

UNIVERSIDADE FEDERAL DO MARANHÃO  
ENGENHARIA DA COMPUTAÇÃO  
CURSO DE INTELIGÊNCIA ARTIFICIAL  
PROF. DR. THALES LEVI AZEVEDO VALENTE

A large, stylized letter 'A' is formed by a trail of small, grey ants. The ants are arranged in a way that they appear to be moving from the top left towards the bottom right, following the shape of the letter. The trail starts with a few ants at the top left and gradually increases in density as it moves down and to the right, forming the vertical and diagonal strokes of the 'A'.

# **MODELAGEM DE FORMIGUEIRO VERSÃO HxH EM NETLOGO**

GABRIEL FELIPE CARVALHO SILVA  
GIORDANO BRUNO DE ARAUJO MOCHEL  
JUDSON RODRIGUES CIRIBELLI

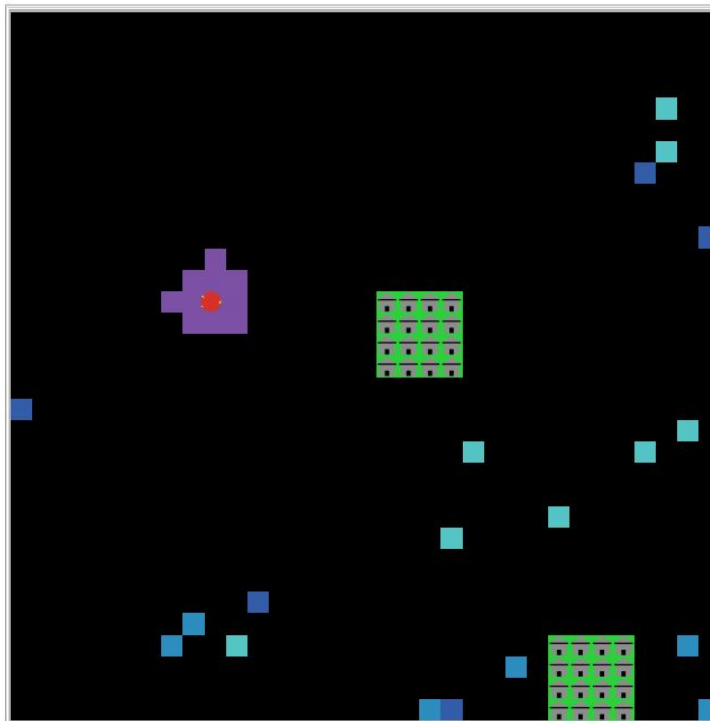
# SUMÁRIO

INTRODUÇÃO

DESENVOLVIMENTO

DISCUSSÃO

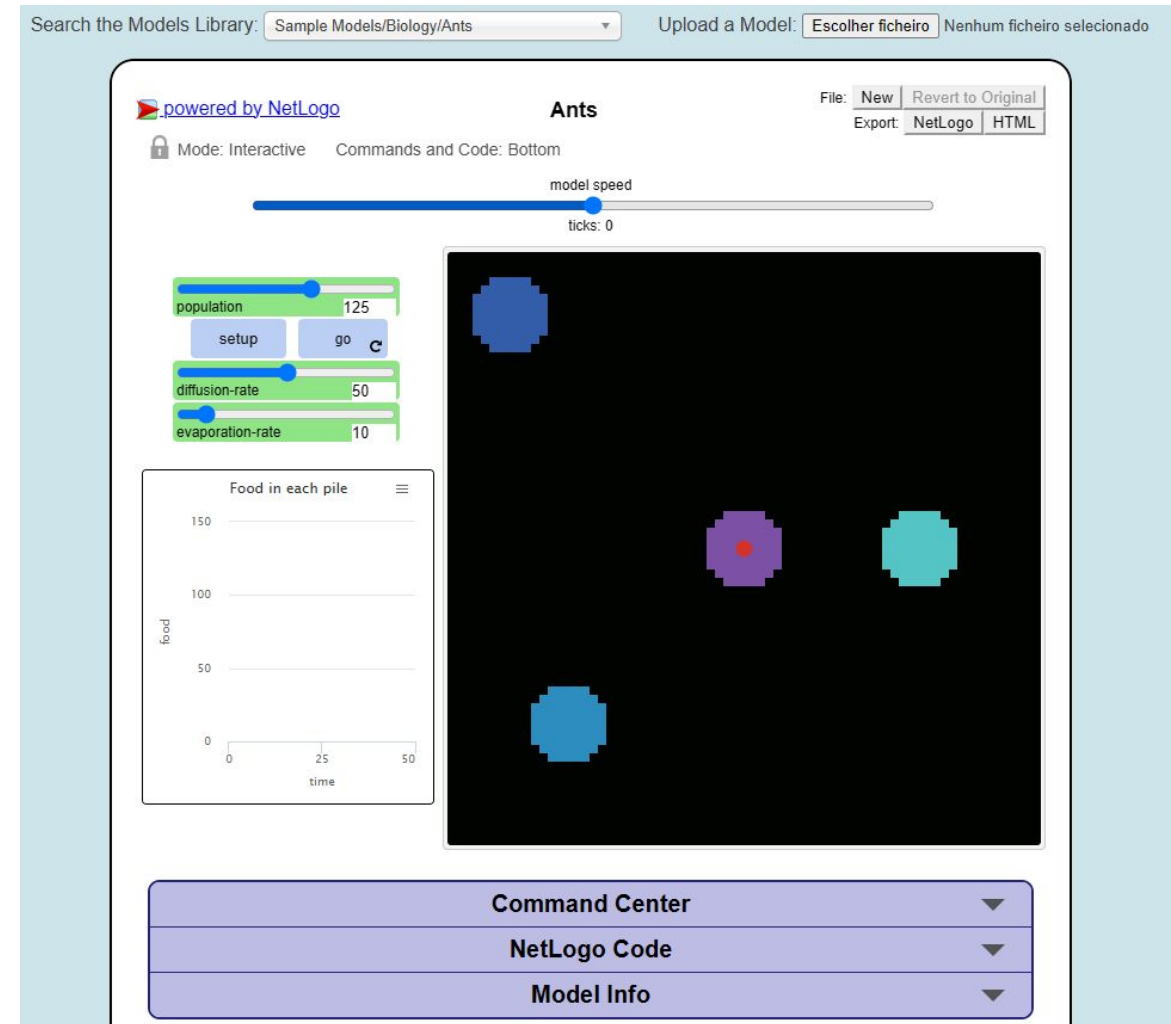
CONCLUSÃO



Fonte:  
[https://hunter-x-hunter.fandom.com/pt/wiki/Formigas\\_Quimera](https://hunter-x-hunter.fandom.com/pt/wiki/Formigas_Quimera)

# INTRODUÇÃO

- Modelo base: Sample Models/Biology/Ants do site do NetLogo;
- Agentes: formigas
- Ações: buscar alimentos, retornar ao formigueiro, deixar rastro de feromônio;



Referência:

<https://www.netlogoweb.org/launch#https://www.netlogoweb.org/assets/modelslib/Sample%20Models/Biology/Ants.nlogo>



# INTRODUÇÃO

- Baseado no arco das formigas Quimeras do anime Hunter x Hunter:
  - **Ascensão das Formigas Quimeras:** Evolução e adaptação das formigas, que ganham habilidades humanas e superam a supremacia humana.
  - **Conflito com os Caçadores:** Caçadores enfrentam as formigas quimeras em batalhas intensas para proteger a humanidade.
  - **Reflexões sobre Humanidade e Moral:** O arco explora dilemas éticos, questionando o que define a humanidade e o impacto da evolução.

# INTRODUÇÃO

## Ambiente de tarefas:

### • Medidas de desempenho:

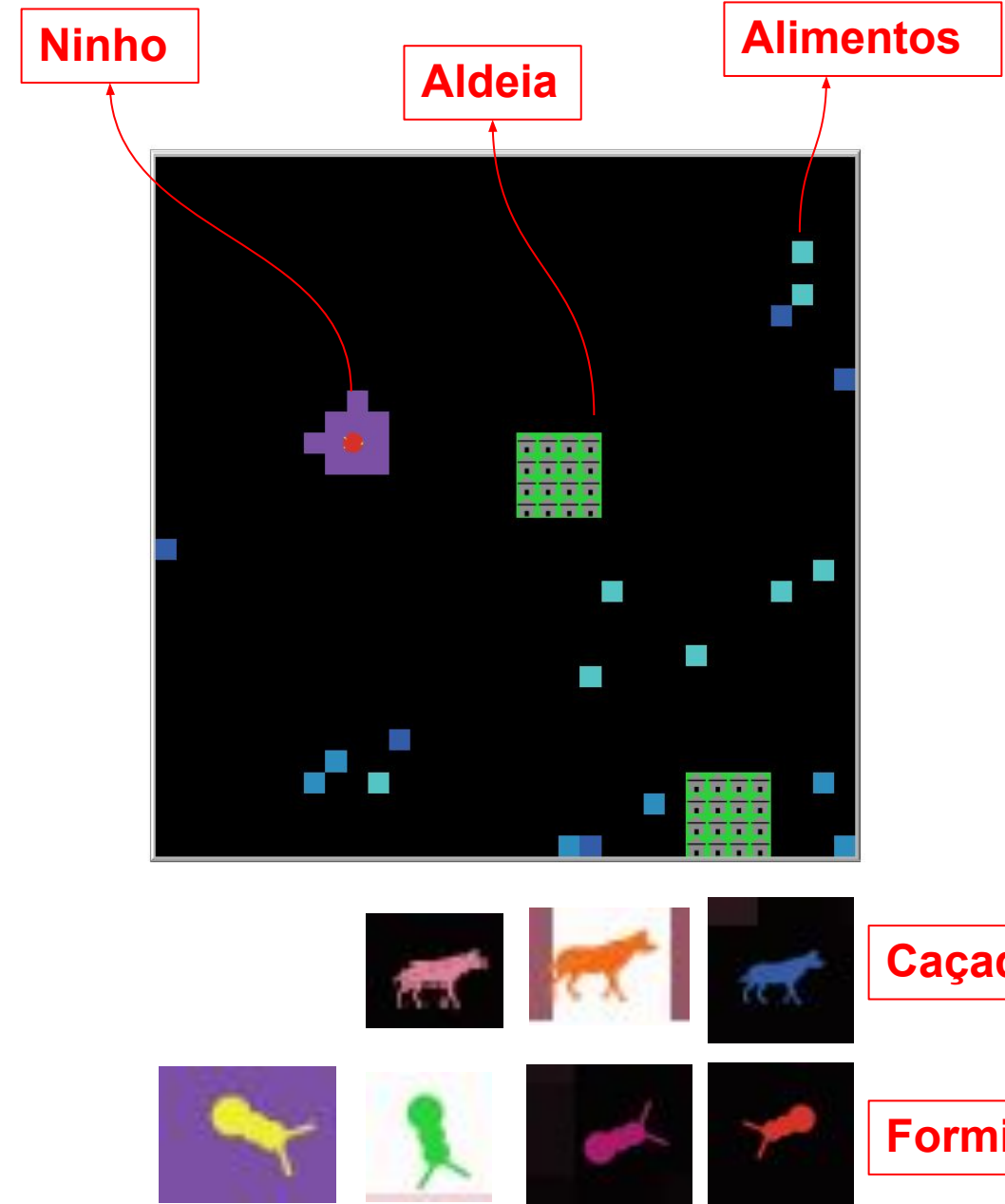
- **Formiga:** Sobrevivência por mais tempo ou até que a comida no mapa se acabe.
- **Caçador:** Proteger as aldeias, eliminando o máximo de formigas; eliminar o rei e/ou rainha das formigas.

### • Ambiente:

- Formigueiro, aldeias e pontos de alimento

### • Atuadores:

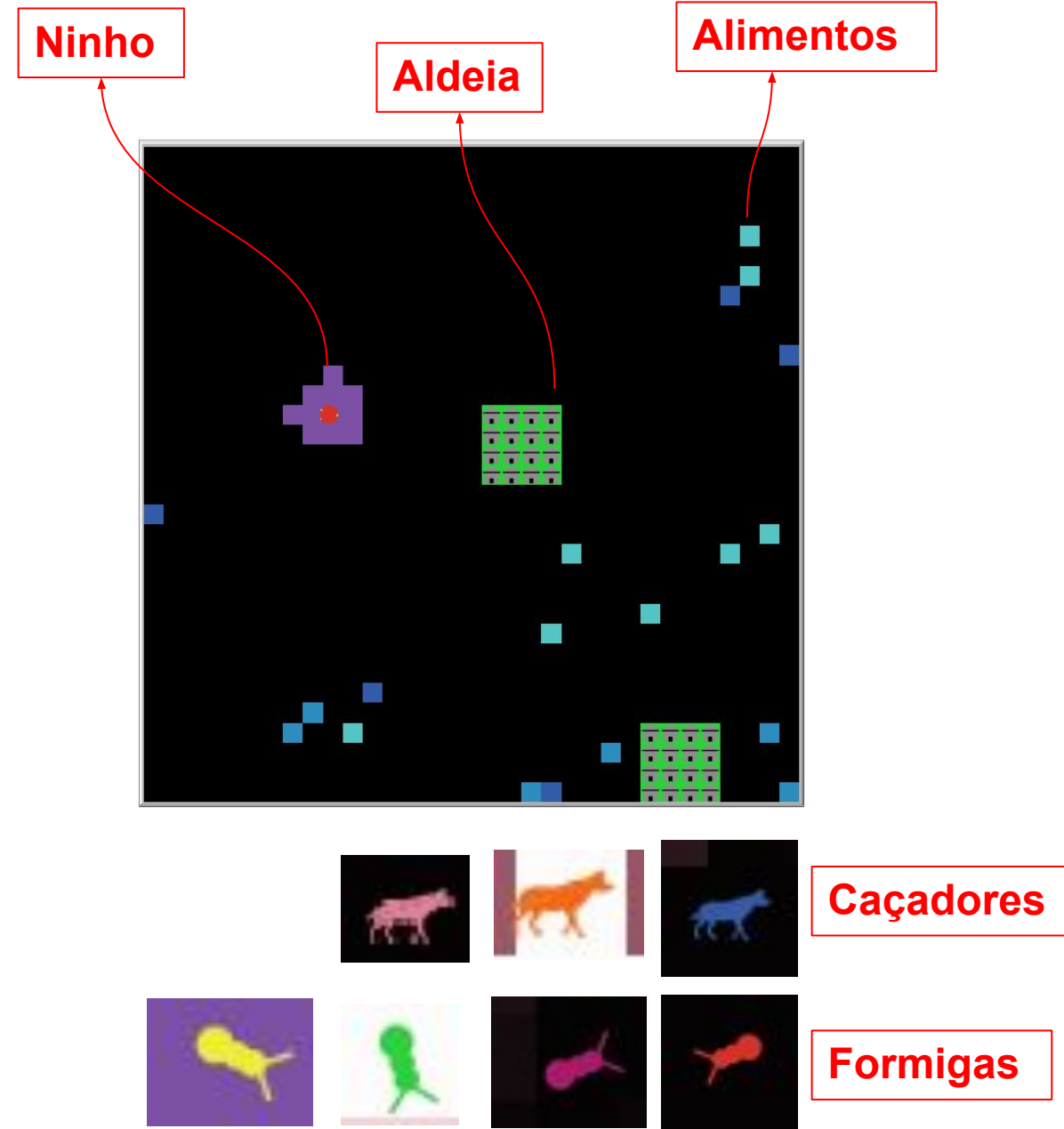
- Formigas: Mover-se no mapa, seguir rastros de feromônio, atacar alvos.
- Caçadores: Mover-se no mapa com foco nas aldeias, atacar alvos;



# INTRODUÇÃO

## • Sensores:

- **Formigas:** Percepção de feromônio, detecção de alvos, detecção de comida, status próprio (vida, ataque, tipo), detecção de estado do alvo (vida).
- **Caçadores:** Detecção de alvos, detecção de áreas para proteção, detecção do estado do alvo (vida)

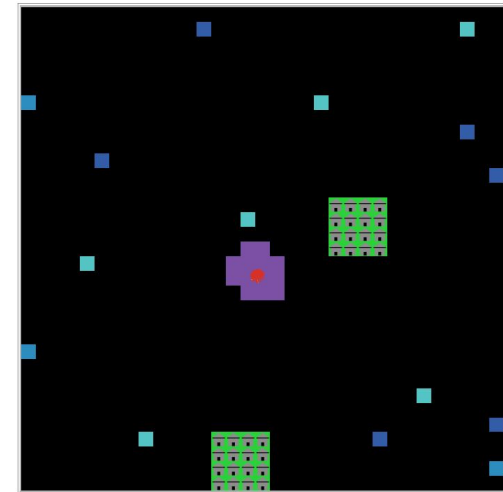
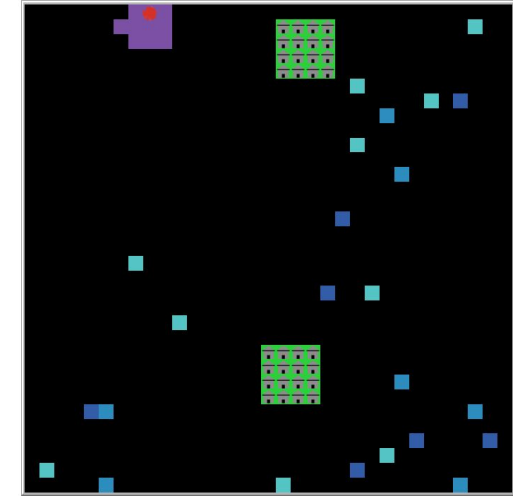




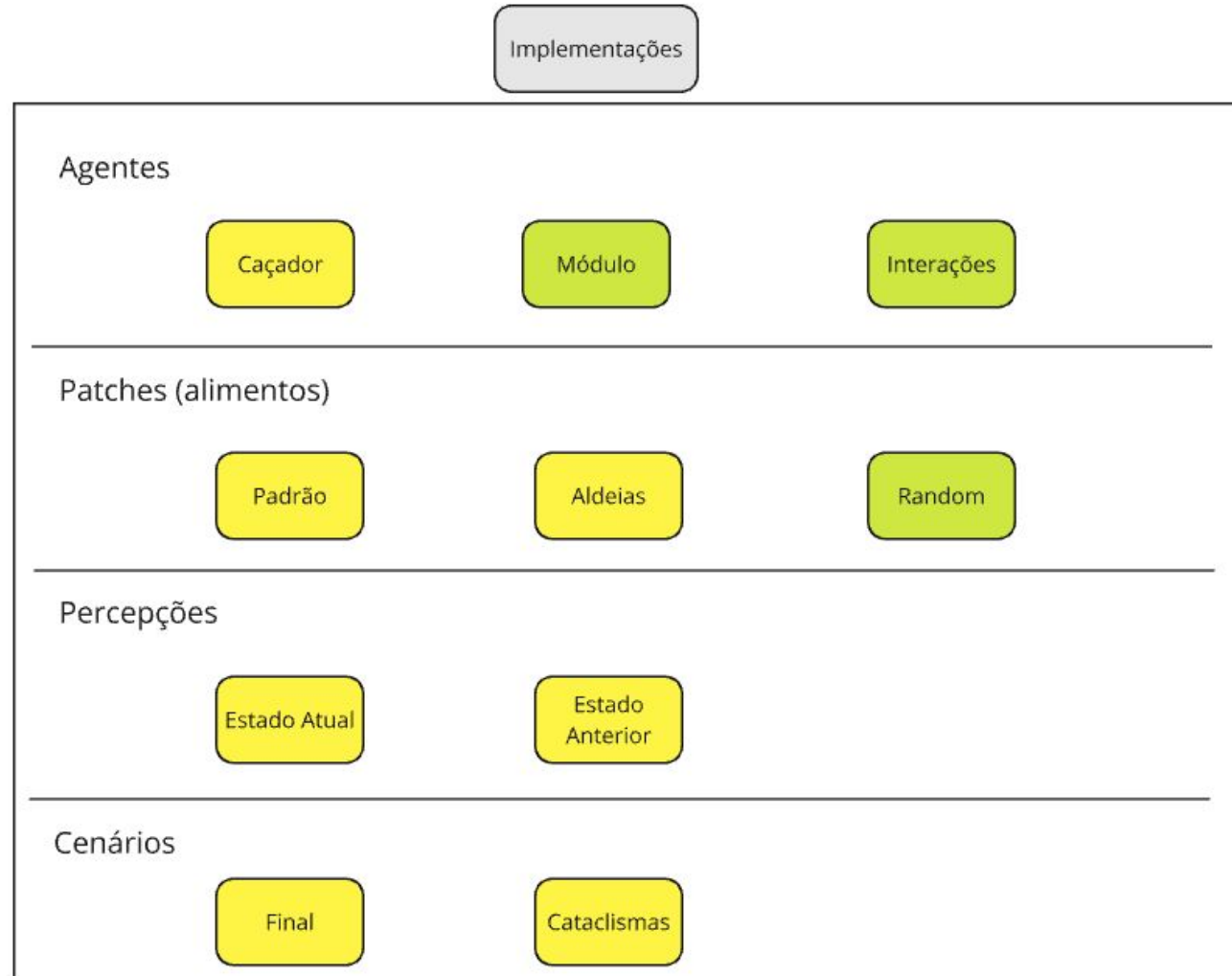
# INTRODUÇÃO

## • PROPRIEDADES DO AMBIENTE:

- **Parcialmente observável:** Os agentes percebem apenas as células que estão próximas.
- **Discreto:** O ambiente é dividido em células ou pixels.
- **Dinâmico:** Os agentes se movem e interagem constantemente.
- **Estocástico:** O comportamento dos agentes pode ser aleatório, como a movimentação ou a seleção de alvos, e o ambiente também pode ser aleatório, com o posicionamento dos recursos.
- **Sistema Multiagente:** Existem dois tipos de agentes globais — formigas e caçadores — divididos em classes que desempenham diferentes papéis dentro da modelagem:
  - Autônomos, Percepção limitada, Comunicação, Ambiente competitivo



# DESENVOLVIMENTO





# DESENVOLVIMENTO

## Definição de novas variáveis

```

[ globals [
  random-x
  random-y
  evaporation-rate
  diffusion-rate
  num-comida-armazenada
  num-humanos-mortos
  num-cacadores-comum-mortos ←
  num-cacadores-elite-mortos
  num-guardas-reais
  ultimo-guardas-reais
  ultimo-humanos-mortos
  ultimo-cacadores-mortos ←
  rei? ←
  rainha?
  num-cacadores-lendarios
  cod-rei
  cod-rainha
  encontrou?
]

```

```

[ turtles-own [
  classe
  tipo
  vida
  dano
  comida?
  foco
]

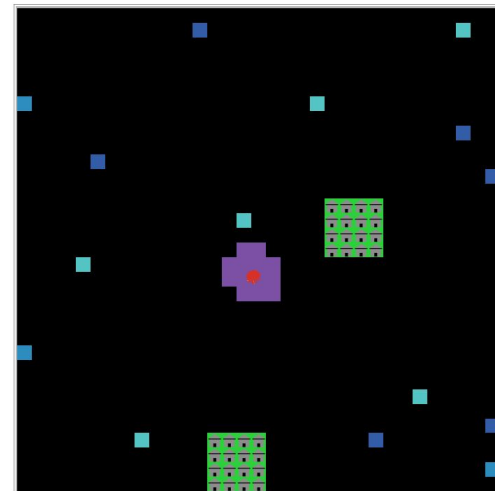
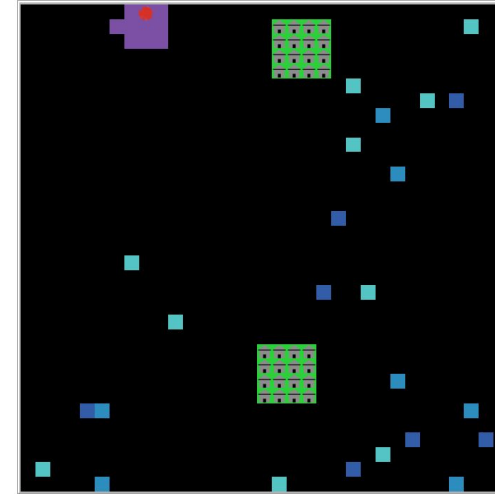
[ patches-own [
  comida
  tipo-de-fonte-de-comida
  aldeoes
  quantidade-de-fontes-de-comida
  chemical
  ninho? ←
  nest-scent
  aldeia?
]

```

# DESENVOLVIMENTO

## Setup

```
to setup
  clear-all
  print "Bem-vindo à Simulação HxH Quimera."
  set random-x random-xcor
  set random-y random-ycor
```



# DESENVOLVIMENTO

## Definição dos Patches

```

to criar-formigueiro
  let propriedades (propriedades-formiga "amarelo")

  create-turtles 1 [
    set classe "formiga"
    set tipo item 0 propriedades
    set vida item 1 propriedades
    set dano item 2 propriedades
    set color item 3 propriedades
    setxy random-x random-y
  ]

  set cod-rainha one-of turtles with [color = yellow]
  set rainha? true

  ; Cria formigas vermelhas (soldados)
  criar-formigas-como "vermelho" 10
end

to destacar-formigueiro
  if random-x > (max-pxcor - 1) [ set random-x (max-pxcor - 1) ]
  if random-x < (min-pxcor + 1) [ set random-x (min-pxcor + 1) ]
  if random-y > (max-pycor - 1) [ set random-y (max-pycor - 1) ]
  if random-y < (min-pycor + 1) [ set random-y (min-pycor + 1) ]
  ask patches [
    if (distancexy random-x random-y) < 2
    [
      set pcolor violet
    ]
  ]
end

```

# DESENVOLVIMENTO

## Definição dos Patches

```

to criar-comida-padrao
  set quantidade-de-fontes-de-comida one-of [0 1 2]

  repeat quantidade-de-fontes-de-comida [
    let float random-float 10 + 1
    let new-x random-xcor * float
    let new-y random-ycor * float

    if (distancexy (new-x * float) (new-y * float)) < (distancexy random-x random-y)
    [
      set tipo-de-fonte-de-comida one-of [1 2 3]
    ]
    if tipo-de-fonte-de-comida > 0
    [
      set comida one-of [10 20]
    ]
  ]
end

to destacar-patch-de-comida
  if comida > 0 [
    if tipo-de-fonte-de-comida = 1 [ set pcolor cyan ]
    if tipo-de-fonte-de-comida = 2 [ set pcolor sky ]
    if tipo-de-fonte-de-comida = 3 [ set pcolor blue ]
  ]
end

```

```

to criar-aldeia [quantidade]
  repeat quantidade [
    let x random-xcor
    let y random-ycor

    ask patches with [abs (pxcor - x) <= 2 and abs (pycor - y) <= 2] [
      set aldeoes random 20 + 10
      set pcolor lime
      set chemical 100
      set aldeia? true
    ]
  ]

  ask patches with [aldeia?] [
    sprout 1 [
      set shape "house"
      set color gray
      set size 1
    ]
  ]
end

```

# DESENVOLVIMENTO

Agente: Formiga

```
to-report propriedades-formiga [formiga-cor]
; if formiga-cor = "lilas" [
;   report ["movel" 150 3 violet]
; ]
if formiga-cor = "rosa" [
  report ["movel" 120 6 magenta]
]
if formiga-cor = "laranja" [
  report ["movel" 150 10 lime]
]
if formiga-cor = "amarelo" [
  report ["imovel" 500 25 yellow]
]
report ["movel" 100 4 red]
end
```

```
to criar-formigas-como [formiga-cor quantidade]
  let propriedades (propriedades-formiga formiga-cor)

  create-turtles quantidade [
    set classe "formiga"
    set tipo item 0 propriedades
    set vida item 1 propriedades
    set dano item 2 propriedades
    set color item 3 propriedades
    set size 1
    setxy random-x random-y
  ]
end
```



# DESENVOLVIMENTO

## Agente: Formiga

```

to gerar-novas-formigas
  if num-comida-armazenada > 0 and num-comida-armazenada mod 5 = 0 and rei? = false [
    criar-formigas-como "vermelho" 1
  ]
  if num-humanos-mortos > 0 and num-humanos-mortos mod 10 = 0 and rei? = false [
    criar-formigas-como "rosa" 1
  ]

  if num-cacadores-elite-mortos > 0 and num-cacadores-elite-mortos mod 5 = 0 and num-guardas-reais < 3 and rei? = false [
    criar-formigas-como "laranja" 1
    set num-guardas-reais num-guardas-reais + 1
    print "Guarda-real nasceu!"
  ]

  if num-guardas-reais > ultimo-guardas-reais and num-guardas-reais mod 3 = 0 and rei? = false [
    ask cod-rainha [
      print "A rainha está morta! Longa vida ao rei!"
      wait 1
      die
    ]
    criar-formigas-como "amarelo" 1
    set cod-rei one-of turtles with [color = yellow]
    set rei? true

    criar-rainha
    set rainha? true

    set ultimo-guardas-reais num-guardas-reais
  ]
end

```



# DESENVOLVIMENTO

Agente: Formiga (rei)

```
to procurar-conjuge
  let alvo one-of turtles in-radius 1 with [self = cod-rainha]
  if alvo != nobody [
    set random-x [xcor] of alvo
    set random-y [ycor] of alvo

    wait 1
    set rei? false
    set encontrou? true
    print "O rei está morto! Longa vida à rainha!"
    die
  ]
end
```

# DESENVOLVIMENTO

## Agente: Caçador

```

to criar-novo-cacador [tipo-cacador quantidade]
  let propriedades (propriedades-cacadores tipo-cacador)

  create-turtles quantidade [
    set shape "wolf"
    set size 1
    set classe "cacador"
    set tipo item 0 propriedades
    set dano item 2 propriedades
    set vida item 1 propriedades
    set color item 3 propriedades
    if (item 4 propriedades) = true [
      set foco one-of patches with [aldeia?]
    ]
    setxy random-xcor random-ycor
  ]
end

to mover-cacadores

to-report propriedades-cacadores [tipo-cacador]
  if tipo-cacador = "cacador-elite" [
    report ["cacador-elite" 300 20 orange true]
  ]
  if tipo-cacador = "cacador-lendario" [
    report ["cacador-lendario" 750 30 pink true]
  ]
  report ["cacador-comum" 200 15 blue true]
end

```

# DESENVOLVIMENTO

Agente: Caçador

```
to criar-cacadores
  if num-humanos-mortos > 0 and num-humanos-mortos mod 5 = 0 and num-humanos-mortos > ultimo-humanos-mortos [
    criar-novo-cacador "cacador-comum" 1
    set ultimo-humanos-mortos num-humanos-mortos
  ]

  if num-cacadores-comum-mortos > 0 and num-cacadores-comum-mortos mod 5 = 0 and num-cacadores-comum-mortos > ultimo-cacadores-mortos [
    criar-novo-cacador "cacador-comum" 1
    criar-novo-cacador "cacador-elite" 1
    set ultimo-cacadores-mortos num-cacadores-comum-mortos
  ]

  if rei? = true and num-cacadores-lendarios <= 4 [
    criar-novo-cacador "cacador-lendario" 2
    set num-cacadores-lendarios num-cacadores-lendarios + 2
    print "Caçador lendario nasceu!"
  ]
end
```

# DESENVOLVIMENTO

## Interação entre agentes

```

to verificar-alvos [classe-agente]
  ask turtles with [classe = classe-agente] [
    if classe = "cacador" [
      let alvo one-of turtles in-radius 1 with [classe = "formiga"]
      if alvo != nobody [
        ask alvo [
          set vida vida - [dano] of myself
          if vida <= 0 [
            if color = orange [set num-guardas-reais num-guardas-reais - 1]
            if self = cod-rei [
              print "O rei foi morto!"
              set rei? false
            ]
            if self = cod-rainha [
              print "A rainha foi morta!"
              set rainha? false
            ]
            die
          ]
        ]
      ]
    ]
  ]
  ;set vida vida - 10
]

```

# DESENVOLVIMENTO

## Interação entre agentes

```

if classe = "formiga" [
  let alvo one-of turtles in-radius 1 with [classe = "cacador"]
  if alvo != nobody [
    ask alvo [
      set vida vida - [dano] of myself
      if vida <= 0 [
        if tipo = "cacador-comum" [set num-cacadores-comum-mortos num-cacadores-comum-mortos + 1]
        if tipo = "cacador-elite" [set num-cacadores-elite-mortos num-cacadores-elite-mortos + 1]
        ; print "Um caçador foi eliminado!"
        die
      ]
    ]
    set vida vida - 10
  ]
]

```

# DESENVOLVIMENTO

## Interação entre agentes

```

to go
  ask turtles with [classe = "formiga" and tipo = "movel"] [
    if who >= ticks [ stop ]           ; sincroniza a saída das formigas do ninho com o tempo
    verificar-alvos "formiga"
    ifelse comida? = false and rei? = false [
      procurar-por-comida               ; procura por comida se não estiver carregando
    ] [
      retornar-ao-formigueiro           ; retorna ao ninho se estiver carregando comida
    ]
    wiggle                             ; movimento aleatório para simular procura
    fd 1                               ; move-se para frente
  ]
  diffuse chemical (diffusion-rate / 100) ; difusão do feromônio entre os patches
  ask patches [
    set chemical chemical * (100 - evaporation-rate) / 100 ; evaporação do feromônio
    recolor-patch                                     ; atualiza a cor do patch após mudanças
  ]
  if rei? = true [
    ask turtles with [ color = yellow ] [
      wiggle
    ]

    ask cod-rei [ procurar-conjuge ]

    if rei? = false and encontrou? = true [
      ask patches [
        set nest-scent 0
        setup-ninho
      ]
      destacar-formigueiro
    ]
  ]

  if rei? = false and encontrou? = false [
    print "Os caçadores mataram o rei! Fim da simulação."
    user-message "Fim!"
    stop
  ]
]

```



# DESENVOLVIMENTO

## Catástrofes (Tempestade)

```
; == Catástrofes (Tempestade) ==  
  
to tempestade  
; Verifica se pode ocorrer uma nova tempestade  
if contador-tempestades >= 2 [  
    stop ; Impede que a tempestade ocorra mais de 2 vezes  
]  
; Incrementa o contador de tempestades  
set contador-tempestades contador-tempestades + 1  
print "A tempestade começou! Nuvens e raios estão no céu."  
; Reduz vida das formigas  
ask turtles with [classe = "formiga"] [  
    set vida max list (vida - 2) 0 ; Reduz a vida em 10, mas não abaixo de 0  
    if vida = 0 [ die ] ; Elimina formigas sem vida  
]  
print "Uma tempestade tirou vida das formigas!"  
; A tempestade dura 5 segundos (50 ciclos de 5s)  
repeat 50 [  
    ask patches [  
        set pcolor gray  
    ]  
    display ; Atualiza a interface para mostrar os efeitos  
    wait 3 ; Aguarda 3.0 segundo antes do próximo ciclo  
]  
display ; Atualiza a interface novamente  
print "A tempestade acabou. O céu está limpo novamente."  
; Marca o fim do evento  
set evento-catastrofe? false  
end
```

# DESENVOLVIMENTO

Verificando catástrofes

```
; === Verificando Catástrofes

to check-catastrophes
  if contador-tempestades < 2 [ ; Limita a tempestade a no máximo 2 ocorrências
    if random 100 < 10 [ ; 10% de chance de ocorrer uma tempestade
      tempestade
    ]
  ]
end
```

# DESENVOLVIMENTO

To go

```
to go
  check-catastrophes ; Verifica se ocorre uma tempestade no início do tick
  ask turtles with [classe = "formiga" and tipo = "movel"] [
    if who >= ticks [ stop ] ; sincroniza a saída das formigas do ninho com o tempo
    verificar-alvos "formiga"
    ifelse comida? = false and rei? = false [
      procurar-por-comida ; procura por comida se não estiver carregando
    ] [
      retornar-ao-formigueiro ; retorna ao ninho se estiver carregando comida
    ]
    wiggle ; movimento aleatório para simular procura
    fd 1 ; move-se para frente
  ]
  diffuse chemical (diffusion-rate / 100) ; difusão do feromônio entre os patches
  ask patches [
    set chemical chemical * (100 - evaporation-rate) / 100 ; evaporação do feromônio
    recolor-patch ; atualiza a cor do patch após mudanças
  ]

  ask turtles with [classe = "cacador"] [
    verificar-alvos "cacador"

    mover-cacadores
    fd 1
  ]

  if rei? = true [
    set encontrou? false
    ask turtles with [ color = yellow ] [
      wiggle
    ]
  ]

  ask cod-rei [ procurar-conjuge ]

  if rei? = false and encontrou? = true [
    ask patches [
      set nest-scent 0
      setup-ninho
    ]
    destacar-formigueiro
  ]

  if rainha? = false [
    print "O rei não consegue mais seguir com a linhagem. Fim da simulação."
    user-message "Fim!"
    stop
  ]
]
```

# DESENVOLVIMENTO

To go

```

    ;set encontrou? false
  ]

  if rainha? = true and rei? = false and encontrou? = false [
    print "A rainha não consegue mais seguir com a linhagem. Fim da simulação."
    user-message "Fim!"
    stop
  ]

  if rainha? = false and rei? = false [
    print "As formigas não conseguem mais procriar! Fim da simulação."
    user-message "Fim!"
    stop
  ]

  ;Verifica população de formigas
  let populacao-formiga count turtles with [classe = "formiga" and color != yellow]
  if populacao-formiga = 0 [
    print "População de Formigas Erradicada. Fim da simulação."
    user-message "Fim!"
    stop
  ]

  ;ações nível observador
  gerar-novas-formigas
  criar-cacadores
  tick
end

```

# RESULTADO

## Interface

queen/king bug

soldier bugs

captain bugs

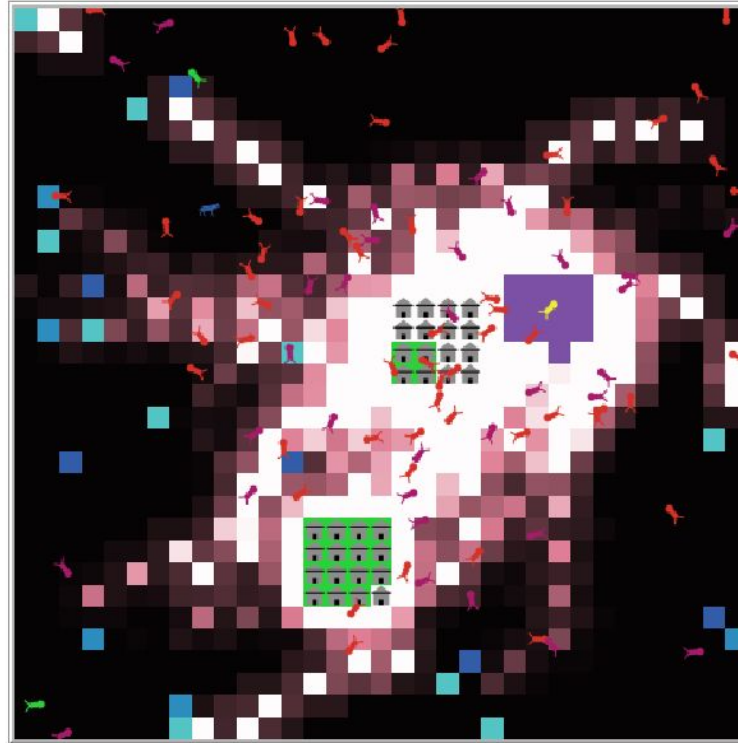
real bugs

normal hunter

elite hunter

legend hunter

num-guardas-reais



```

^v
Central de Comandos
Bem-vindo a Simulação HXH Quimera.
Guarda-real nasceu!
Guarda-real nasceu!
Guarda-real nasceu!
A rainha está morta! Longa vida ao rei!
Caçador lendário nasceu!
Caçador lendário nasceu!
Caçador lendário nasceu!
O rei está morto! Longa vida à rainha!
observer>
    
```

# RESULTADO

Fator: Distância

setup

go

queen/king bug  
1

soldier bugs  
67

captain bugs  
32

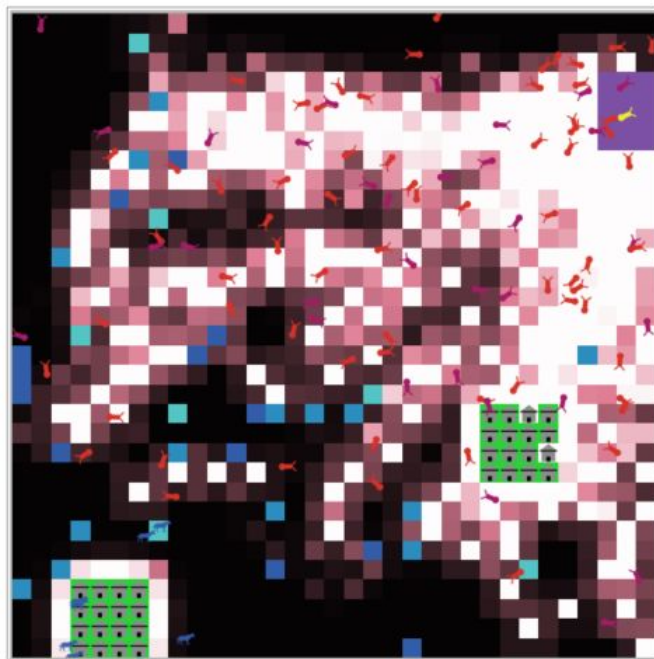
real bugs  
0

normal hunter  
7

elite hunter  
0

legend hunter  
0

num-guardas-reais  
0



setup

go

queen/king bug  
1

soldier bugs  
7

captain bugs  
7

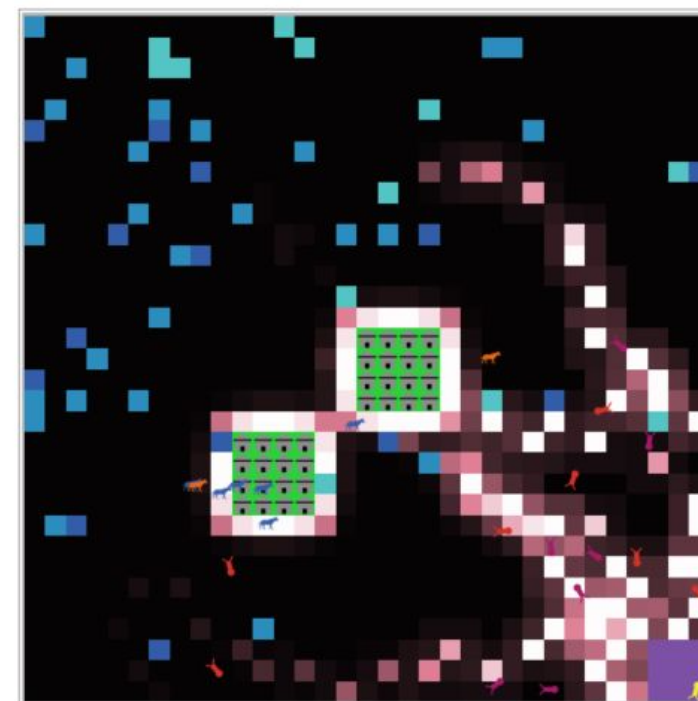
real bugs  
0

normal hunter  
6

elite hunter  
2

legend hunter  
0

num-guardas-reais  
0

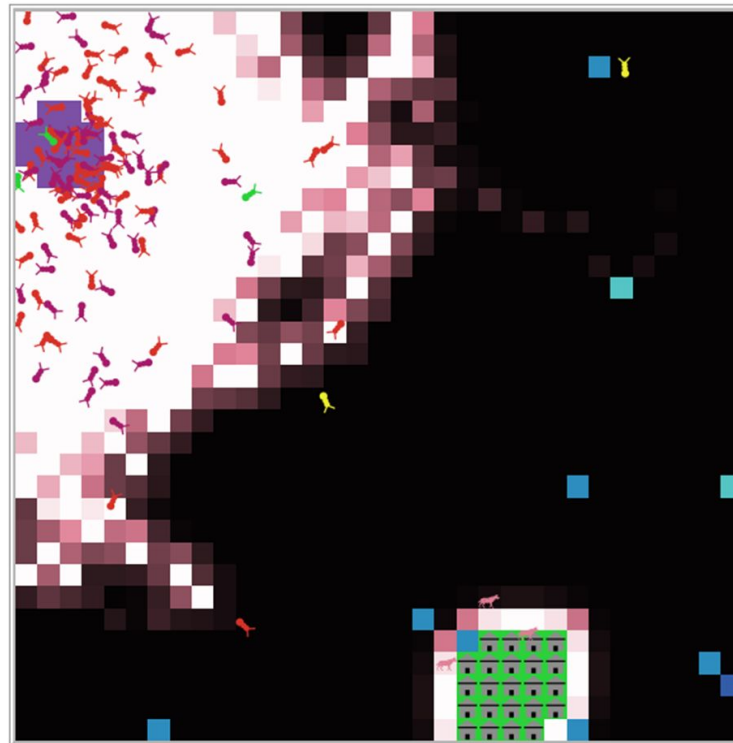




# RESULTADO

Fator: Surgimento do Rei

|  |  |
|--|--|
| <input type="button" value="setup"/>             |  |
| <input type="button" value="go"/>                |  |
| <input type="text" value="queen/king bug"/> 2    |  |
| <input type="text" value="soldier bugs"/> 62     | <input type="text" value="normal hunter"/> 0 |
| <input type="text" value="captain bugs"/> 56     | <input type="text" value="elite hunter"/> 0  |
| <input type="text" value="real bugs"/> 0         | <input type="text" value="legend hunter"/> 3 |
| <input type="text" value="num-guardas-reais"/> 3 |  |



# RESULTADO

## Cenário Final: Morte do rei ou rainha

setup

go

queen/king bug  
2

solder bugs  
32

captain bugs  
15

real bugs  
3

normal hunter  
2

elite hunter  
0

legend hunter  
6

num-guardas-reais  
3

User Message

Fim!

OK Parar

Central de Comandos

Uma tempestade tirou vida das formigas!  
Guarda-real nasceu!  
Guarda-real nasceu!  
Guarda-real nasceu!  
A rainha está morta! Longa vida ao rei!  
Caçador lendario nasceu!  
Caçador lendario nasceu!  
Caçador lendario nasceu!  
O rei foi morto pelos caçadores!  
A rainha não consegue mais seguir com a linhagem. Fim da simulação.

setup

go

queen/king bug  
1

solder bugs  
4

captain bugs  
4

real bugs  
0

normal hunter  
5

elite hunter  
1

legend hunter  
0

num-guardas-reais  
0

User Message

Fim!

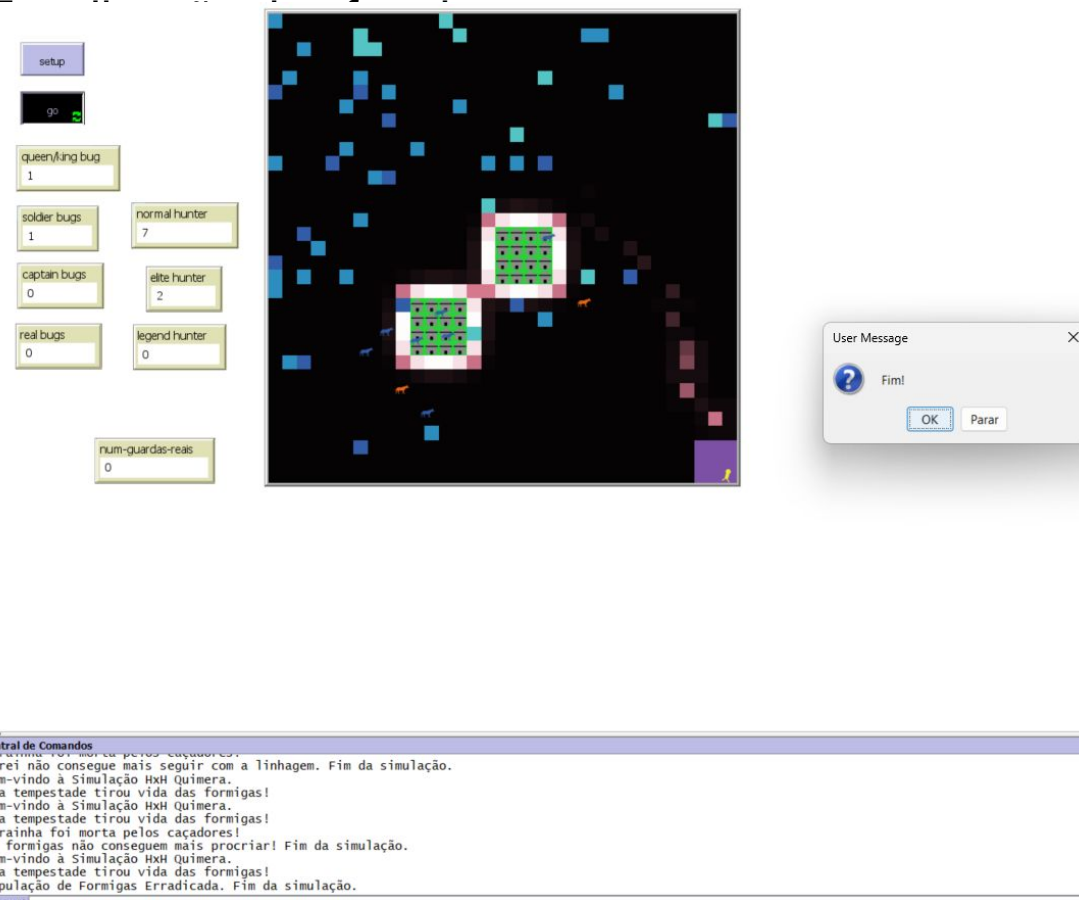
OK Parar

Central de Comandos

Uma tempestade tirou vida das formigas!  
A rainha está morta! Longa vida ao rei!  
Caçador lendario nasceu!  
Caçador lendario nasceu!  
Caçador lendario nasceu!  
O rei foi morto pelos caçadores!  
A rainha não consegue mais seguir com a linhagem. Fim da simulação.  
Bem-vindo à Simulação HxH Quimera.  
Uma tempestade tirou vida das formigas!  
A rainha foi morta pelos caçadores!  
As formigas não conseguem mais procriar! Fim da simulação.

# RESULTADO

Cenário Final:



# CONCLUSÃO

- **Objetivo e Análise:** Modelagem baseada no universo de *Hunter x Hunter* para analisar comportamentos de agentes autônomos em um ambiente competitivo e dinâmico.
- **Impacto das Condições:** A posição dos agentes e a dispersão dos recursos influenciam diretamente as interações, com proximidade do ninho favorecendo as formigas e a distância favorecendo os caçadores.
- **Momentos Críticos:** A morte da rainha e o nascimento do rei criam um ponto vulnerável para as formigas, favorecendo os caçadores e gerando um equilíbrio dinâmico no sistema.
- **Implementação e Parâmetros:** Adição de novos parâmetros e interação entre variáveis de estado, como comida e caçadores mortos, impactando o comportamento dos agentes e os resultados.
- **Contribuições:** O trabalho proporciona insights sobre dinâmicas de sistemas complexos e a interação entre agentes, mostrando como diferentes cenários influenciam os resultados.



**Obrigado!**



