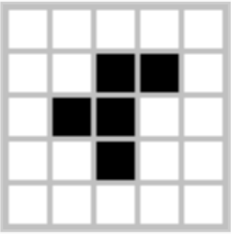
1. Code a well-structured code in C that implements the Game of Life.
2. Use the ncurses library for visualization
3. Simulate the following structures:

Challenge

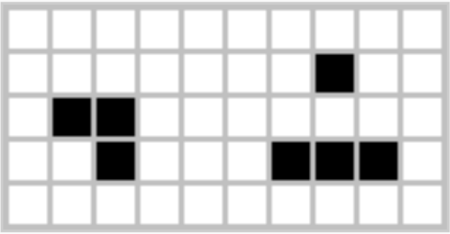
Challenge

1. Code a well-structured code in C that implements the Game of Life.
2. Use the ncurses library for visualization
3. Simulate the following structures:

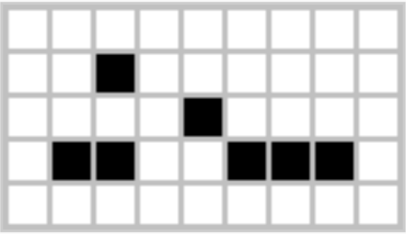
Simulation



The R-pentomino



Diehard



Acorn

Challenge Evaluation

Evaluation

Code *must* be original.

30% Code Clarity

20% Team Work (groups of 3)

50% Functionality

Important

- Code must be correctly commented.
- Explain in comments in the main.c the team-work strategy
- The creator of each code must be included in the comments, identified as *coder 1*, *coder 2*, *coder 3*
- Upload two files:

  .txt A *authors.txt* file with the corresponance between real authors and coders

  .tar.gz A *gameoflife.tar.gz* file containing the code of your project, uncompiled.