

**BC-0005**

**Bases Computacionais da Ciência**

## **Aula 09 – Simulação computacional (parte 2)**

Prof. Jesús P. Mena-Chalco  
[jesus.mena@ufabc.edu.br](mailto:jesus.mena@ufabc.edu.br)

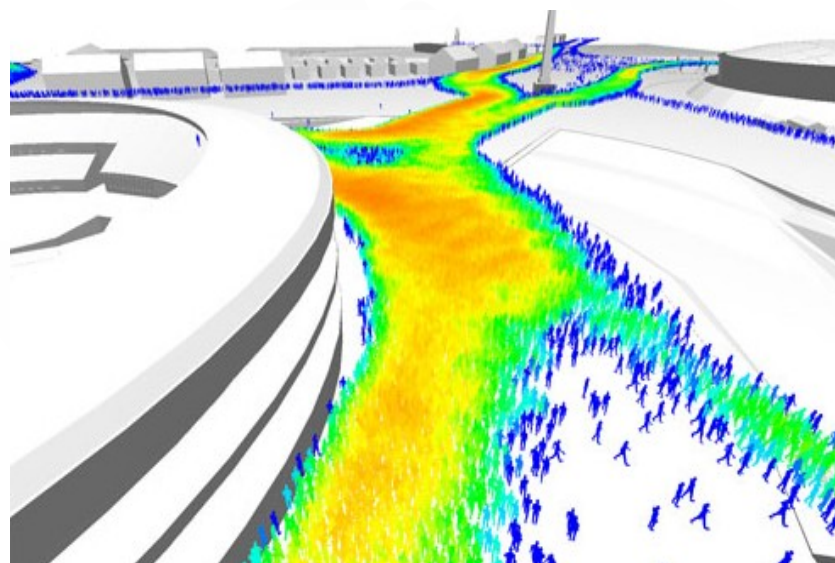
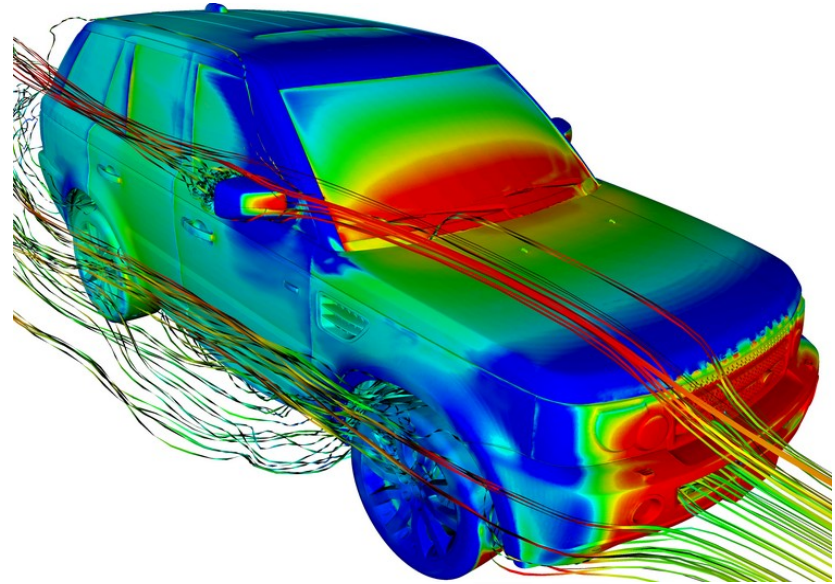
2Q-2014

# Motivação

A área de **Modelagem e Simulação Computacional** está cada vez mais sendo utilizada nos diferentes campos do saber.

Isto deve-se à possibilidade de: **Estudar sistemas reais de maneira aproximada, criando-se modelos matemáticos que os representem**

Tais modelos são implementados em simulações computacionais, que são executadas visando obter um melhor entendimento do sistema real.



# Definição de sistema

O termo sistema vem do grego **snistánai** e significa “**fazer ficar junto**”.

Um sistema é:

- Um conjunto de elementos interconectados que interagem entre si.
- Um sistema e seus elementos estão inseridos em um ambiente



# Sistema

Há três formas de se estudar um sistema:

- (1) Experimentos com o **Sistema Real**.
- (2) Experimentos com **Modelos Físicos**.
- (3) Experimentos com **Modelos Matemáticos**.

# (1) Experimentos com o sistema real

Experimentos com o sistema real ocorrem quando é possível trabalhar diretamente com o sistema real:

- Atuando em seus elementos e/ou
- Alterando sua configuração para fazê-lo operar sob estas novas condições propostas

Como exemplo tem-se um **experimento real de *crash-test*** realizado em um veículo da General Motors





# (1) Experimentos com o sistema real

Entretanto, tratar diretamente com o sistema real pode não ser possível:

- **O experimento pode ser muito caro ou perigoso.** Por exemplo, analisar pessoas em uma situação de incêndio
- **Pode ser impossível tratar diretamente com sistemas reais.** Exemplo: a análise dos buracos negros, descritos pela Astrofísica
- **Situações onde não há evidências da existência do sistema.**



# (1) Experimentos com o sistema real

Em muitas situações é necessário **construir um modelo** que represente **parcialmente** o **sistema**.

E **realizar experimento** com este **modelo**  
Desta forma, é possível **estudar o sistema** real  
de maneira **indireta**, deixando-o inalterado.

Um **modelo** é uma **representação parcial** de um objeto, sistema ou ideia.

## (2) Experimentos com modelos físicos

Os **modelos físicos** consideram:

- experimentos com **objetos reais**
- tais objetos atuam como **representações parciais** do **sistema** que se deseja estudar

Como exemplo de modelos físicos cita-se **mapas** e **maquetes de aviões**





# (3) Experimentos com modelos matemáticos

Modelos matemáticos usam símbolos em lugar de dispositivos físicos

- procurando representar as principais características e comportamentos do sistema alvo que se deseja analisar

Há duas formas de solução de modelos matemáticos:

- Solução analítica
- Solução numérica, **via simulação**

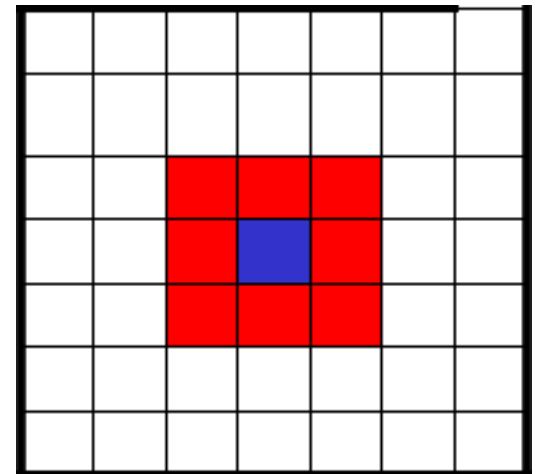
# Jogo da Vida (*Game of life*)

Jogo de simulação  
Jogo que recria processo do mundo-real

<http://to-campos.planetaclix.pt/fractal/celular/Vida.html>

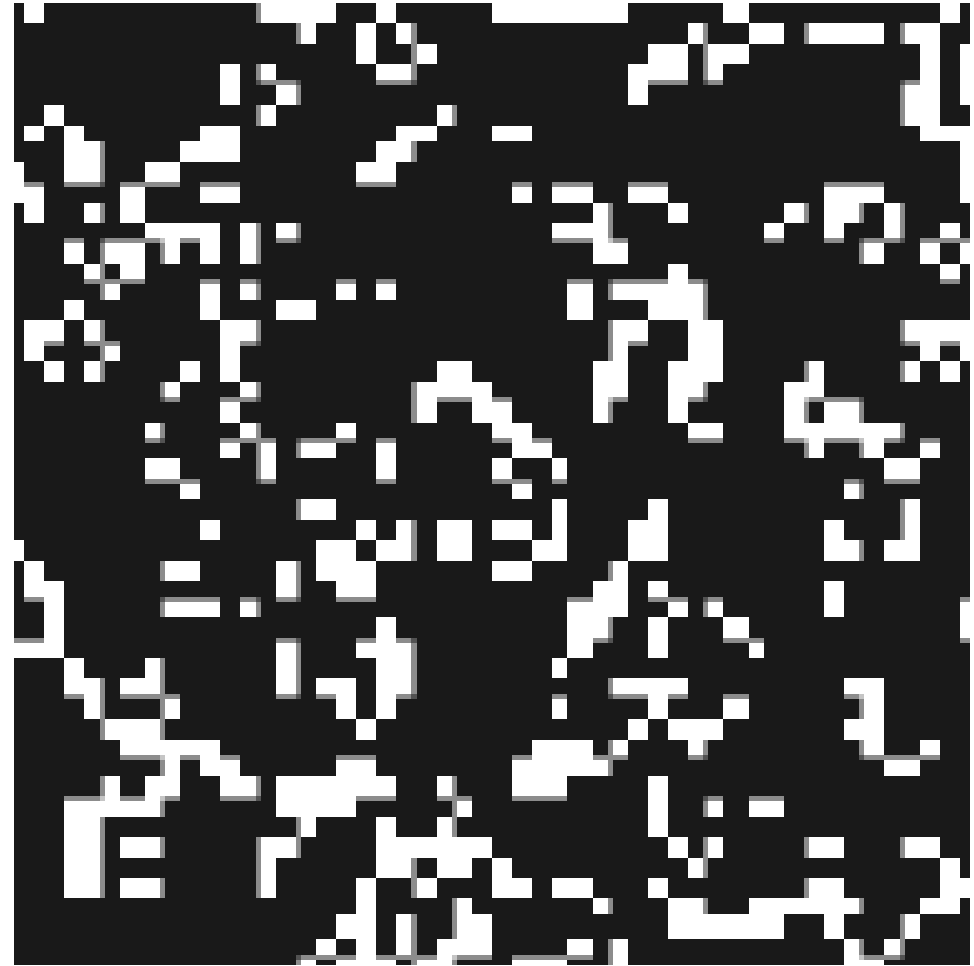
# Jogo da vida

- ❖ O **Jogo da Vida** foi desenvolvido pelo matemático **John Conway**, em 1970
- ❖ Conway explorava a ideia do **construtor universal**, de forma que uma **máquina** hipotética pudesse construir **cópias de si mesma**
- ❖ O Jogo da Vida é um **autômato celular** que se desenvolve em um espaço de **duas dimensões**, dividido em células quadrangulares
- ❖ É governado por **regras simples** que definem **nascimentos, mortes e sobrevivências de células**

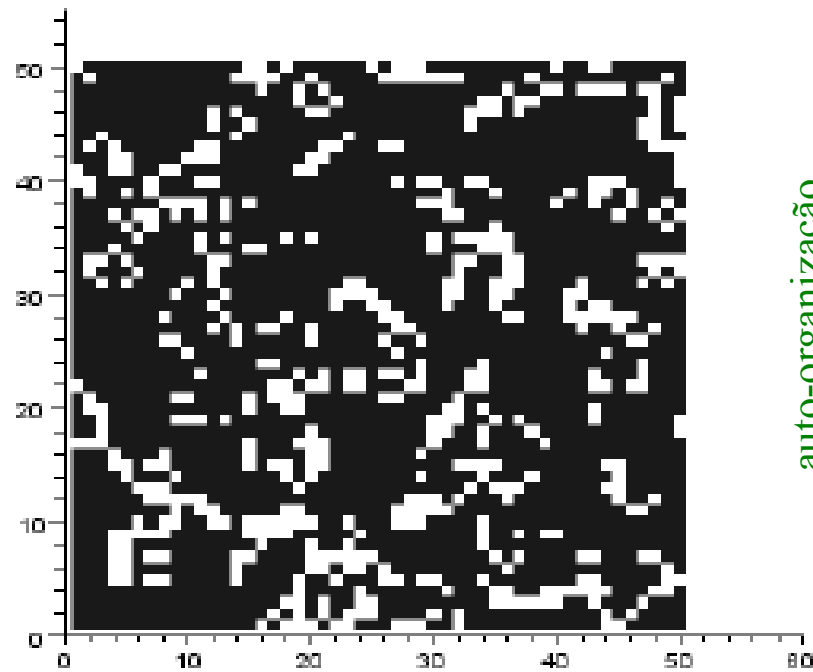


# Jogo da vida

- ◆ Cada uma das **células** do universo bidimensional pode estar em dois **estados** possíveis:
  - viva (cor branca)
  - morta (cor preta)
- ◆ Se uma célula **sobrevive, morre ou nasce** será determinado pelo **número de vizinhos vivos** ao redor de uma célula

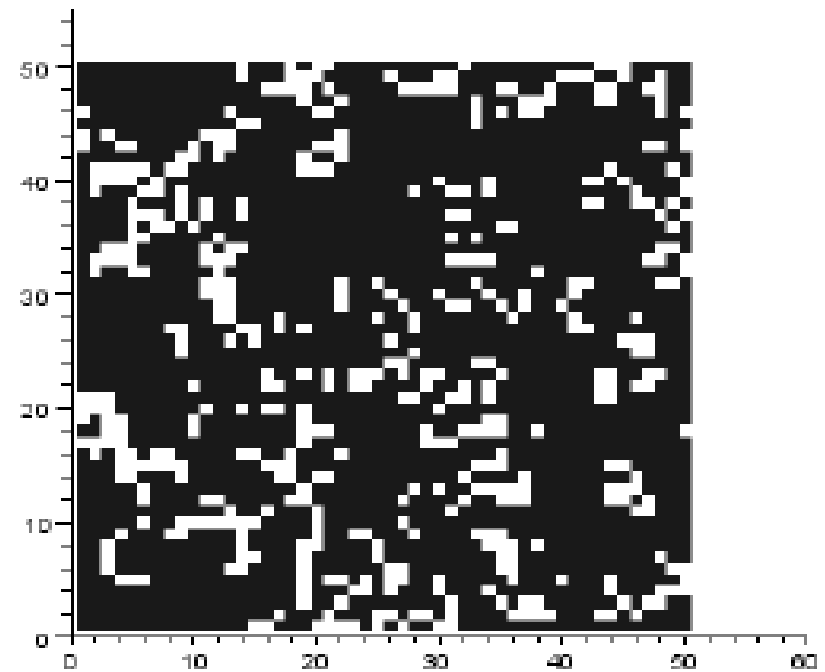


# Jogo da vida



(a) Jogo da Vida no Passo N.

auto-organização

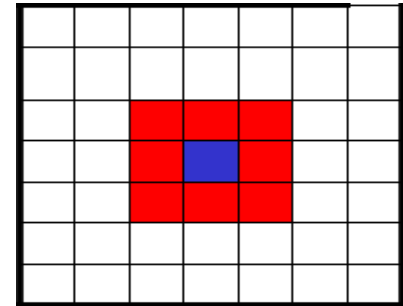


(b) Jogo da Vida no Passo N+1.



# Jogo da vida de Conway

- ◆ Cada célula pode ter **oito células vizinhas**
- ◆ As **regras** envolvem três tópicos:



- **Sobrevivência:** Se a quantidade de **vizinhos vivos** é igual a dois (02) ou três (03)
- **Nascimento:** Se a célula está **morta**, mas tem **três (03) vizinhos vivos**, então ela nasce na próxima fase
- **Morte:** Se a quantidade de **vizinhos vivos** é menor que dois (solidão) ou **maior que três** (superpopulação)

As regras não são arbitrárias: evitam comportamento caótico, crescimento infinito ou rápida estabilidade

Preto: viva

Branca: morta

# Simulação

	1	2	3	4	5
1					
2					
3					
4					
5					

Estado inicial

# Simulação

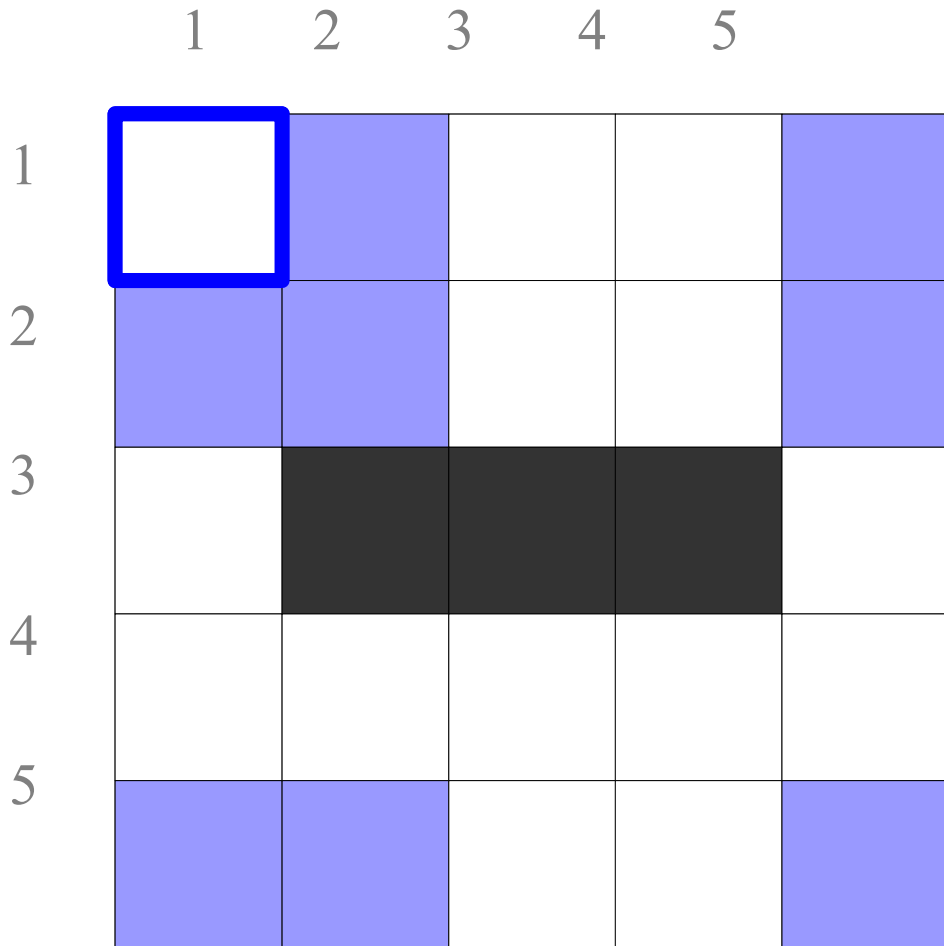
Preto: viva

Branca: morta

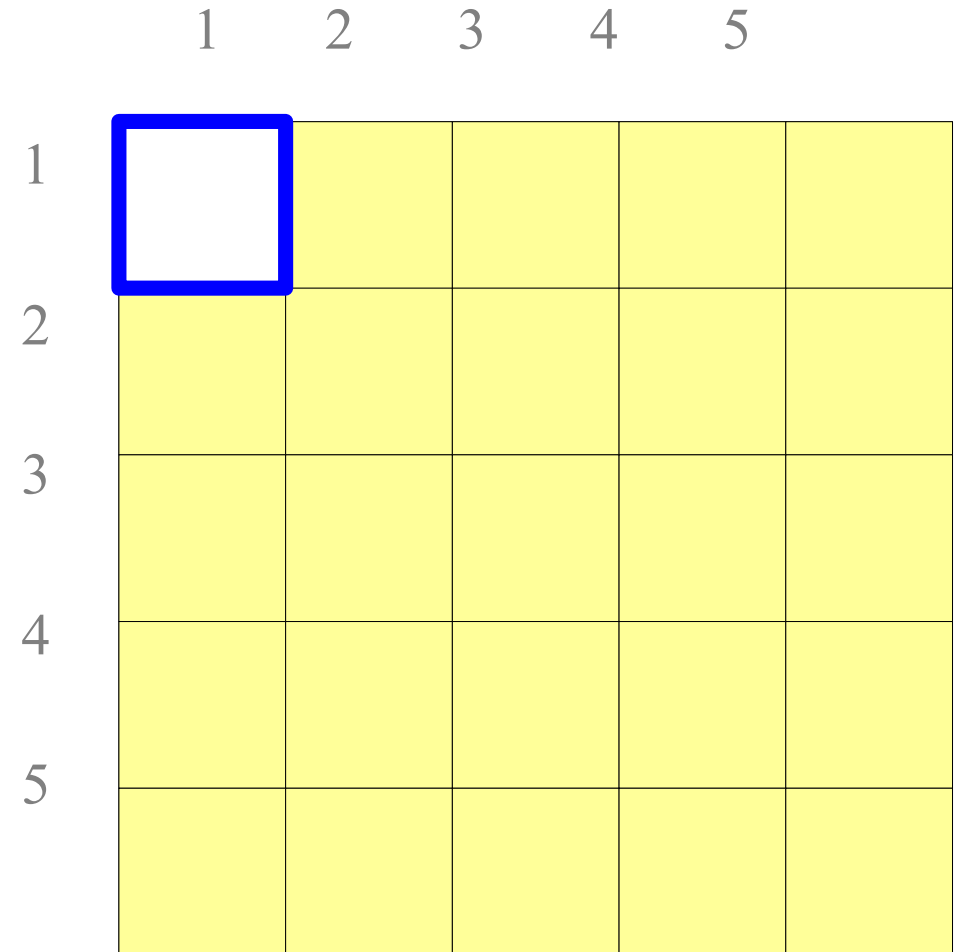
Sobrevivência:  $2 \leq N \leq 3$

Nascimento:  $N=3$

Morte:  $N < 2$  ou  $N > 3$



Estado inicial



Iteração 1

# Simulação

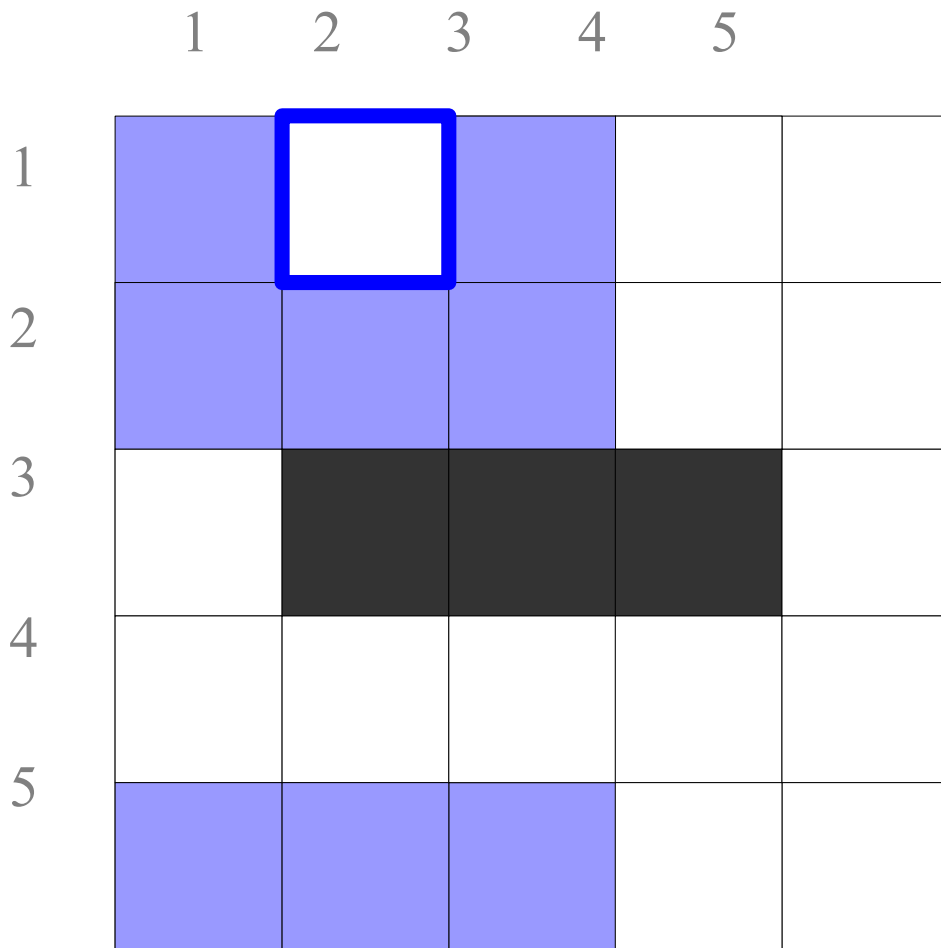
Preto: viva

Branca: morta

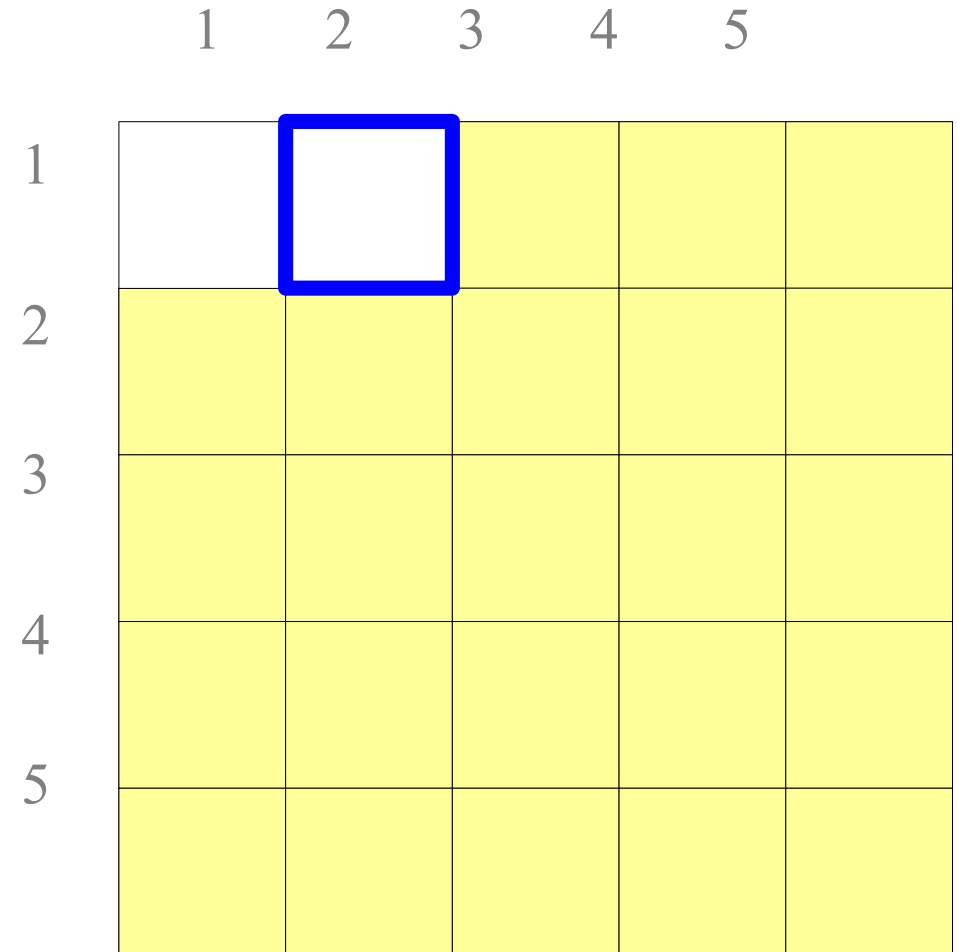
Sobrevivência:  $2 \leq N \leq 3$

Nascimento:  $N = 3$

Morte:  $N < 2$  ou  $N > 3$



Estado inicial



Iteração 1

# Simulação

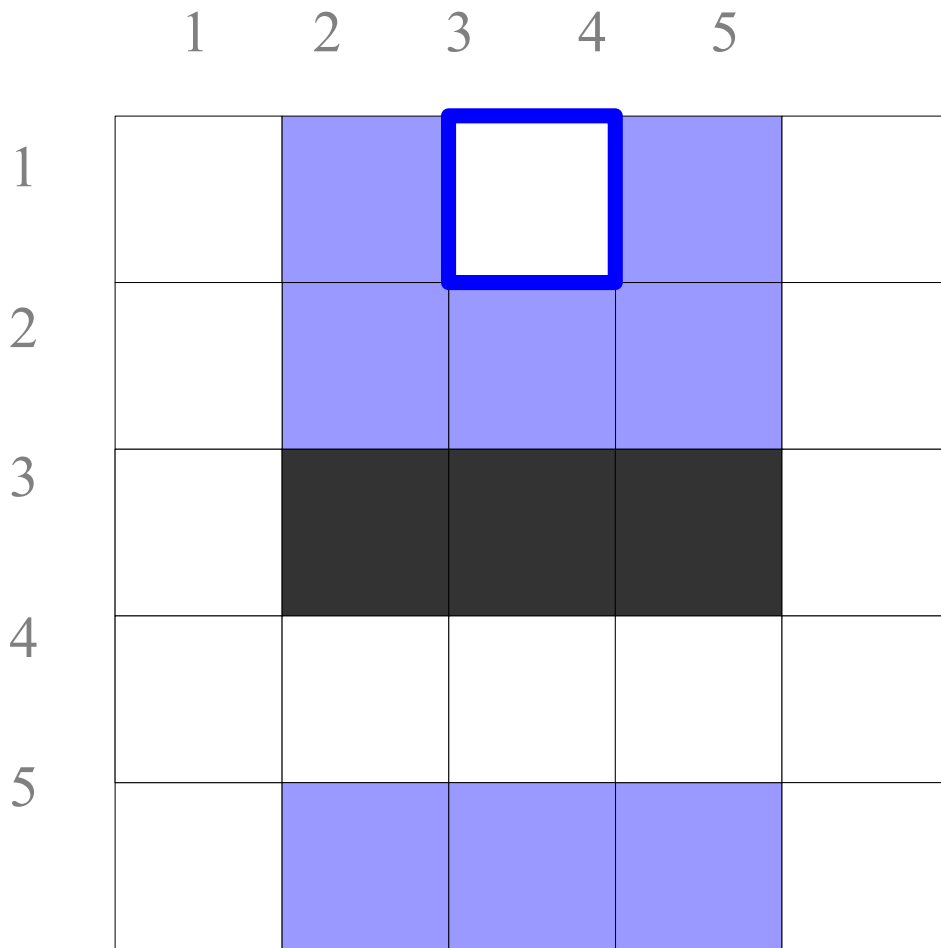
Preto: viva

Branca: morta

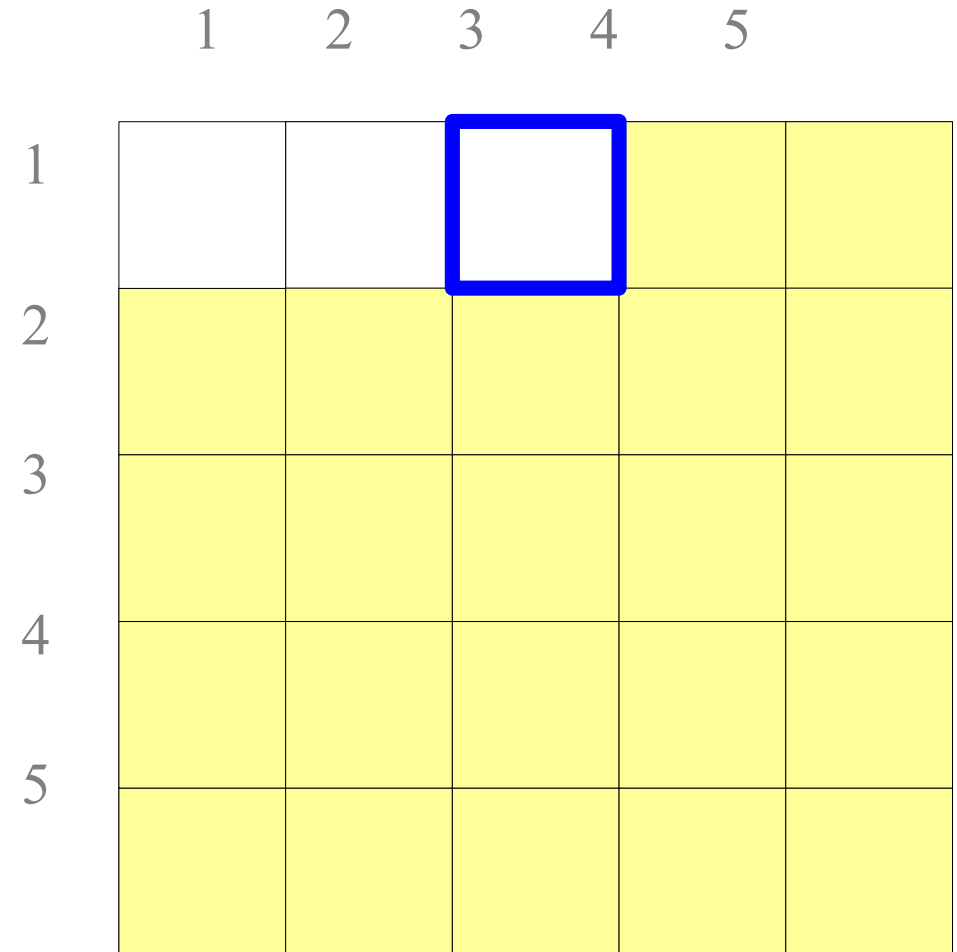
Sobrevivência:  $2 \leq N \leq 3$

Nascimento:  $N=3$

Morte:  $N < 2$  ou  $N > 3$



Estado inicial



Iteração 1



# Simulação

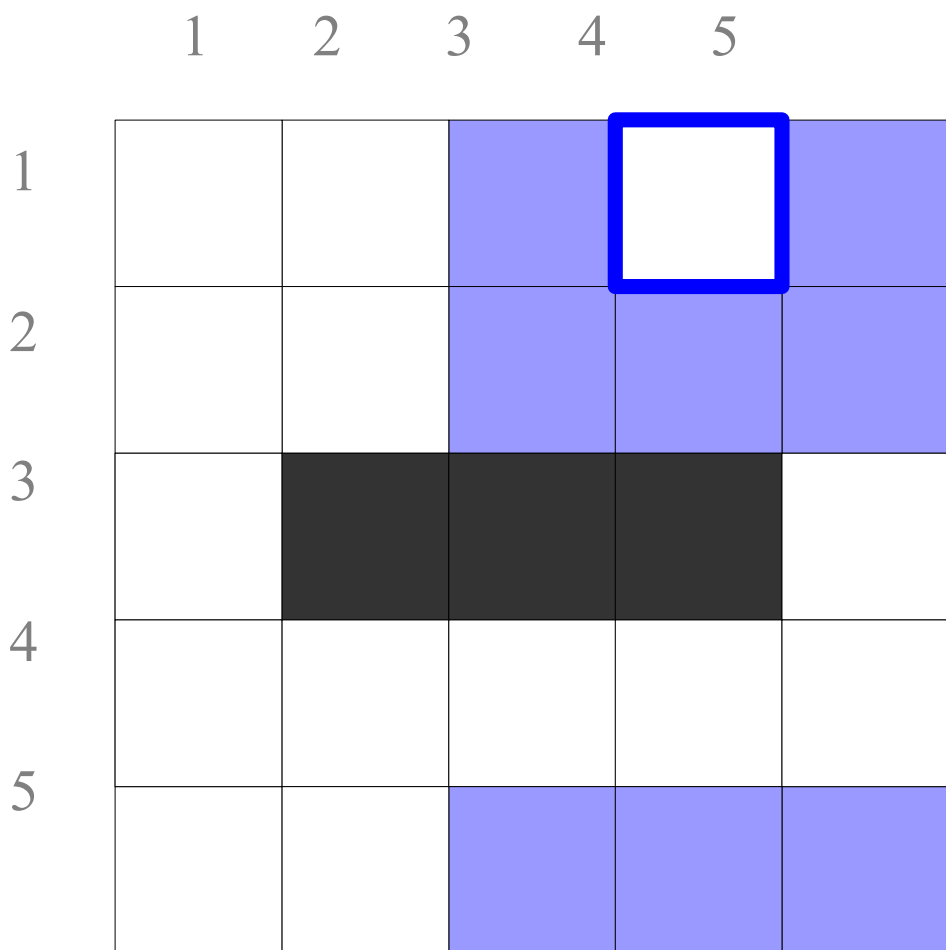
Preto: viva

Branca: morta

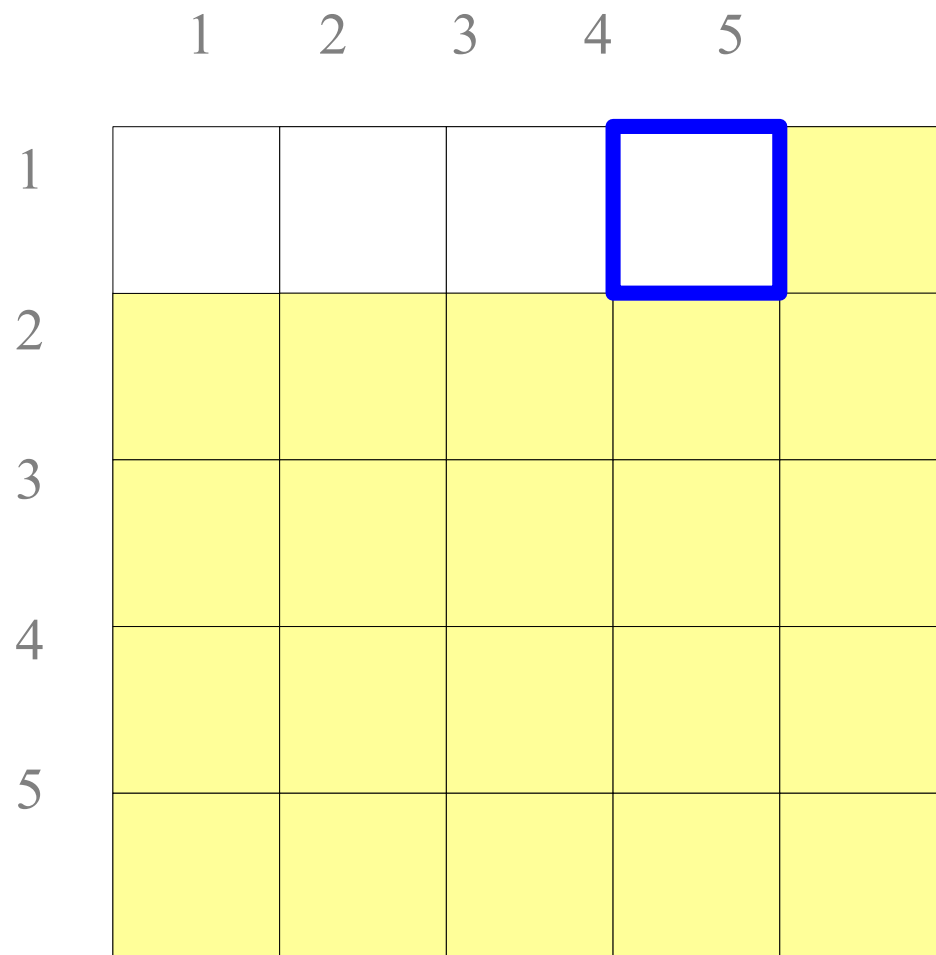
Sobrevivência:  $2 \leq N \leq 3$

Nascimento:  $N=3$

Morte:  $N < 2$  ou  $N > 3$



Estado inicial



Iteração 1

# Simulação

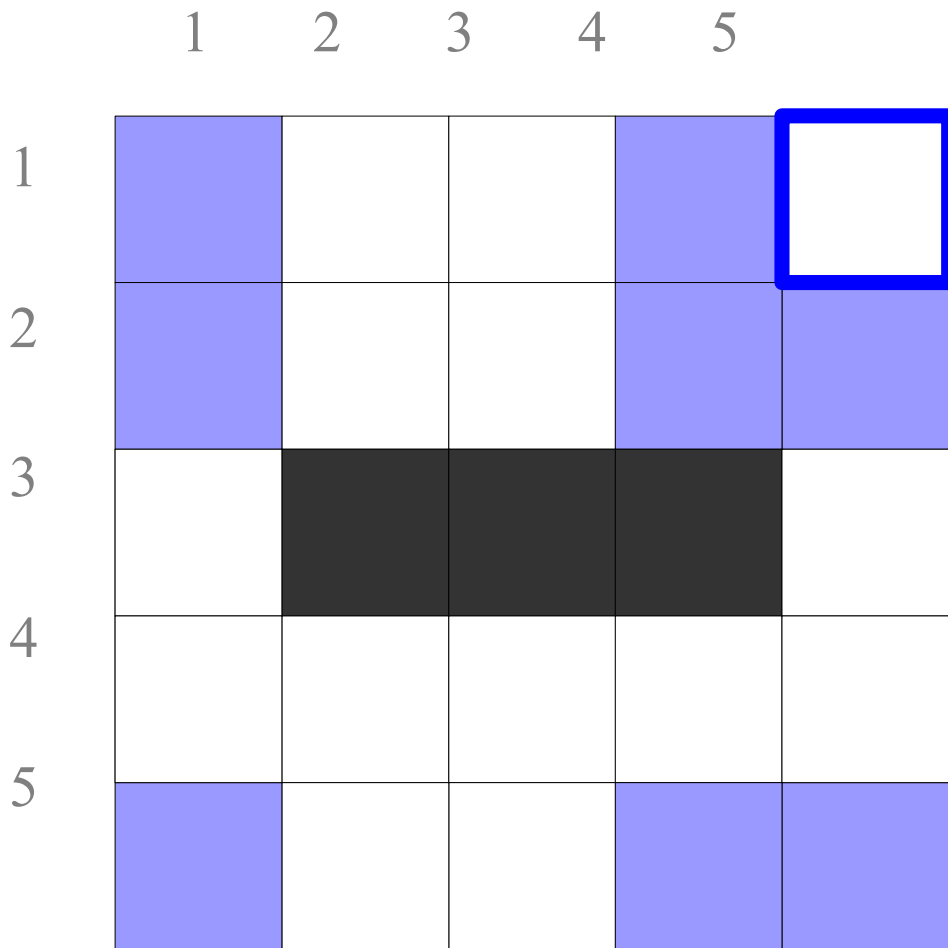
Preto: viva

Branca: morta

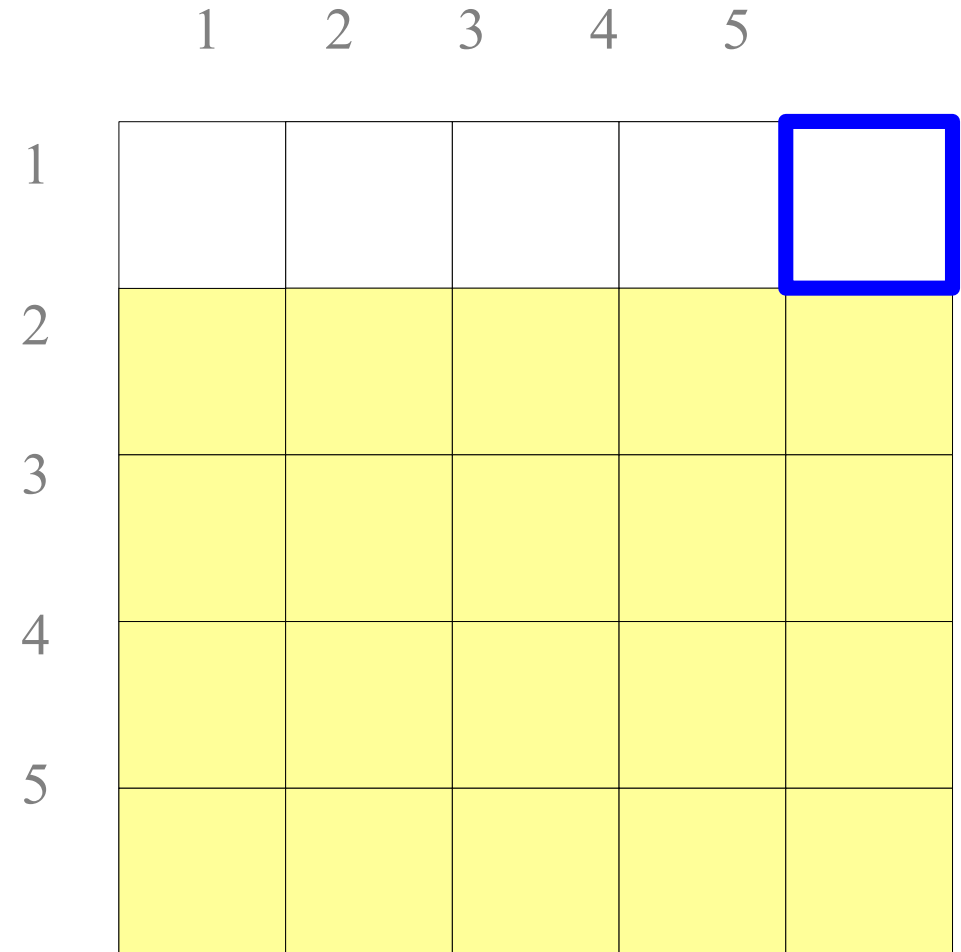
Sobrevivência:  $2 \leq N \leq 3$

Nascimento:  $N=3$

Morte:  $N < 2$  ou  $N > 3$



Estado inicial



Iteração 1

# Simulação

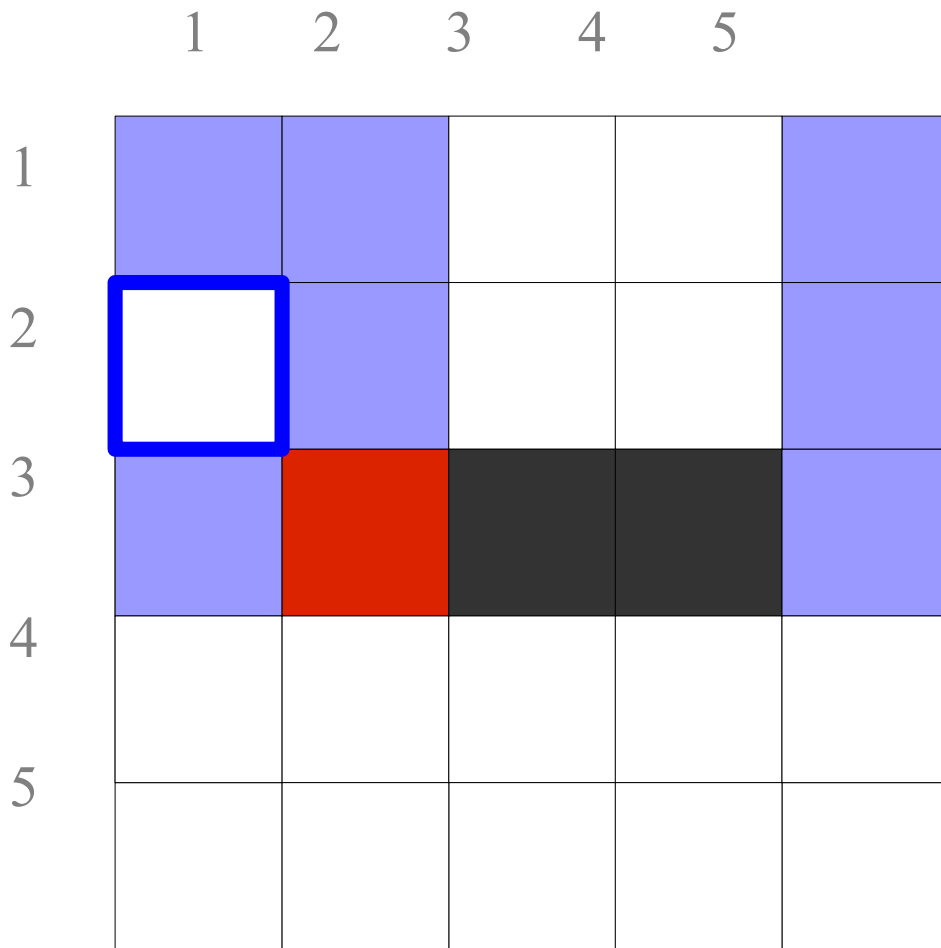
Preto: viva

Branca: morta

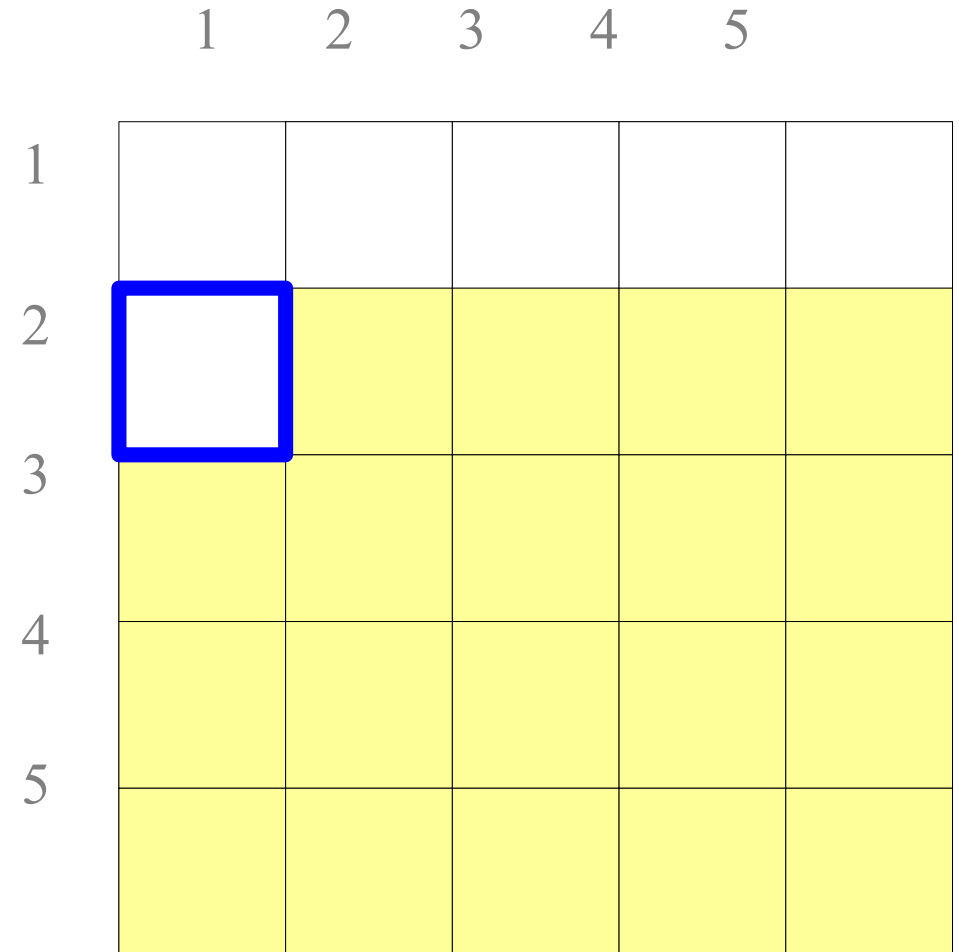
Sobrevivência:  $2 \leq N \leq 3$

Nascimento:  $N=3$

Morte:  $N < 2$  ou  $N > 3$



Estado inicial



Iteração 1

# Simulação

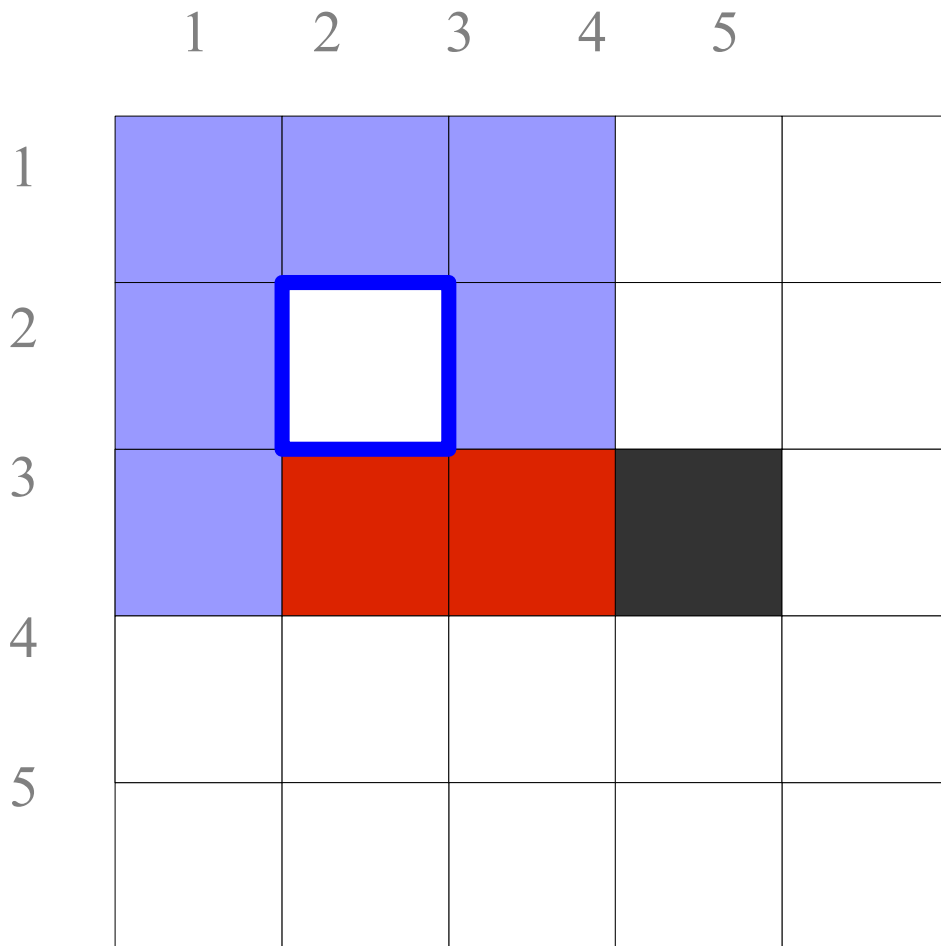
Preto: viva

Branca: morta

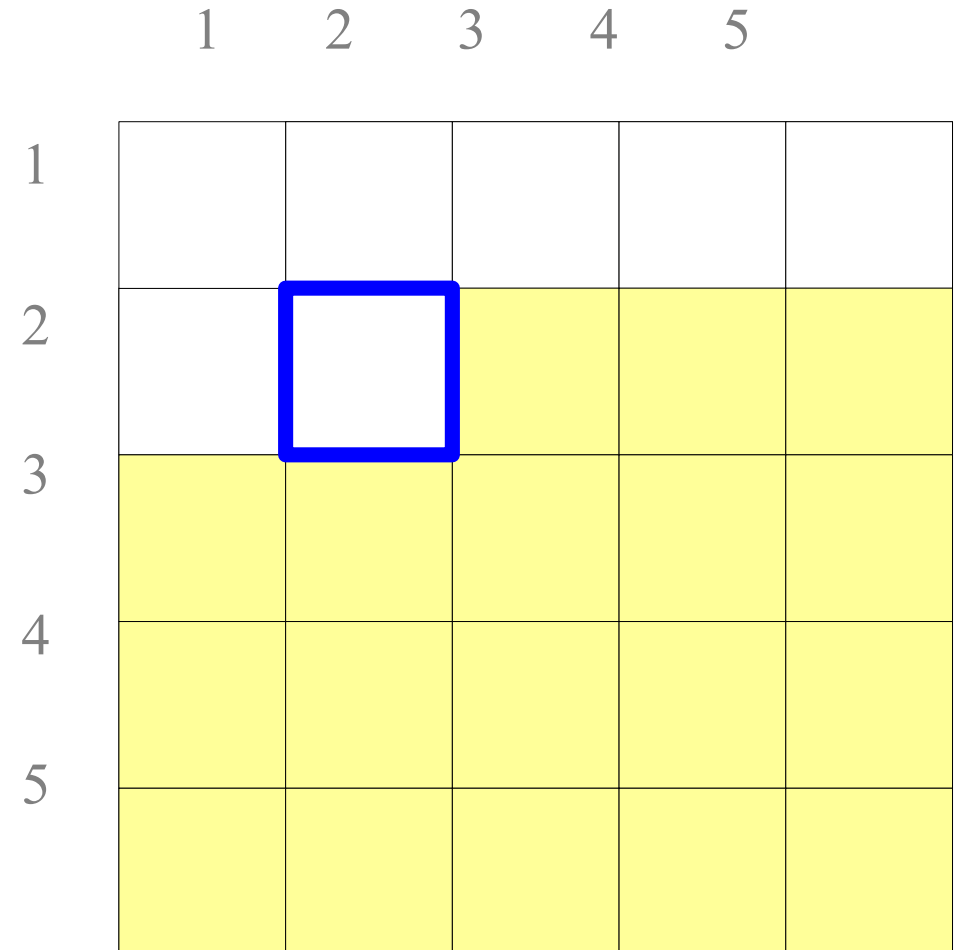
Sobrevivência:  $2 \leq N \leq 3$

Nascimento:  $N=3$

Morte:  $N < 2$  ou  $N > 3$



Estado inicial



Iteração 1

# Simulação

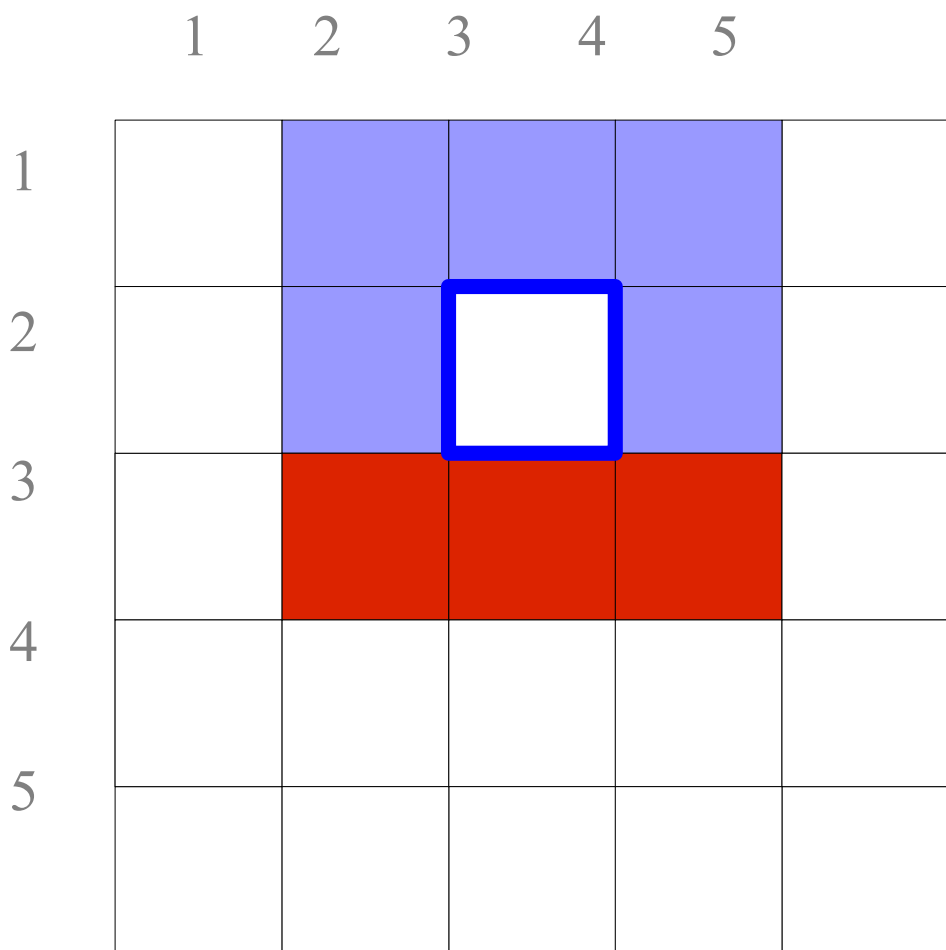
Preto: viva

Branca: morta

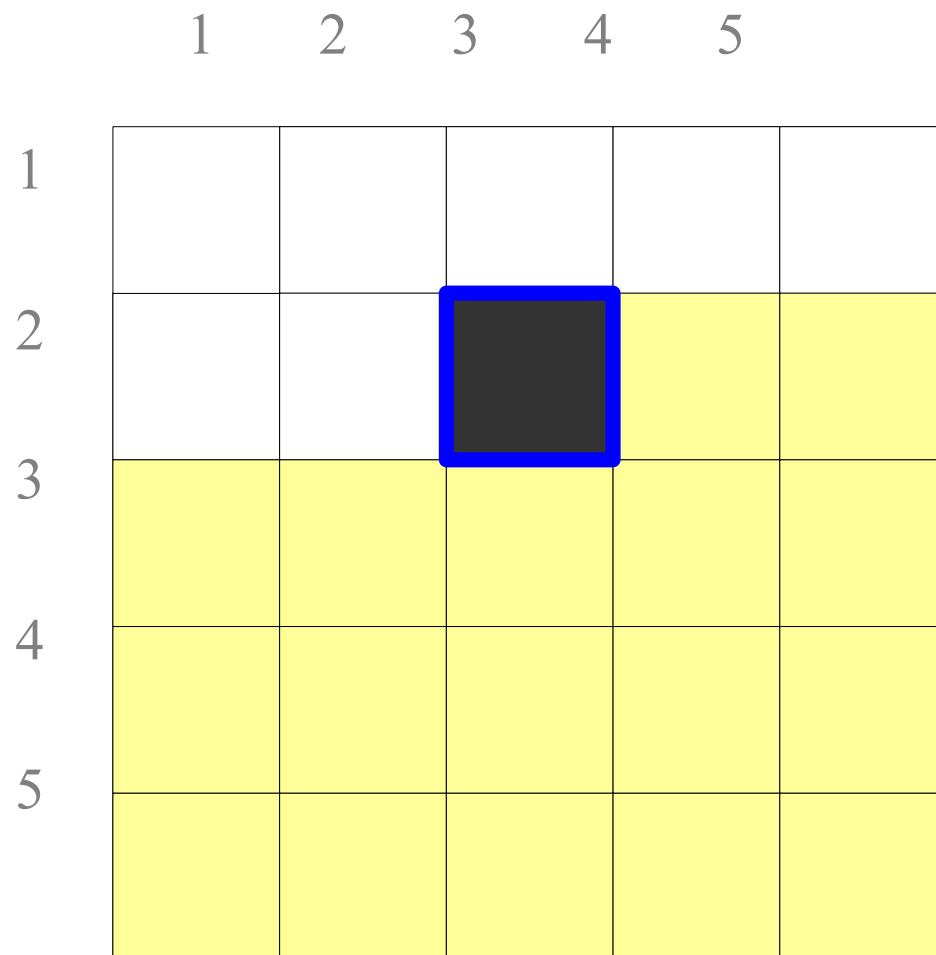
Sobrevivência:  $2 \leq N \leq 3$

Nascimento:  $N=3$

Morte:  $N < 2$  ou  $N > 3$



Estado inicial



Iteração 1



# Simulação

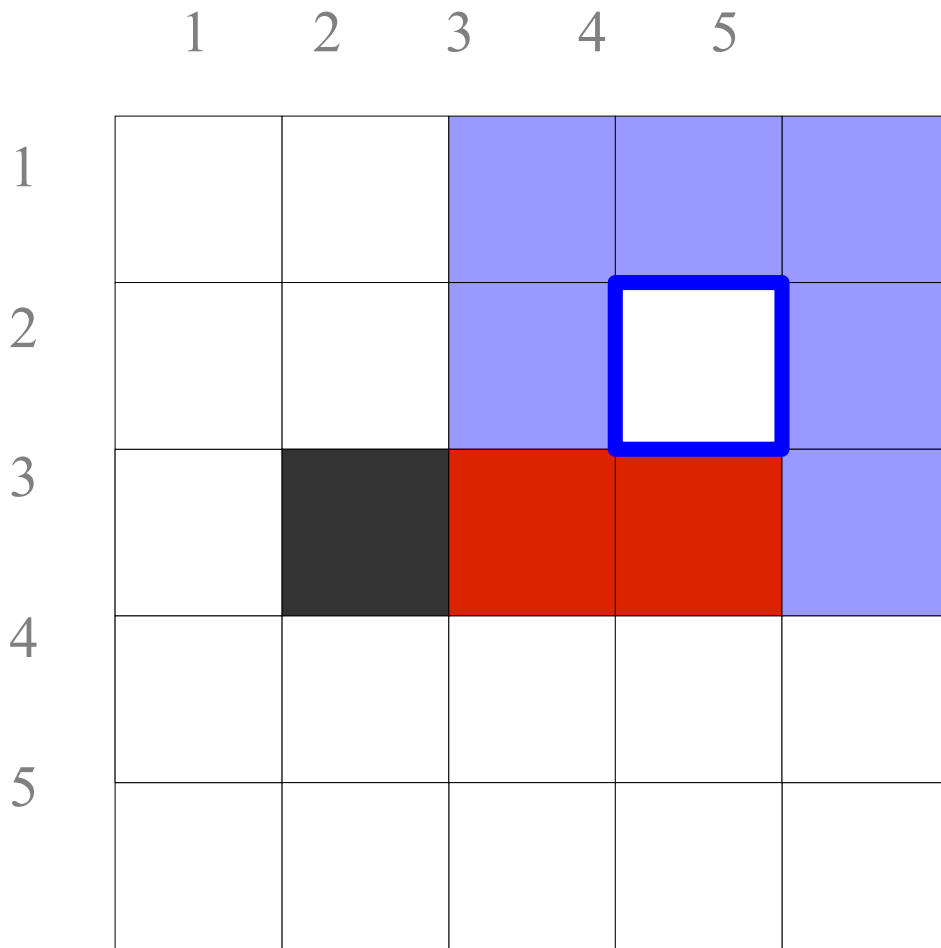
Preto: viva

Branca: morta

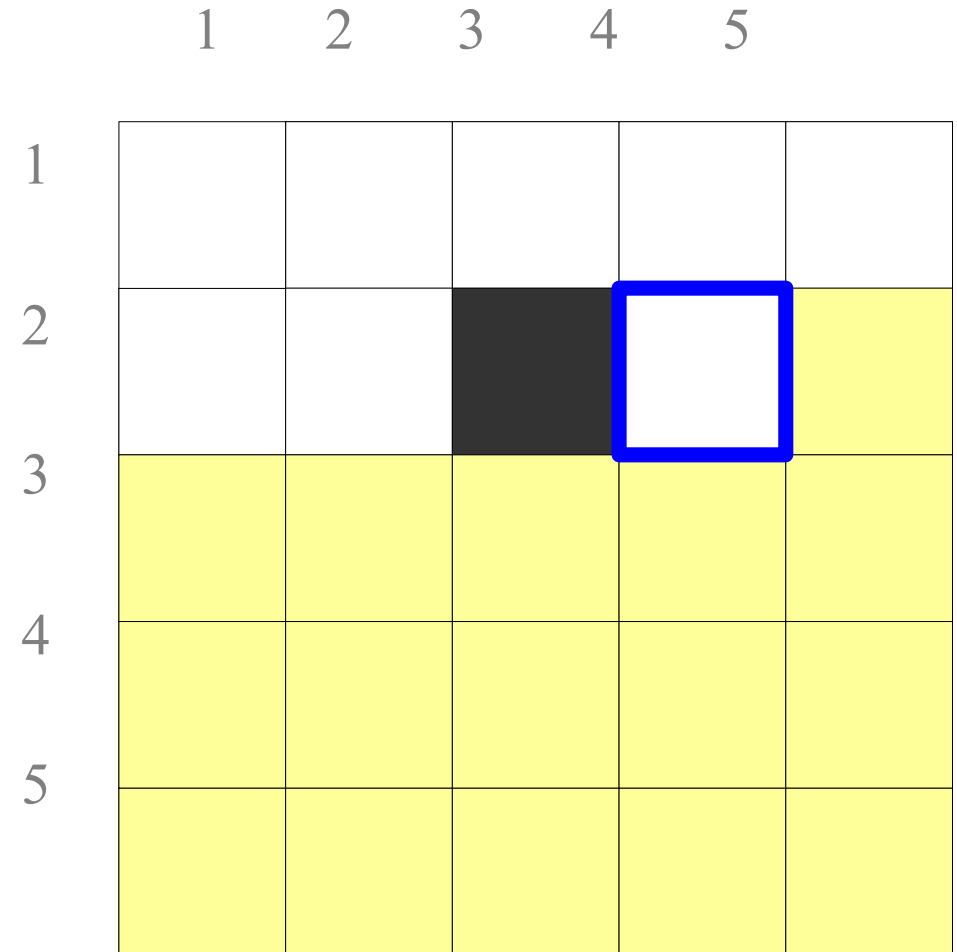
Sobrevivência:  $2 \leq N \leq 3$

Nascimento:  $N=3$

Morte:  $N < 2$  ou  $N > 3$



Estado inicial



Iteração 1

# Simulação

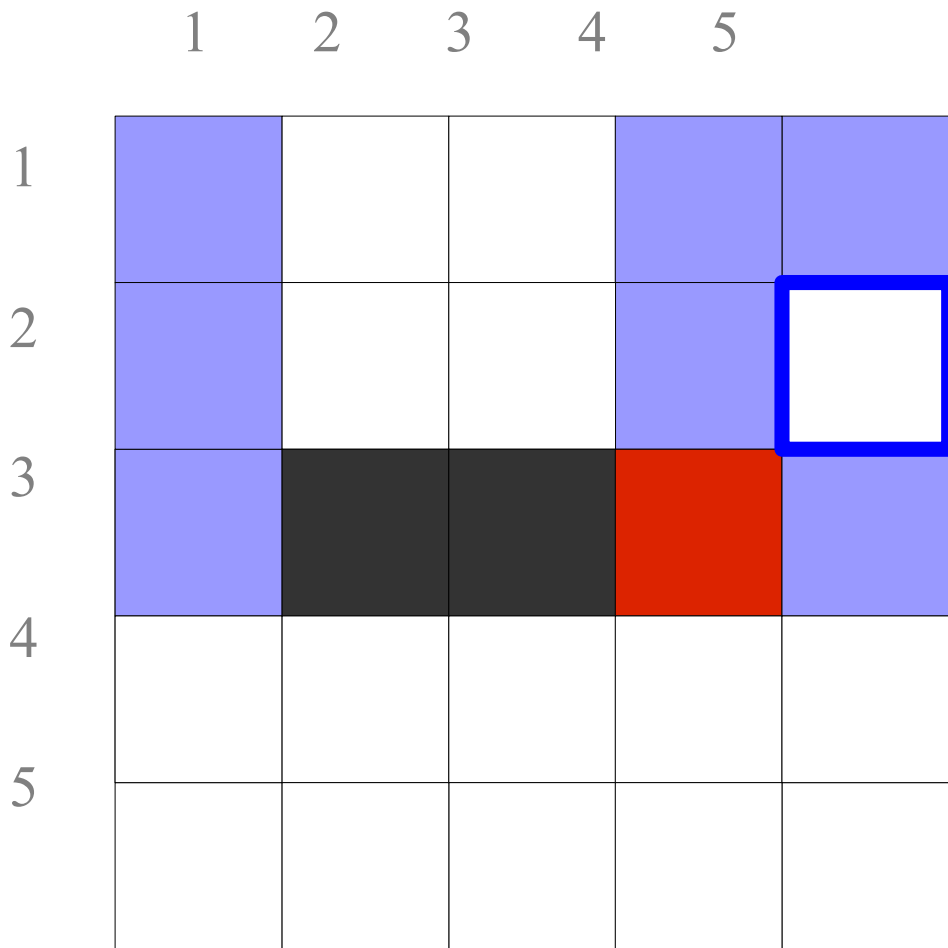
Preto: viva

Branca: morta

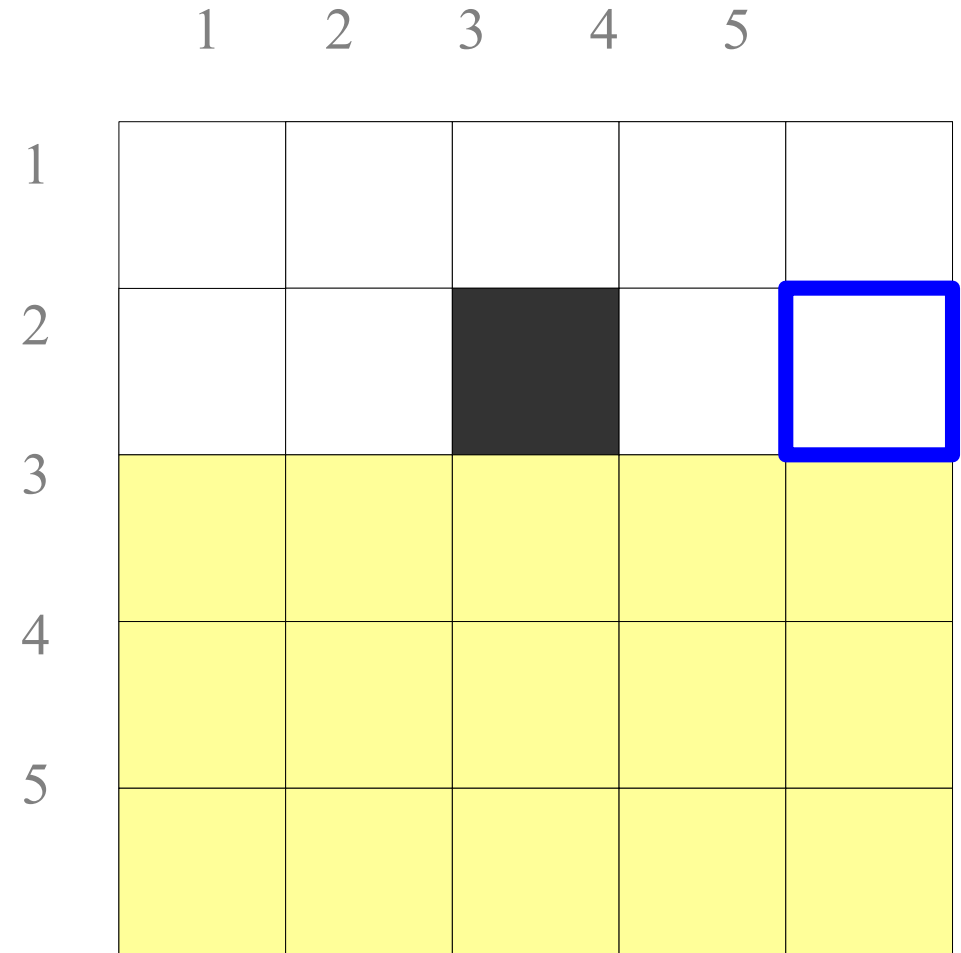
Sobrevivência:  $2 \leq N \leq 3$

Nascimento:  $N=3$

Morte:  $N < 2$  ou  $N > 3$



Estado inicial



Iteração 1

# Simulação

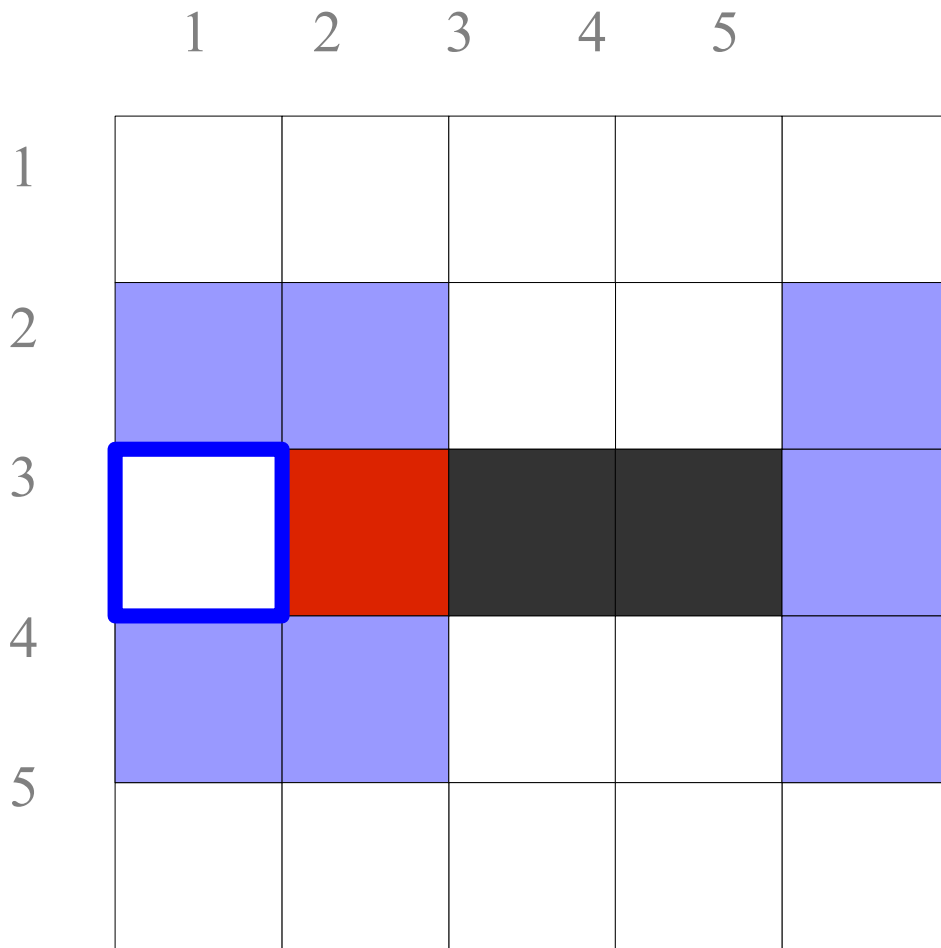
Preto: viva

Branca: morta

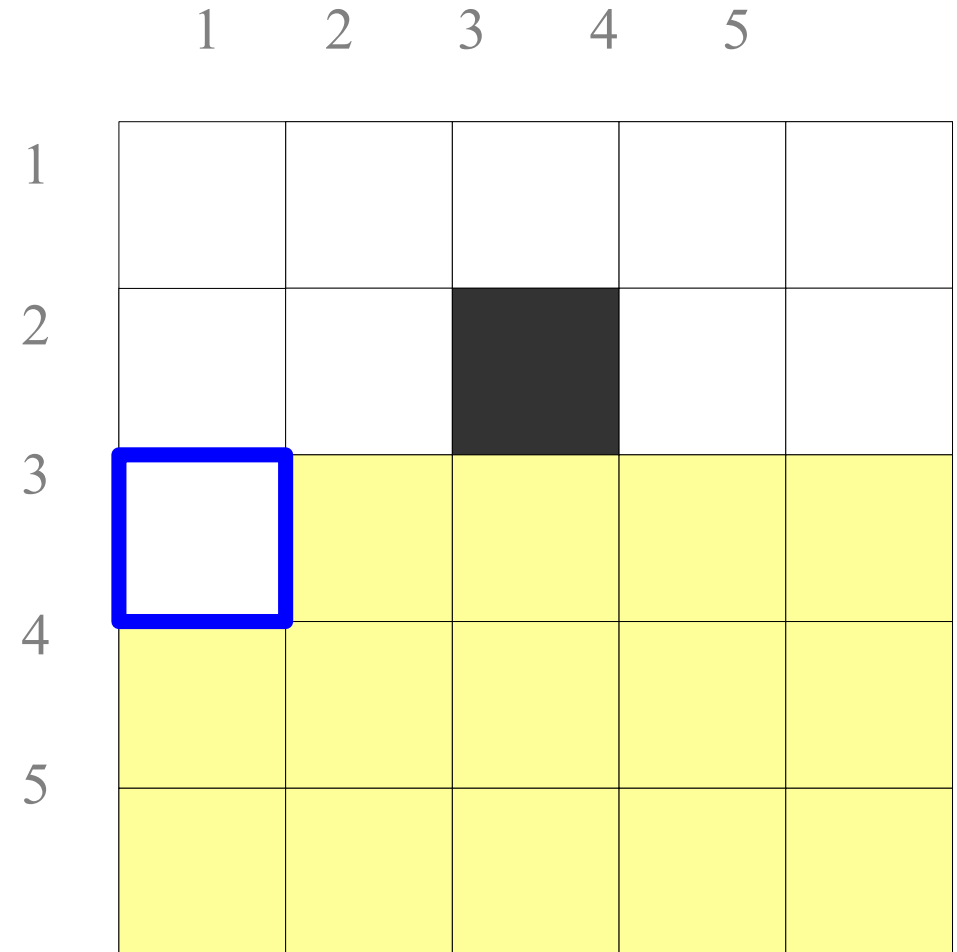
Sobrevivência:  $2 \leq N \leq 3$

Nascimento:  $N=3$

Morte:  $N < 2$  ou  $N > 3$



Estado inicial



Iteração 1

# Simulação

Preto: viva

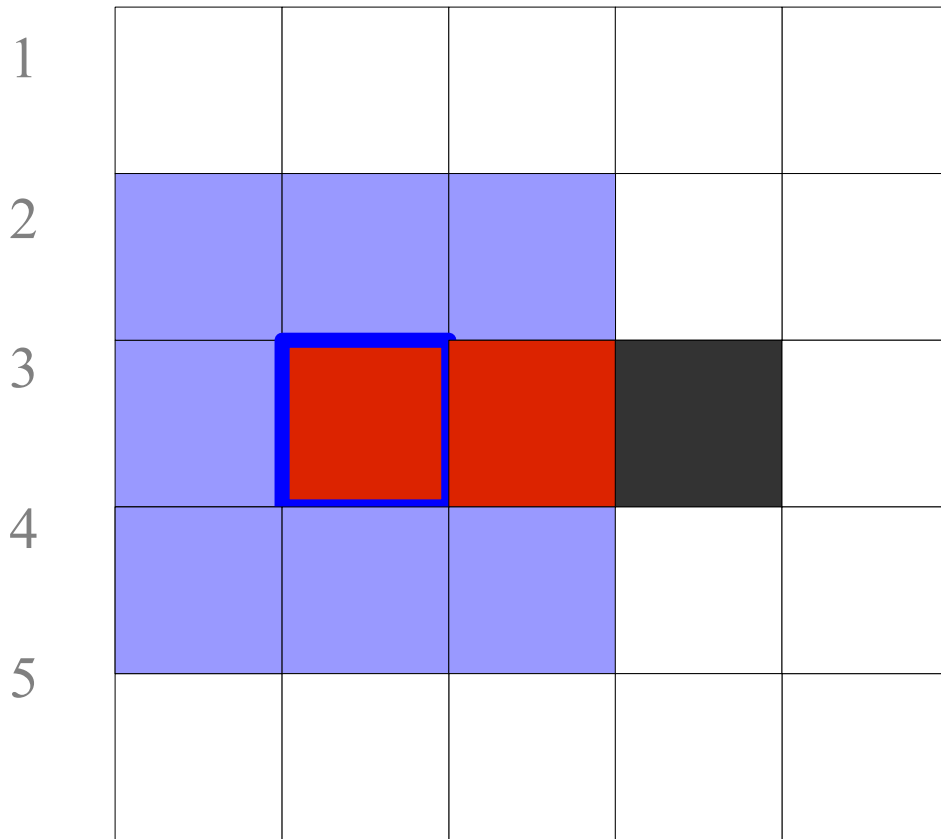
Branca: morta

Sobrevivência:  $2 \leq N \leq 3$

Nascimento:  $N=3$

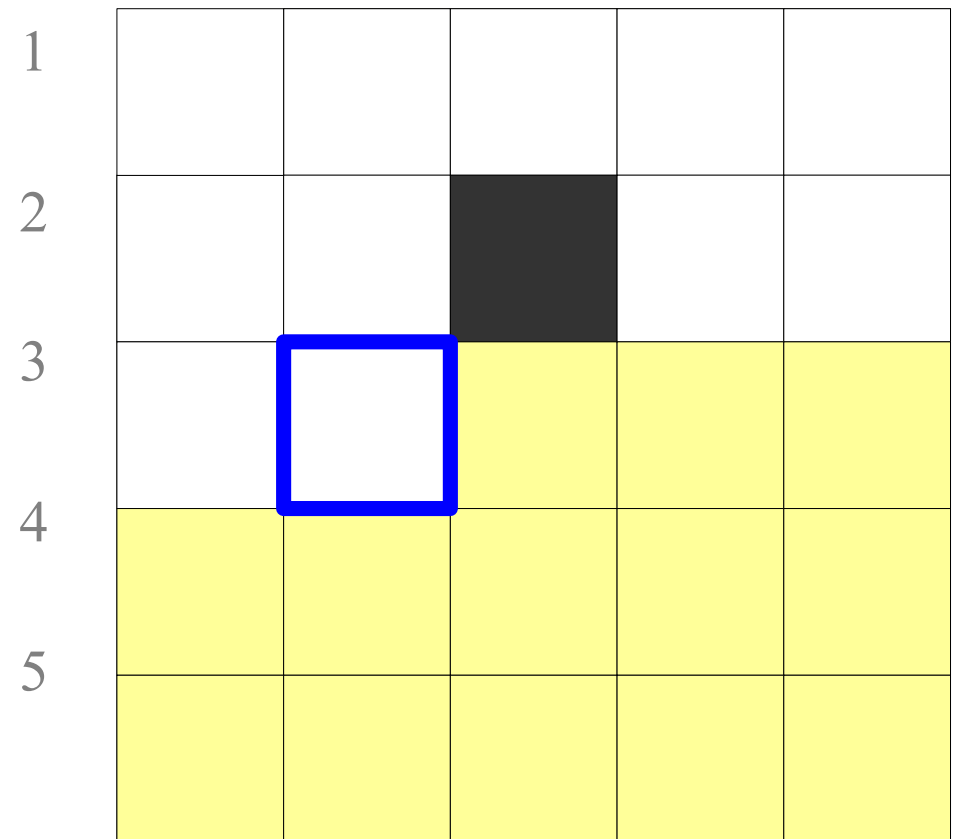
**Morte:**  $N < 2$  ou  $N > 3$

1 2 3 4 5



Estado inicial

1 2 3 4 5



Iteração 1

# Simulação

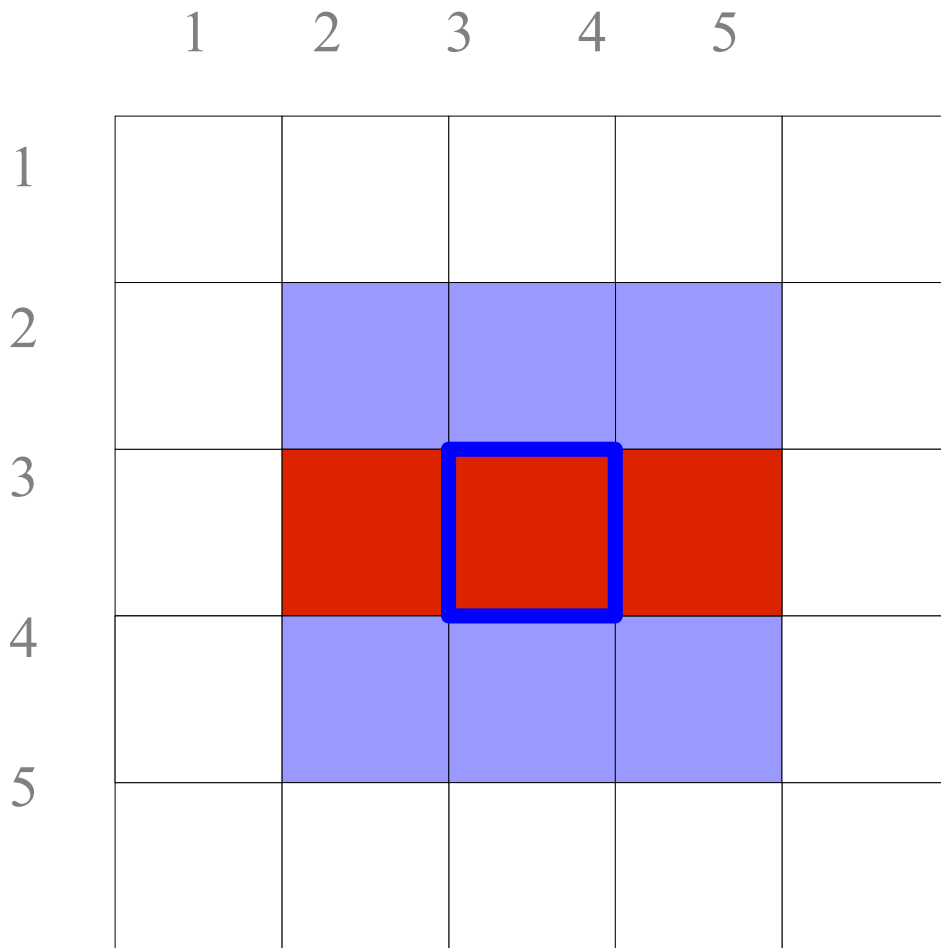
Preto: viva

Branca: morta

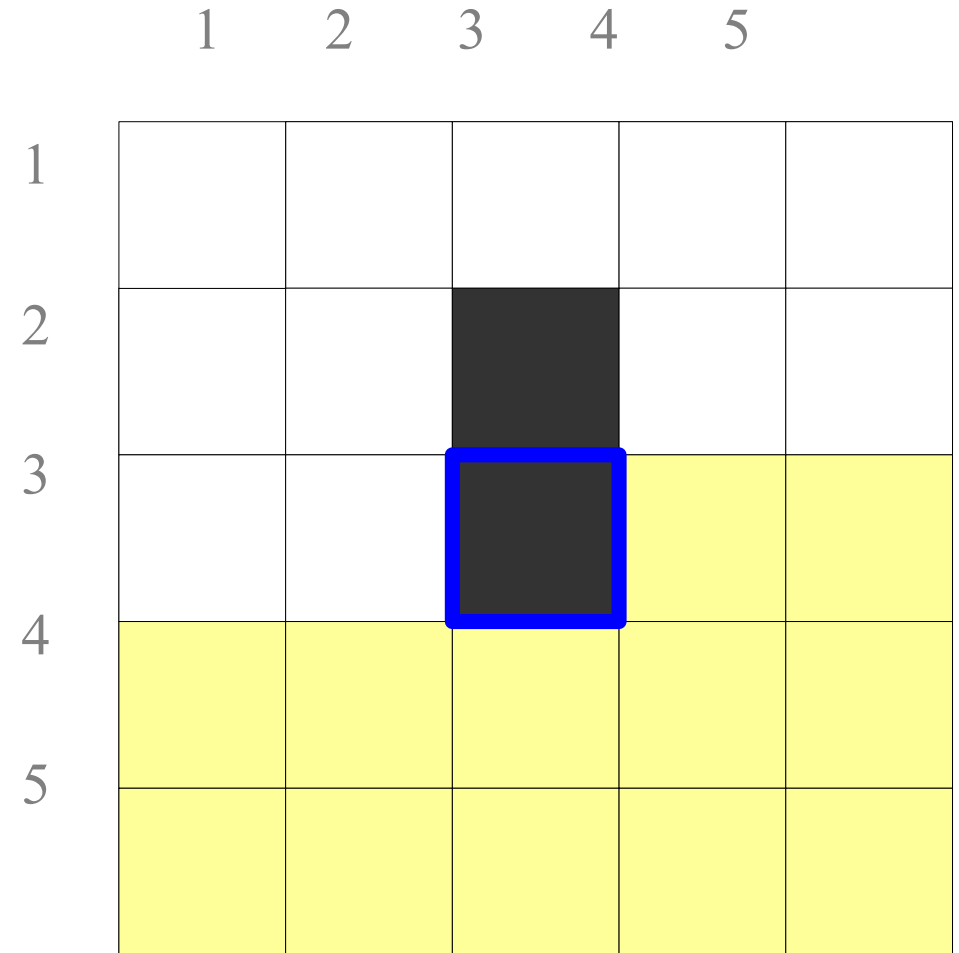
**Sobrevivência:**  $2 \leq N \leq 3$

Nascimento:  $N=3$

Morte:  $N < 2$  ou  $N > 3$



Estado inicial



Iteração 1



# Simulação

Preto: viva

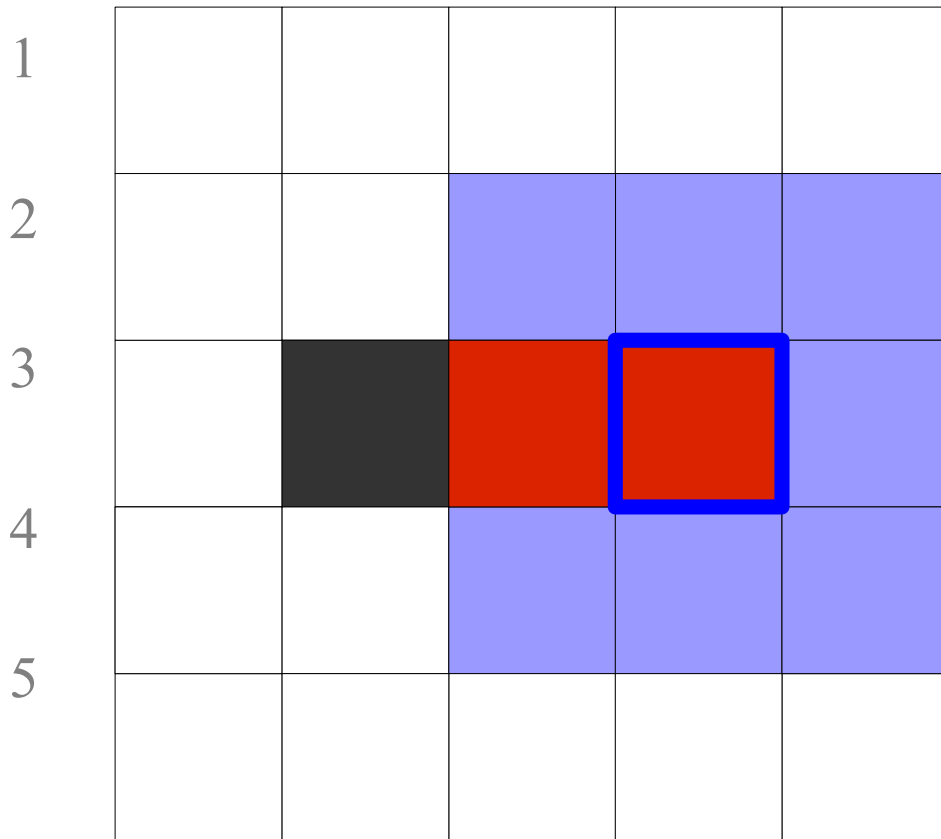
Branca: morta

Sobrevivência:  $2 \leq N \leq 3$

Nascimento:  $N=3$

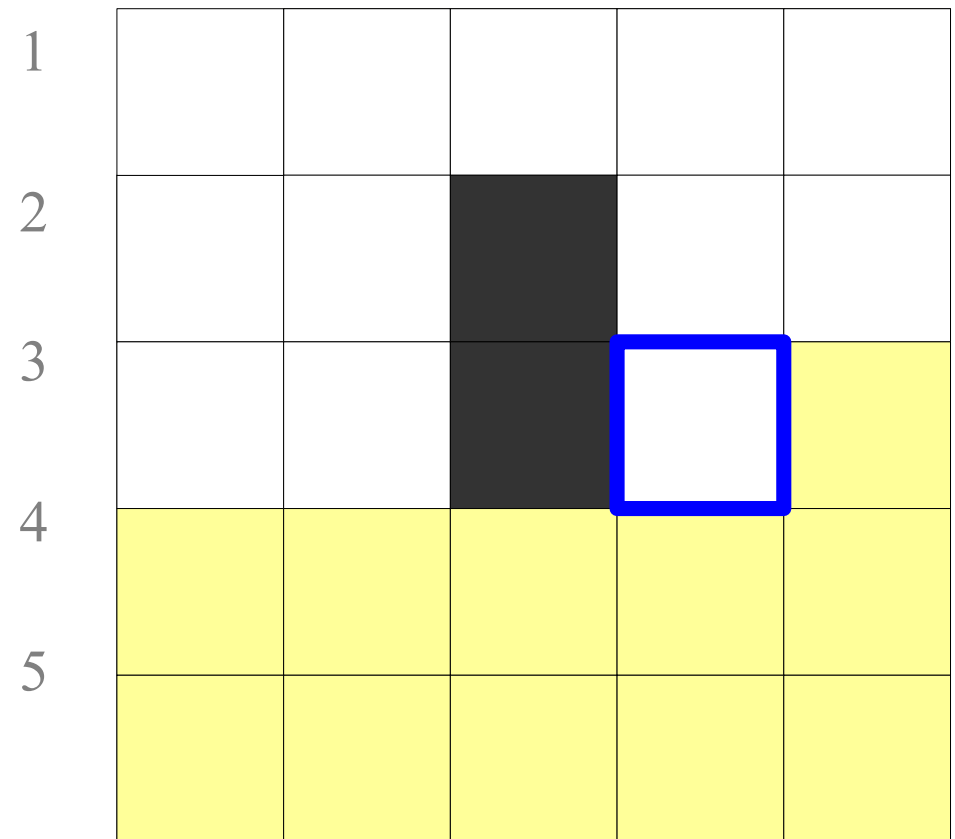
**Morte:**  $N < 2$  ou  $N > 3$

1 2 3 4 5



Estado inicial

1 2 3 4 5



Iteração 1

# Simulação

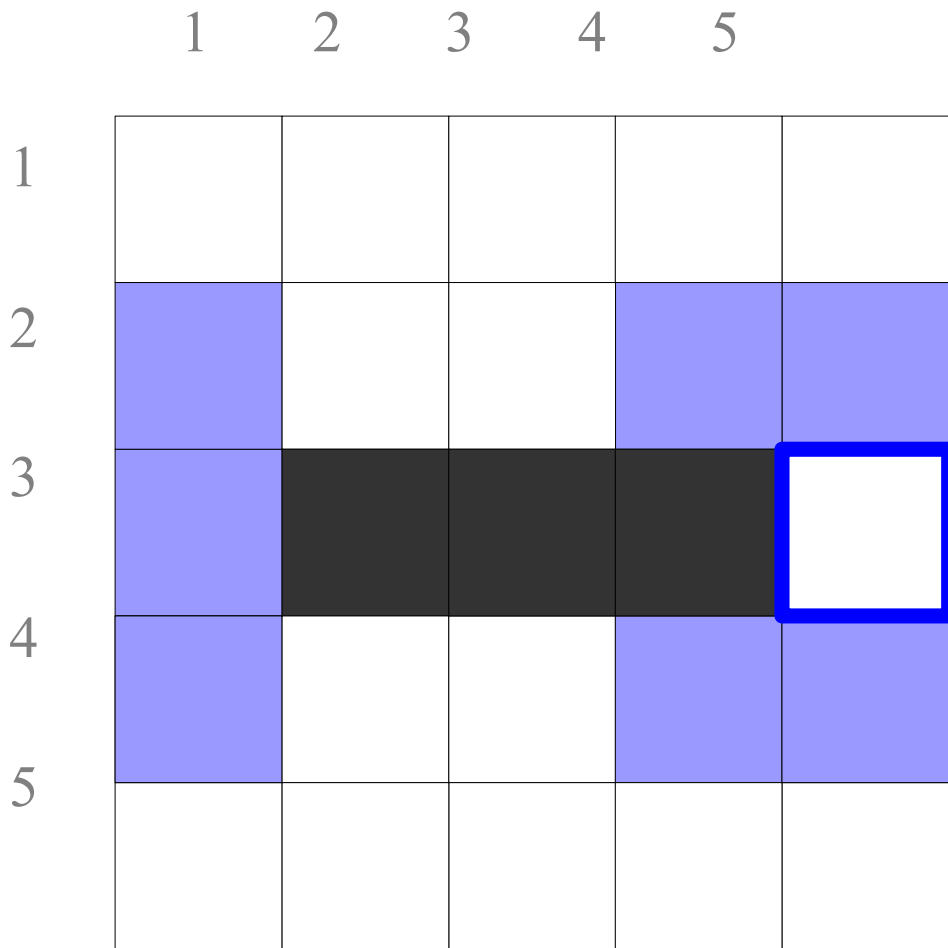
Preto: viva

Branca: morta

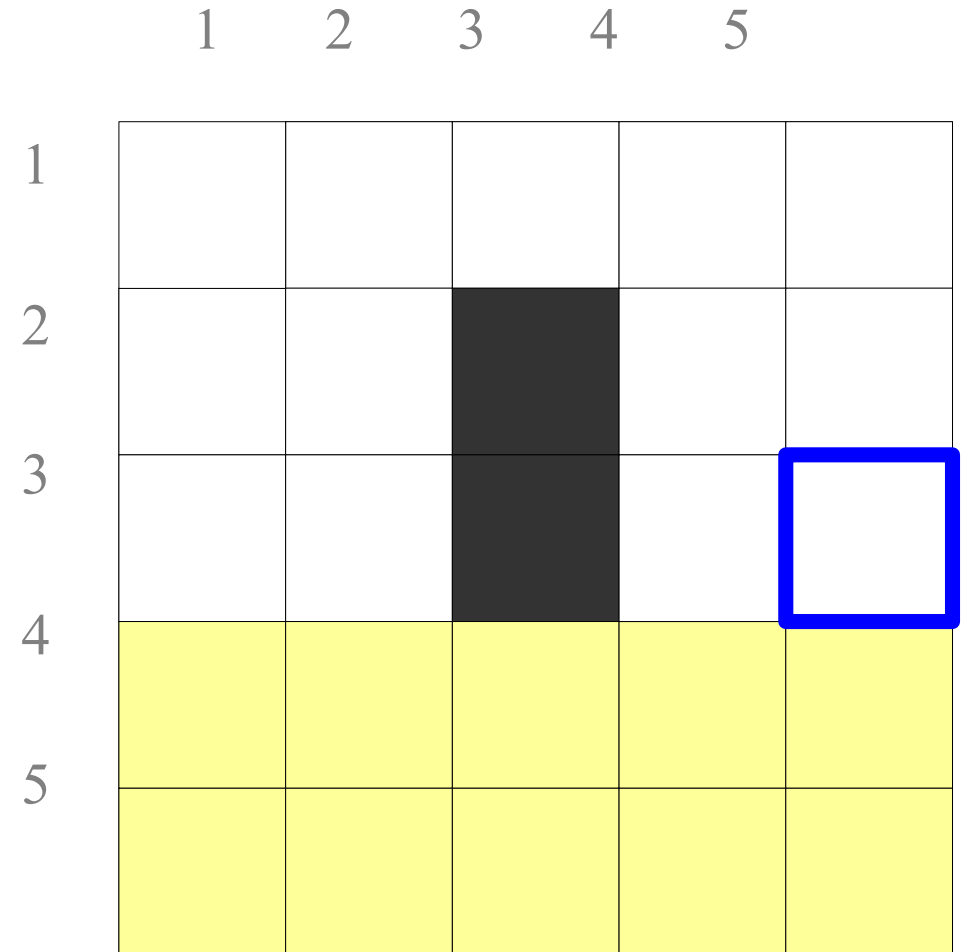
Sobrevivência:  $2 \leq N \leq 3$

Nascimento:  $N=3$

Morte:  $N < 2$  ou  $N > 3$



Estado inicial



Iteração 1

# Simulação

Preto: viva

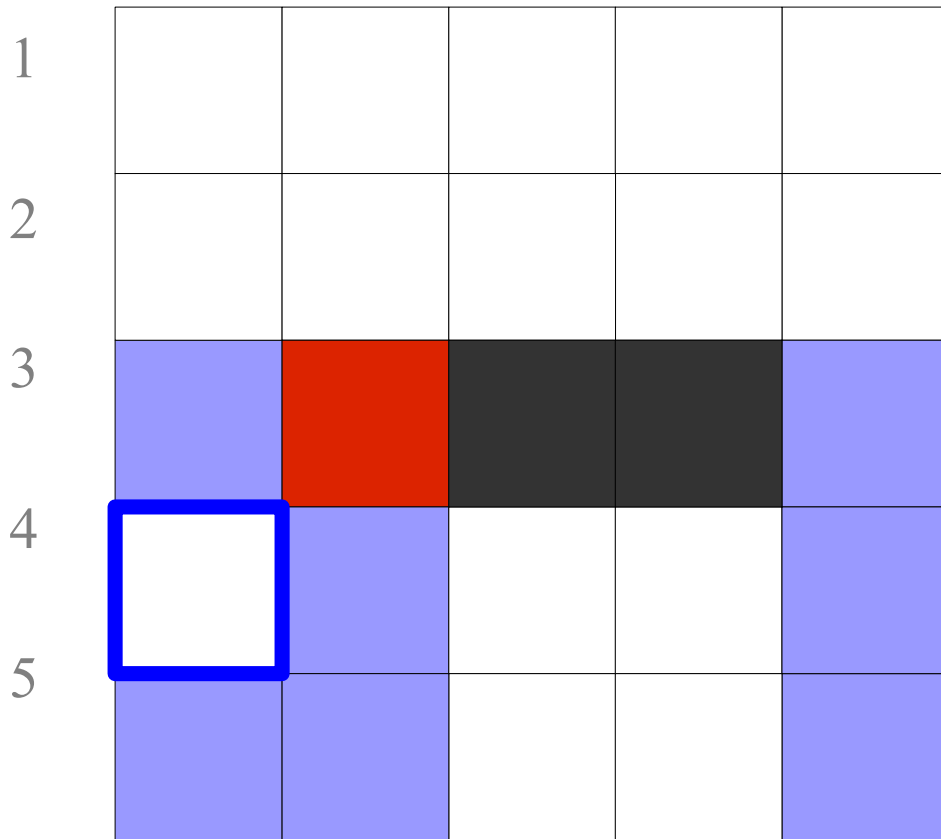
Branca: morta

Sobrevivência:  $2 \leq N \leq 3$

Nascimento:  $N=3$

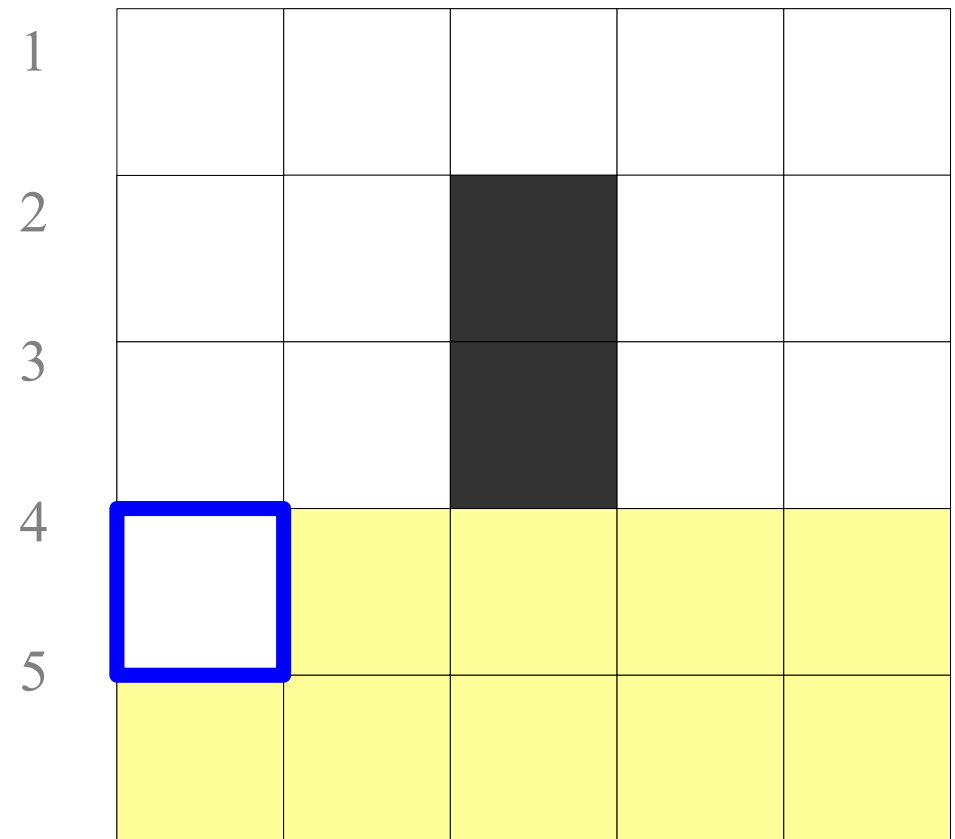
Morte:  $N < 2$  ou  $N > 3$

1 2 3 4 5



Estado inicial

1 2 3 4 5



Iteração 1

# Simulação

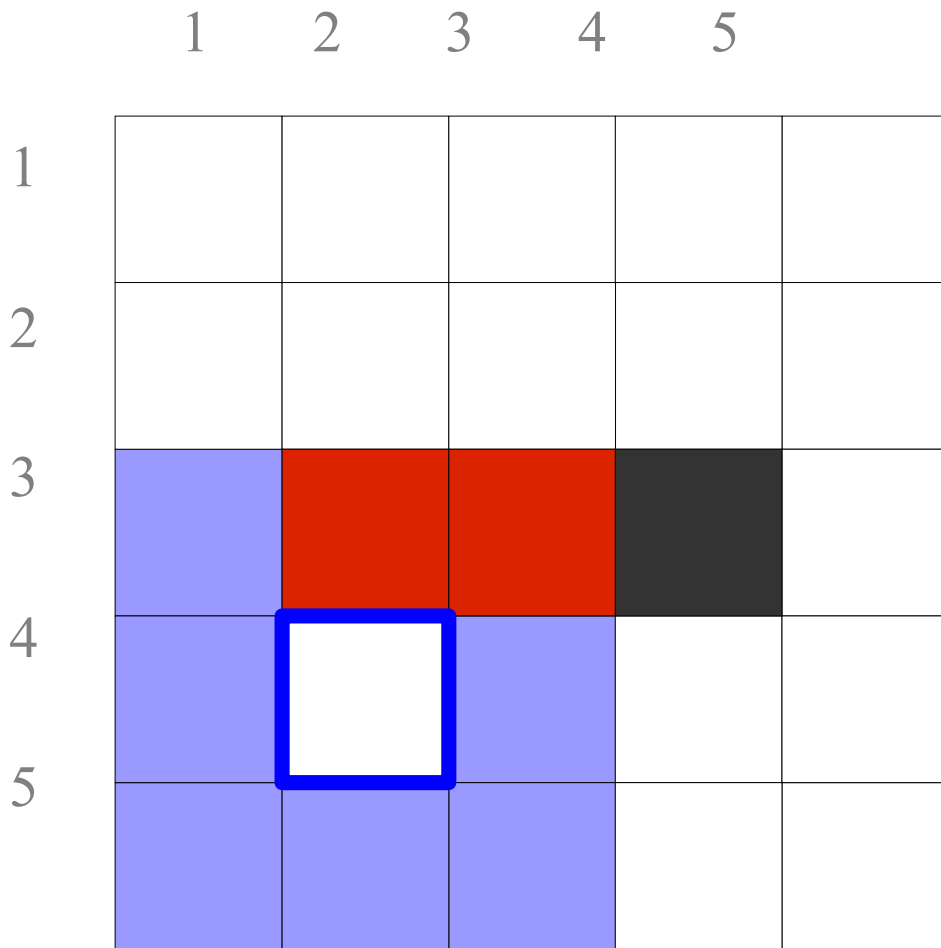
Preto: viva

Branca: morta

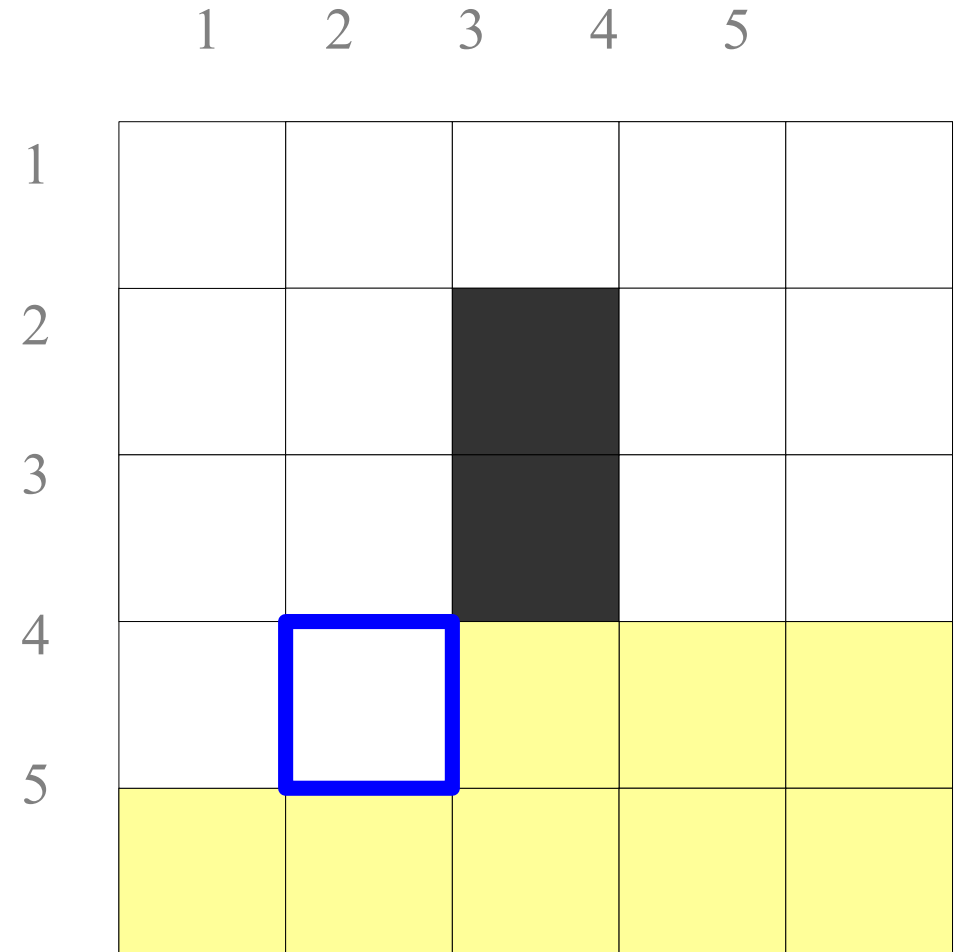
Sobrevivência:  $2 \leq N \leq 3$

Nascimento:  $N=3$

Morte:  $N < 2$  ou  $N > 3$



Estado inicial



Iteração 1

# Simulação

Preto: viva

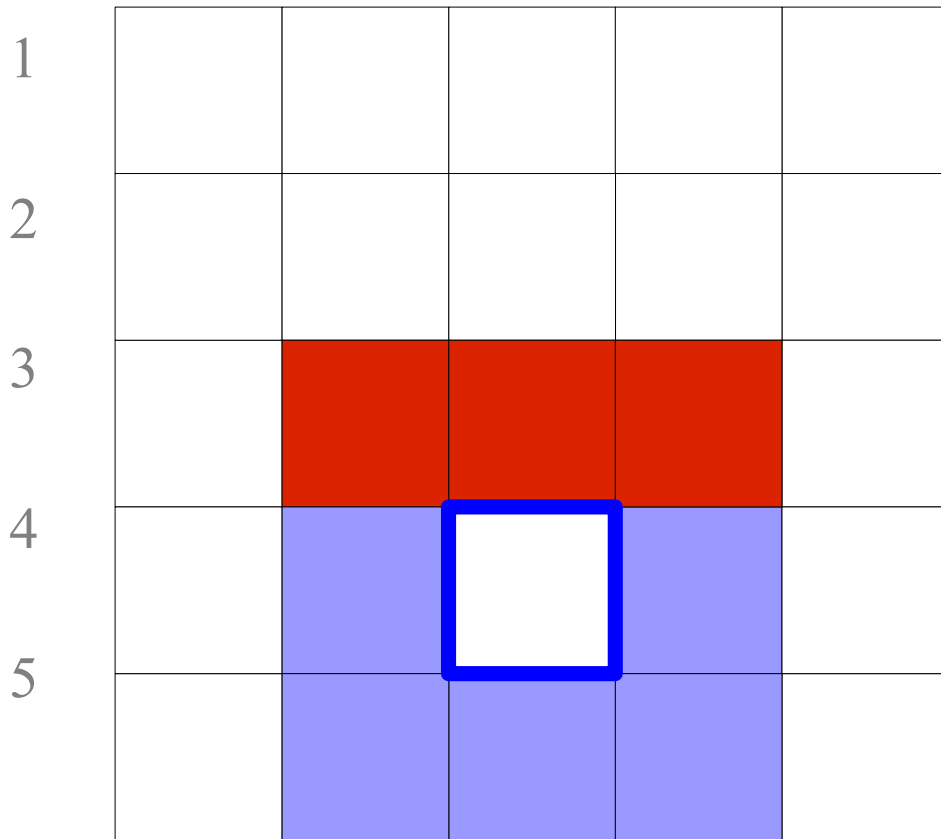
Branca: morta

Sobrevivência:  $2 \leq N \leq 3$

Nascimento:  $N=3$

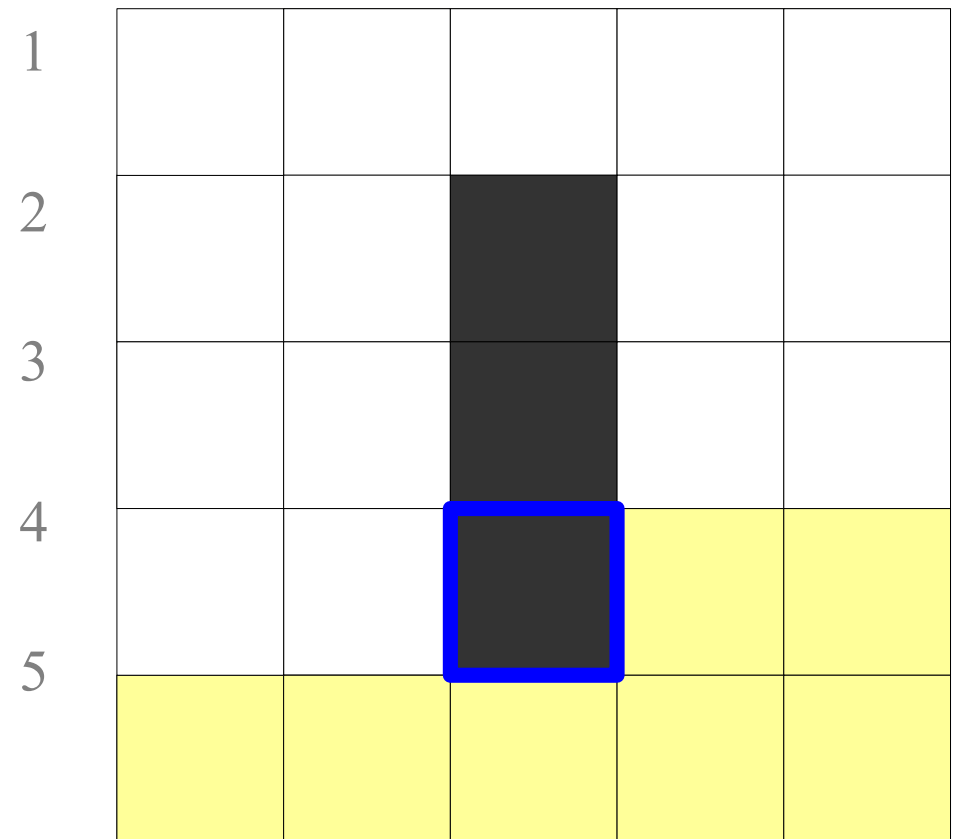
Morte:  $N < 2$  ou  $N > 3$

1 2 3 4 5



Estado inicial

1 2 3 4 5



Iteração 1

# Simulação

Preto: viva

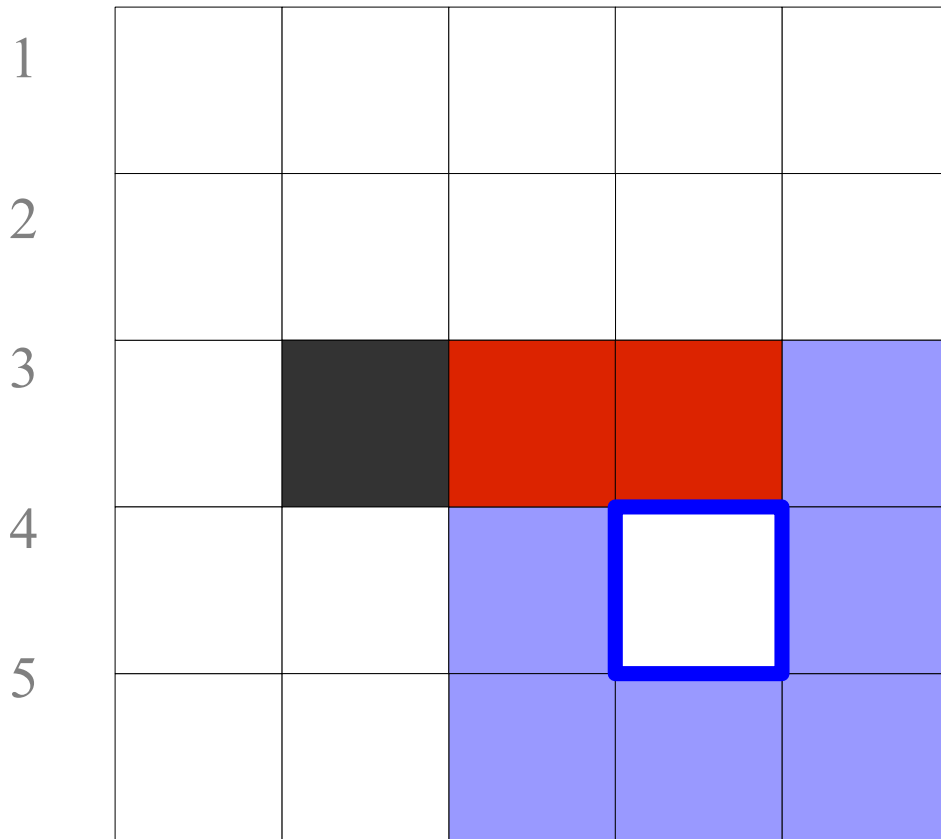
Branca: morta

Sobrevivência:  $2 \leq N \leq 3$

Nascimento:  $N=3$

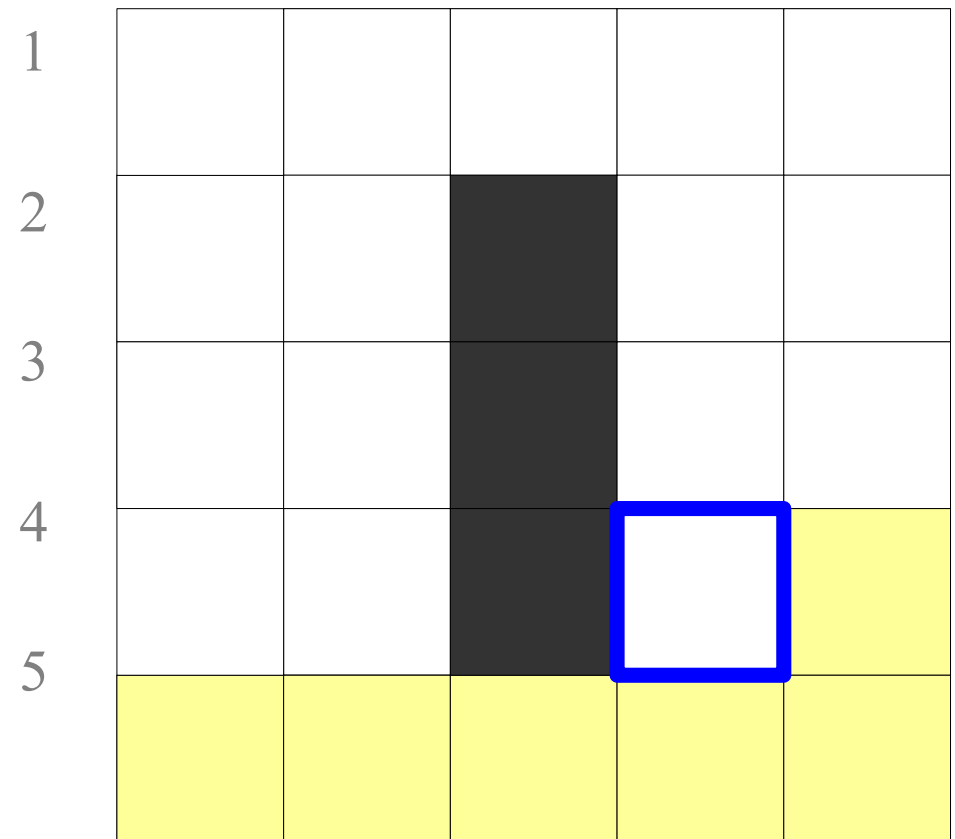
Morte:  $N < 2$  ou  $N > 3$

1 2 3 4 5



Estado inicial

1 2 3 4 5



Iteração 1

# Simulação

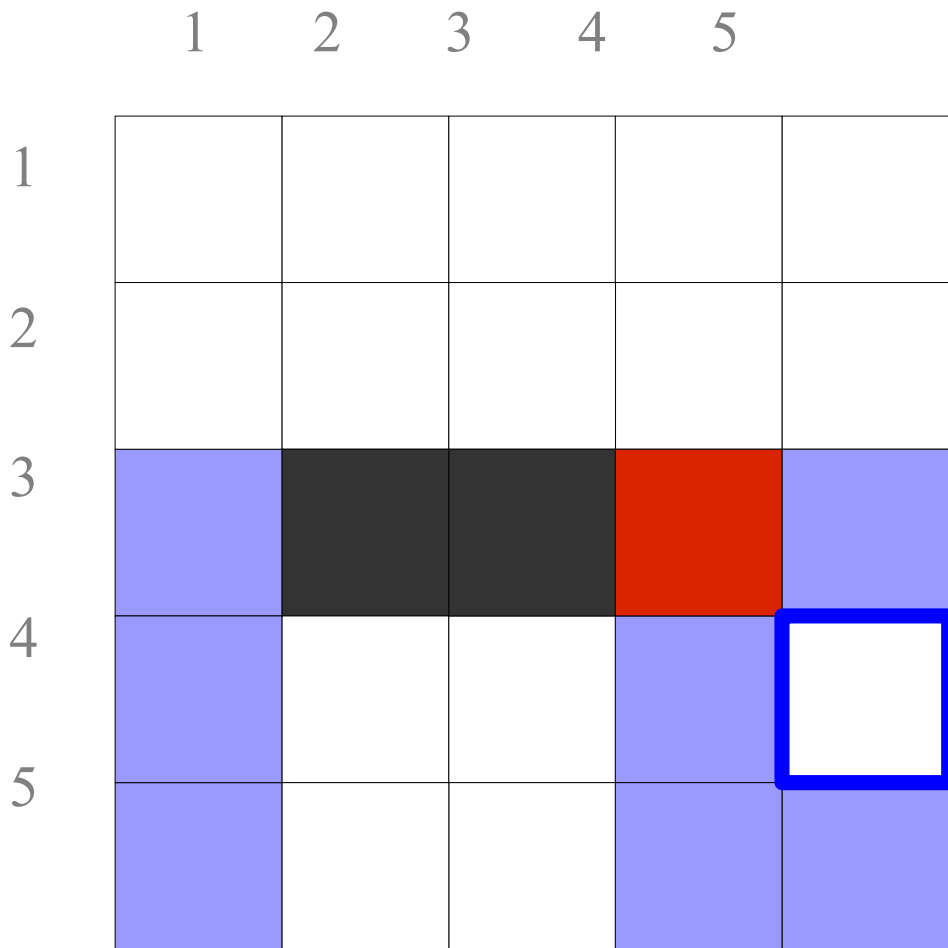
Preto: viva

Branca: morta

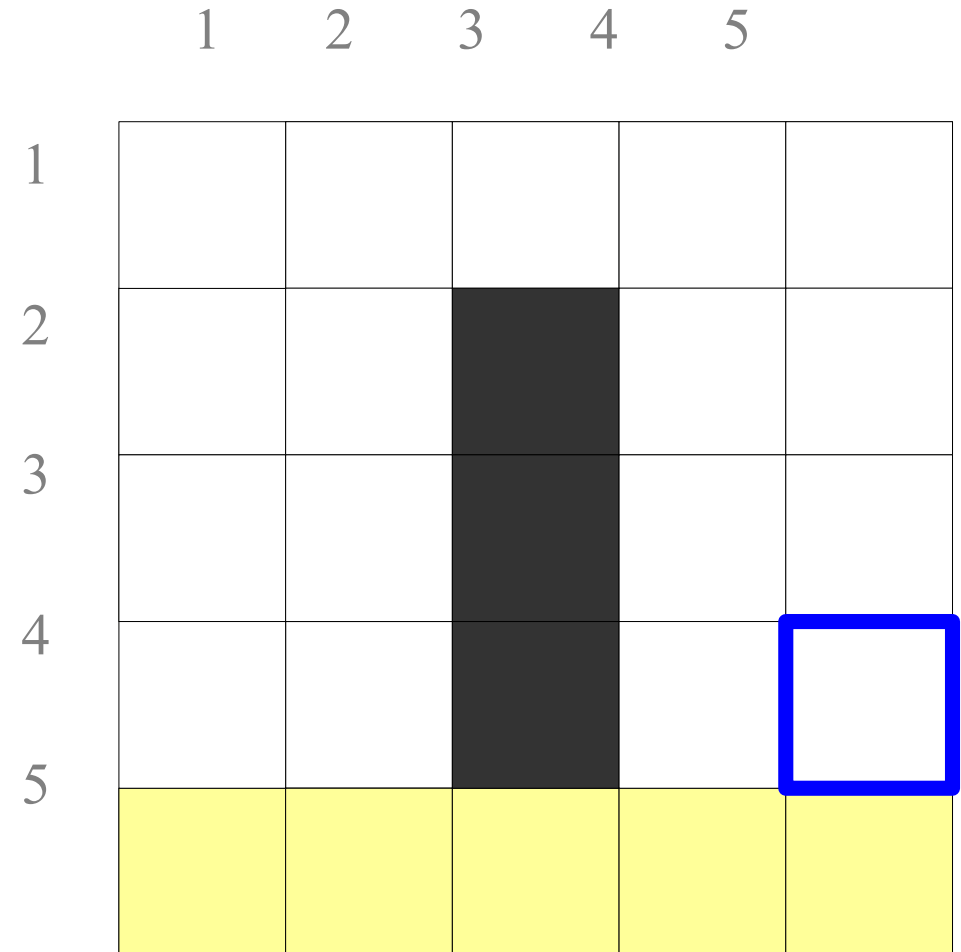
Sobrevivência:  $2 \leq N \leq 3$

Nascimento:  $N=3$

Morte:  $N < 2$  ou  $N > 3$



Estado inicial



Iteração 1

# Simulação

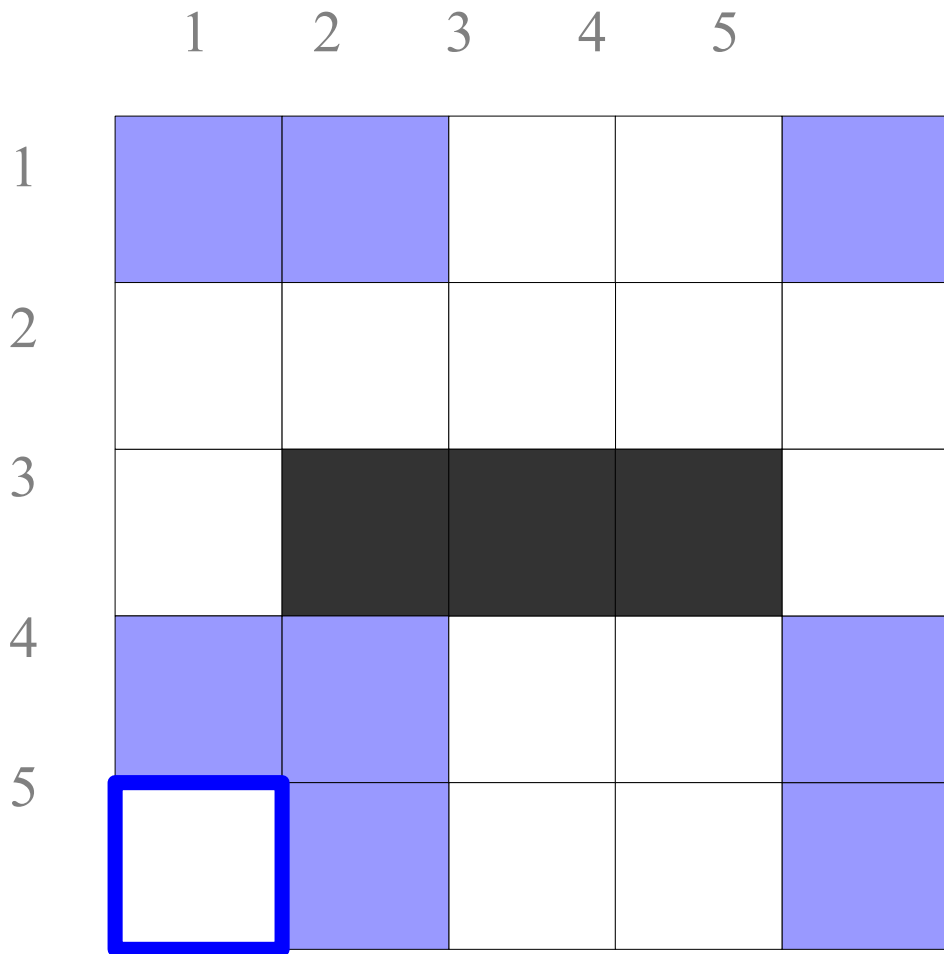
Preto: viva

Branca: morta

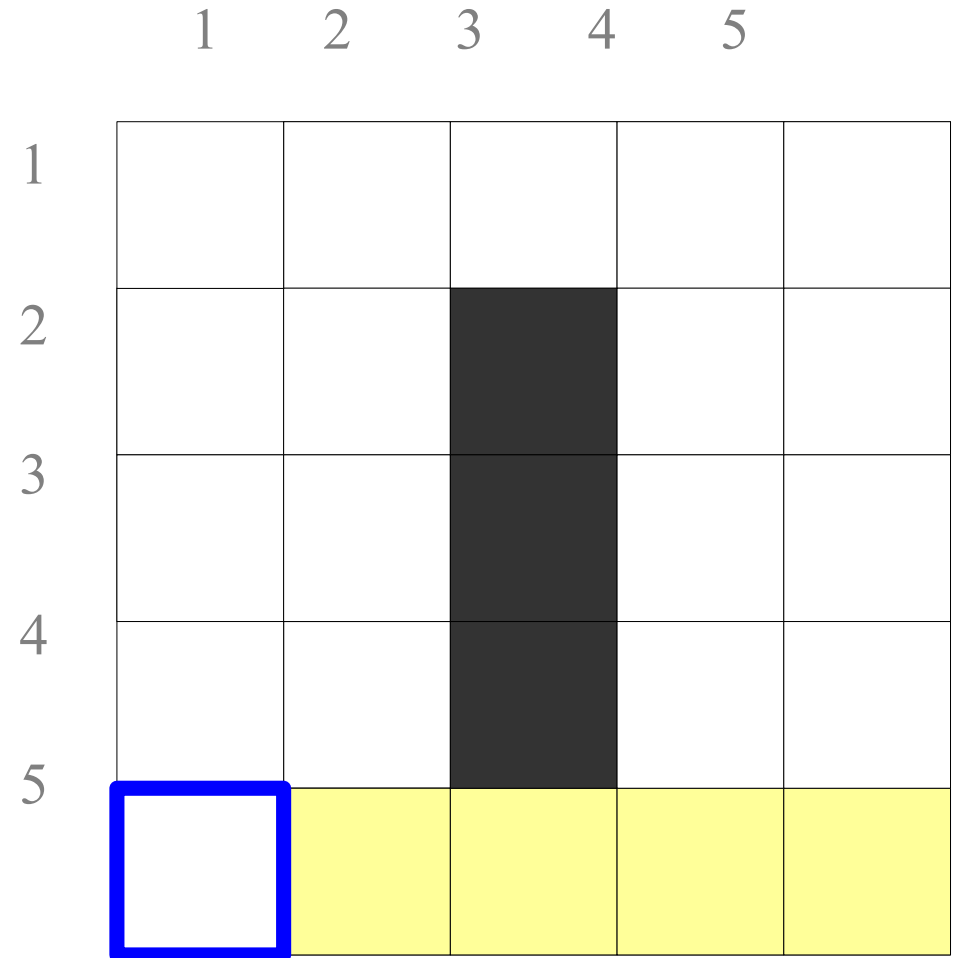
Sobrevivência:  $2 \leq N \leq 3$

Nascimento:  $N=3$

Morte:  $N < 2$  ou  $N > 3$



Estado inicial



Iteração 1



# Simulação

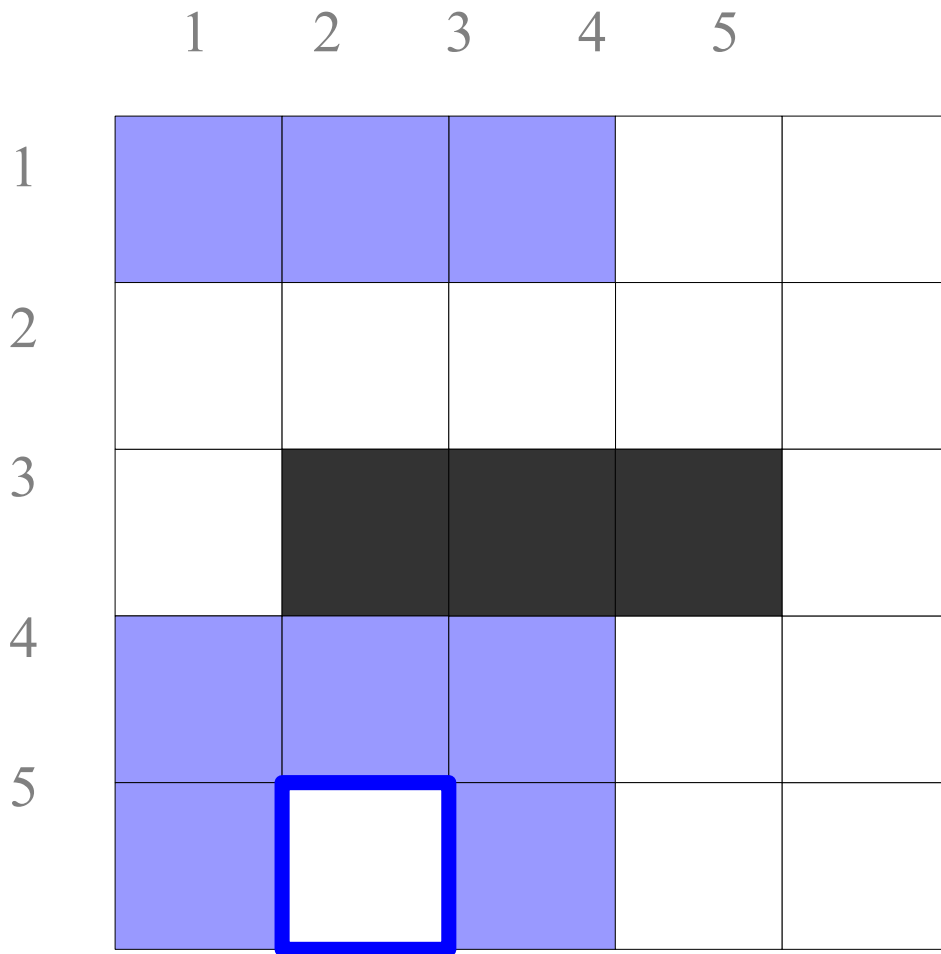
Preto: viva

Branca: morta

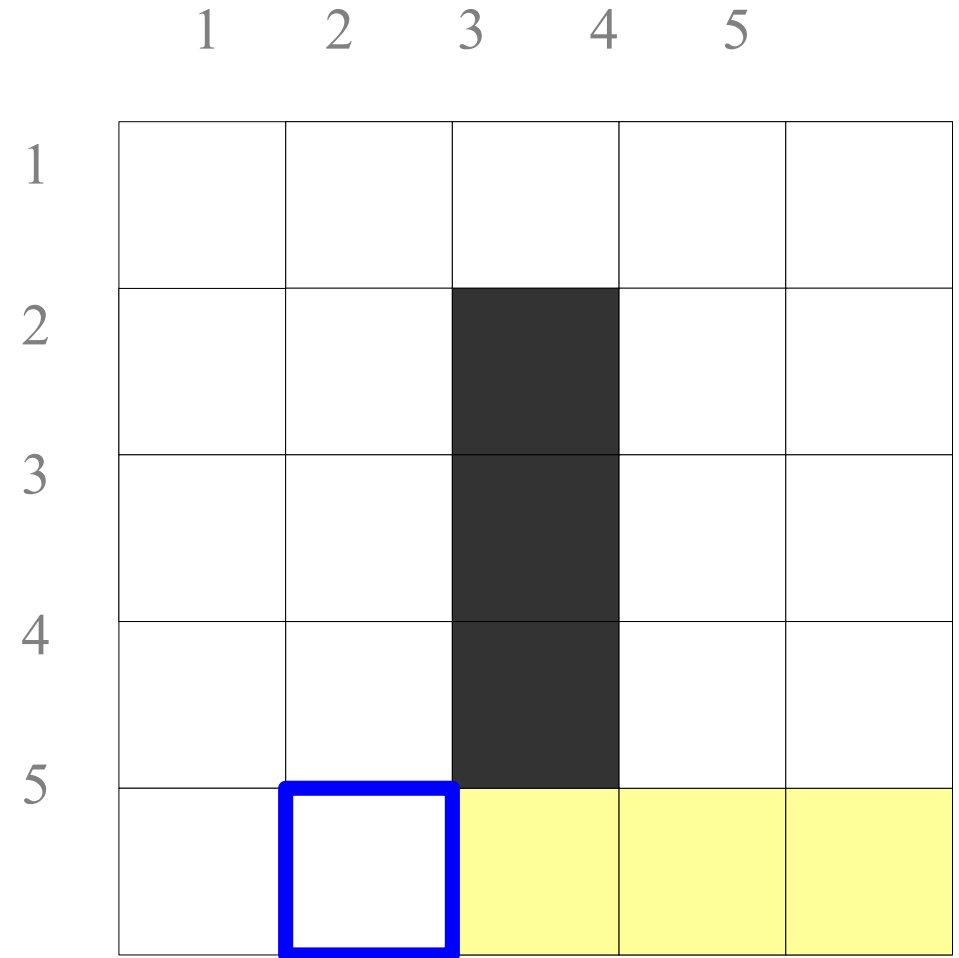
Sobrevivência:  $2 \leq N \leq 3$

Nascimento:  $N=3$

Morte:  $N < 2$  ou  $N > 3$



Estado inicial



Iteração 1

# Simulação

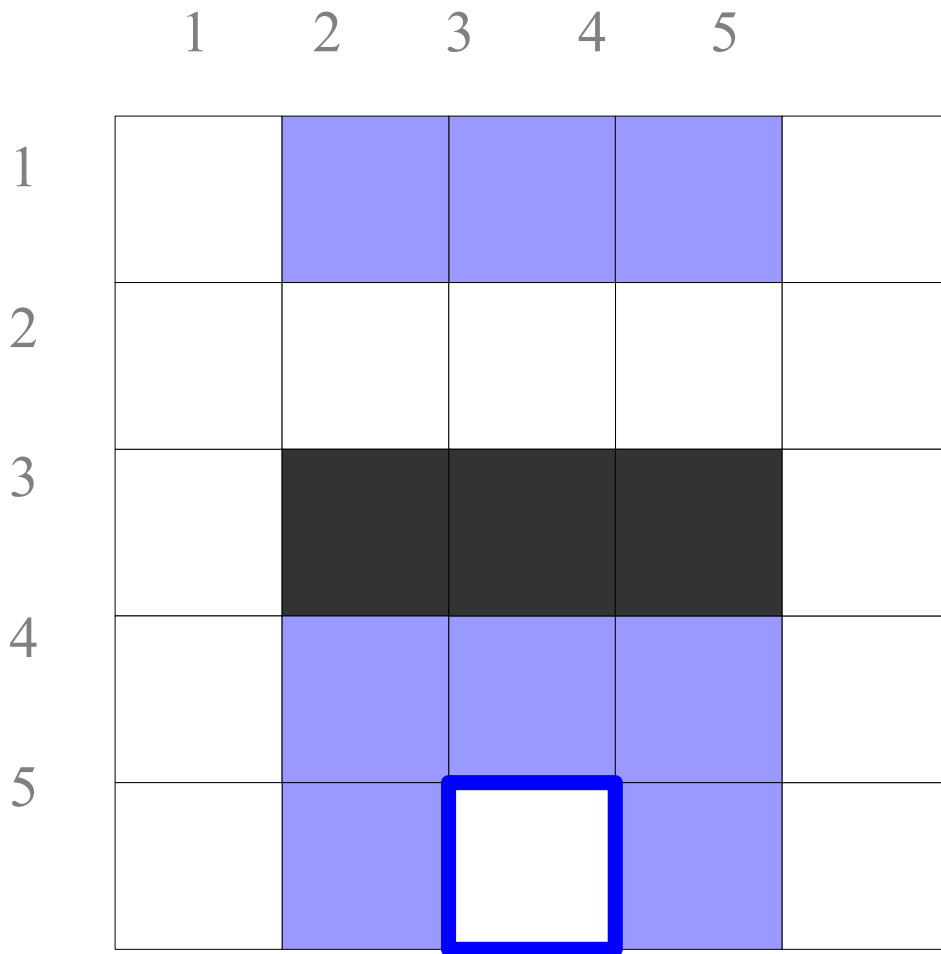
Preto: viva

Branca: morta

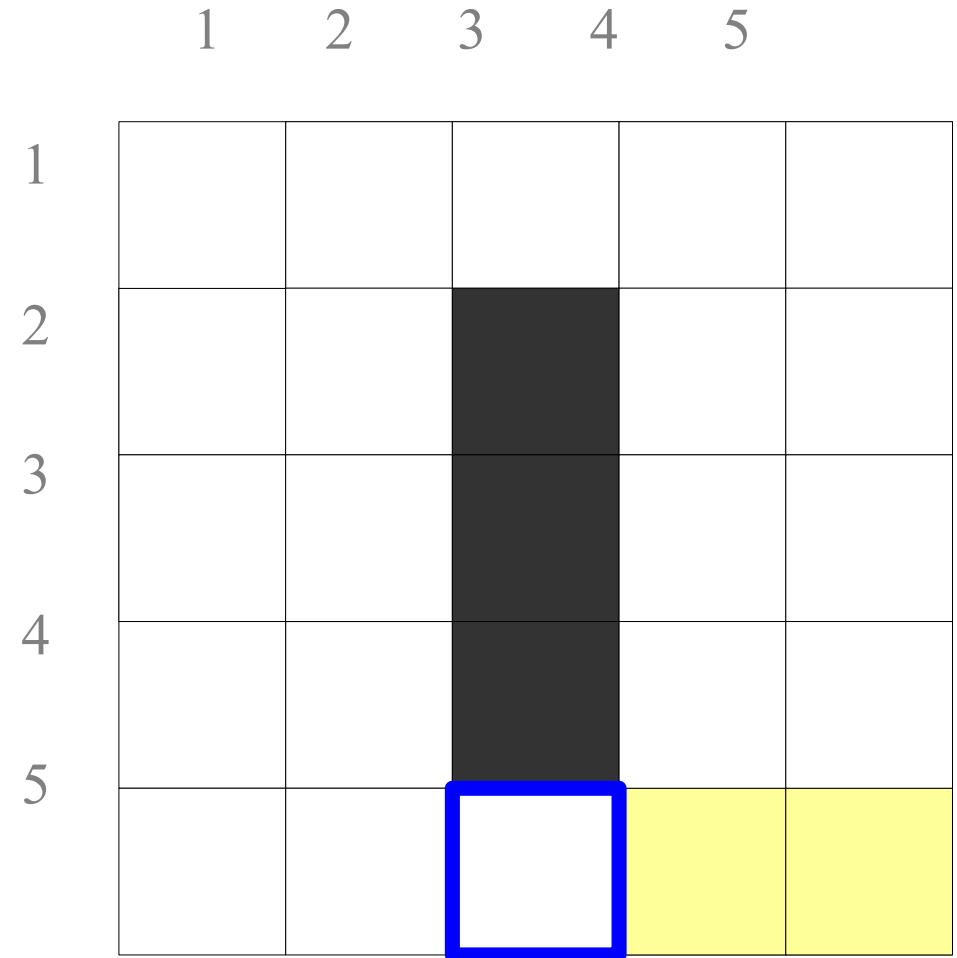
Sobrevivência:  $2 \leq N \leq 3$

Nascimento:  $N=3$

Morte:  $N < 2$  ou  $N > 3$



Estado inicial



Iteração 1

# Simulação

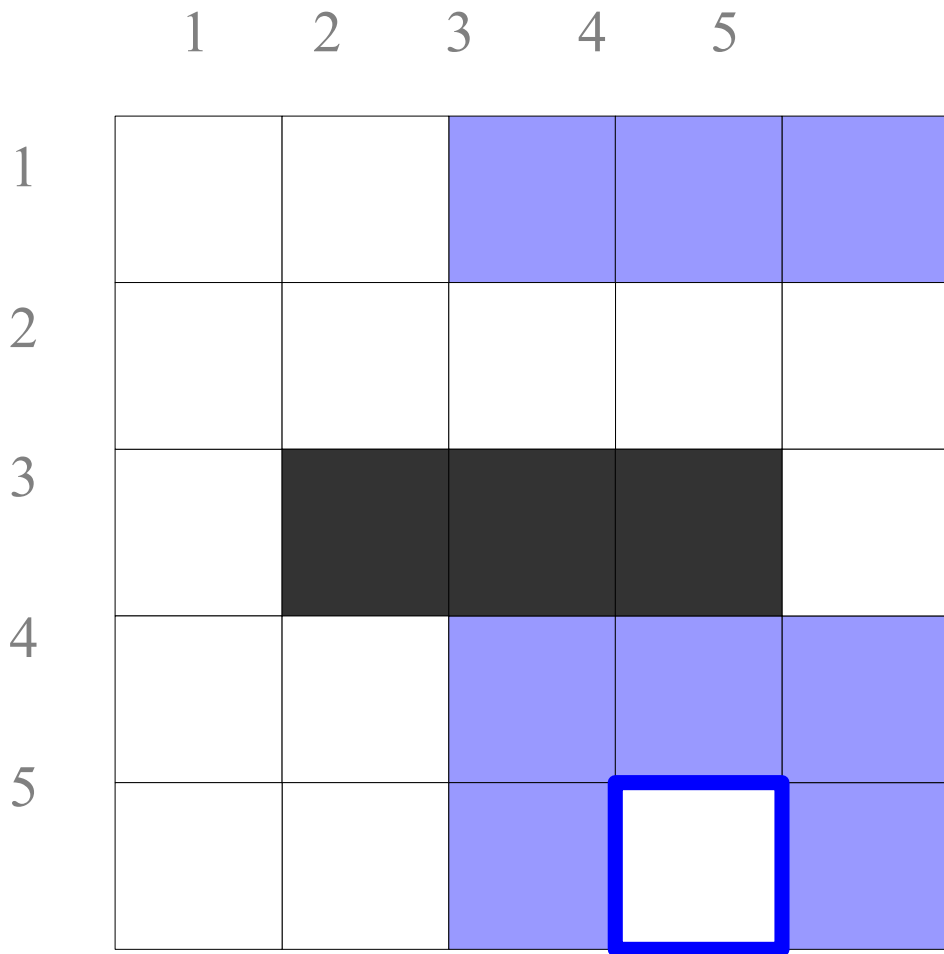
Preto: viva

Branca: morta

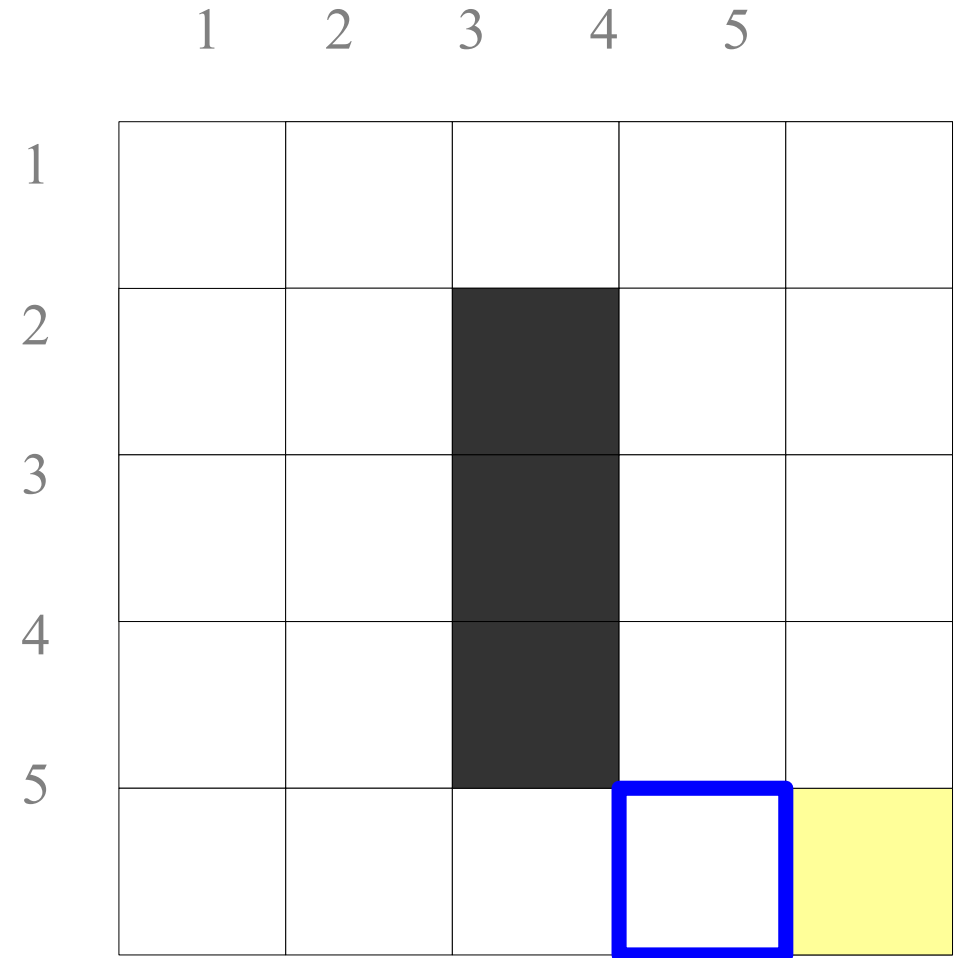
Sobrevivência:  $2 \leq N \leq 3$

Nascimento:  $N=3$

Morte:  $N < 2$  ou  $N > 3$



Estado inicial



Iteração 1

# Simulação

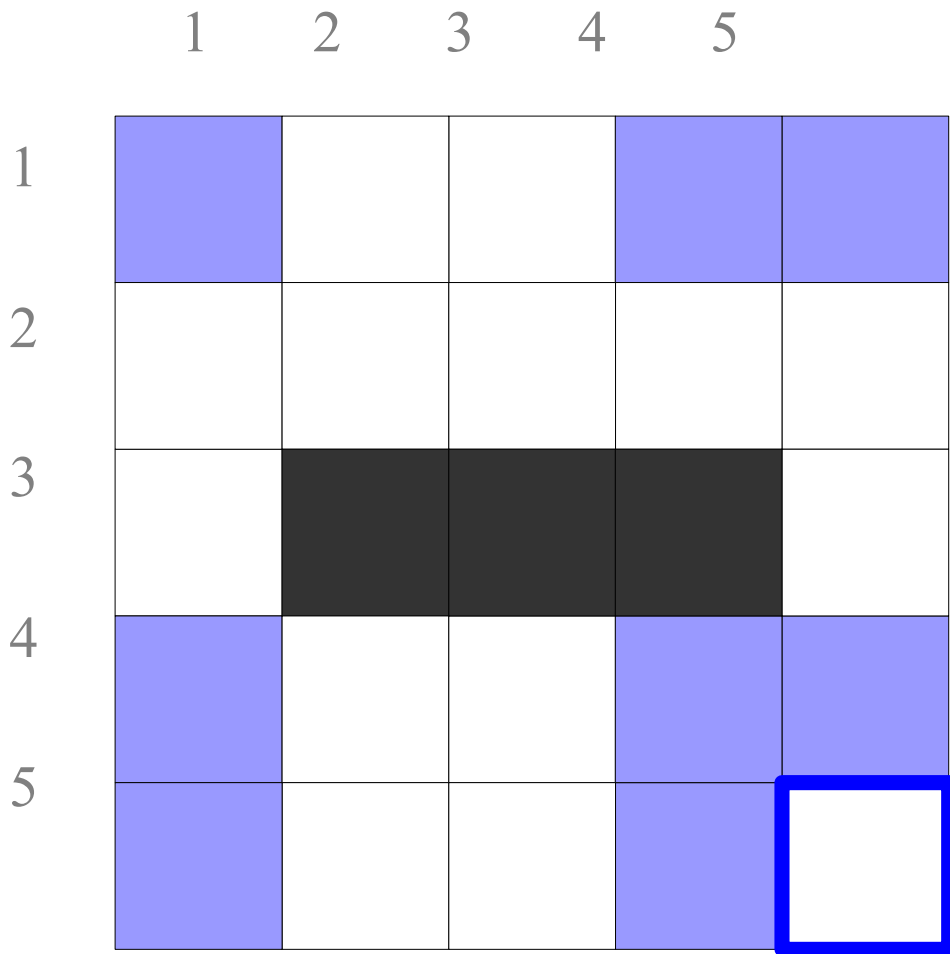
Preto: viva

Branca: morta

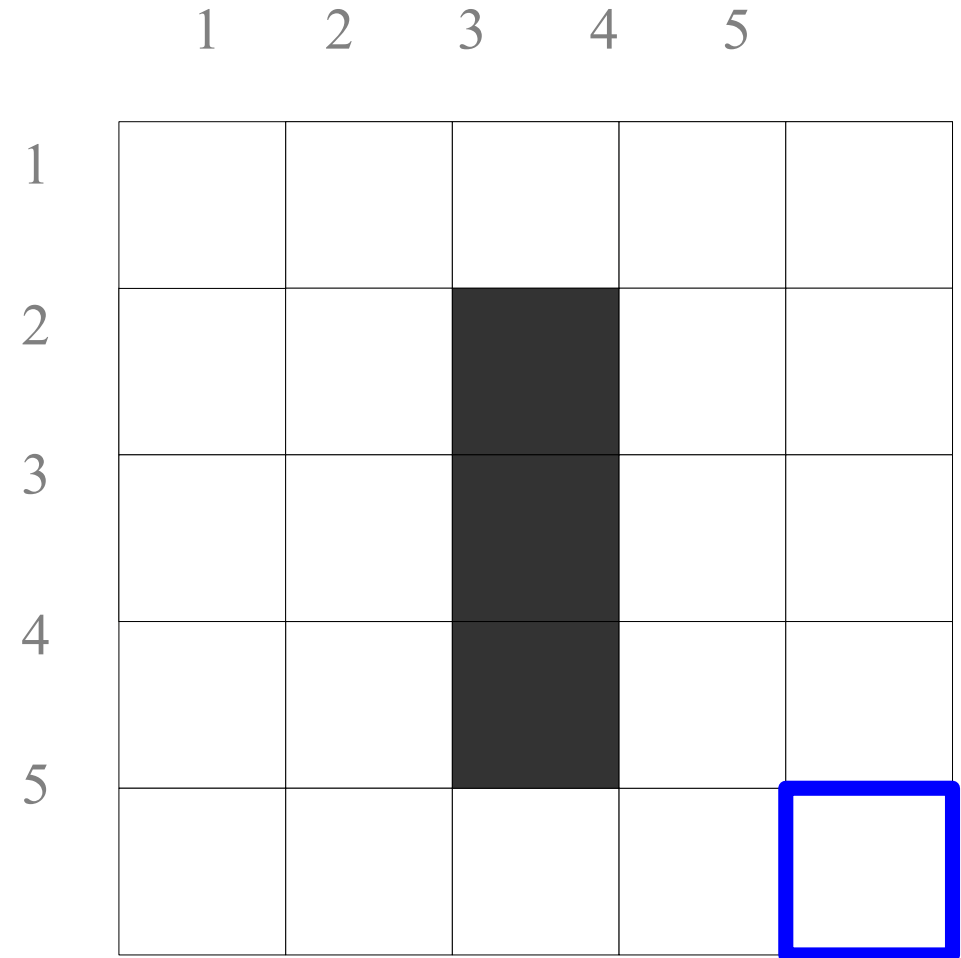
Sobrevivência:  $2 \leq N \leq 3$

Nascimento:  $N=3$

Morte:  $N < 2$  ou  $N > 3$



Estado inicial



Iteração 1

# Simulação

Preto: viva

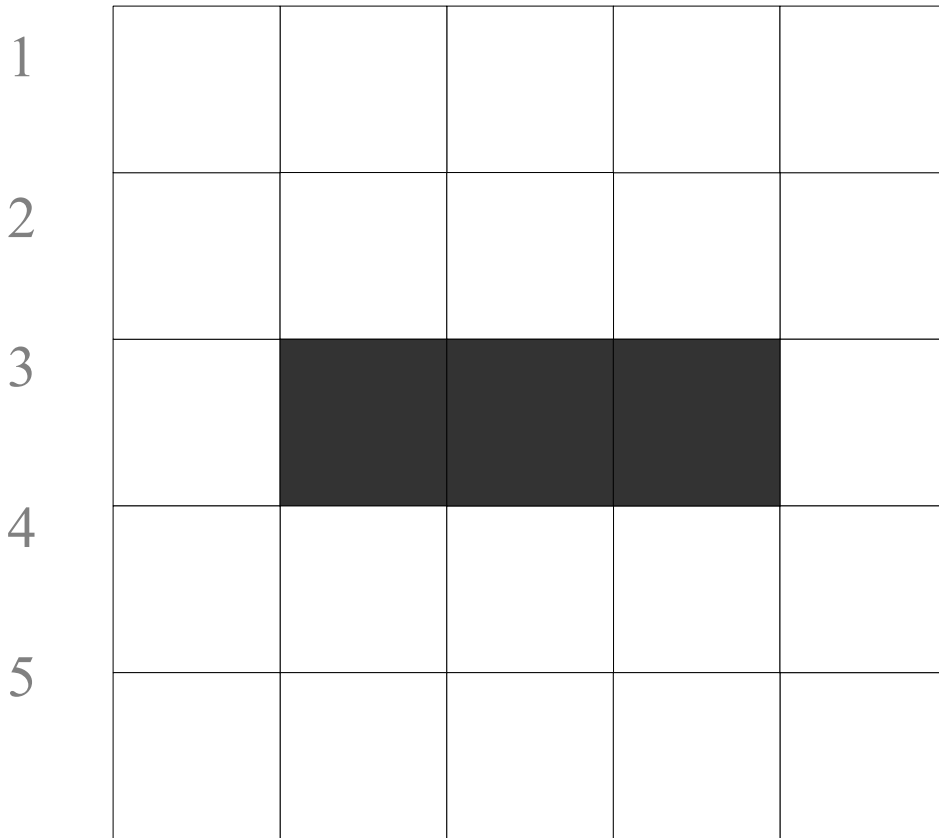
Branca: morta

Sobrevivência:  $2 \leq N \leq 3$

Nascimento:  $N=3$

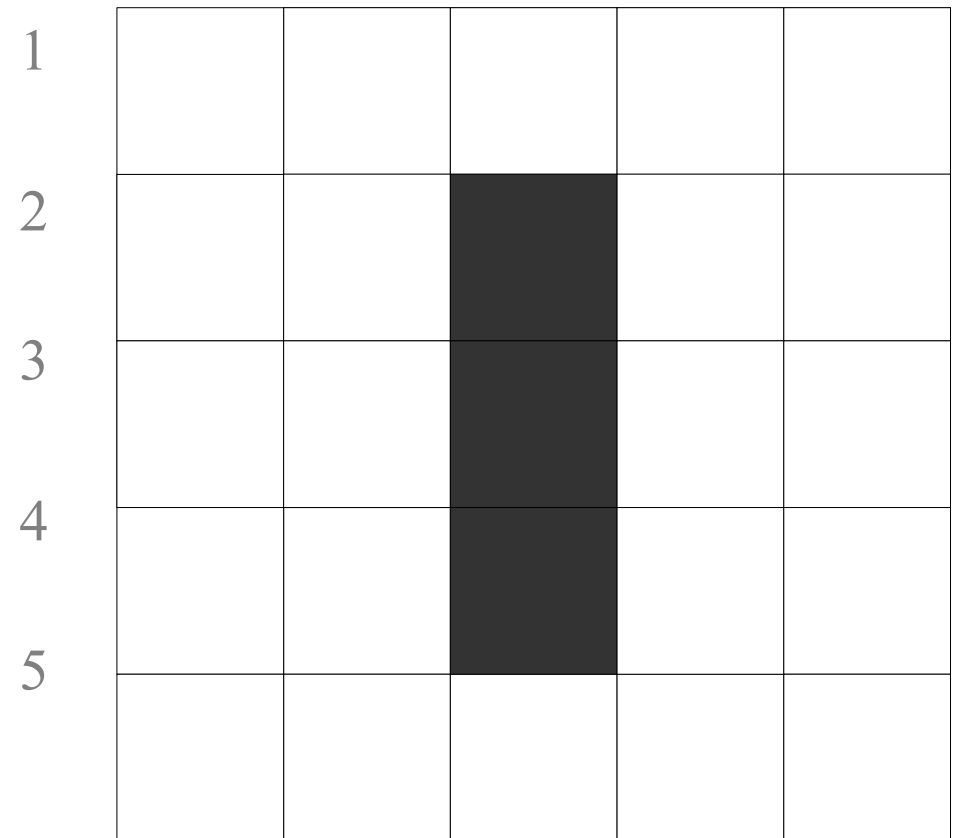
Morte:  $N < 2$  ou  $N > 3$

1 2 3 4 5



Estado inicial

1 2 3 4 5



Iteração 1

FIM

# Exercício 01

Sobrevivência:  $2 \leq N \leq 3$

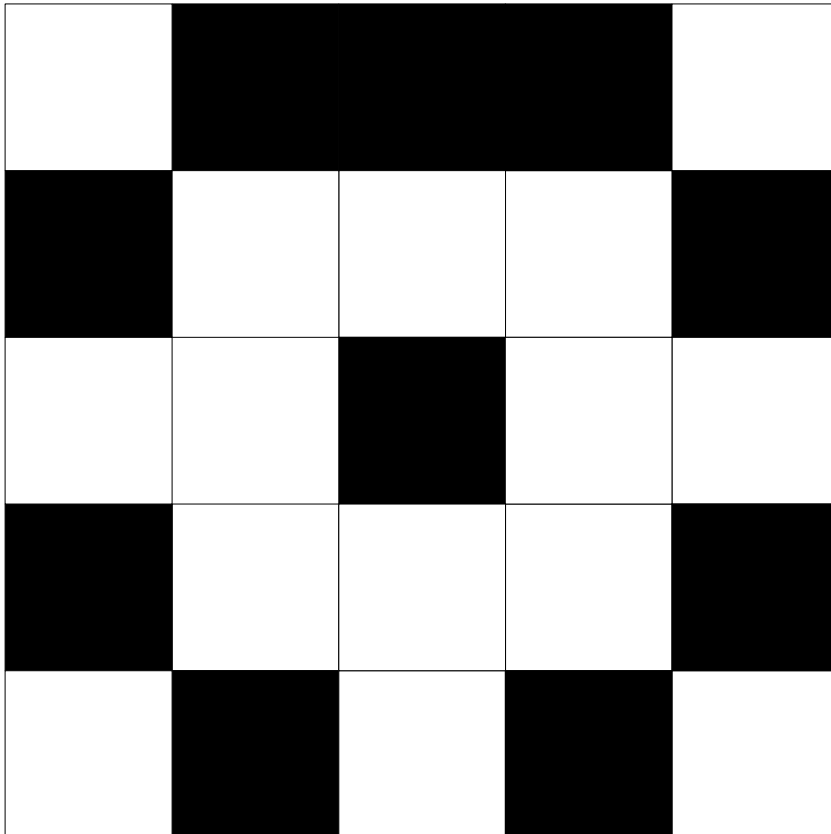
Nascimento:  $N = 3$

Morte:  $N < 2$  ou  $N > 3$

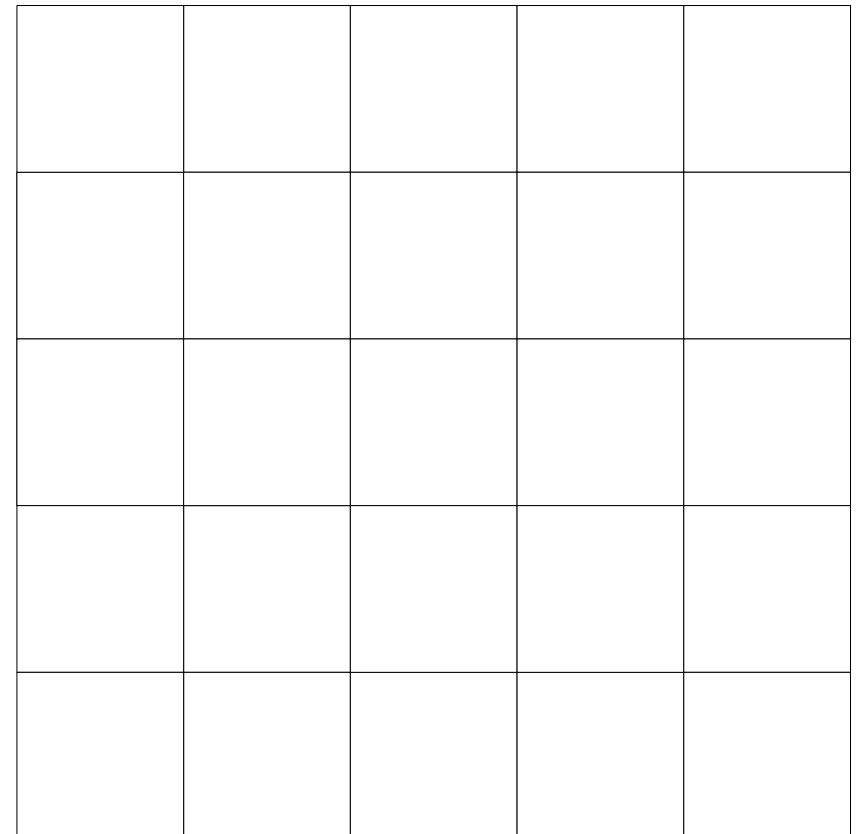
Preto: viva

Branca: morta

Utilize as regras de Conway para determinar **a primeira iteração** da população indicada pela matriz de A



Matriz A: Estado inicial



Iteração 1?

# Exercício 02

Sobrevivência:  $2 \leq N \leq 3$

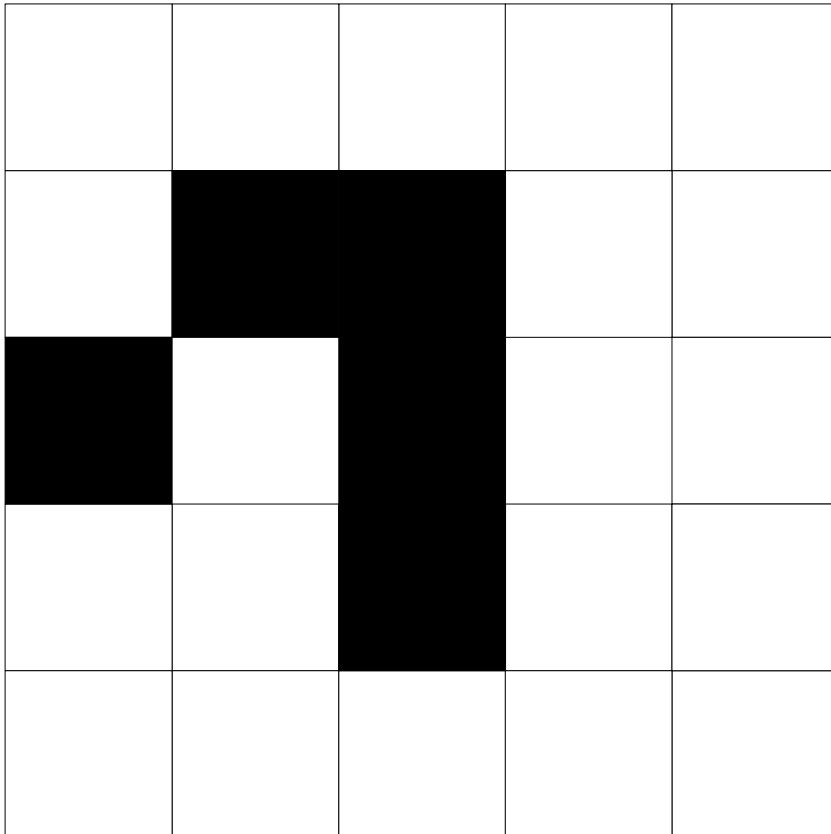
Nascimento:  $N = 3$

Morte:  $N < 2$  ou  $N > 3$

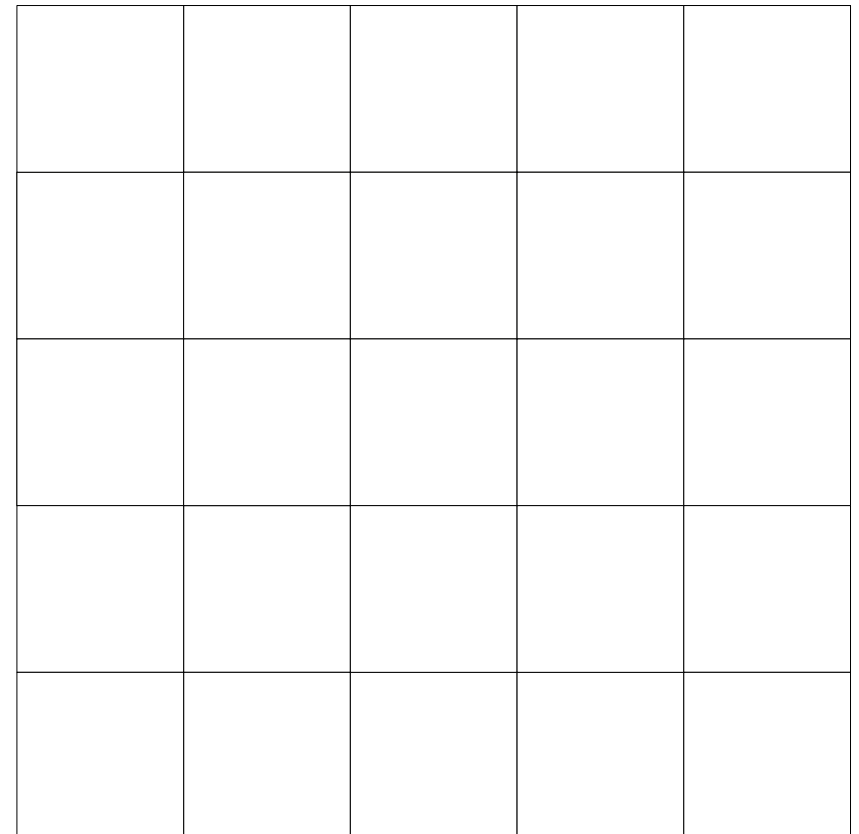
Preto: viva

Branca: morta

Utilize as regras de Conway para determinar as **duas primeiras iterações** da população indicada pela matriz de A



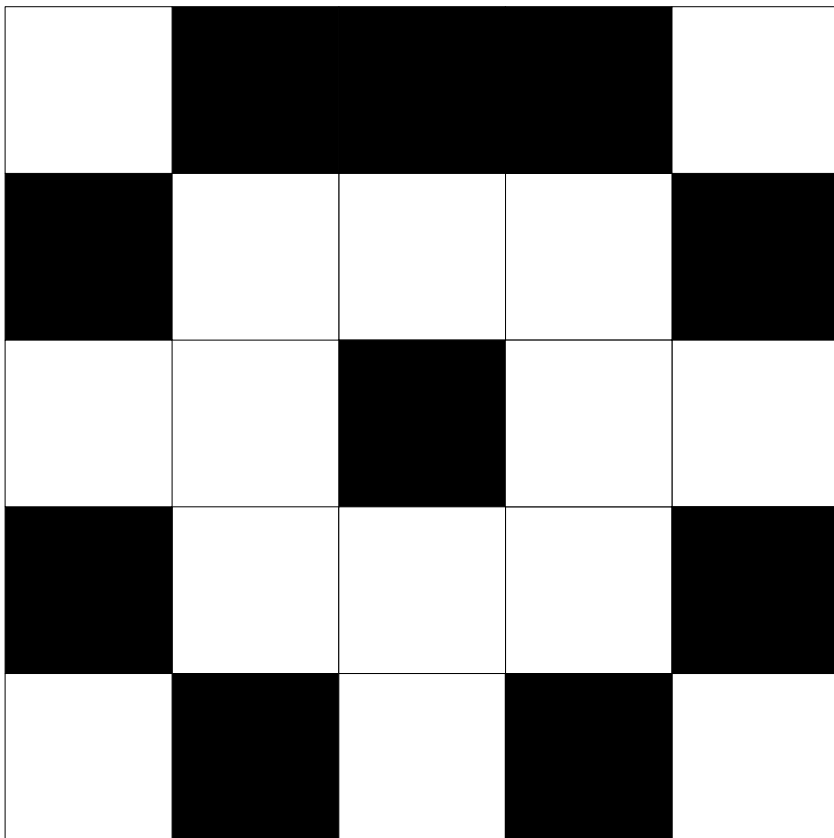
Matriz A: Estado inicial



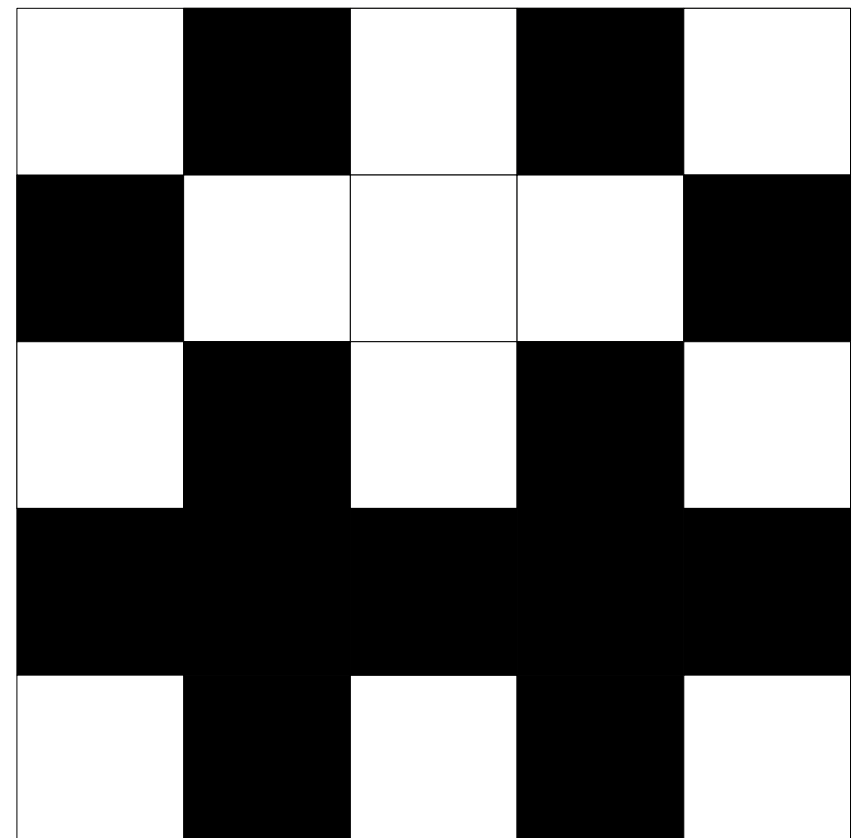
Iteração 2?

# Exercício 01- Solução

Avaliação por pares:  
-1 ponto por erro



Matriz A: Estado inicial

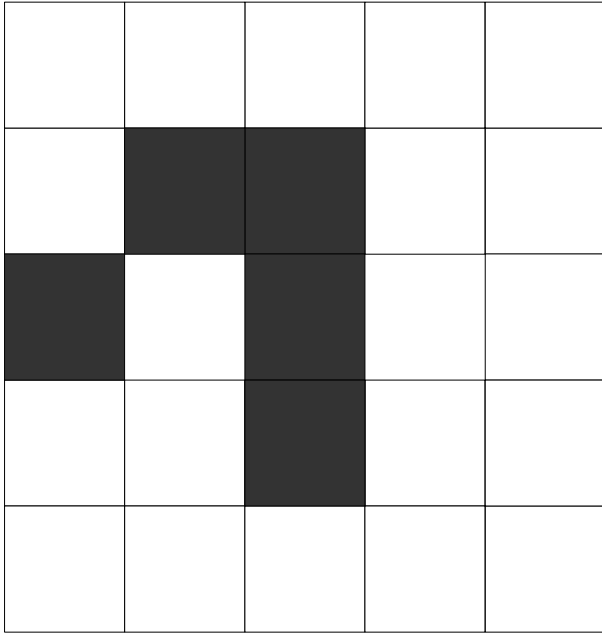


Iteração 1

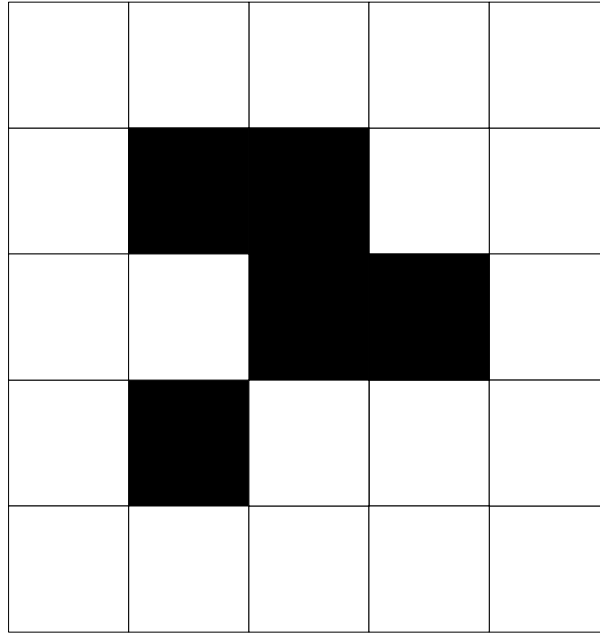


# Exercício 02 - Solução

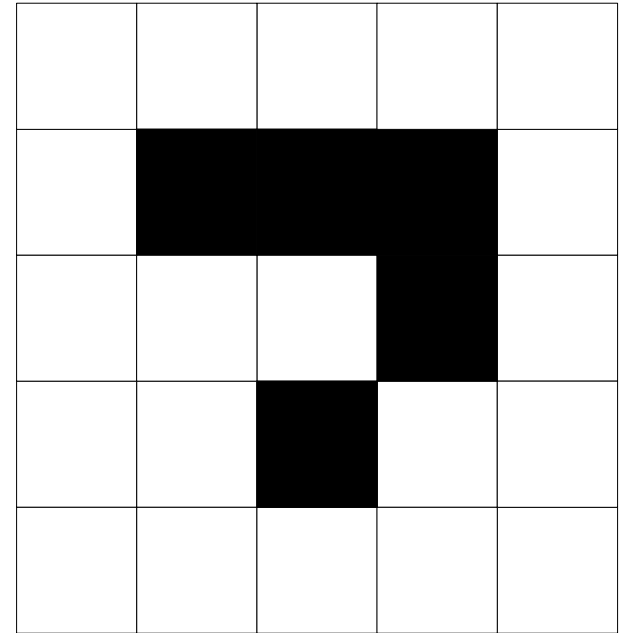
Avaliação por pares:  
-1 ponto por erro



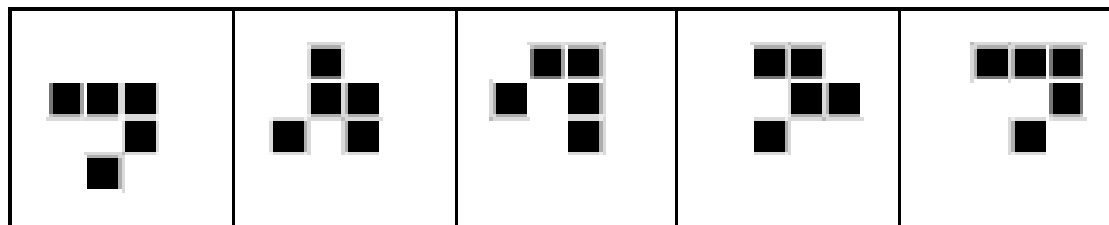
Estado inicial



Iteração 1



Iteração 2



t = 0

t = 1

t = 2

t = 3

t = 4

Padrão spaceship

# Jogo da vida

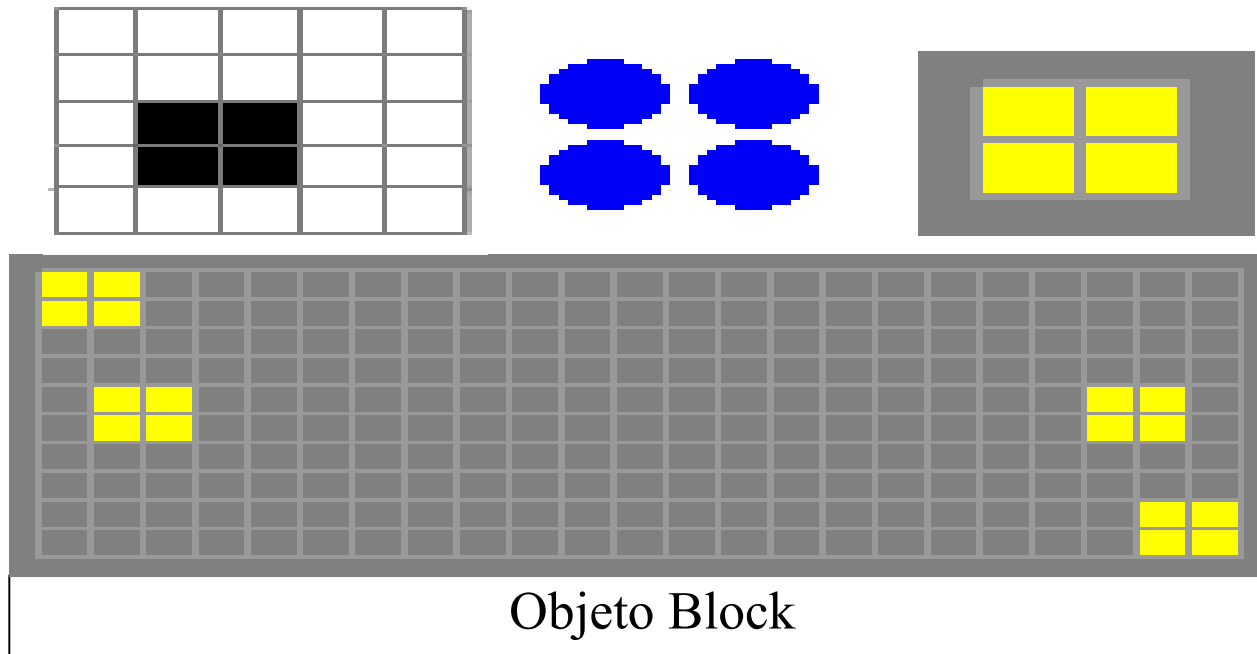
- ◆ Abra no Scilab o arquivo: [life.sce](#) (Tidia)
- ◆ Execute a simulação:
  - Veja o que ocorre quando você altera o valor das variáveis **N** e **Niteracoes** no script
  - Observe os padrões formados: estáticos, oscilatórios e spaceships
  - Analise os padrões já definidos

# Padrões de comportamento

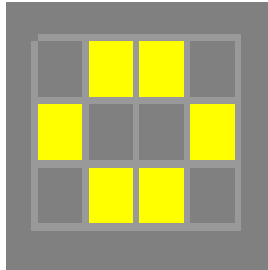
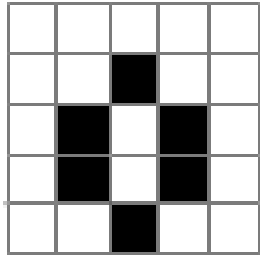
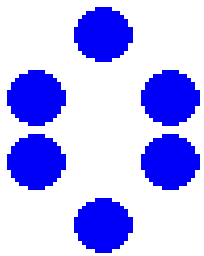
- ◆ Durante a execução do Jogo da Vida as células organizam-se seguindo alguns padrões, formando objetos visuais
- ◆ Existem vários tipos de padrões identificados, dentre eles:
  - Tipo I: estáveis
  - Tipo II: oscilatórios
  - Tipo III: spaceships

# Padrão estável

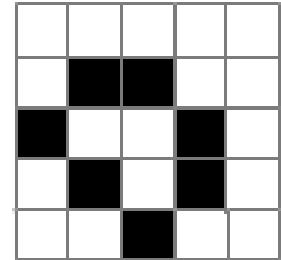
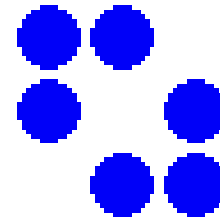
- ◆ Os **objetos** do padrão Tipo I (**estáveis**) são aqueles que **não mudam**, que são estáticos
- ◆ Os objetos estáveis ocorrem quando **nenhuma célula viva tende a morrer**, e **nenhuma célula tende a nascer**
- ◆ Como exemplo tem-se os seguintes objetos: block, beehive, boat, ship, loaf



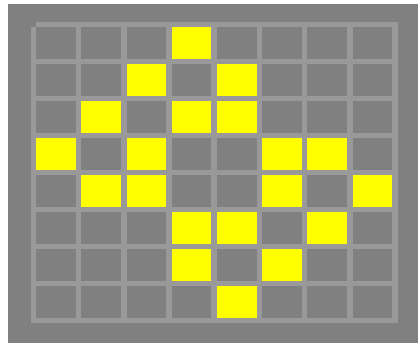
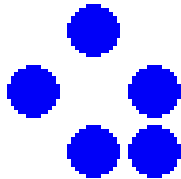
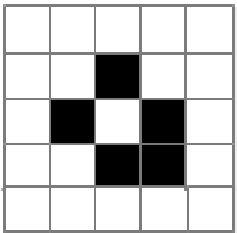
# Padrão estável



Objeto Beehive

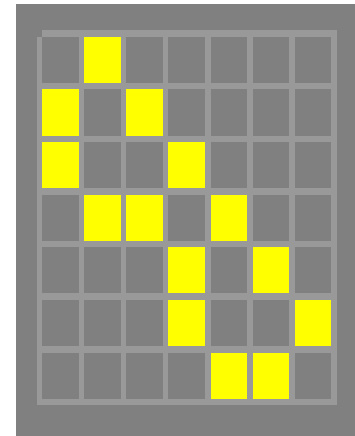
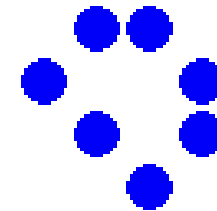
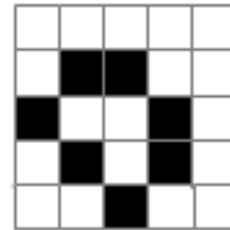


Objeto Ship



Objeto Boat

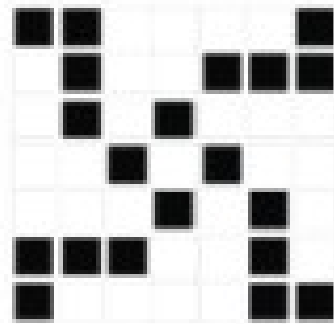
Boats



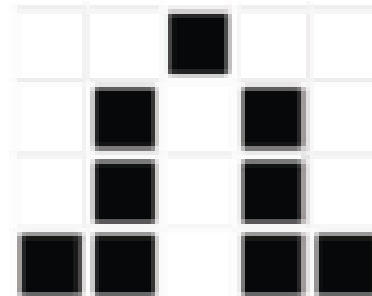
Objeto Loaf

Loafs

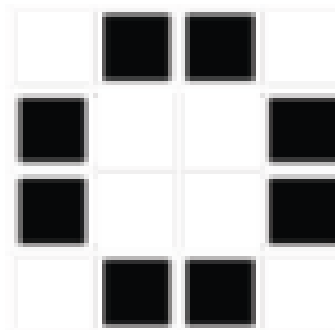
# Padrão estável



Objeto Spiral



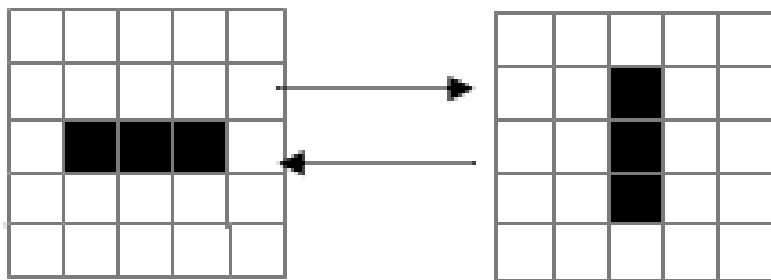
Objeto Hat



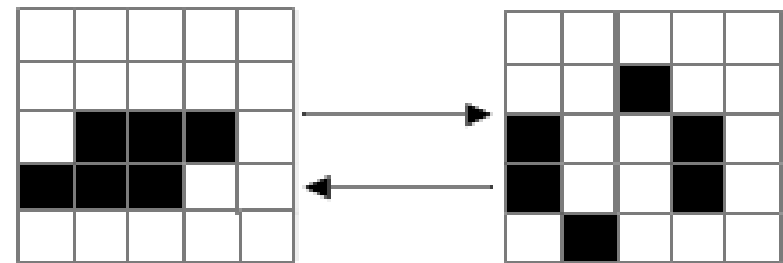
Objeto Pond

# Padrão oscilatório

- ◆ Os objetos **oscilatórios** são formas que mudam da etapa em etapa até atingir um **ciclo constante**
- ◆ O tipo mais simples é o oscilador de dois períodos, ou aqueles que se repetem após duas etapas



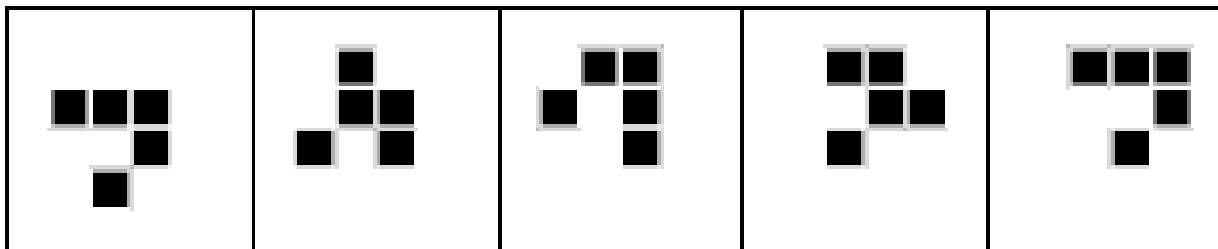
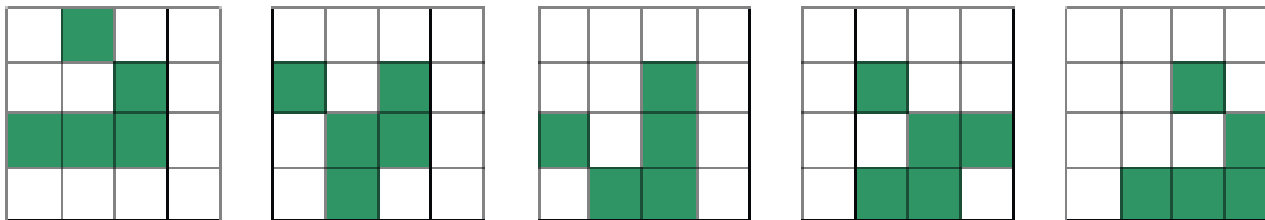
Objeto Blinker



Objeto Toad

# Padrão spaceship

- ◆ Padrões que se **repetem** depois de uma determinada sequência e **retornam a seus estado original**, e se transformam no espaço



t = 0

t = 1

t = 2

t = 3

t = 4

Objeto Glider



# Simulação Cyber-Anasazi

# Simulação Cyber-Anasazi

- ◆ **Anasazi:** Sociedade indígena norte-americanada extinta no final do século XIII.
  - Sociedade bem conhecida mas não se sabe porque foi extinta
- ◆ **Modelo computacional de simulação Cyber-Anasazi:**
  - Reproduz comportamento da cultura Anasazi e declínio da sociedade.
  - Busca testar hipóteses sobre o que pode ter levado a extinção

# Simulação Cyber-Anasazi

## ◆ Simulação:

- Réplica da região onde moravam (Long Valley)
- Agentes: famílias
  - Idade
  - Tempo de vida
  - Visão
  - Habilidade de locomoção
  - Consumo, etc.
  - Interação com outras famílias e com o ambiente



# Simulação Cyber-Anasazi

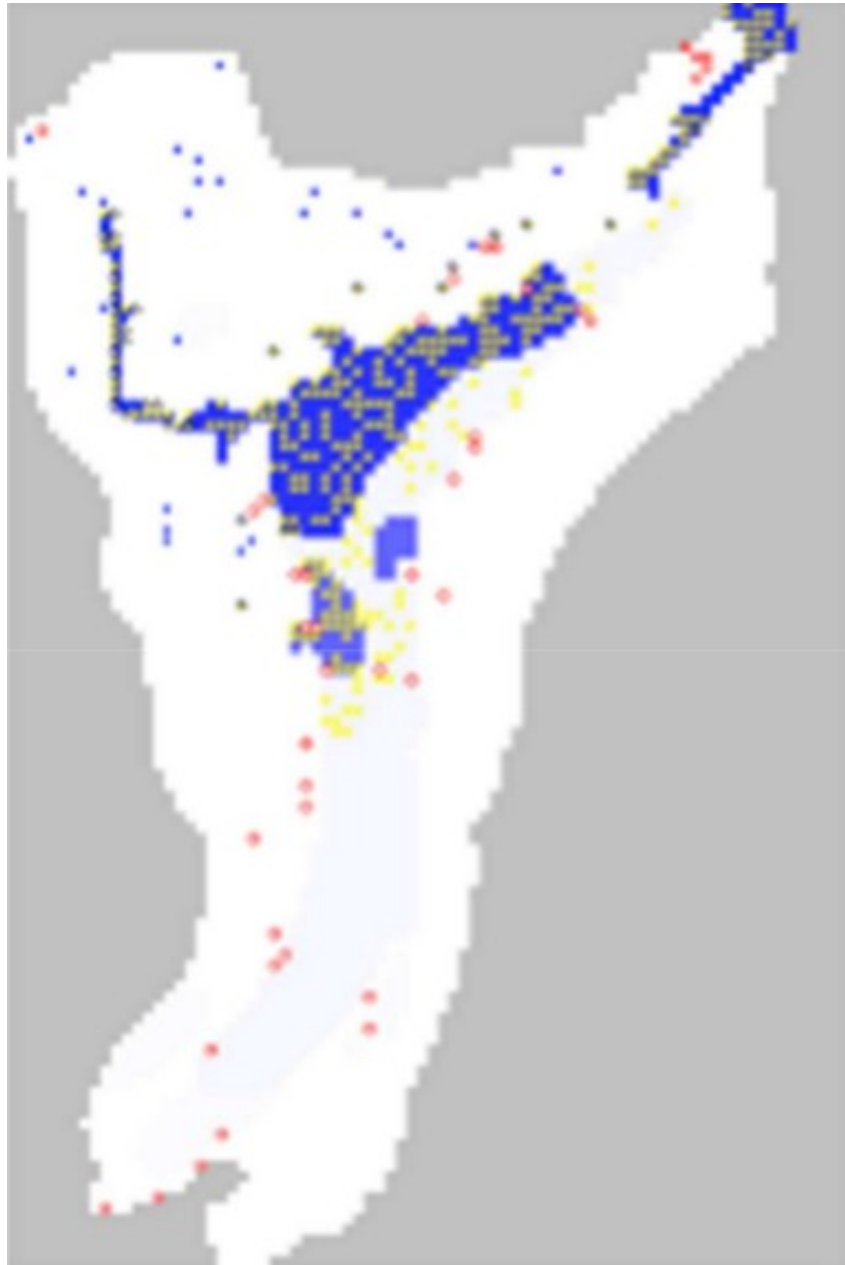
## ◆ Simulação do comportamento de cada agente:

- Colheitas de milho para se alimentar guardando o excedente para o inverno:
  - Foi suficiente: permanece no mesmo local
  - Não foi: muda para outro local promissor, reinicia o ciclo no ano seguinte.

## ◆ Se continuar não sendo suficiente, família desaparece.

## ◆ Regras para novas famílias, nascimentos, etc.

# Simulação Cyber-Anasazi



# Simulação Cyber-Anasazi

- ◆ Simulações ajudaram a melhorar o entendimento da evolução da civilização.
- ◆ Não permitiram concluir o que levou a extinção.
- ◆ **Simulações de sociedades artificiais:** ferramenta importante na reprodução da gênese, evolução e extinção de sociedades reais.
- ◆ São capazes de prever algumas tendências

# Atravessar um terreno

(inspirado no filme Wargames)

# Simulação computacional

Este exemplo mostra como as simulações computacionais são usadas para **identificar / calcular / aproximar** soluções a problemas.

## Problema:

- Programar um robô para atravessar um terreno (desconhecido)



# Um rob tem que atravessar um terreno

## Algumas considerações:

- O robô não tem sensores.
- O robô pode mover-se para frente e ser programado para ir aleatoriamente para a esquerda ou para a direita (em uma percentagem das vezes).
- O robô faz o seu percurso de essencialmente cegas.

O objetivo é determinarmos a frequência com que o robô deve virar à esquerda ou à direita para, assim, otimizar sua viagem através do terreno.

Simple Simulation Game inspired by [Wargames](#)

Game Settings (you can change some of the settings):

Board Length:  Walls (% of Board):  Speed (ms/step):

Times to Play:  Number Games Played: 1

Simulator Values: (you set chance of turning left or right)

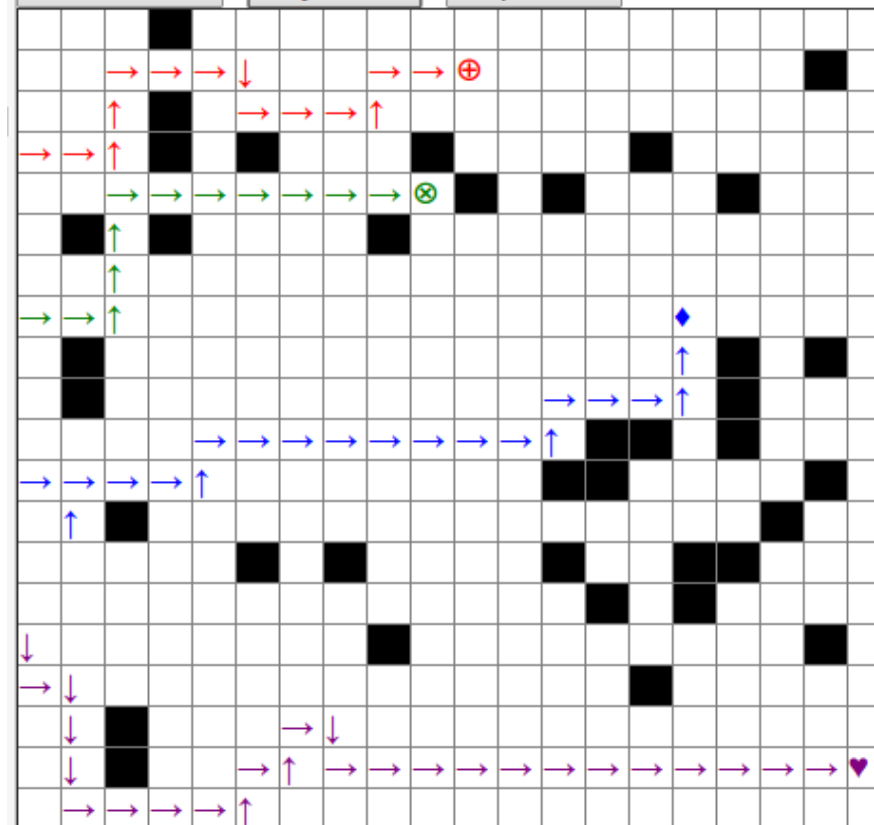
Player Red (⊕): Left %  Right %  Red:

Player Green (⊗): Left %  Right %  Green:

Player Blue (♦): Left %  Right %  Blue:

Player Pink (♥): Left %  Right %  Pink Wins: 1

Number of Simulation Steps: 30



<http://jsfiddle.net/reaglin/yJzT7/>

Simple Simulation Game inspired by [Wargames](#)

Game Settings (you can change some of the settings):

Board Length:  Walls (% of Board):  Speed (ms/step):

Times to Play:  Number Games Played: 103

Simulator Values: (you set chance of turning left or right)

Player Red (⊕):	Left %	<input type="text" value="5"/>	Right %	<input type="text" value="5"/>	Red Wins: 28
Player Green (⊗):	Left %	<input type="text" value="10"/>	Right %	<input type="text" value="10"/>	Green Wins: 43
Player Blue(♦):	Left %	<input type="text" value="15"/>	Right %	<input type="text" value="15"/>	Blue Wins: 21
Player Pink (♥):	Left %	<input type="text" value="20"/>	Right %	<input type="text" value="20"/>	Pink Wins: 16

Simple Simulation Game inspired by [Wargames](#)

Game Settings (you can change some of the settings):

Board Length:  Walls (% of Board):  Speed (ms/step):

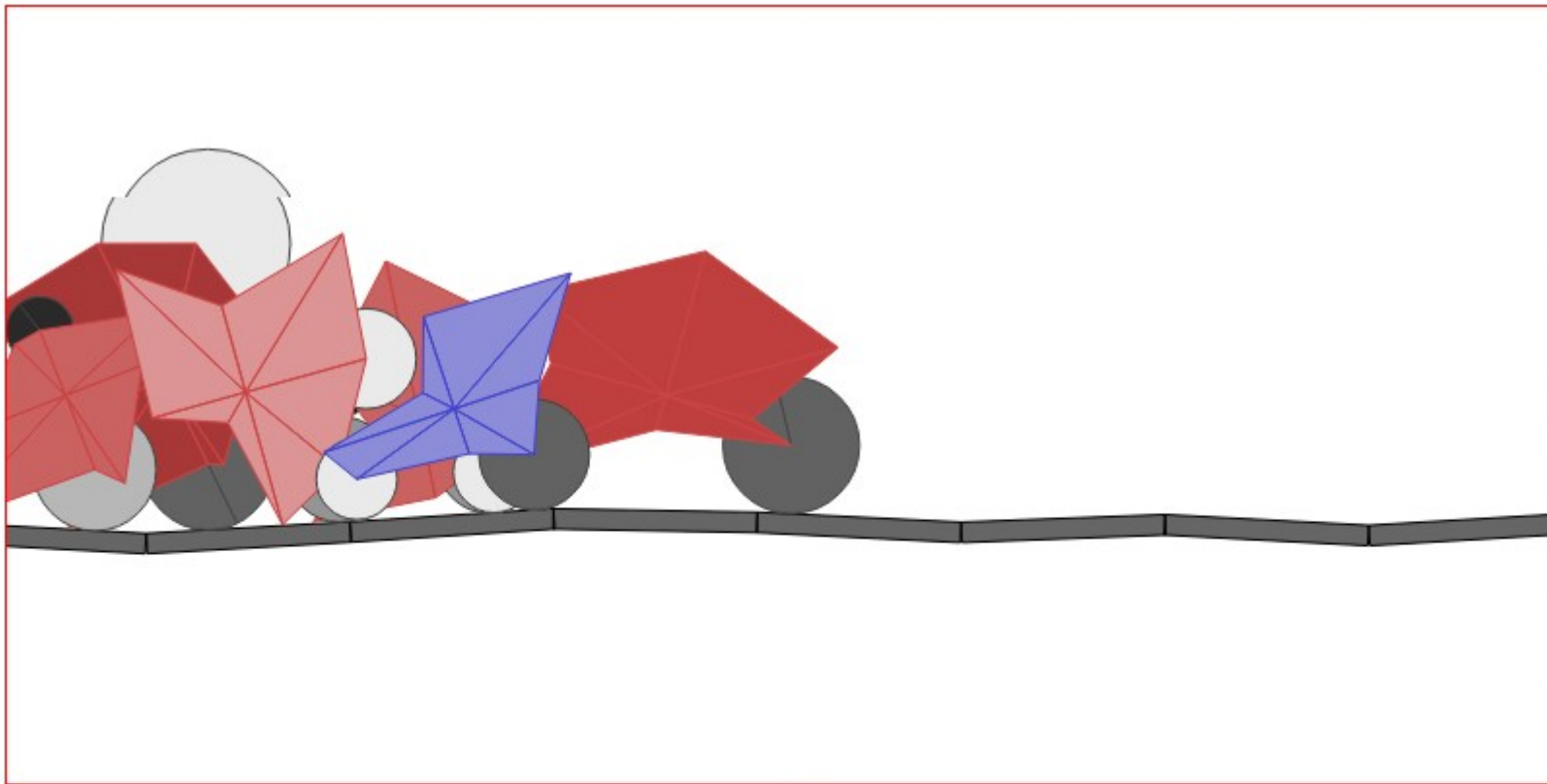
Times to Play:  Number Games Played: 1000

Simulator Values: (you set chance of turning left or right)

Player Red (⊕):	Left %	<input type="text" value="8"/>	Right %	<input type="text" value="8"/>	<input type="text" value="Red Wins: 293"/>
Player Green (⊗):	Left %	<input type="text" value="10"/>	Right %	<input type="text" value="10"/>	Green Wins: 263
Player Blue(♦):	Left %	<input type="text" value="12"/>	Right %	<input type="text" value="12"/>	Blue Wins: 259
Player Pink (♥):	Left %	<input type="text" value="14"/>	Right %	<input type="text" value="14"/>	Pink Wins: 232

# Evoluindo um carro

(algoritmos genéticos)



Create new world with seed:

Mutation rate:

Mutation size:

Floor:

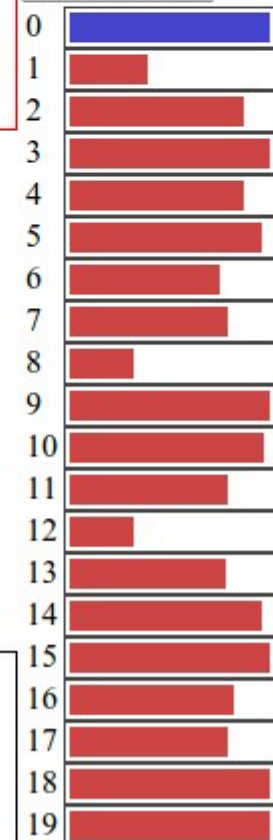
Gravity:

Elite clones:

generation 2  
 cars alive: 20  
 distance: 7.93 meters  
 height: 0.89 meters



Top Scores:  
 #1: 71.72 d:67.24 h:-2.55/3.19m (gen 0)  
 #2: 71.67 d:67.2 h:-2.55/3.19m (gen 1)



[http://rednuht.org/genetic\\_cars\\_2/](http://rednuht.org/genetic_cars_2/)