

# Introductory Guide to Agent-Based Simulation Development on the GAMA Platform

**Aline Rodrigues Santos**

Aluna de Engenharia de Software / CEAVI

[aline.rs@edu.udesc.br](mailto:aline.rs@edu.udesc.br)

**Fernando Santos**

Professor / CEAVI

[fernando.santos@udesc.br](mailto:fernando.santos@udesc.br)



18<sup>th</sup> Workshop-School on Agents, Environments and Applications  
Brasilia, DF, Brazil

August 14 - 16, 2024

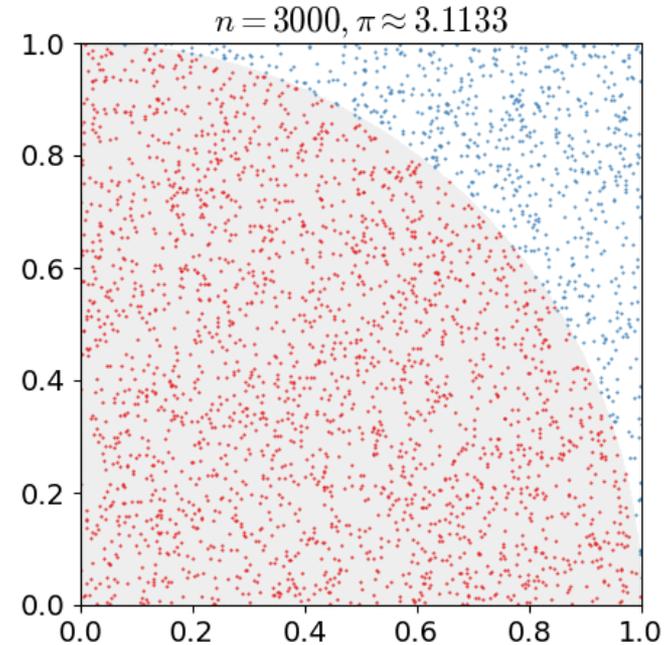
# Introdução

## Simulações baseadas em agentes (SBA):

- Visualização e compreensão de cenários

### Elementos:

- Agentes
- Ambiente
- Interações



# Introdução

## Plataformas para a criação de SBAs

- Tornar o desenvolvimento simples e intuitivo



[ccl.northwestern.edu/netlogo](http://ccl.northwestern.edu/netlogo)



[repast.github.io](http://repast.github.io)



[www.swarm.org](http://www.swarm.org)



[cs.gmu.edu/~eclab/projects/mason](http://cs.gmu.edu/~eclab/projects/mason)

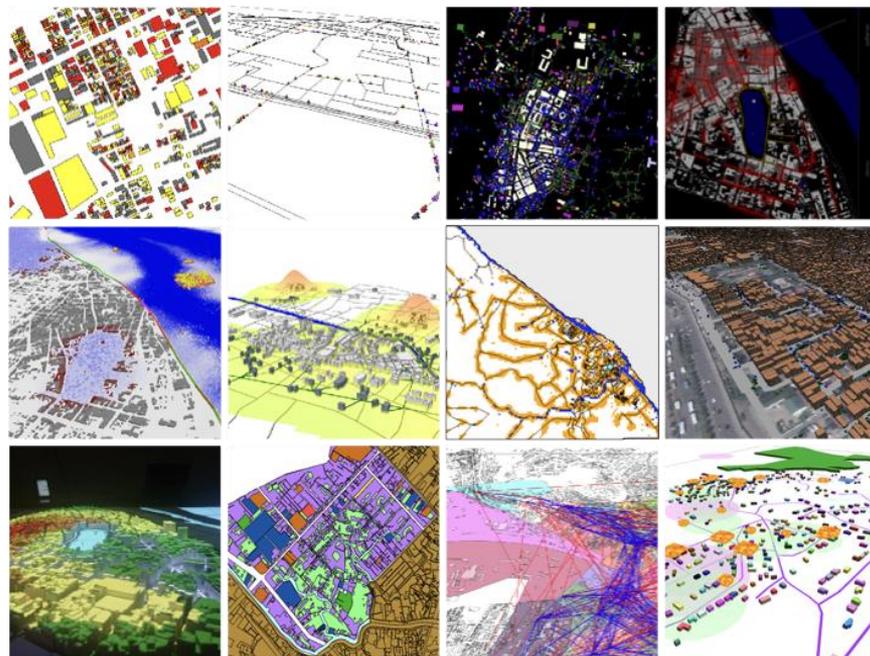


[gama-platform.github.io](http://gama-platform.github.io)

# Introdução

## Plataforma GAMA

- Taillandier, Gaudou e Grignard (2007)
- Disponibiliza recursos para a criação de SBAs
- Foco: ambiente = landscape



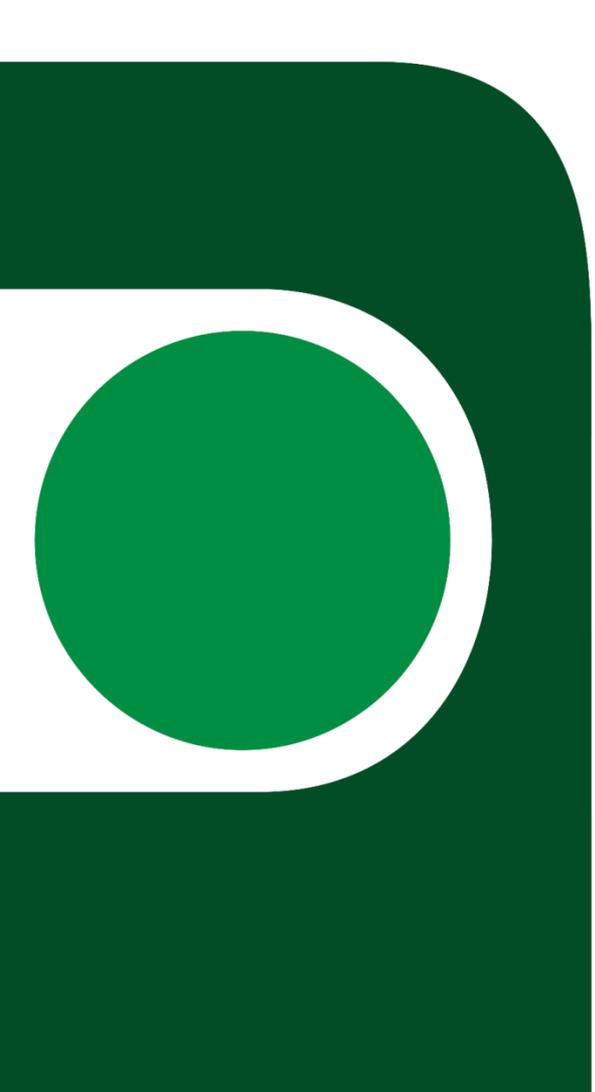
# ■ Problema e Objetivo

## **Problema:**

Falta de um guia introdutório que explique elementos essenciais para implementar simulações na GAMA.

## **Objetivo do artigo:**

Desenvolver um guia introdutório para a plataforma GAMA, utilizando uma simulação simples e amplamente conhecida (SugarScape)



# **Fundamentação Teórica**

# SugarScape

Simulação criada por Joshua M. Epstein e Robert Axtell (1996) para estudar dinâmicas sociais

**Objetivo:** Simular o comportamento de formigas

## Ambiente

- Células com açúcar (recurso)
- Reaparecimento de açúcar (total ou parcialmente)



# SugarScape

## Agentes (formigas)

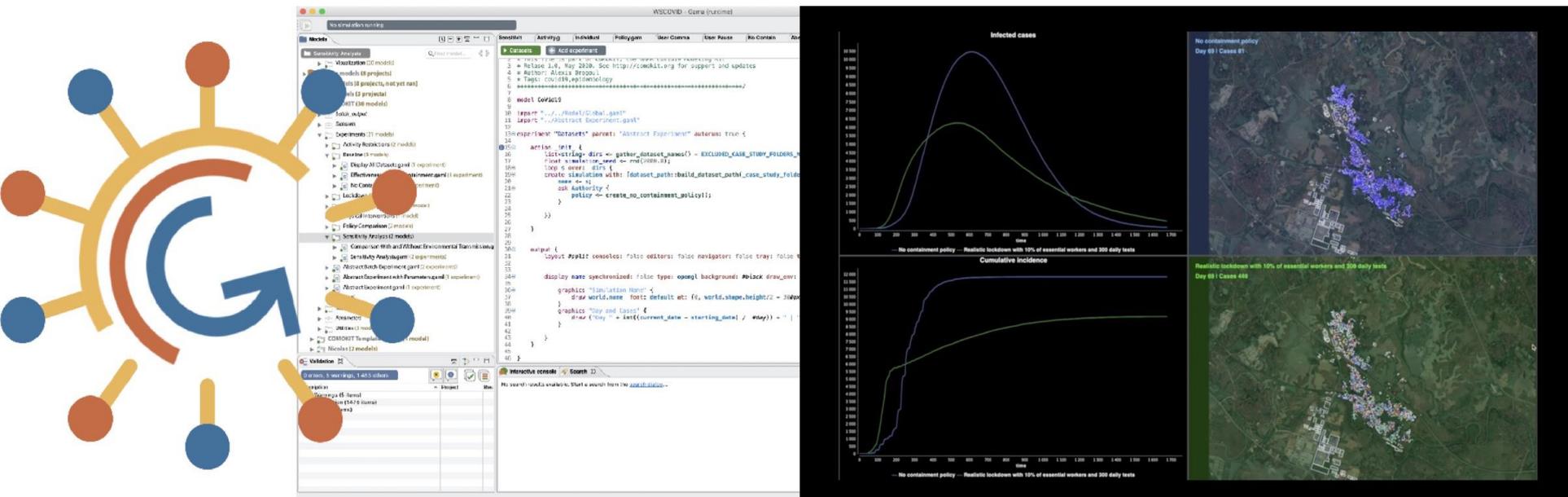
- Heterogêneos (visão, energia, metabolismo)
- Vagam pelo ambiente em busca de recursos (açúcar)
- Consomem açúcar para repor energia
- Morrem quando energia zero

**Sugarscape** reproduz fenômeno da **capacidade de carga** do ambiente



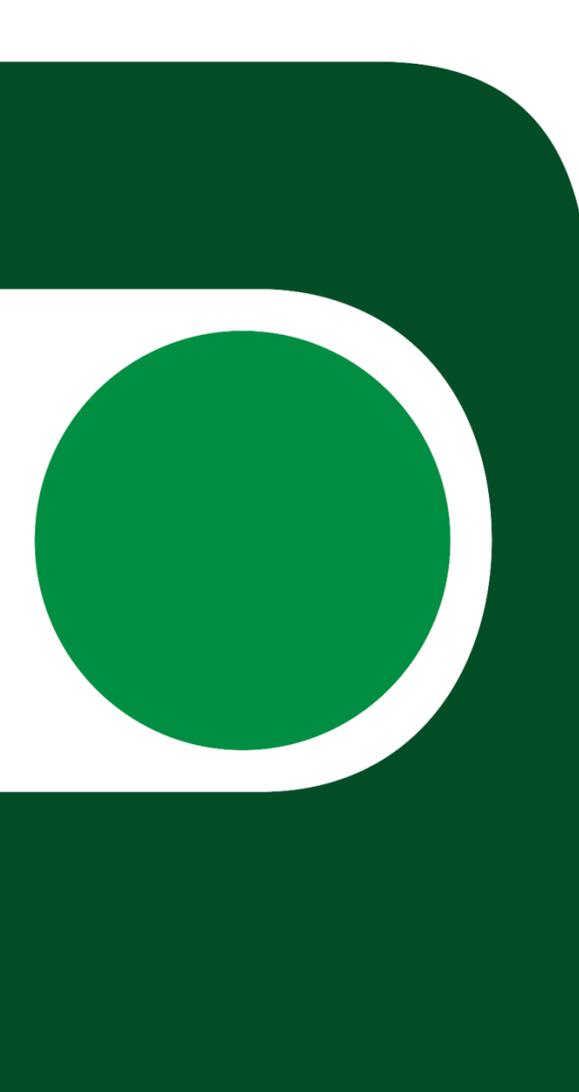
# Plataforma GAMA

- **Objetivo:** facilitar a criação de SBAs, a partir de funcionalidades específicas e interface intuitiva
- Linguagem de programação própria: GAML
- Exemplo de simulação: COMOKIT (Covid19)



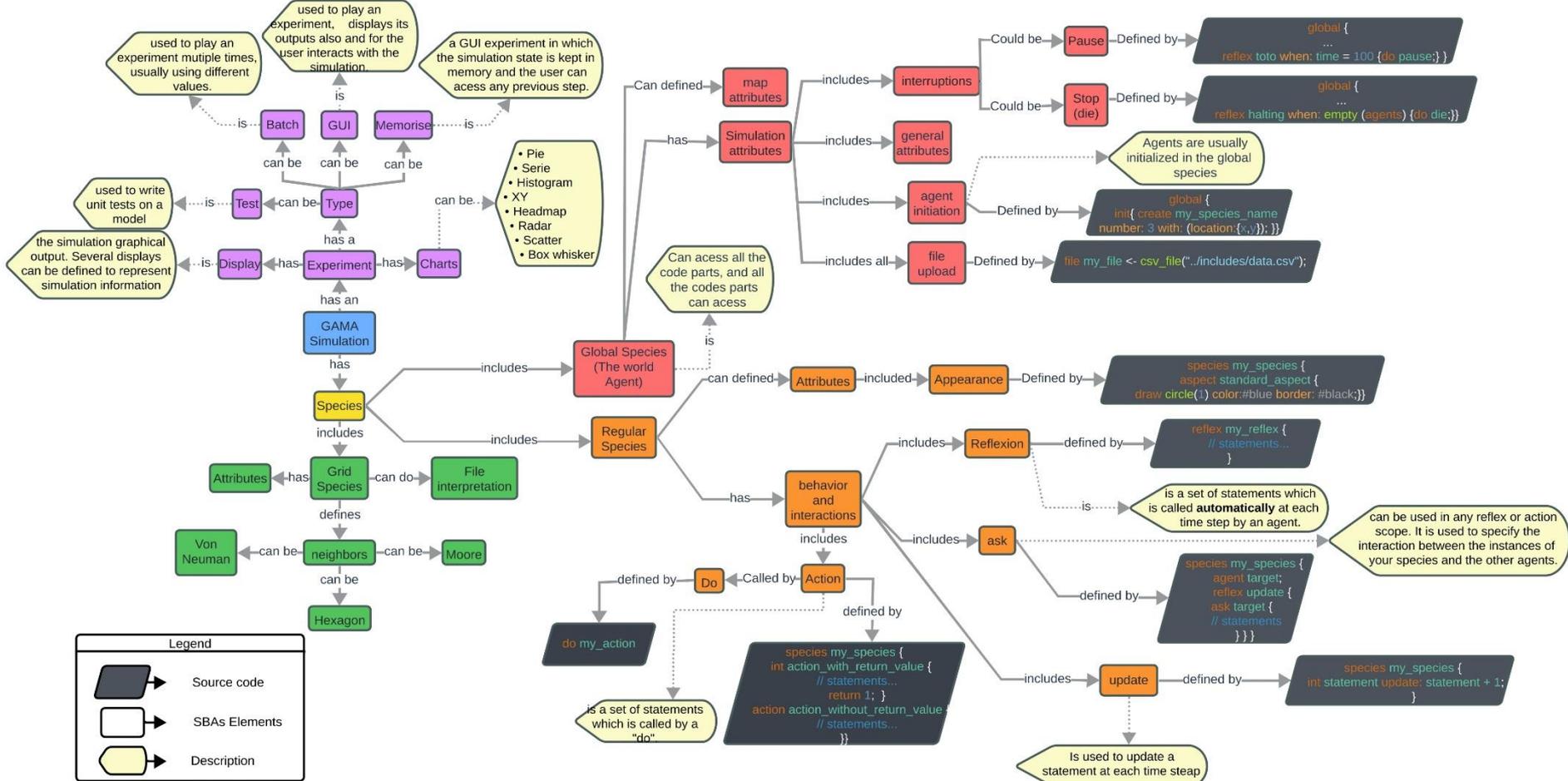
# Plataforma GAMA: desafios

- **Desafios enfrentados por um iniciante em SBA+GAMA:**
  - Linguagem de programação própria: GAML
  - Documentação extensa, porém a "digestão" não é simples
  - Ausência de artefato com visão geral dos elementos da SBA
  - Comportamento diferente da documentação
  - Curva de aprendizado



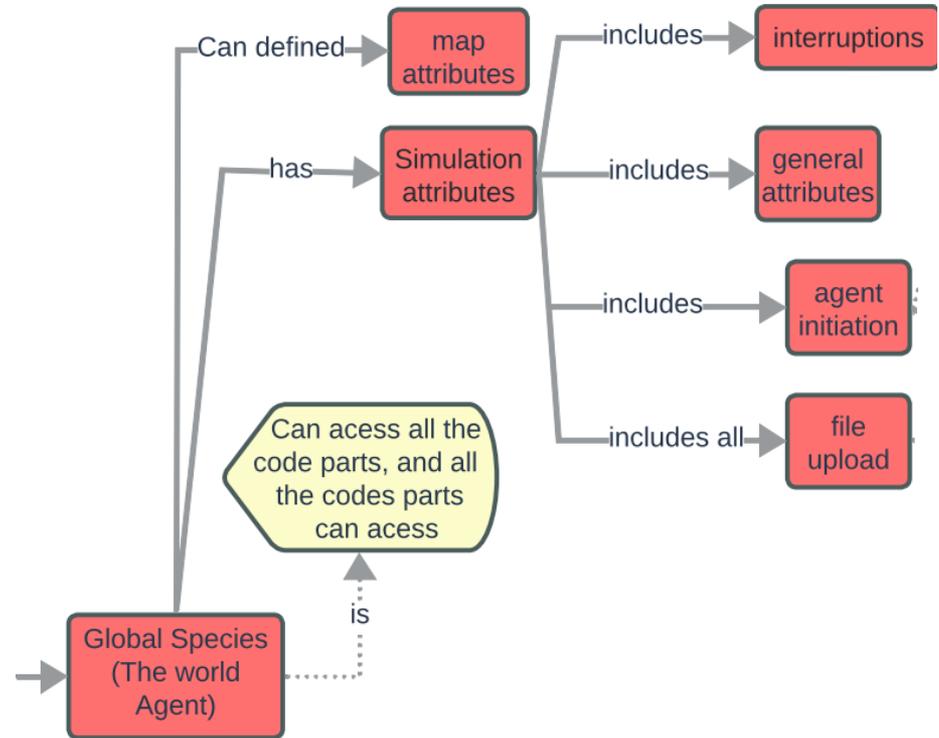
# **Guia Introdutório SBA em GAMA**

# SBA em GAMA: infográfico com elementos essenciais



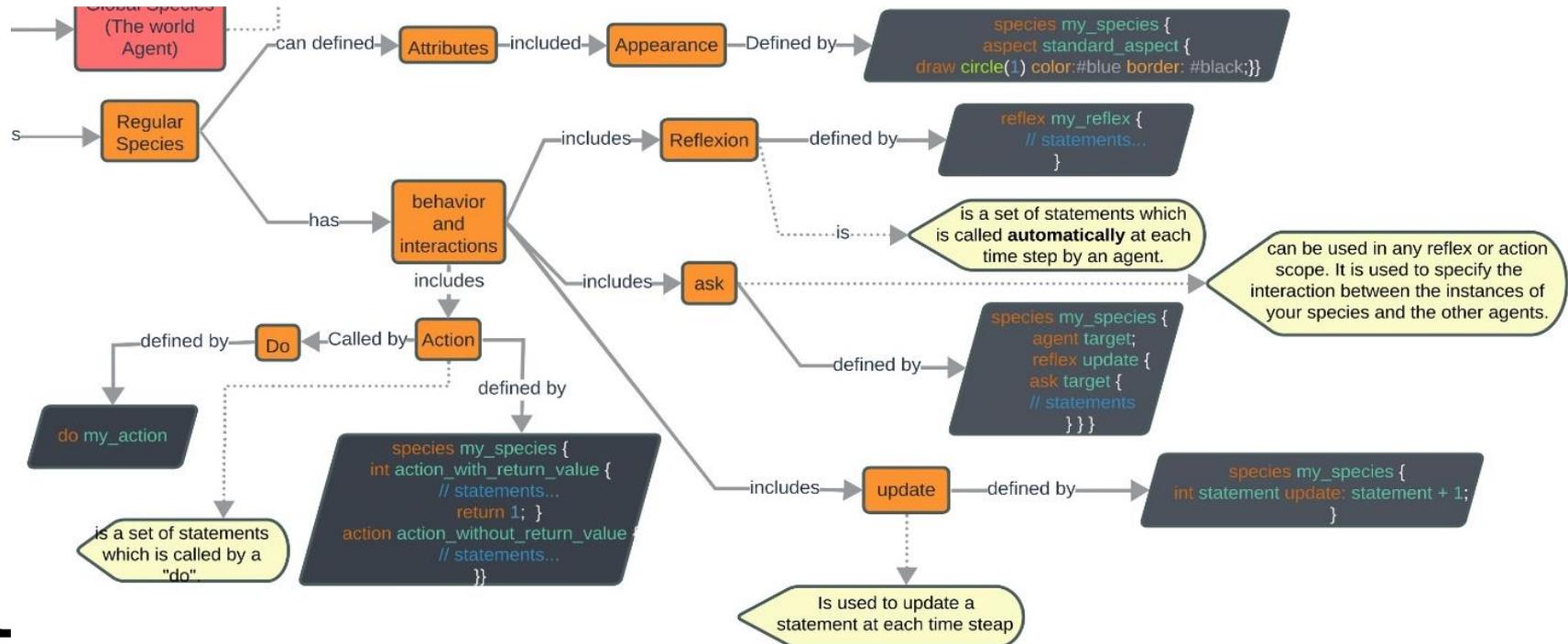
# Global Species

- Espécie (agente)-mestre
- Parâmetros gerais da SBA
- Inicializações e arquivos
- Detalhes do ambiente



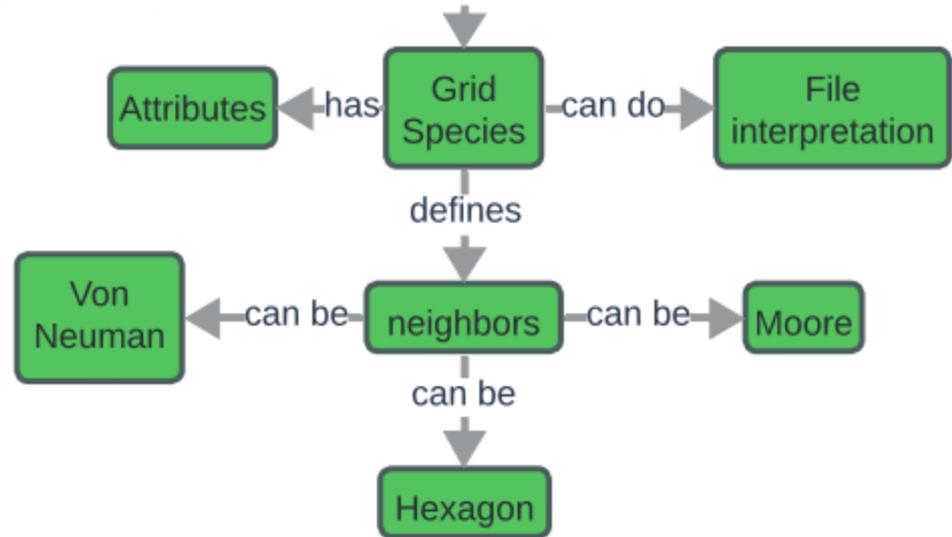
# Regular Species

- Analogia: classes na orientação a objetos
- Características e comportamentos do agente
- Primitivas para implementação (init, reflex, update, ask)



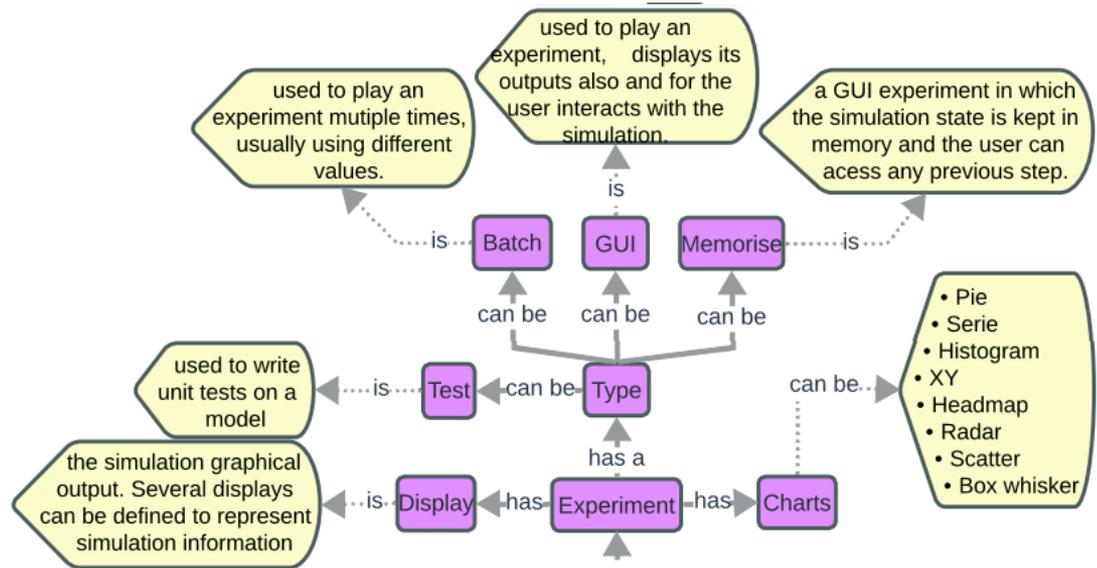
## Grid Species

- Ambiente da SBA (grid)
- Atributos pré-definidos (x, y, color, etc)
- Vizinhança das células
- Inicialização por arquivo



# Experiment

- Detalha a execução da simulação (GUI, batch, memorize, test)
- Especifica as saídas (coletadas e/ou exibidas)



# SugarScape em GAMA

The screenshot displays the GAMA (Geometric Agent Modeling Architecture) software interface. The main window shows the GAMA workspace with the file "SugarScape.gaml" open. The interface includes a menu bar (File, Edit, Search, Views, Model, Support), a toolbar with a play button and a status indicator "No experiment running", and a search bar for GAML references. A left sidebar shows a project tree with "SugarScape" selected under "User models". Below the sidebar is a validation panel showing "1508 items" and a description panel with "Information (1507 items)" and "Tasks (1 item)". The main editor area displays the GAML code for the "species ant" model, which includes initialization of attributes, movement logic, and a reflex for death when sugar levels are low.

```
55 species ant{
56   // The minimum values for some ants attributes and the random assignment of the ants' characteristics
57   int vision_ant min:1 <- rnd (vision);
58   float matabolism_ant min: 1.0<- rnd(metabolism);
59   float inicial_energy min: 5.0<- rnd(max_sugar);
60   // Ant energy variation
61   cell my_cell <- one_of(cell);
62   float sugar min: 0.00 max: max_energy <- inicial_energy update: sugar - matabolism_ant + my_cell.pSugar;
63   // movement of ants
64   init{
65     location <- choose_cell().location;
66   }
67   cell choose_cell {
68     list<cell> available_cell <- my_cell.neighbours[vision_ant] where (empty(ant inside (each)));
69     cell cell_with_max_sugar <- available_cell with_max_of (each.pSugar);
70     if (my_cell = nil){
71       return one_of(my_cell.neighbors);}
72     if (cell_with_max_sugar.pSugar < my_cell.pSugar or cell_with_max_sugar.pSugar = my_cell.pSugar) {
73       return my_cell;
74     } else {
75       return cell_with_max_sugar;}
76   }
77   // ant appearance
78   aspect default {
79     draw circle(1.0) color: #darkred;
80     draw string (sugar with_precision 1 ) size: 3 color: #black;
81   }
82   // The end of agent's life
83   reflex end_of_life when: (sugar < 1) {
84     do die;
85   }
86 }
```

# SugarScape em GAMA – Global Species

- Parâmetros da simulação
- Inicialização do ambiente
- Métodos para coletar dados

```
sugarscape2.gaml X
simulation + Add Experiment
4 * Author: Aline Santos
5 * Tags:
6 */
7 model sugarscape1
8
9 global {
10   int vision <- 6;
11   int nb_initial_ant <- 400;
12   int metabolism <- 4;
13   float max_energy <- 200.0;
14   int nb_ant -> {length(ant)};
15   float average_vision <- 6.0 update: calculate_average_vision();
16   float average_metabolism <- 5.0 update: calculate_average_metabolism();
17   float calculate_average_vision {
18     float totalVision <- 0.0;
19     ask ant {
20       totalVision <- totalVision + vision_ant;
21     }
22     return totalVision / nb_ant;
23   }
24
25   float calculate_average_metabolism {
26     float totalmetabolism <- 0.0;
27     ask ant {
28       totalmetabolism <- totalmetabolism + metabolism_ant;
29     }
30     return totalmetabolism / nb_ant;
31   }
32
33   csv_file arquivo <- csv_file("../includes/map.csv");
```

# SugarScape em GAMA – Regular Species

## Agente Formiga

- Atributos + Comportamentos + Aspecto visual

```
53 species ant {
54   int vision_ant min: 1 <- rnd(vision);
55   cell my_cell <- one_of(cell);
56   int metabolism_ant min: 1 <- rnd(metabolism);
57   float inicial_energy min: 5.0 <- rnd(max_energy);
58   float energy min: 0.0 <- inicial_energy update: energy - metabolism_ant + my_cell.sugar max: max_energy;
59
60   init {
61     location <- my_cell.location;
62     my_cell <- choose_cell();
63   }
64
65   cell choose_cell {
66     list<cell> available_cells <- my_cell.neighbours[vision_ant] where (empty(ant inside (each)));
67     cell cell_with_max_sugar <- available_cells with_max_of (each.sugar);
68     if (cell_with_max_sugar.sugar < my_cell.sugar or cell_with_max_sugar.sugar = my_cell.sugar) {
69       return my_cell;
70     } else {
71       return cell_with_max_sugar;
72     }
73
74 }
```

```
76 aspect default {
77   draw circle(1.0) color: #darkred;
78 }
79
80 reflex end_of_life when: (energy <= 0) {
81   do die;
82 }
83 }
```

## SugarScape em GAMA – Grid Species

- Dimensões do ambiente grid
- Vizinhança
- Açúcar disponível nas células
- Aspecto visual

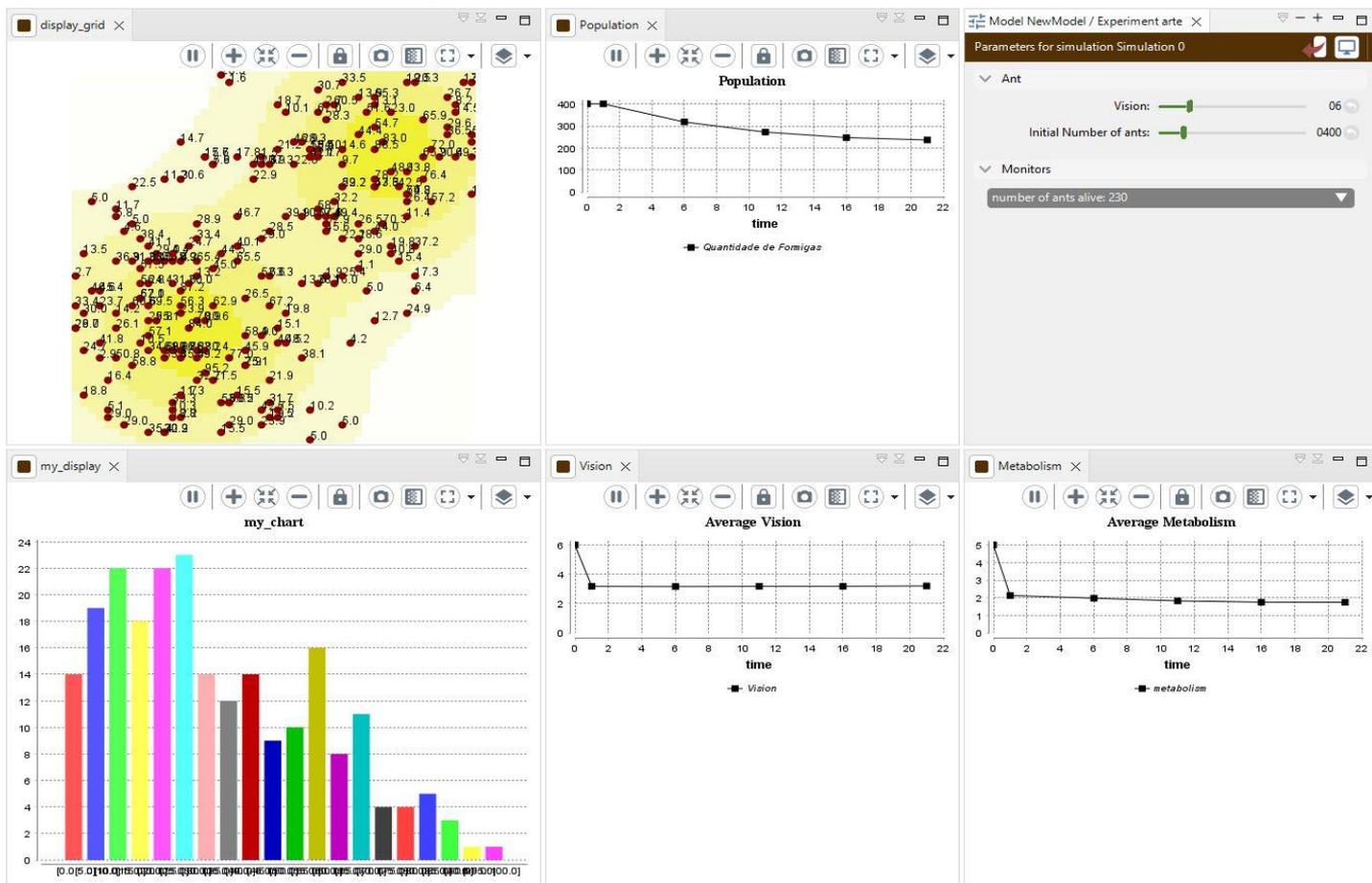
```
79 @ grid cell width: 50 height: 50 neighbors: 4 {
80     float sugar;
81     map<int, list<cell>> neighbours;
82
83 @     reflex updateColor {
84 @         if (sugar = 1) {
85             color <- rgb(250, 250, 210);
86 @         } else if (sugar = 2) {
87             color <- rgb(247, 246, 167);
88 @         } else if (sugar = 3) {
89             color <- rgb(243, 242, 126);
90 @         } else if (sugar = 4) {
91             color <- rgb(240, 241, 50);
92 @         } else if (sugar = 0) {
93             color <- rgb(254, 254, 251);
94         } }
95
96     // A way to deal with neighbours bug
97 @     init {
98 @         loop i from: 1 to: vision {
99             neighbours[i] <- self neighbors_at i;
100         }
101     } }
```

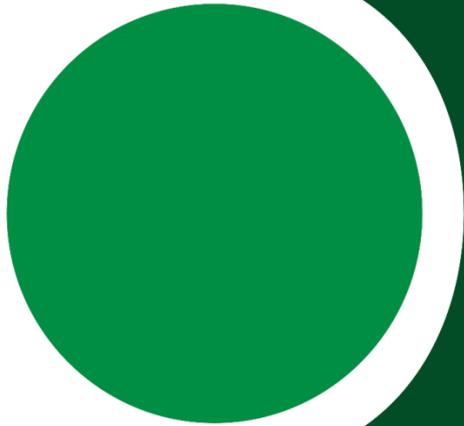
## SugarScape em GAMA - Experiment

- GUI dos parâmetros para o usuário
- Definição da visualização e saídas

```
104 experiment simulation type: gui {
105     parameter "Max values of vision: " var: vision min: 1 max: 25 category: "Ant";
106     parameter "Initial number of ants: " var: nb_initial_ant min: 1 max: 2500 category: "Ant";
107     output {
108         display display_grid {
109             grid cell;
110             species ant aspect: default;
111         }
112
113         display "my_display" {
114             chart "my_chart" type: histogram {
115                 datalist (distribution_of(ant collect each.energy, 10, 0, 200) at "legend") value: (di
116             }
117         }
118     }
```

# SugarScape em GAMA - Experiment





# **Recomendações para um iniciante em GAMA**

# Recomendações

- **Use o fórum:**

- A sua pergunta pode levar um tempo para ser respondida, mas acessar o fórum te ajuda a verificar problemas similares e achar alternativas.
- Encontre usuários da plataforma e os peça ajuda, nada melhor do que alguém experiente para ajudar.

- **Use os modelos prontos:**

- Ver os modelos prontos disponíveis ajudam a se habituar com a plataforma e te ajuda a encontrar alternativas para funcionalidades não funcionais

# Recomendações

- **Utilizar o fórum em casos de comportamentos inesperados**



fds...@gmail.com

para GAMA

27 de out. de 2023, 13:39:18

Hello there,

I'm getting an unexpected return from the neighbors\_at function in the Sugarscape toy sample included within the Gama Library models.

In the basic\_move reflex, I'm trying to get the neighbors of a place (sugar\_cell) using the neighbors\_at function. However, the returned number of sugar\_cells is not the same as that returned during the initialization of the grid.

Please see the basic\_move code below, in particular the two write statements.



alexis...@gmail.com

para gama-p...@googlegroups.com

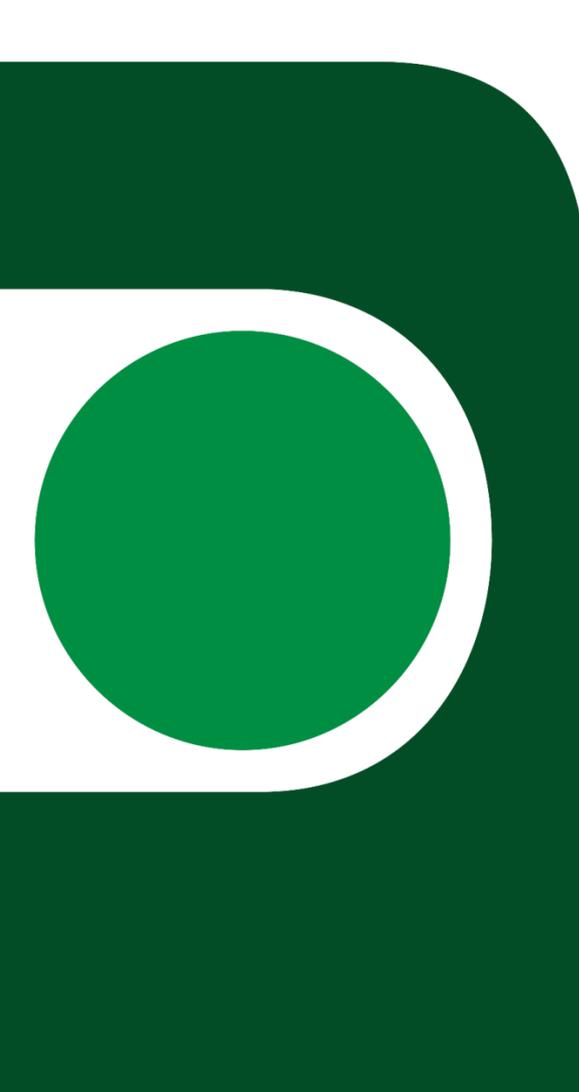
31 de out. de 2023, 12:01:22

Hi,

Let's investigate if this is a bug -- in the meantime, you can force the topology with the 'using' statement, followed by your grid species name . See: <https://gama-platform.org/wiki/OperatorsSZ#using>

Cheers

Alexis



# Conclusão

# Conclusão

- Curva de aprendizado para iniciantes em SBA
- Artigo apresentou um tutorial baseado em SBA conhecida: SugarScape
- Infográfico com elementos essenciais pode suavizar a curva de aprendizado

## Trabalhos Futuros

- Guia introdutório em vídeo
- Experimento para avaliar esforço de desenvolvimento

# Introductory Guide to Agent-Based Simulation Development on the GAMA Platform

**Aline Rodrigues Santos**

Aluna de Engenharia de Software / CEAVI

[aline.rs@edu.udesc.br](mailto:aline.rs@edu.udesc.br)

**Fernando Santos**

Professor / CEAVI

[fernando.santos@udesc.br](mailto:fernando.santos@udesc.br)



18<sup>th</sup> Workshop-School on Agents, Environments and Applications  
Brasilia, DF, Brazil

August 14 - 16, 2024