Projeto Captsone

-

GOL

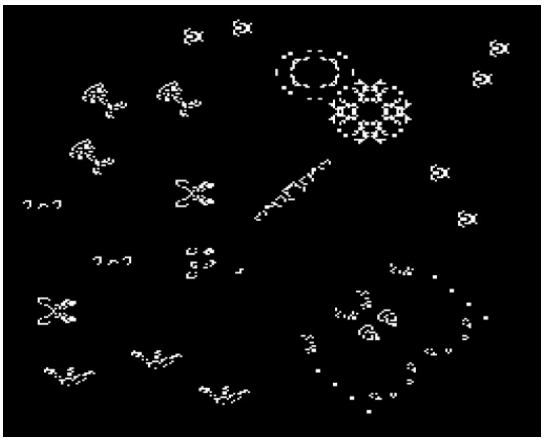
Matheus Victor Henrique da Silva

Mariah Bocoli

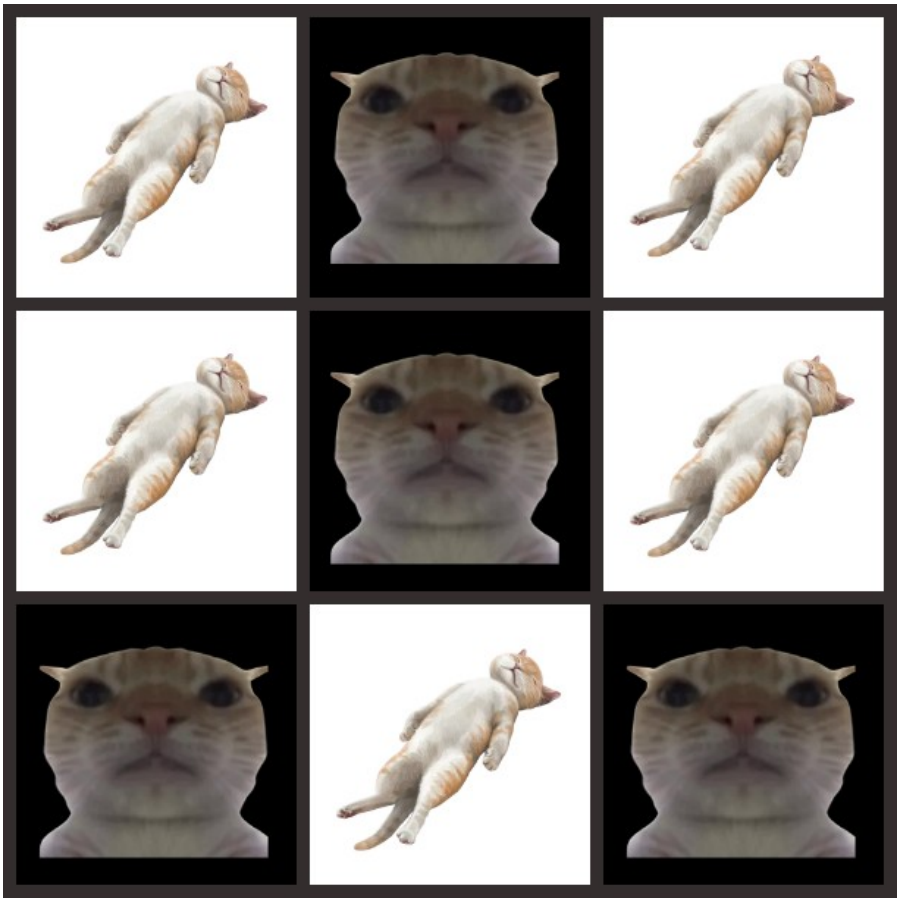
Gustavo Henrique de Jesus da Silva

Como ele é?

Ele é um jogo em que apenas não tem jogador, na verdade precisa apenas que haja uma entrada inicial para que o jogo comece a rodar.



Exemplo de do GRID



Estados das células

VIVO

MORTO

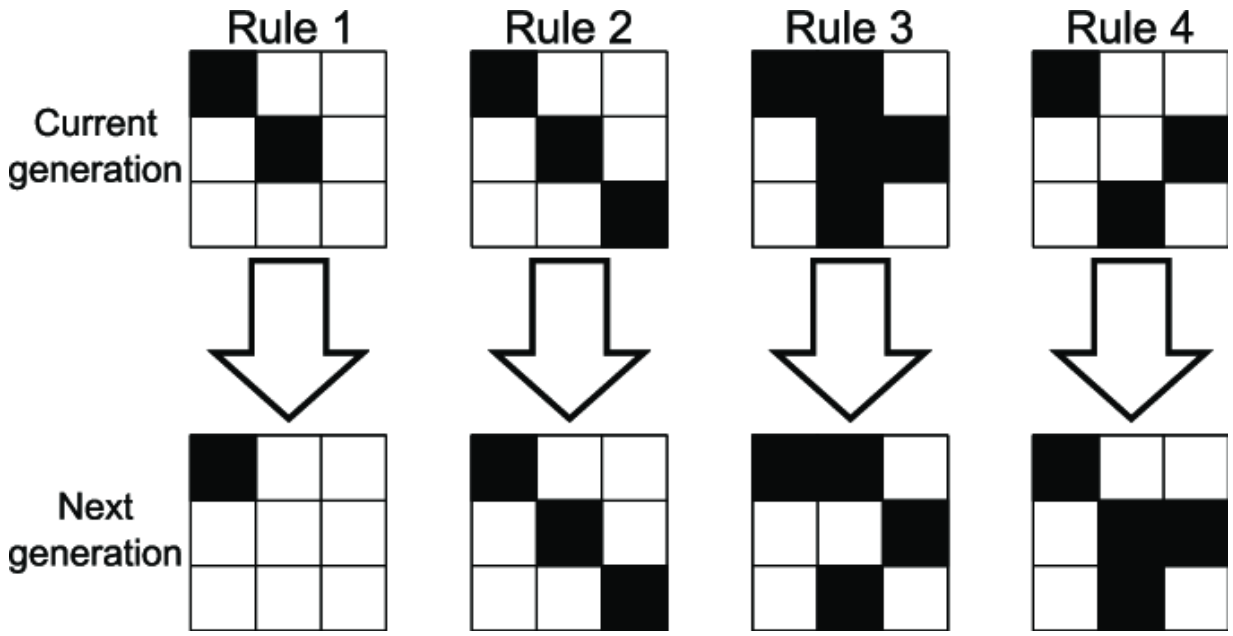
Regras do Jogo

Regra 1: Uma celula viva com menos de dois vizinhos ela more na próxima geração

Regra 2: Uma celula viva com dois ou três vizinhos vivos vive na proxima geração

Regra 3: Uma celula viva com mais de três vizinhos morre na próxima geração

Regra 4: Uma celula morta com três vizinhos vivos ganhara vida na próxima geração



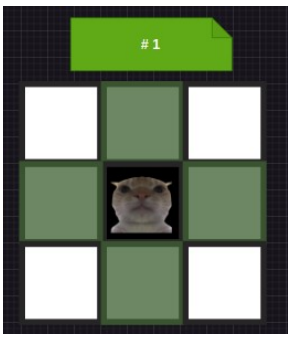
Funcionamento do Programa

Ao abrir o IntellJ, vou clicar com o botão direito no sinal de **“Run”** e clicar em seguida em **“Run with arguments”** (ou qualquer coisa do tipo pareça me dizer que é para rodar e passar argumentos). Com isso ao aparecer a tela de passar eles, **tenho obrigatoriamente que passar os seguintes argumentos**:

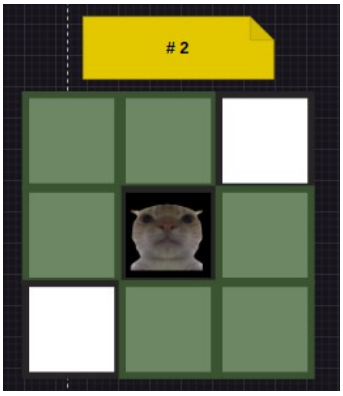
- w=20
- h=20
- g=100
- s=300
- p=”101#010#111”
- n=1

Argumentos

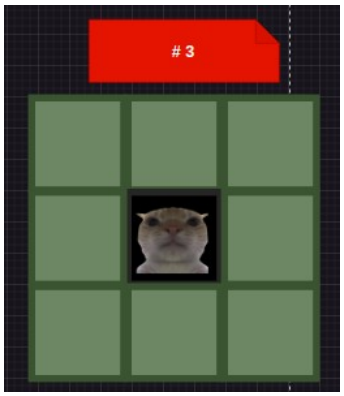
Esses **argumentos,** podem ser divididos em duas **propriedades principais** do programa:

- **Grid:**
  - **w** → É um número inteiro passado para a **largura** do Grid
    - Com os seguintes valores possíveis: **20, 40 ou 80**
  - **h** → É um número inteiro da **altura** do Grid
    - Com os seguintes valores possíveis: **20, 40**
- **Automatos:**
  - **g** → É um número inteiro das **gerações**, definira quantas gerações a serem executadas
    - Com os seguintes possíveis valores:
      - Valores **maiores ou iguais a zero**
        - Caso o valor passado seja **zero (0)** então ficara **executando** até que o usuário **pressione alguma tecla**
  - **s** → É a **velocidade** das gerações, quanto tempo em **milissegundos** passara entre as gerações
    - Com os seguintes possíveis valores: maiores ou iguais que 250 até menores ou iguais a 1000
  - **p** →É um texto que representa como deve ser a **população inicial**, sendo que este argumento será passado a partir de alguns modelos específicos
    - Padrões aceitos: **“001#010#100”**; **“01#11#10”**; **”##110”**
  - **n** → É um valor numérico que corresponde a qual padrão de vizinhança (os quadrados que estão nas **diagonais**, **acima**, **baixo**, **direita**, **esquerda**, **direita**) **somente esses quadrados serão computados nas regras** → valendo para cada célula
    - Isso é informado no Anexo 2
    - Padrões aceitos – *eles são valores numéricos inteiros positivos*:
      - **1, 2, 3, 4, 5** → cada um representa *um padrão especifico*, um desses valores é que sera passado por CLI
        - **Padrão 1:** 

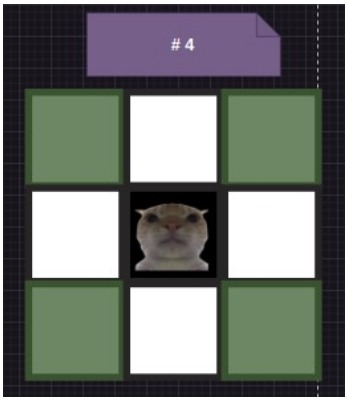
- **Padrão 2:**



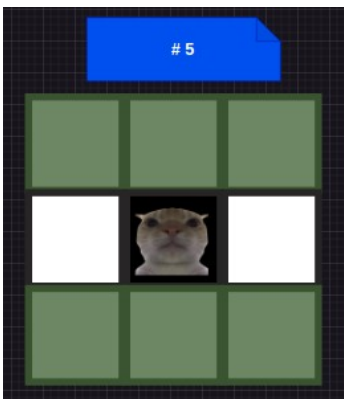
- **Padrão 3:**
  - **É** o **valor padrão** que vai ser a vizinhança completa:



- **Padrão 4:**



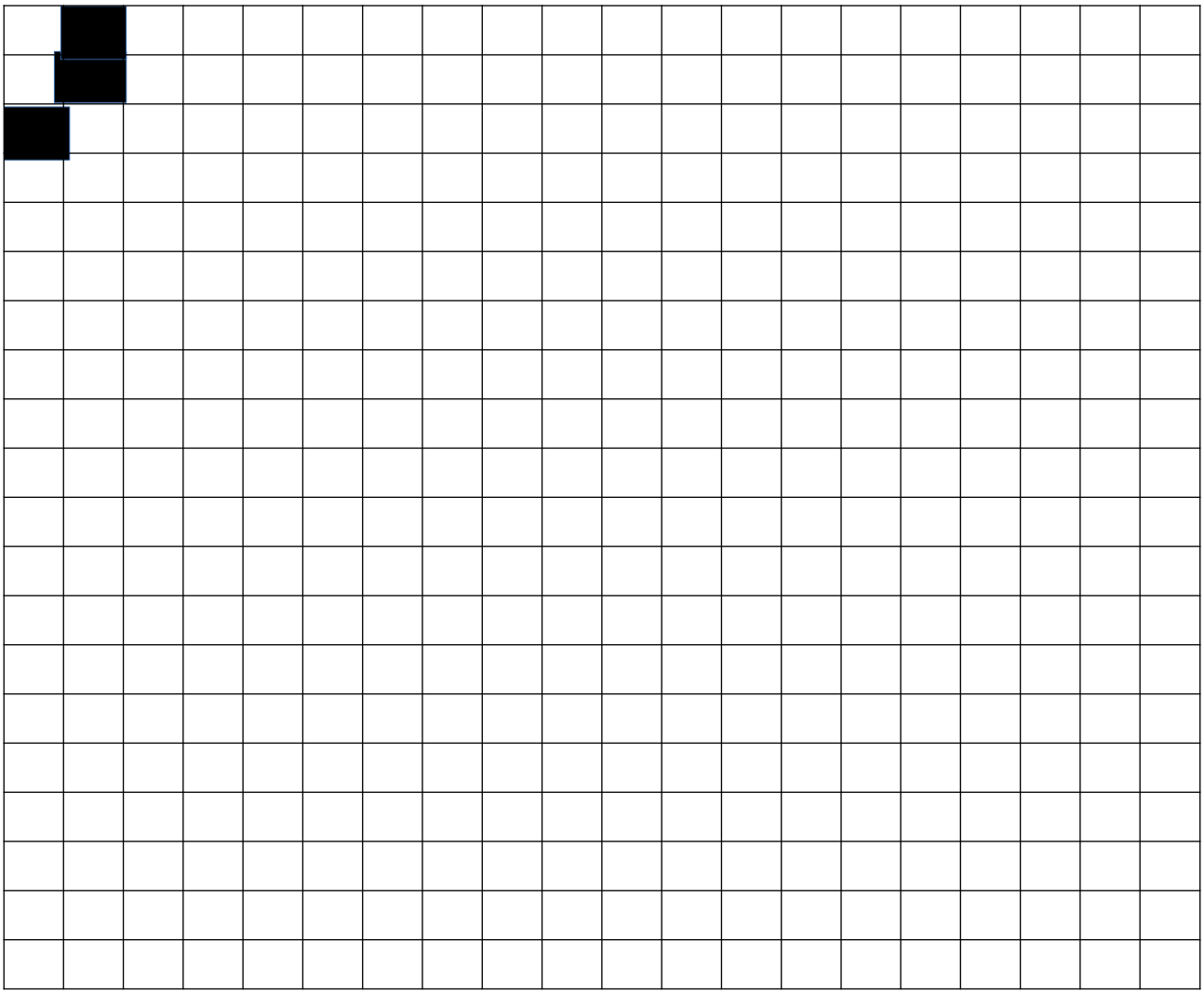
- **Padrão 5:**



Com os argumentos já passados e apertando em “Run”. Em seguida **deve** me **aparecer** uma **interface** como essa:

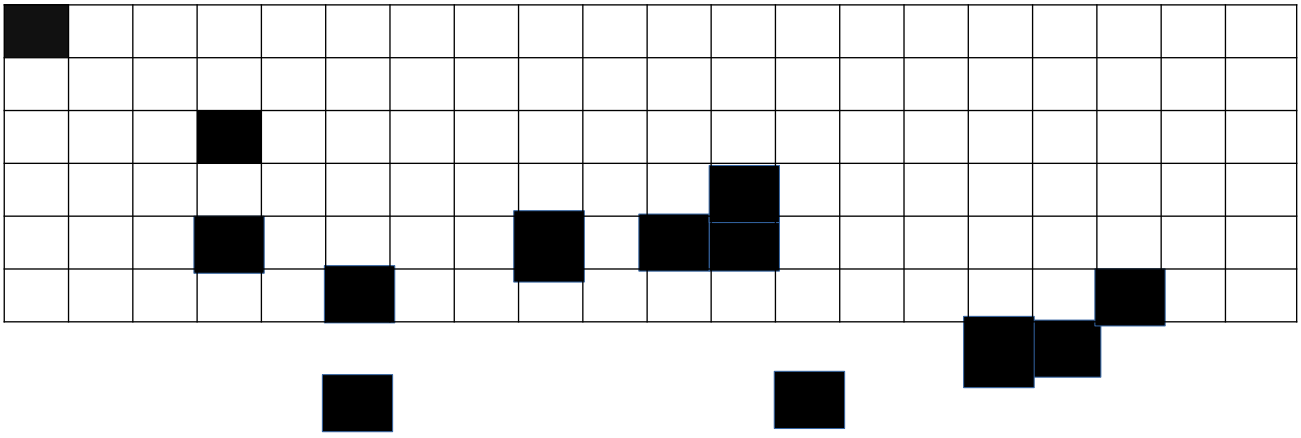
Gride de Exemplo 1

- Relembrando os argumentos passados: w=20 h=20 g=100 s=300 p=”101#010#111”



Gride de Exemplo 2

Chamada: w=20 h=10 g=0 s=500 p=”rnd”



Para continuar vou usar um outro exemplo:

- **Chamada:** w=20 h=10 g=0 s=500 p=”rnd” Assim será a primeira geração:

A		
	B	
C		

Com isso o programa irá rodar em 500 milissegundos por geração (s=500) e seguindo as Regras do Jogo:

- Como a celula **B** dois vizinhos vivos ela continua viva,
- Mas as celulas **A** e **C** não possuem pelo menos dois vizinhos vivos então irão morrer

Assim a segunda geração estará:

	B	

Assim, quando passar os 500ms chegara a terceira geração que vai resultar na morte de **B**, já que nessa geração ela está sozinha e sem pelo menos dois vizinhos mortos. Logo a grade fica apenas com celulas mortas:

Abstração

O que é importante?

Para o projeto será importante:

- Exibir a Grade ou Grid, na primeira vez e renderizar toda vez que houver uma atualização das populações
- Inserir a popuçação inicial
- Atualizar as gerações, respeitando as regras e o tempo fornecido previamente

Decomposição

Quais são as partes?

Entrada

- Vai precisar receber os parametros: `w h g s p` na chamada do arquivo, veja mais em Anexo 1.

Processamento

- Pegar essa entrada de largura e altura: `w h` → para fazer a construção do Grid
- Colocar no Grid a população inicial → `p`
- Iniciar as gerações com o tempo informado → `s`
- Aplicar as regras no Grid atual
- Ir para a próxima geração:
  - Se o numero de gerações for menor ou igual ao informado → `g`
  - Caso contrario, ele para o programa

Anexo 1 - Passando argumentos para um programa Java através da linha de comando

Qualquer número de argumentos pode ser passado para um programa java através da linha de comando. Ao executar o programa todas as **coisas escritas após o nome da classe** na linha de comando **são argumentos**. Os **argumentos** são **limitados pelo espaço**.

Este exemplo de código apresenta como resultado na tela todos os parâmetros passados para o program:

```
public class CmdLnArgmntExp {

   public static void main(String[] args) {
      System.out.println("d");
      for (int i = 0; i < args.length; i++)
         System.out.println(args[i]);
   }

}
```

Execute o programa da seguinte maneira:

```
$ Java CmdLnArgmntExp arg1 arg2 arg3 arg4
```

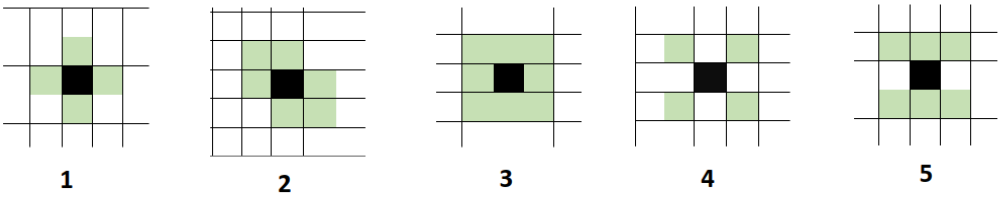
Saída gerada pelo programa:

```
arg1
arg2
arg3
arg4
```

Anexo 2 - Onde está esses padrões do argumento N?

Está na página 19 a 22: do **Material de Leitura Obrigatória → Projeto Final Capstone – PT-BR** (PDF).

- n =>(int) um dos valores entre 1e 5. Cada um representa uma das vizinhanças listadas:



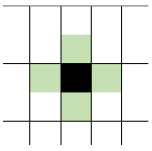
- Se não se define o parâmetro, assumir como default a vizinhança (3).

Página - 19

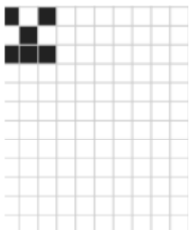
- Exemplo 1 de como executar GOL

  - `java GameOfLife w=10 h=20 g=100 s=300 p= "101#010#111" n=1`

  - Executará GOL com um tamanho de matriz 10x20, durante 100 gerações, a uma velocidade de 300 ms. por geração, a vizinhança correspondente ao valor 1 e com a seguinte população:



Incluir nas regras só as caixas verdes



Página - 20