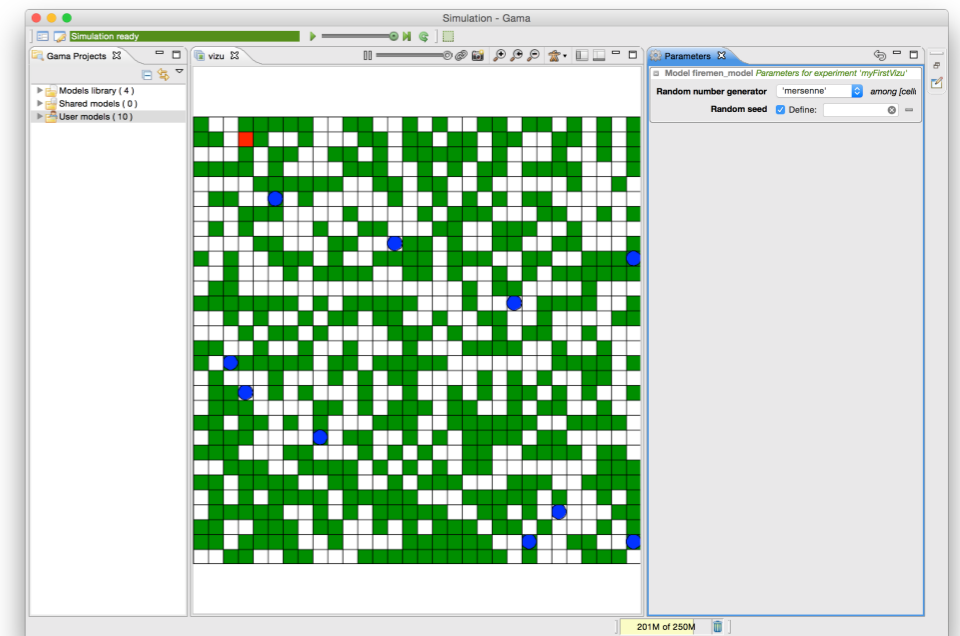
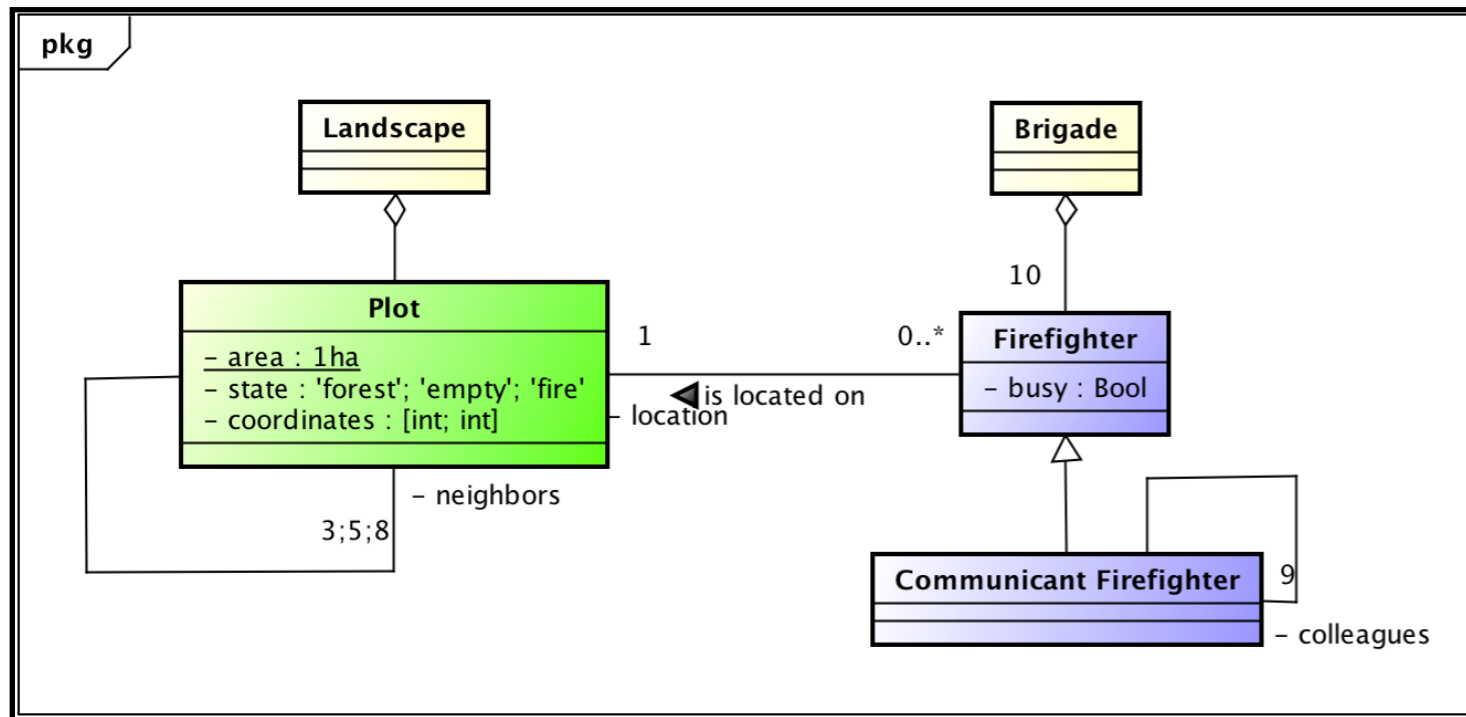
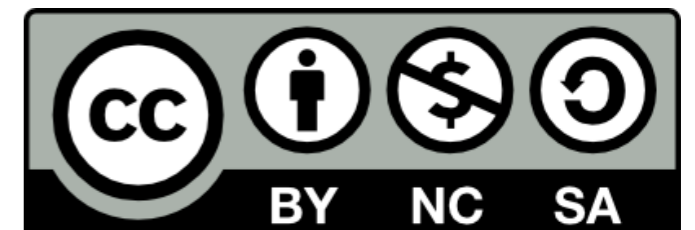


# Firefighter model: Structure and initialisation



Benoit Gaudou (Univ. Toulouse 1)  
Patrick Taillandier (INRAE)



# GAMA: first demo

- ▶ Run GAMA
  - ▶ workspace
- ▶ Open a model in the model library
  - ▶ show editor
    - ▶ compilation errors
  - ▶ run it (show the link between button and experiments)
    - ▶ simple example
    - ▶ multiple display example
    - ▶ batch mode
- ▶ Create a model
  - ▶ Create a first GAMA project
  - ▶ Create a first GAMA model

# Loading an experiment

Click on the desired **experiment** button to load it: an experiment define a simulation execution context

The screenshot displays the GAMA software interface. On the left, the 'Gama Projects' pane shows a tree view of models, with 'Ant Foraging (Complex).gaml' selected. Below it, the 'Outline' pane shows the project structure, including 'ants', 'global', 'entities', and various experiment types like 'Displays', 'Complete', 'Batch', 'Genetic', 'Quadtree', and 'Callback'. The main editor window shows the GAML code for 'Ant Foraging (Complex).gaml'. A green circle highlights the 'Run' menu, and a green arrow points to the 'Exp. Callback' button. The code editor contains the following GAML script:

```
216 display Ants background: rgb('white') refresh_every: 1 {
217     chart "Food Gathered" type: series {
218         data "Food" value: food_gathered;
219     }
220 }
221 }
222 }
223 }
224 experiment Genetic type: batch repeat: 2 keep_seed: true until: (food_gathered = food_placed) or (time >
225     parameter name: 'Size of the grid:' var: gridsize init: 75 unit: '(width and height)';
226     parameter name: 'Number:' var: ants_number init: 200 unit: 'ants';
227     parameter name: 'Evaporation:' var: evaporation_rate among: [0.1, 0.2, 0.5, 0.8, 1.0] unit: 'rate eve
228     parameter name: 'Diffusion:' var: diffusion_rate min: 0.1 max: 1.0 unit: 'rate every cycle (1.0 means
229     method genetic maximize: food_gathered pop_dim: 5 crossover_prob: 0.7 mutation_prob: 0.1 nb_prelim_ge
230 }
231 }
232 experiment Quadtree type: gui {
233     output {
234         monitor name: 'Food gathered' value: food_gathered;
235         display QuadTree {
236             quadtree 'qt';
237         }
238     }
239     display Ants background: rgb('white') refresh_every: 1 {
240         grid ant_grid lines: rgb('black');
```

At the bottom, the 'Problems' window shows 2 errors, 9 warnings, and 485 other items. The table below summarizes the error and warning counts:

Description	Resource	Path	Location	Type
Errors (2 items)				
Warnings (9 items)				
Infos (100 of 485 items)				

# Simulation Interface

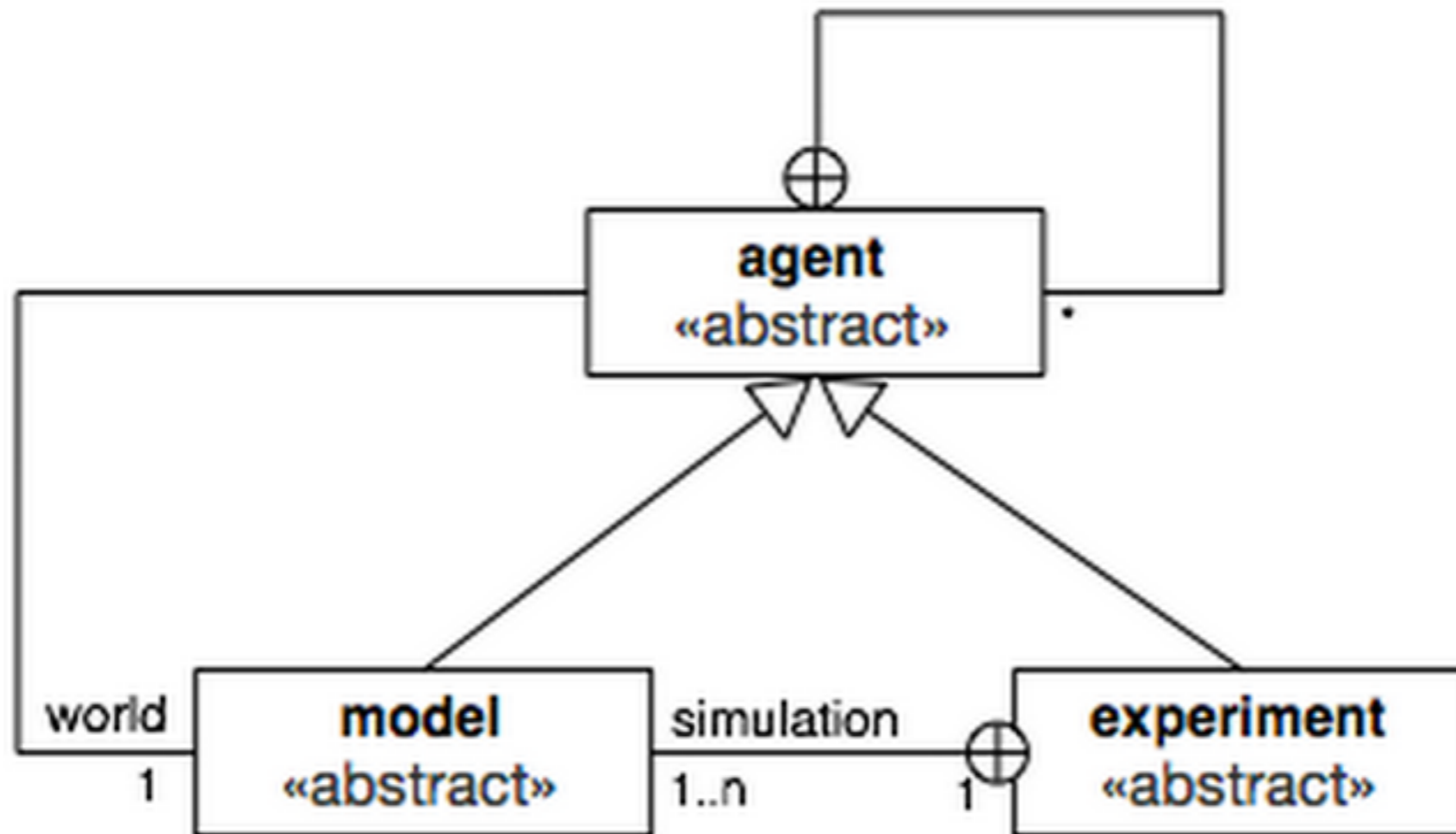


The screenshot displays the GAMA simulation environment. At the top, a progress bar shows '14 cycles elapsed'. The main window features a 3D view of a flock of blue birds flying against a sky background. A red dot in the center of the flock is highlighted with a yellow circle. The interface includes several panels and controls:

- Control Panel (Top):** Contains buttons for 'Run/pause the current simulation', 'Step by step execution', 'Launch a new simulation', and 'Stop simulation'. A slider below these buttons is labeled 'Slows down the execution of the simulation'.
- Parameters Panel (Right):** A 'Parameters' window with a 'General' tab. It lists various simulation parameters such as 'Number of agents' (100), 'Number of obstacles' (5), 'Maximal speed' (15.0), 'Cohesion Factor' (200), 'Alignment Factor' (100), 'Minimal Distance' (10.0), 'Maximal Turn' (90), and 'Width/Height of the Environment' (800). It also includes checkboxes for 'Apply Cohesion?', 'Apply Alignment?', 'Apply Separation?', 'Follow Goal?', 'Apply Avoidance?', 'Apply Wind?', and 'Moving Obstacles?'. A 'Random number generator' is set to 'mer: among (ce)' and a 'Random seed' is set to '0.0'.
- Console View (Bottom Left):** A yellow panel labeled 'Console view' with a scrollable text area.
- Code Editor (Bottom):** A text area showing GAMA code for the 'boids.gaml' model, including 'model boids' and 'global torus: torus\_environment'.
- Navigation Tools (Top Right):** A toolbar with icons for 'Synchronize the simulation and the display', 'Snapshot', 'Zoom in, zoom out, fit the view', and 'Inspect agents'.

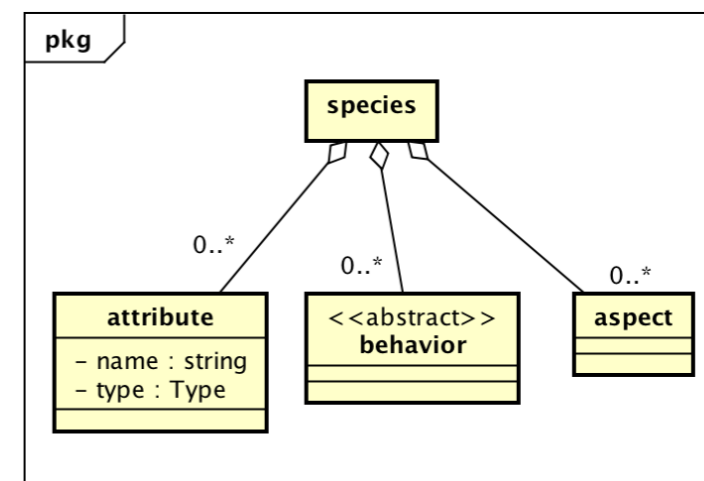
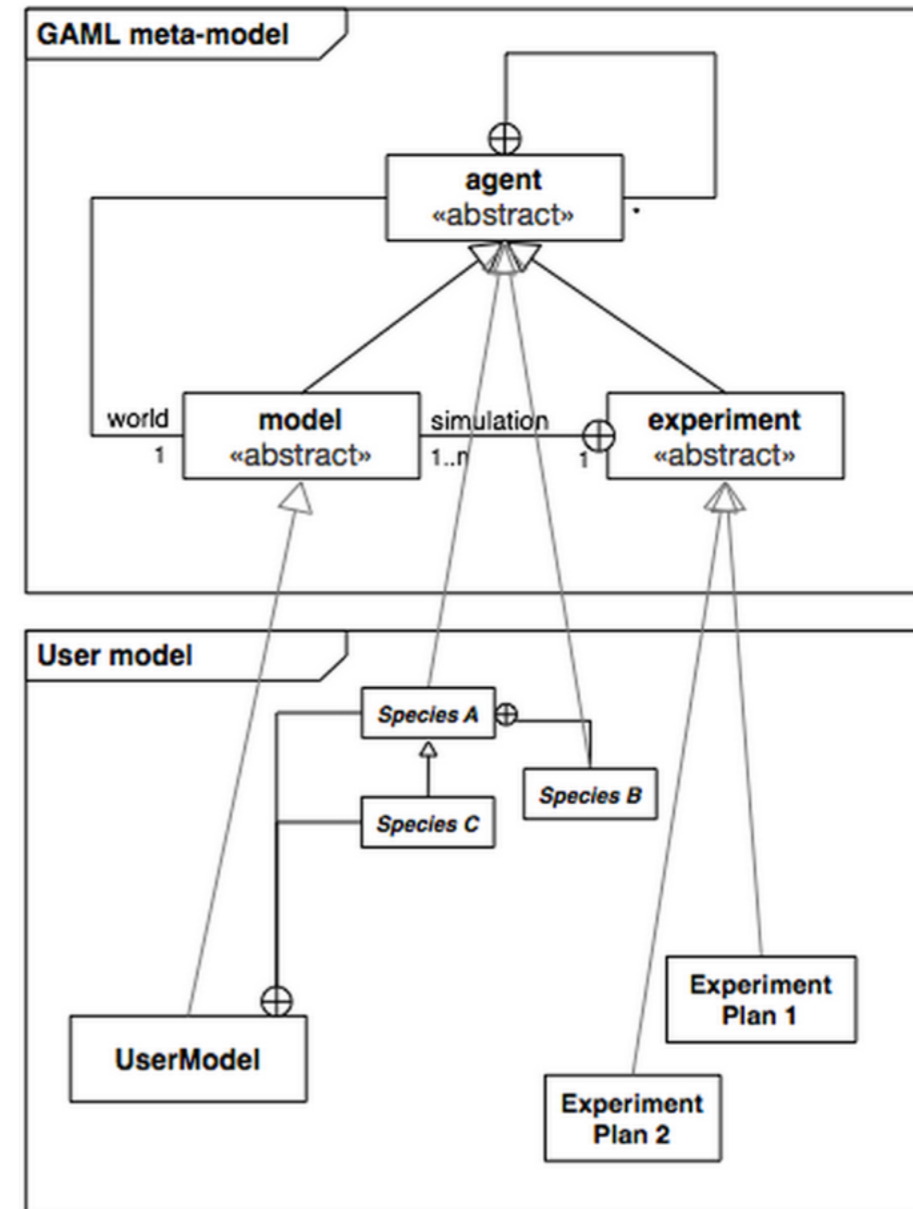
# GAMA metamodel / a framework

► In GAMA: everything is agent!



# Instantiation of GAMA metamodel on a particular model

- ▶ **Model** (a.k.a. global): global variables, actions, dynamics environment and initialization.
- ▶ **Species** (and **Grid**): agent species. A species/grid is a UML class. Several species blocks can be defined.
- ▶ **Experiment** : simulation execution context, in particular inputs and outputs. Several experiment blocks can be defined.



# Implementation of the model

```
model firemen

global { }

grid plot {
  list<plot> neighbors;
  string state;

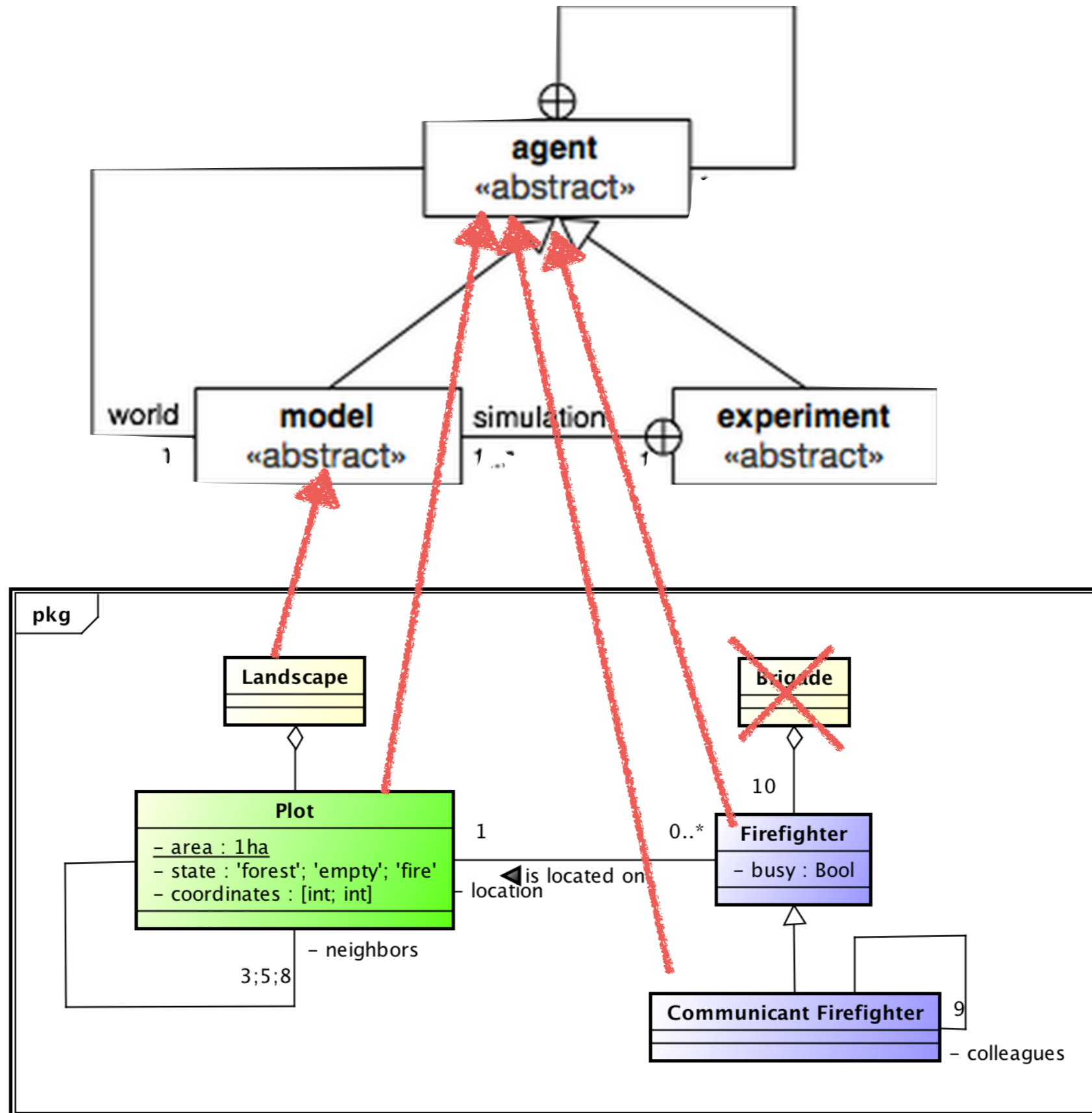
  rgb color;
}

species firefighter {
  bool busy;

  plot my_plot;
}

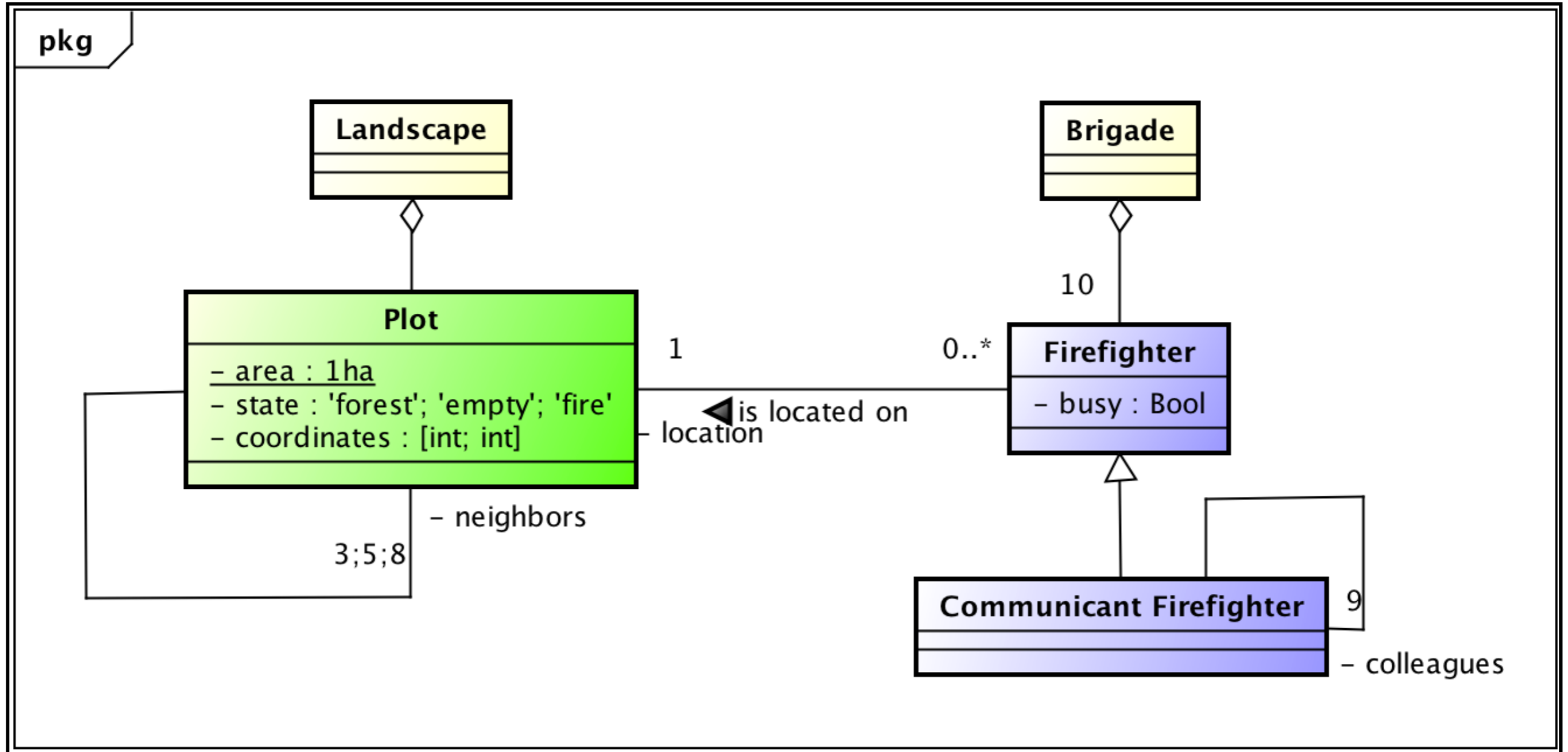
species communicant_firefighter parent:firefighter {
  list<communicant_firefighter> colleagues;
}
```

# Structure: Mapping Firefighter model to GAMA Meta-model

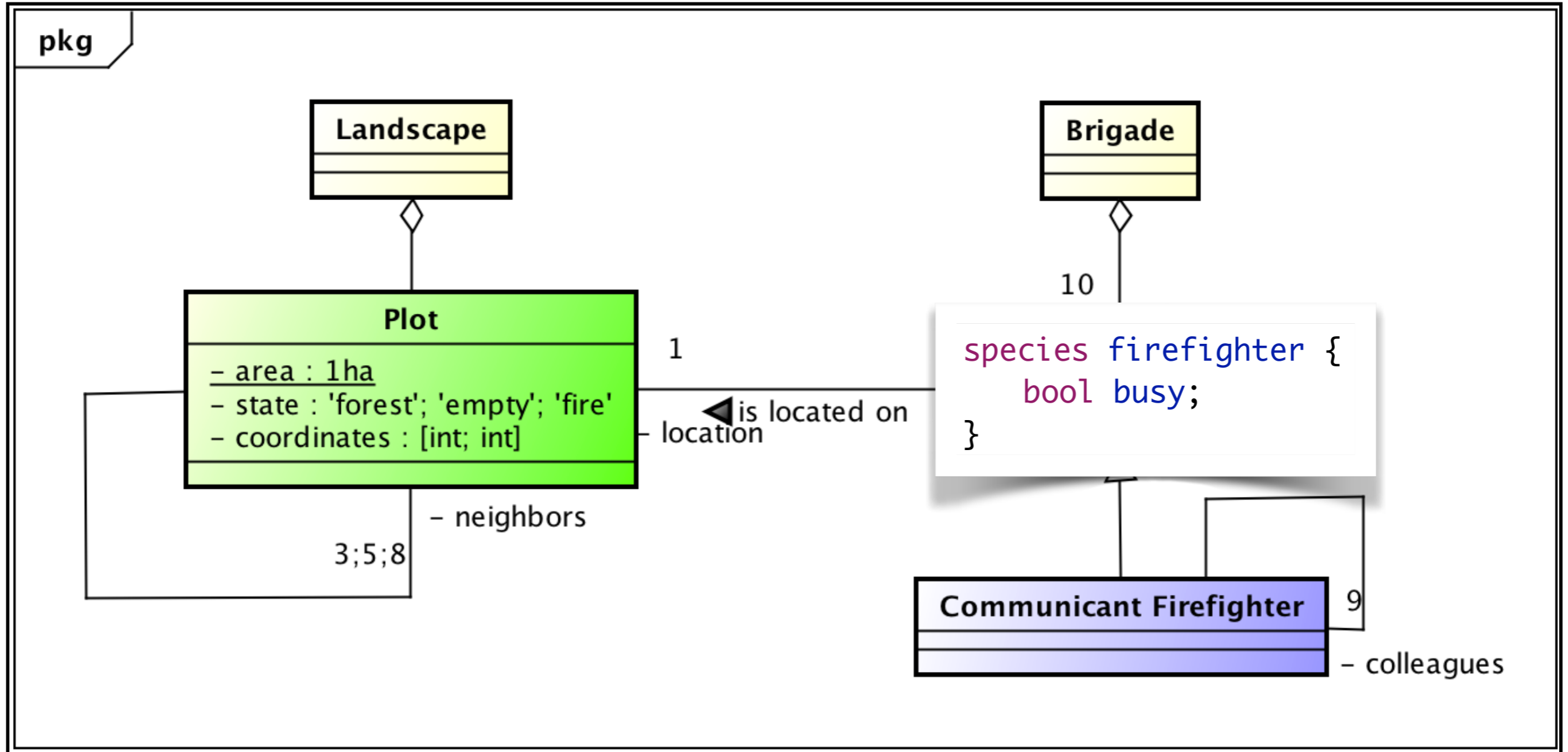




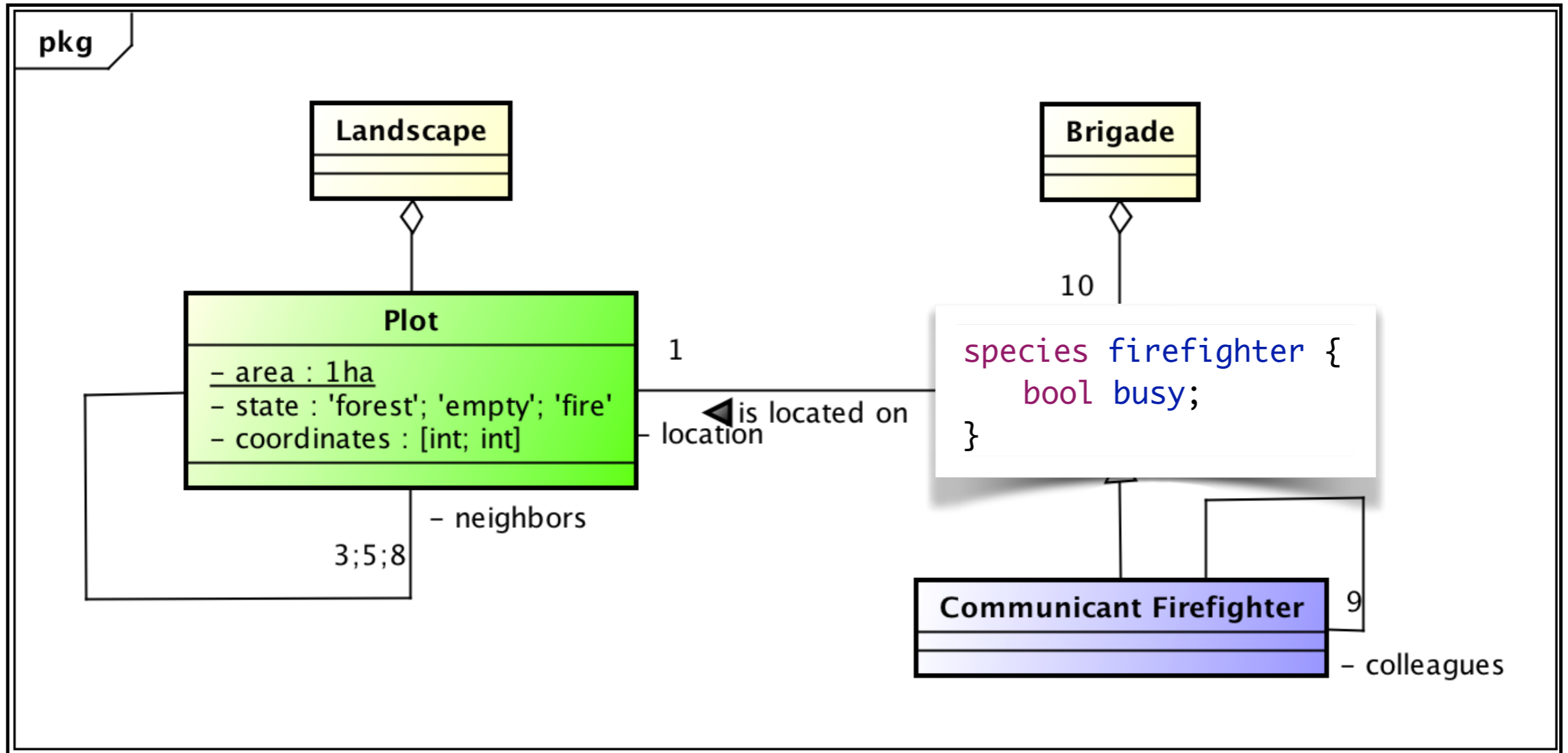
# Structure: Mapping Firefighter model to GAMA Meta-model



# Structure: Mapping Firefighter model to GAMA Meta-model



# Structure: Mapping Firefighter model to GAMA Meta-model



- Create a first GAMA model with the model structure representing this UML class diagram.

# Notes on the model.

- ▶ Every kind of agent has **built-in attributes**:
  - ▶ name (a string)
  - ▶ shape (a geometry) (default value = a point)
  - ▶ location (a point) (value = the centroid of its shape)
- ▶ In addition, **grid** agents have additional built-in attributes:
  - ▶ grid\_x (an integer)
  - ▶ grid\_y (an integer)
  - ▶ color (a color)
  - ▶ grid\_value (used when grid is created from a data file)

```
model firemen

global { }

grid plot {
  list<plot> neighbors;
  string state;

  rgb color;
}

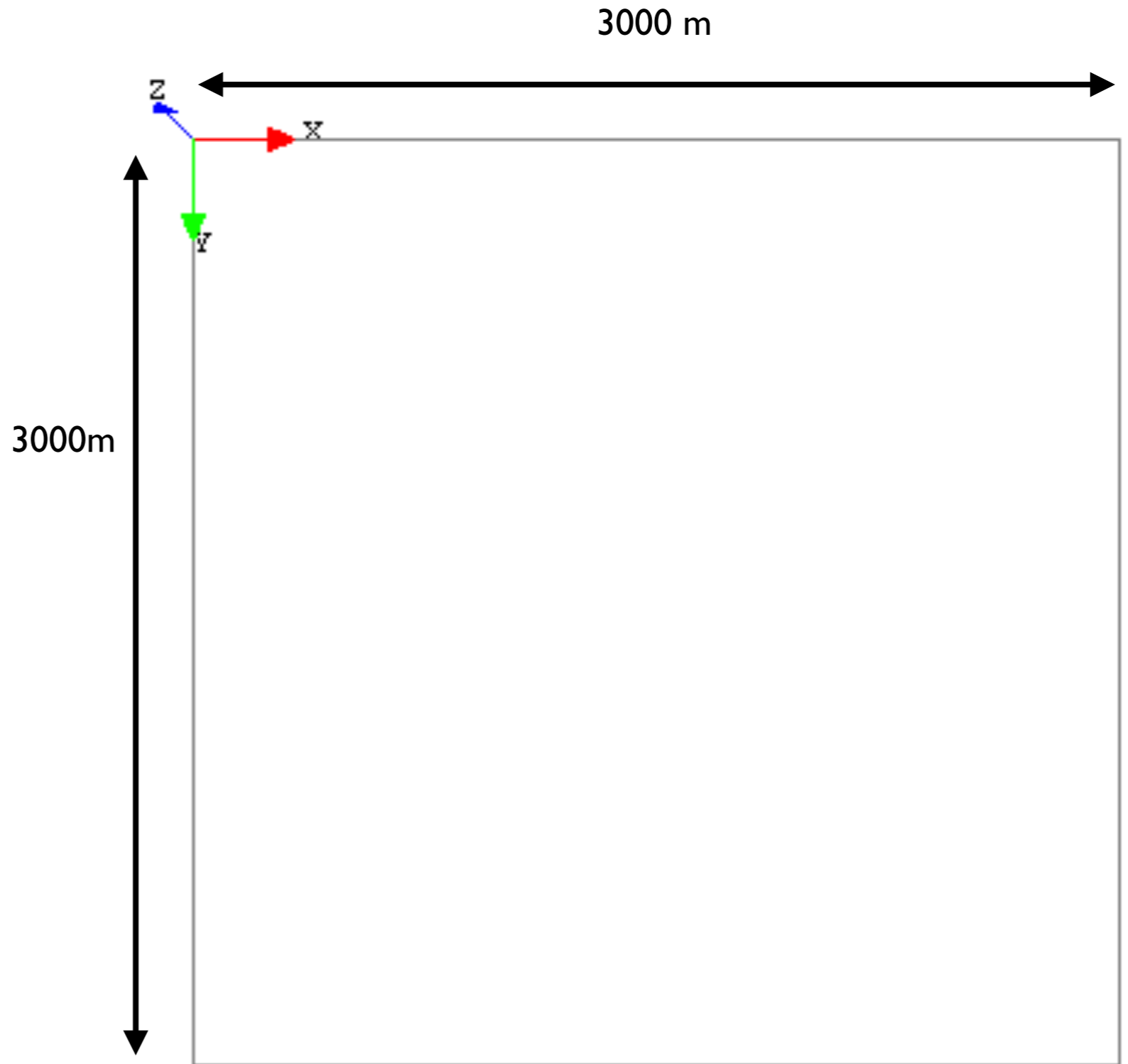
species firefighter {
  bool busy;

  plot my_plot;
}

species communicant_firefighter parent:firefighter {
  list<communicant_firefighter> colleagues;
}
```

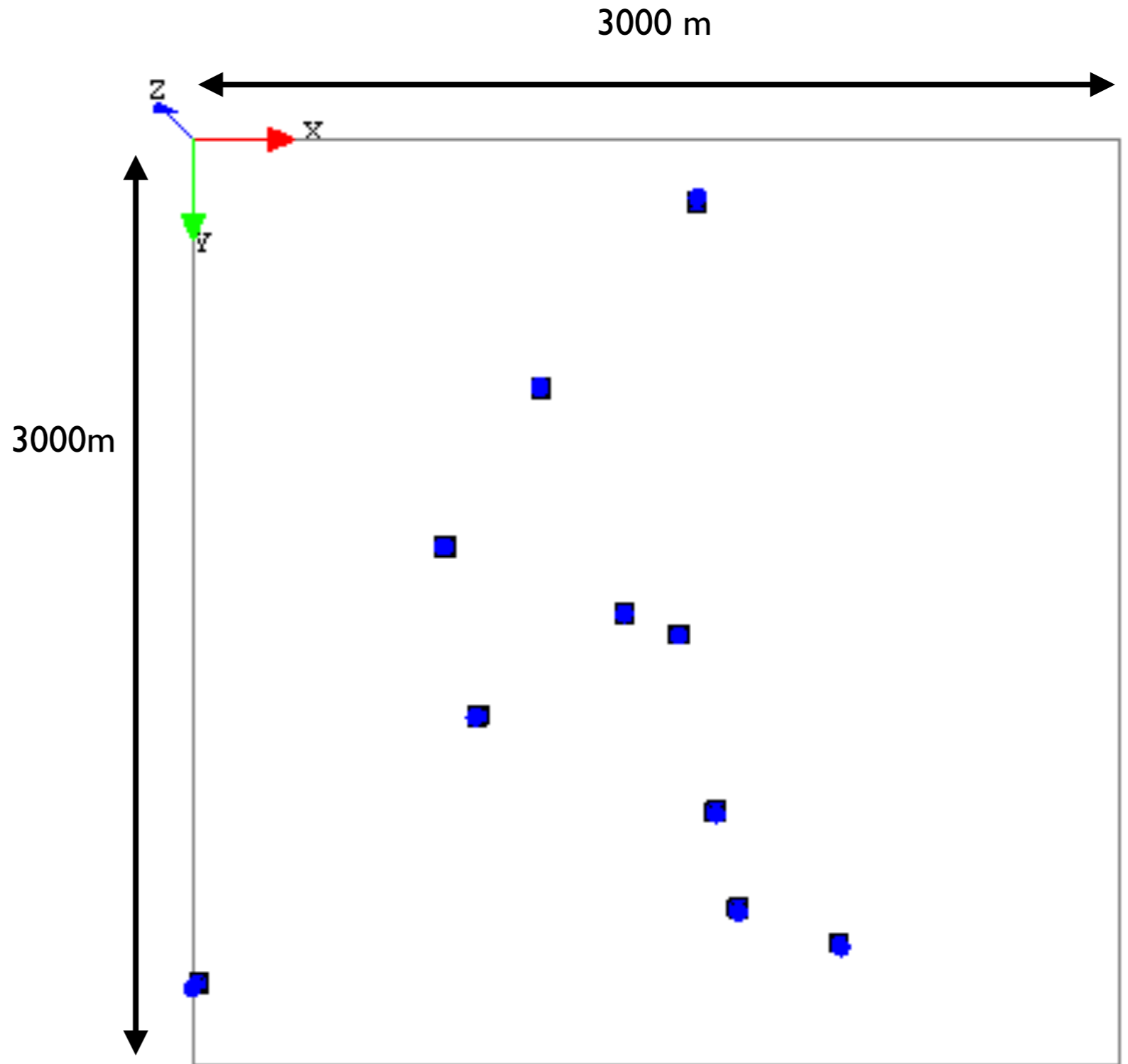
# Space in GAMA

- ▶ In GAMA, agents have a **location in a reference continuous space.**
- ▶ To create a grid of **cells**, we need to create explicitly a new species with a particular **spatial organisation** (a particular topology).



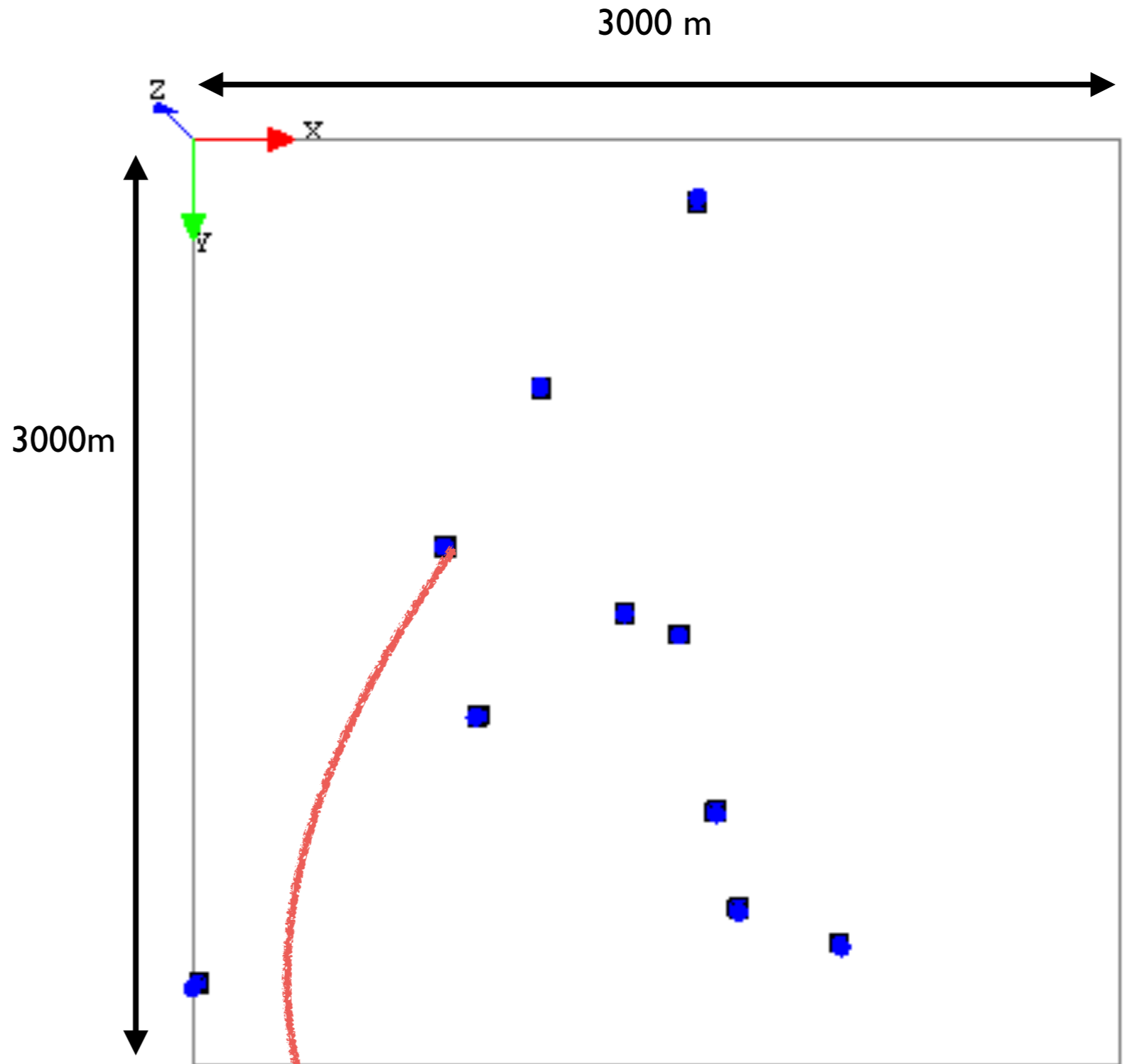
# Space in GAMA

- ▶ In GAMA, agents have a **location in a reference continuous space.**
- ▶ To create a grid of **cells**, we need to create explicitly a new species with a particular **spatial organisation** (a particular topology).



# Space in GAMA

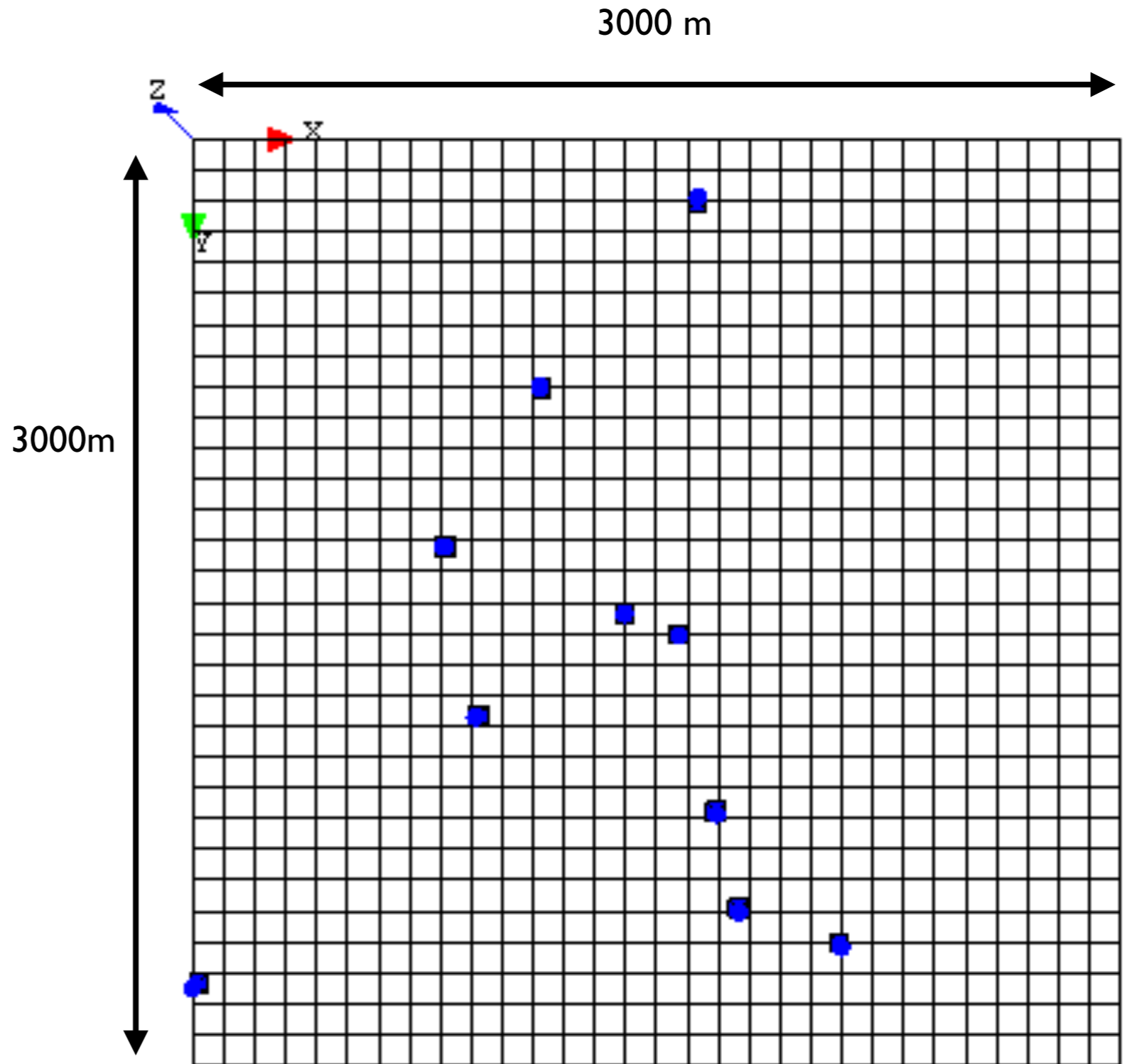
- ▶ In GAMA, agents have a **location** in a **reference continuous space**.
- ▶ To create a grid of **cells**, we need to create explicitly a new species with a particular **spatial organisation** (a particular topology).



```
name = "communicant_firefighter2"  
location = {1400.1, 44.02, 0.0}  
shape = {27.198, 44.02, 0.0}
```

# Space in GAMA

- ▶ In GAMA, agents have a **location** in a **reference continuous space**.
- ▶ To create a grid of **cells**, we need to create explicitly a new species with a particular **spatial organisation** (a particular topology).

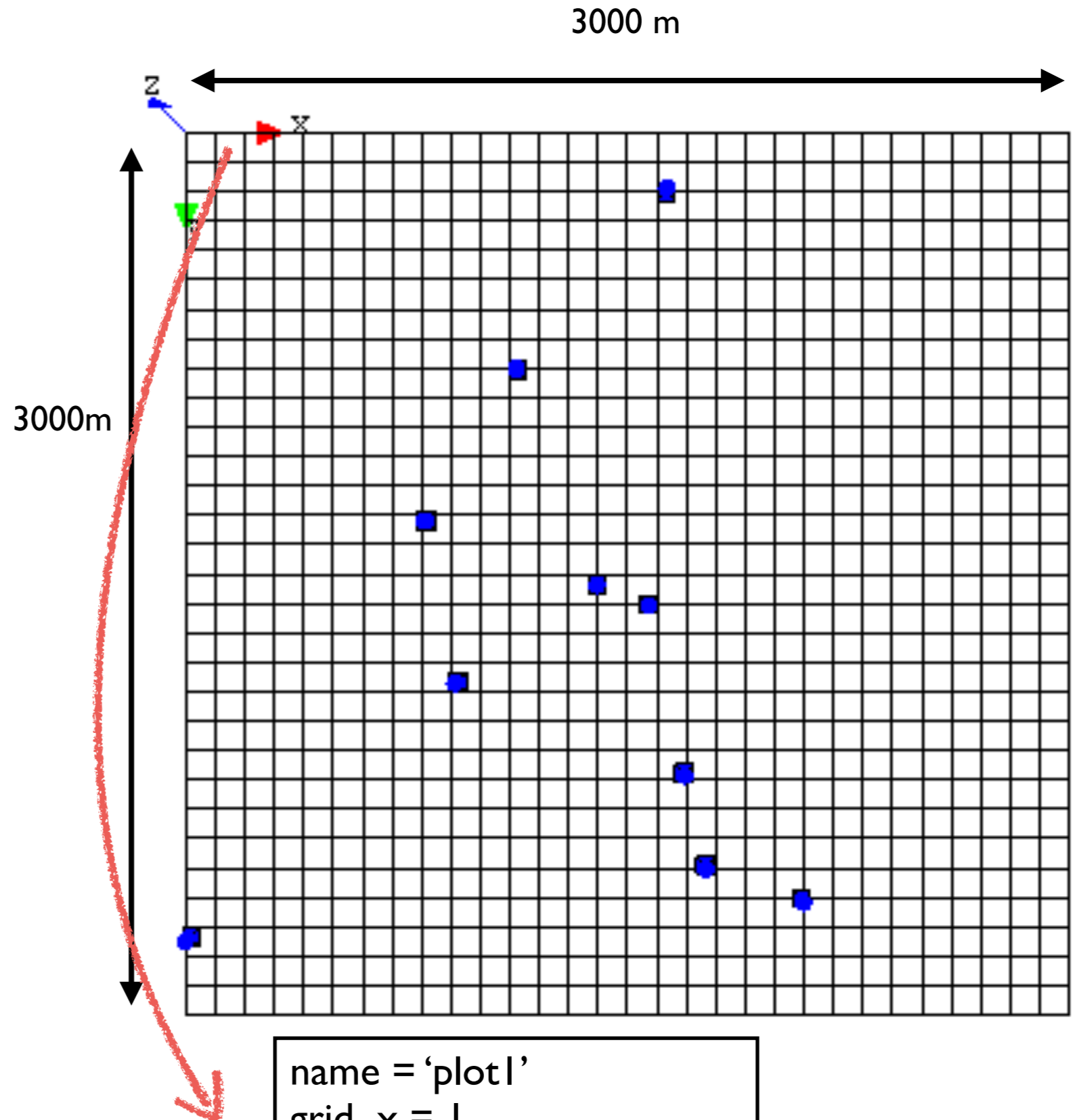


Addition of a 30x30 grid



# Space in GAMA

- ▶ In GAMA, agents have a **location** in a **reference continuous space**.
- ▶ To create a grid of **cells**, we need to create explicitly a new species with a particular **spatial organisation** (a particular topology).



```
name = 'plot1'  
grid_x = 1  
grid_y = 0  
location = {155.0, 50.0, 0.0}  
shape = a square
```

Addition of a 30x30 grid

# Firefighter model - Initialization

## ▶ Initialization of the global:

- ▶ Add a global variable to choose if we will play scenario 1 or 2.
- ▶ creation of agents
- ▶ Initialization of the environment size

## ▶ Initialization of plot agents:

- ▶ create 900 plots of 1ha, i.e. 30x30 plots
- ▶ Setting randomly 50% patches to forest / green and 50% patches to clear / white

## ▶ Initialization of firemen

- ▶ create 10 firemen randomly located, depending of the chosen scenario

## ▶ Setting fire:

- ▶ chose 1 plot

# Firefighter model - Initialization

The screenshot shows the GAMA simulation environment. The main window displays a grid-based environment with green obstacles and a red fire source. Blue circles represent firefighter agents. The interface includes a 'Simulation ready' status bar, a 'Gama Projects' sidebar, a 'Parameters' panel, and a progress indicator at the bottom.

**Simulation - Gama**

Simulation ready

Gama Projects

- Models library (4)
- Shared models (0)
- User models (10)

vizu

Parameters

Model firemen\_model Parameters for experiment 'myFirstVizu'

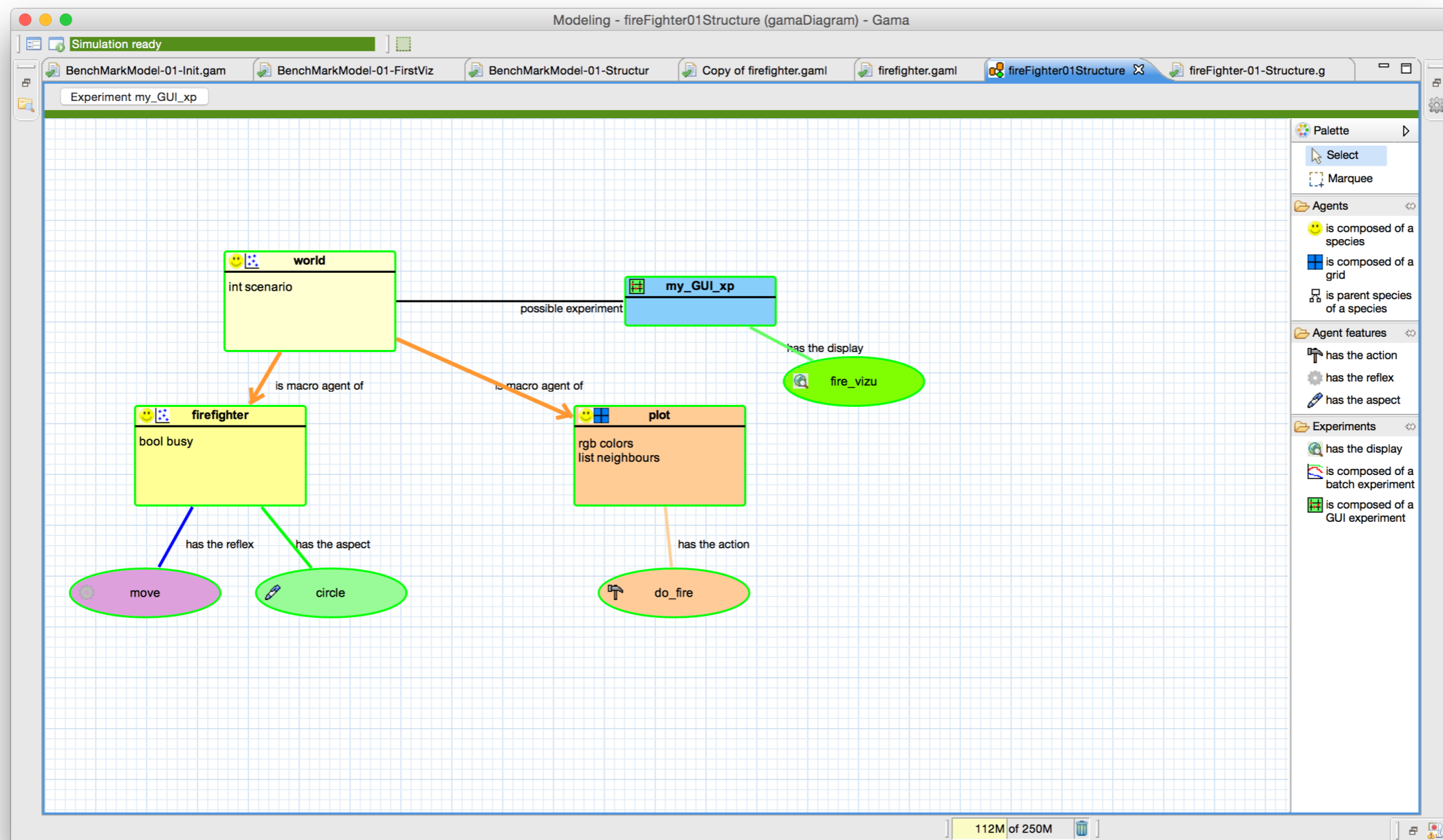
Random number generator 'mersenne' among [cell]

Random seed  Define: [ ]

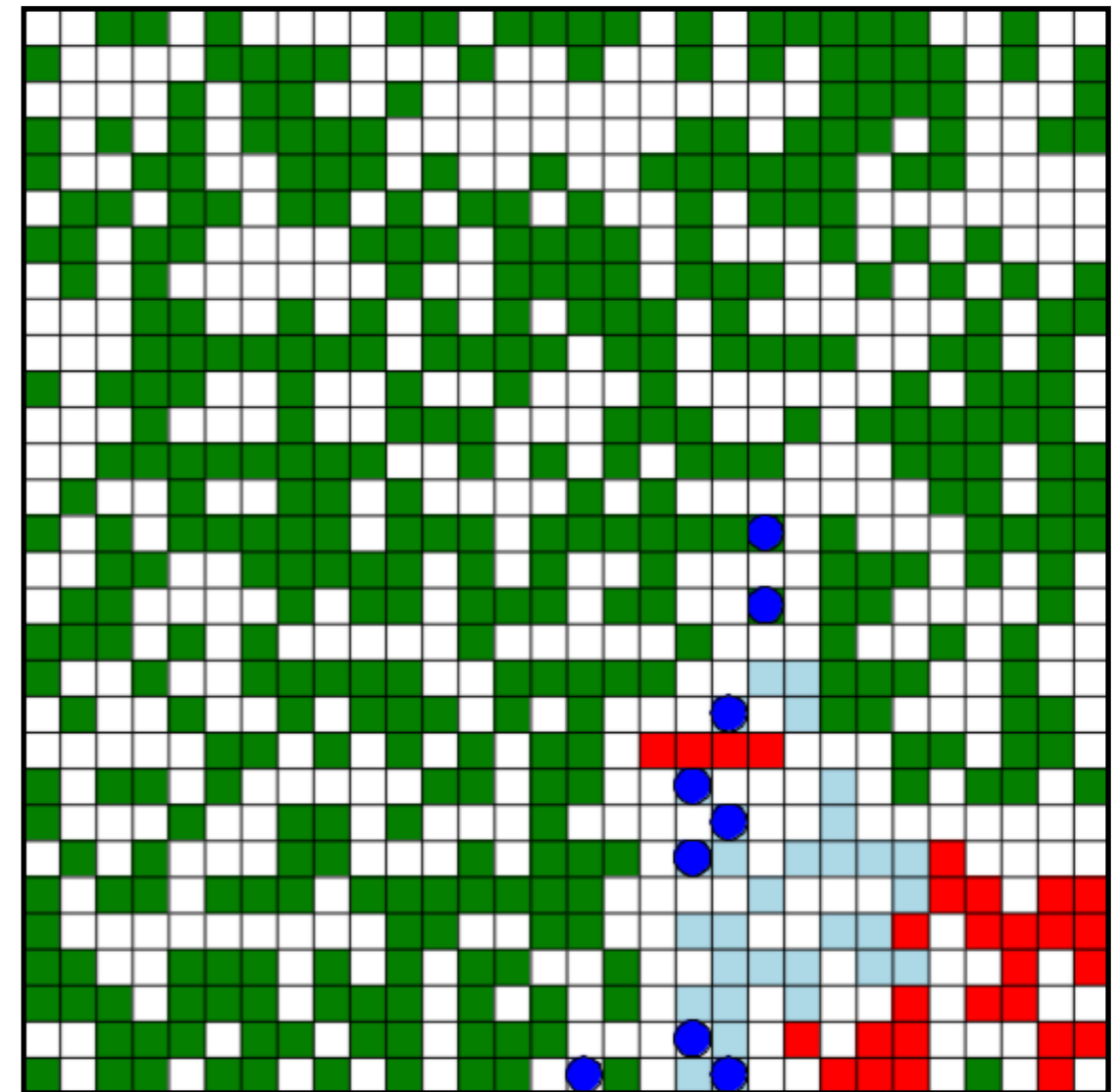
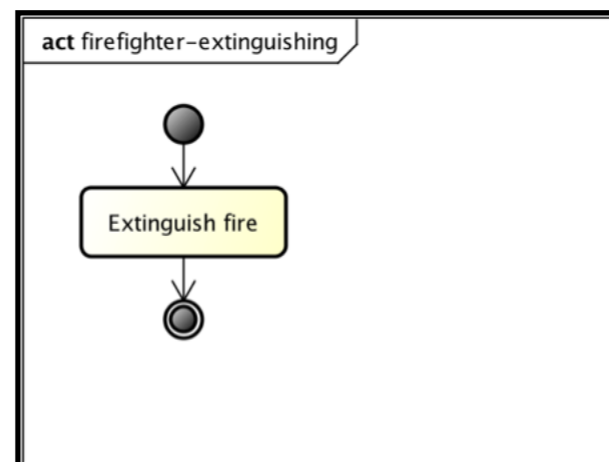
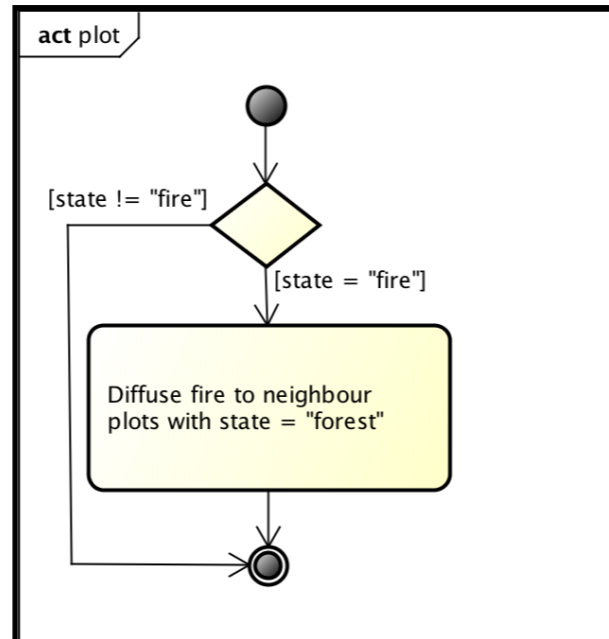
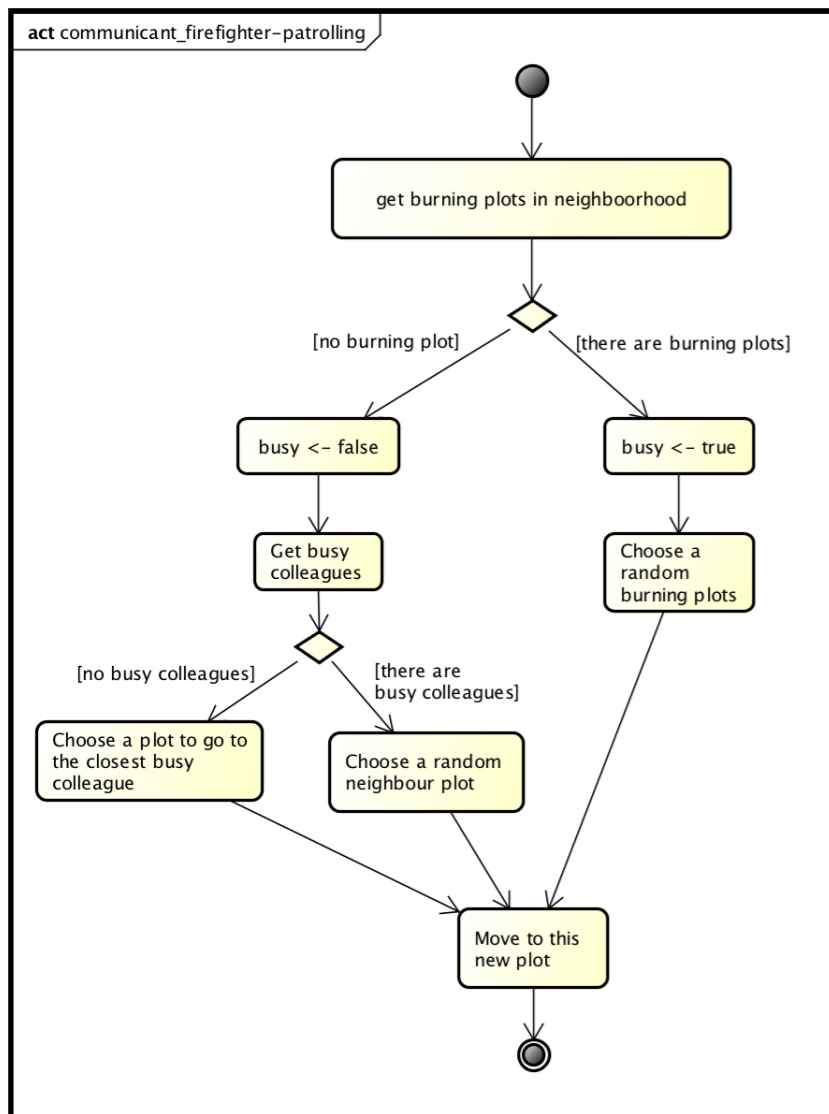
201M of 250M

# Do it with the Graphical Modeling plugin (still a beta version).

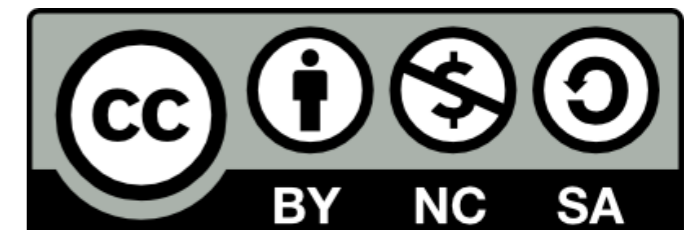
- ▶ GAMA: install extension Graphical Modelling
- ▶ Create new diagram
- ▶ Generate GAML model



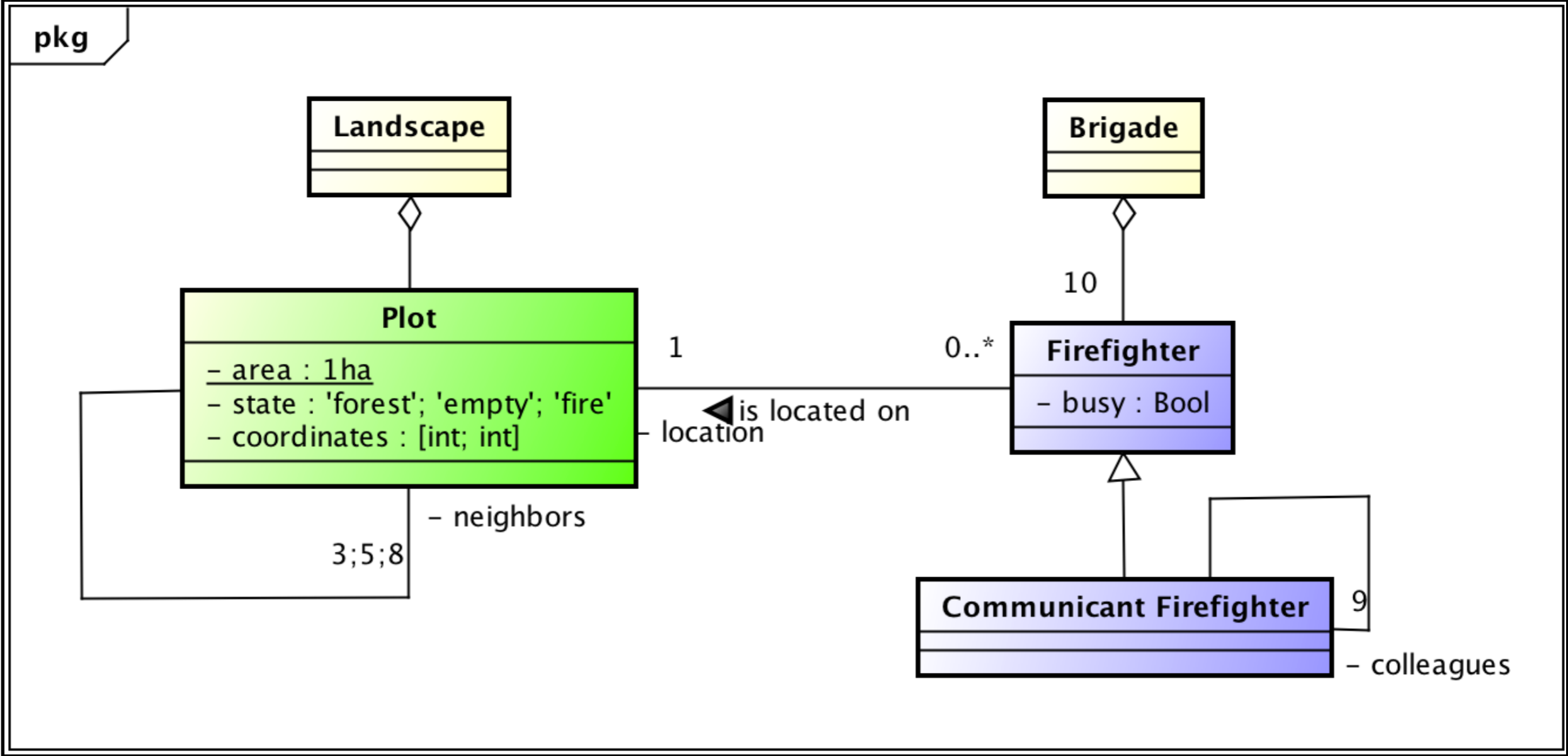
# Firefighter model : Dynamics



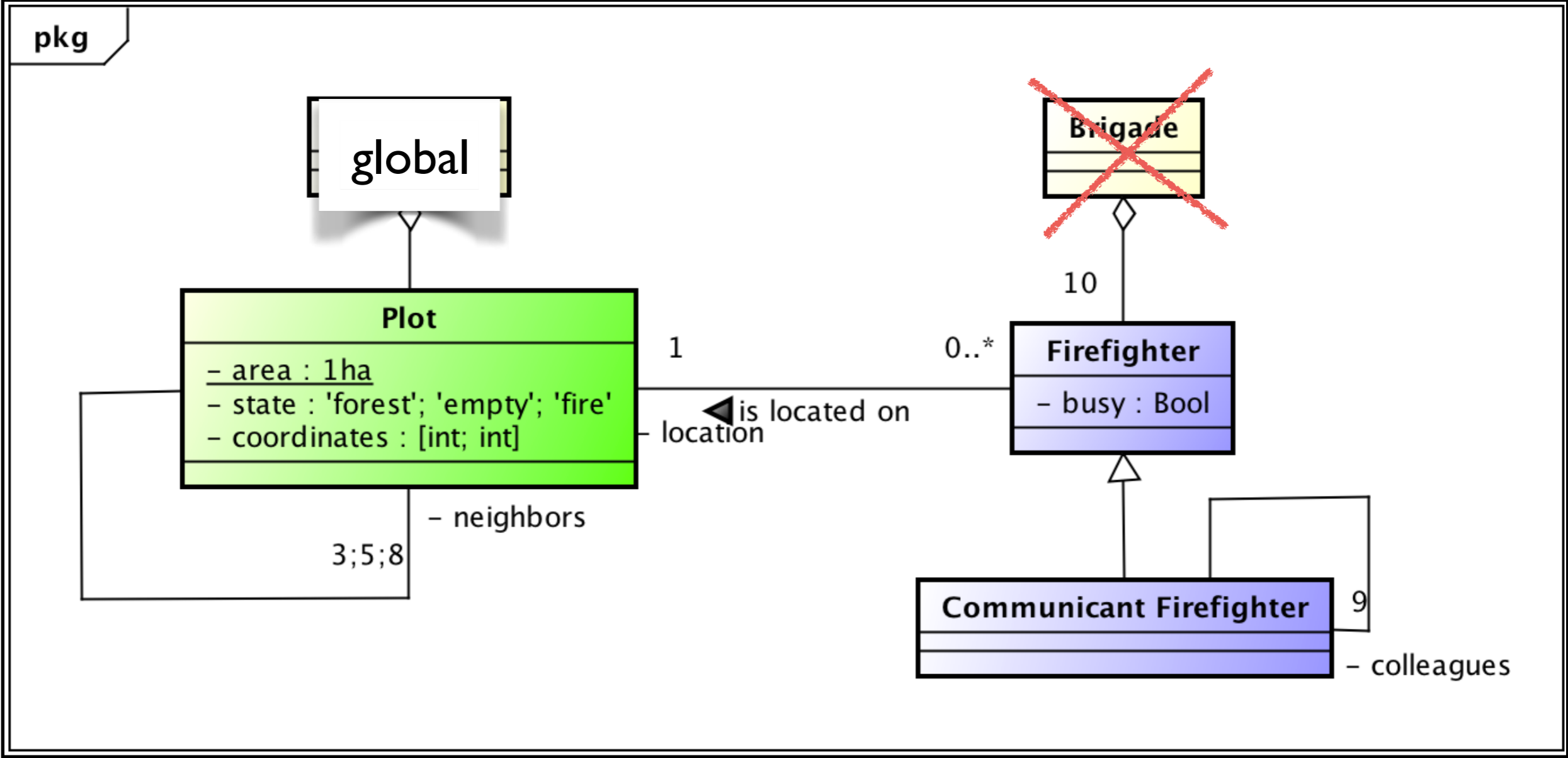
Benoit Gaudou (Univ. Toulouse 1)  
Patrick Taillandier (INRAE)



# Reminder



# Reminder



# Firefighter model - Result of initialization

The screenshot displays the GAMA simulation environment. The main window, titled "Simulation - Gama", shows a 2D grid with green and white cells. A red cell is located at the top left, representing a fire source. Several blue circles are scattered across the grid, representing firefighters. The interface includes a "Simulation ready" status bar, a "Gama Projects" sidebar with "Models library (4)", "Shared models (0)", and "User models (10)", and a "Parameters" panel on the right. The parameters panel shows "Model firemen\_model Parameters for experiment 'myFirstVizu'" with "Random number generator" set to "mersenne" and "Random seed" checked with "Define:".

Simulation - Gama

Simulation ready

Gama Projects

- Models library (4)
- Shared models (0)
- User models (10)

vizu

Parameters

Model firemen\_model Parameters for experiment 'myFirstVizu'

Random number generator 'mersenne' among [cell]

Random seed  Define: [ ]

201M of 250M



# Scheduling: at each simulation step, GAMA executes agents in the following order.

**For** the world agent,  
its behaviors (e.g. its reflexes, in the  
order)

**For each** species *s* **Do**

**For each** agent *a* **of** *s* **Do**

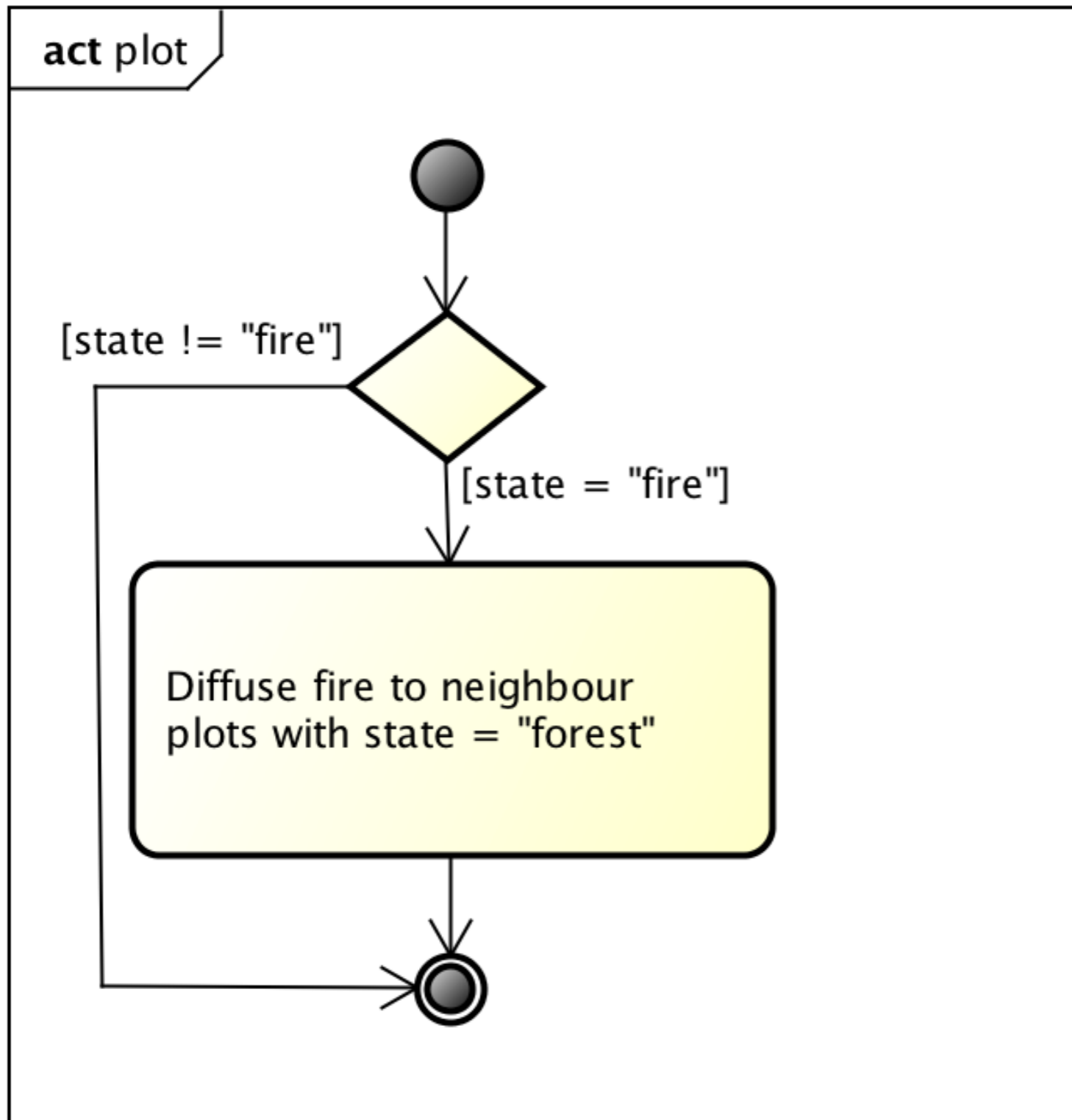
its behaviors (e.g. its reflexes, in the  
order)

**For each** agent of grid **Do**

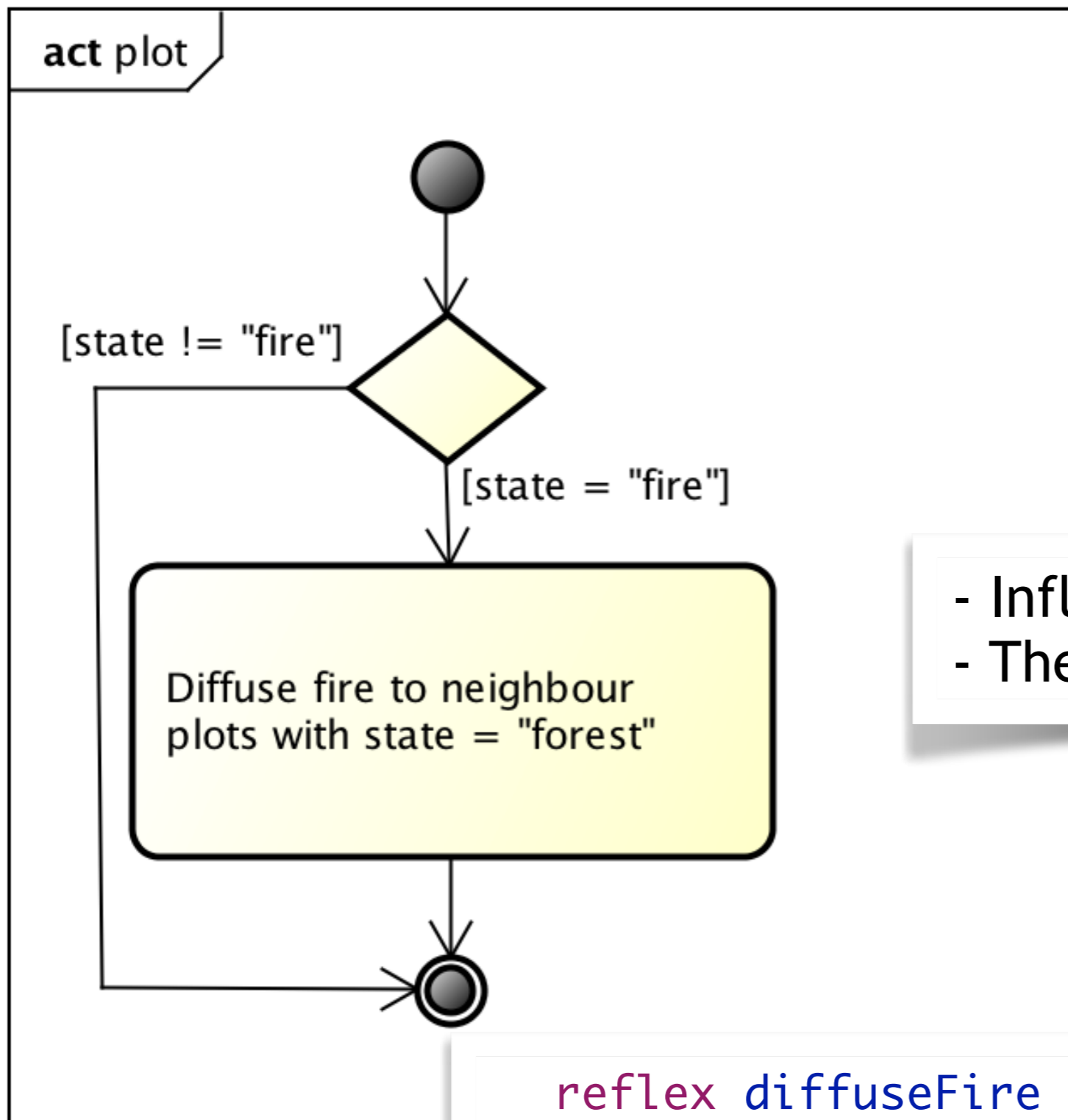
its behaviors (e.g. its reflexes, in the  
order)

The scheduler can be modified at hand

# Dynamics of plots



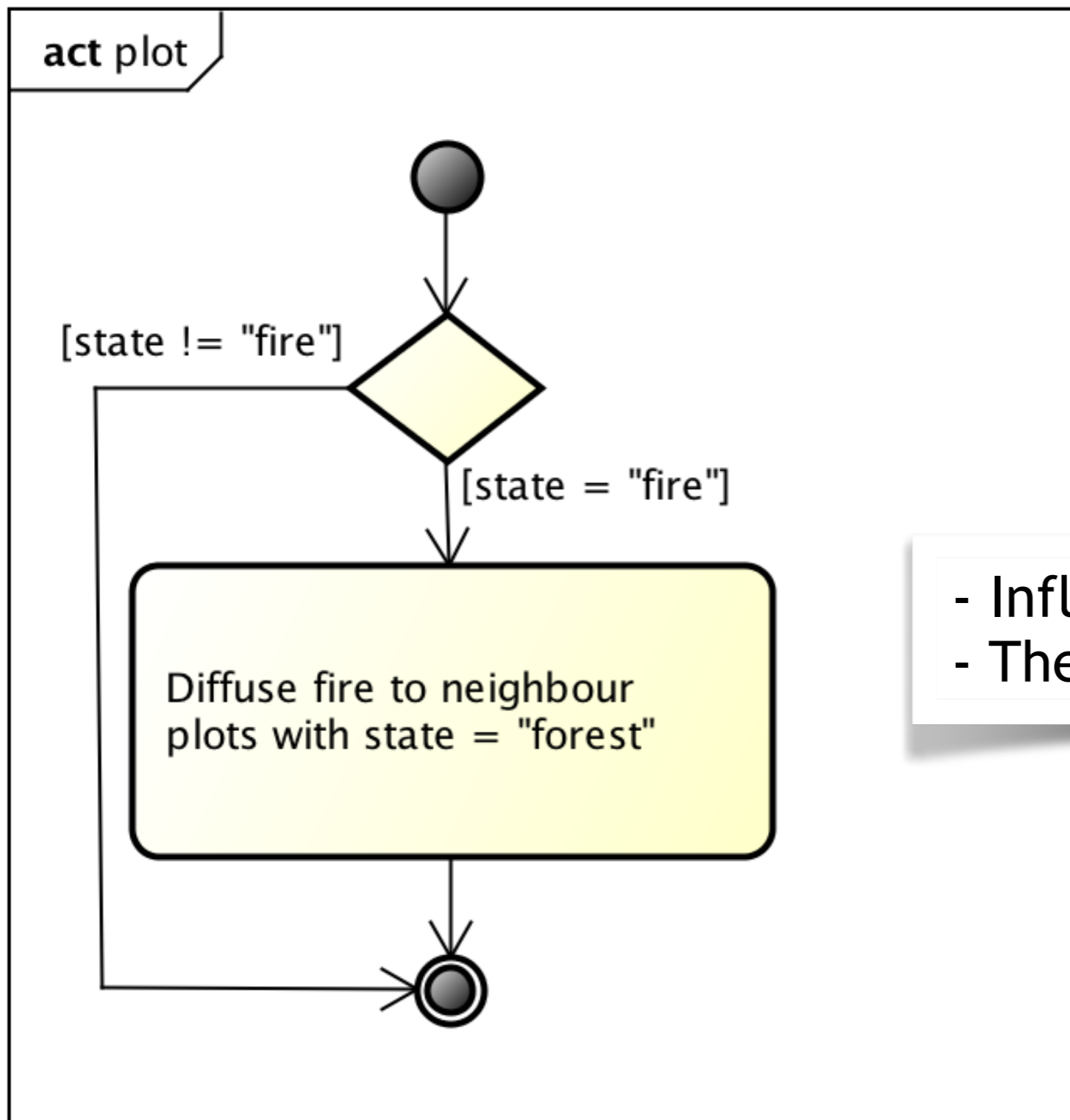
# Dynamics of plots



- Influence of the number of neighbors
- The diffusion is too fast!

```
reflex diffuseFire when: (state = "fire") {
  ask (neighbors where (each.state = "forest")) {
    state <- "fire" ;
    color <- #red ;
  }
}
```

# Dynamics of plots

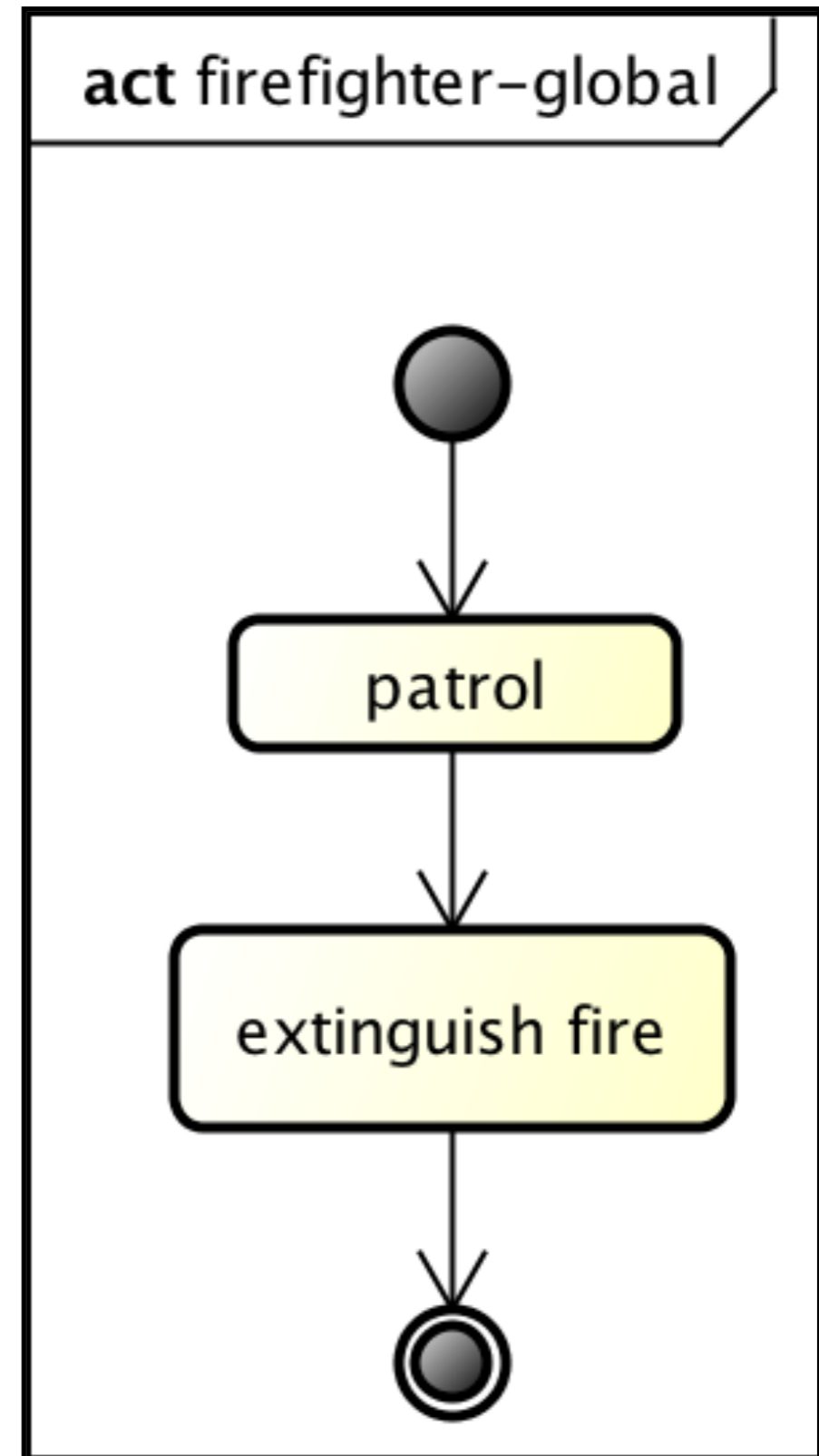


- Influence of the number of neighbors
- The diffusion is too fast!

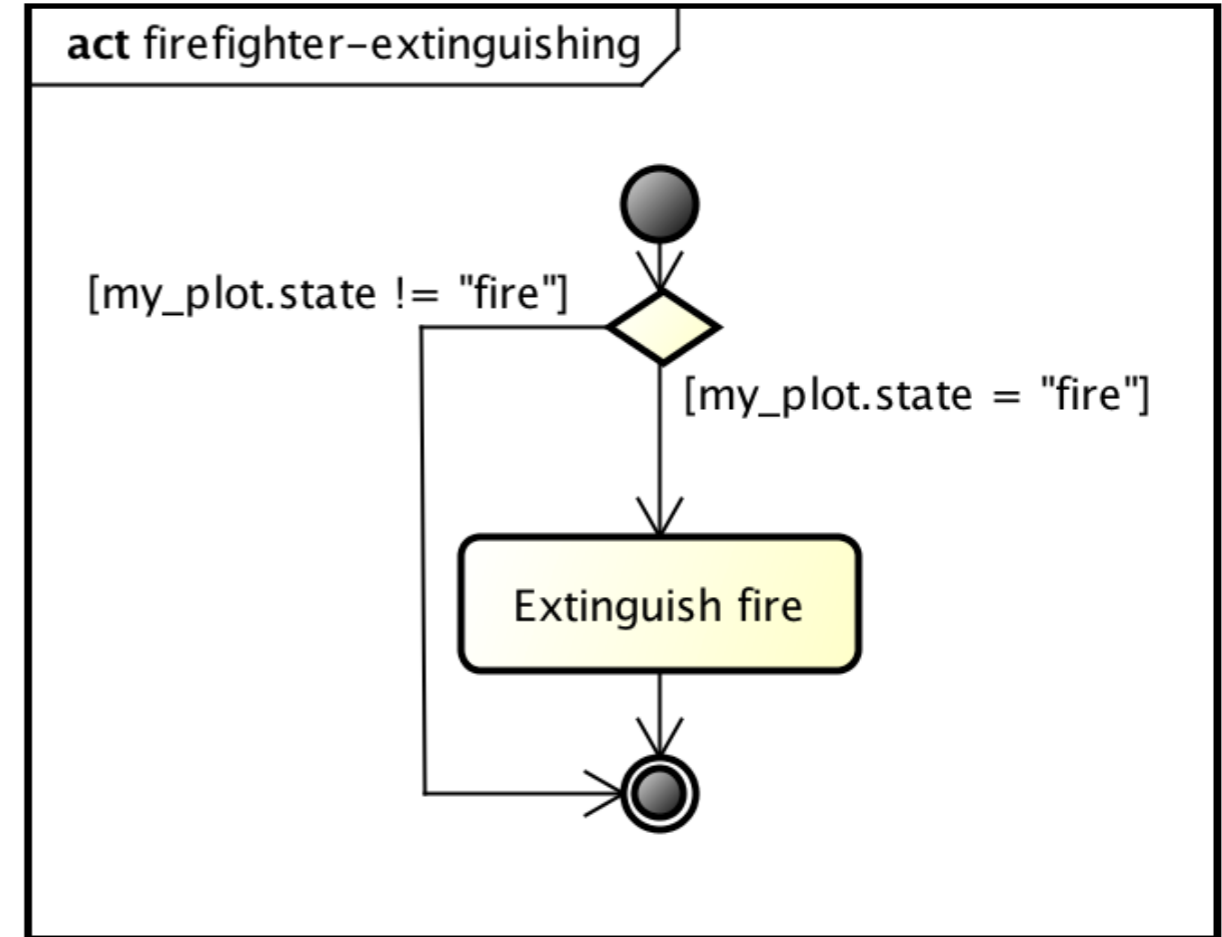
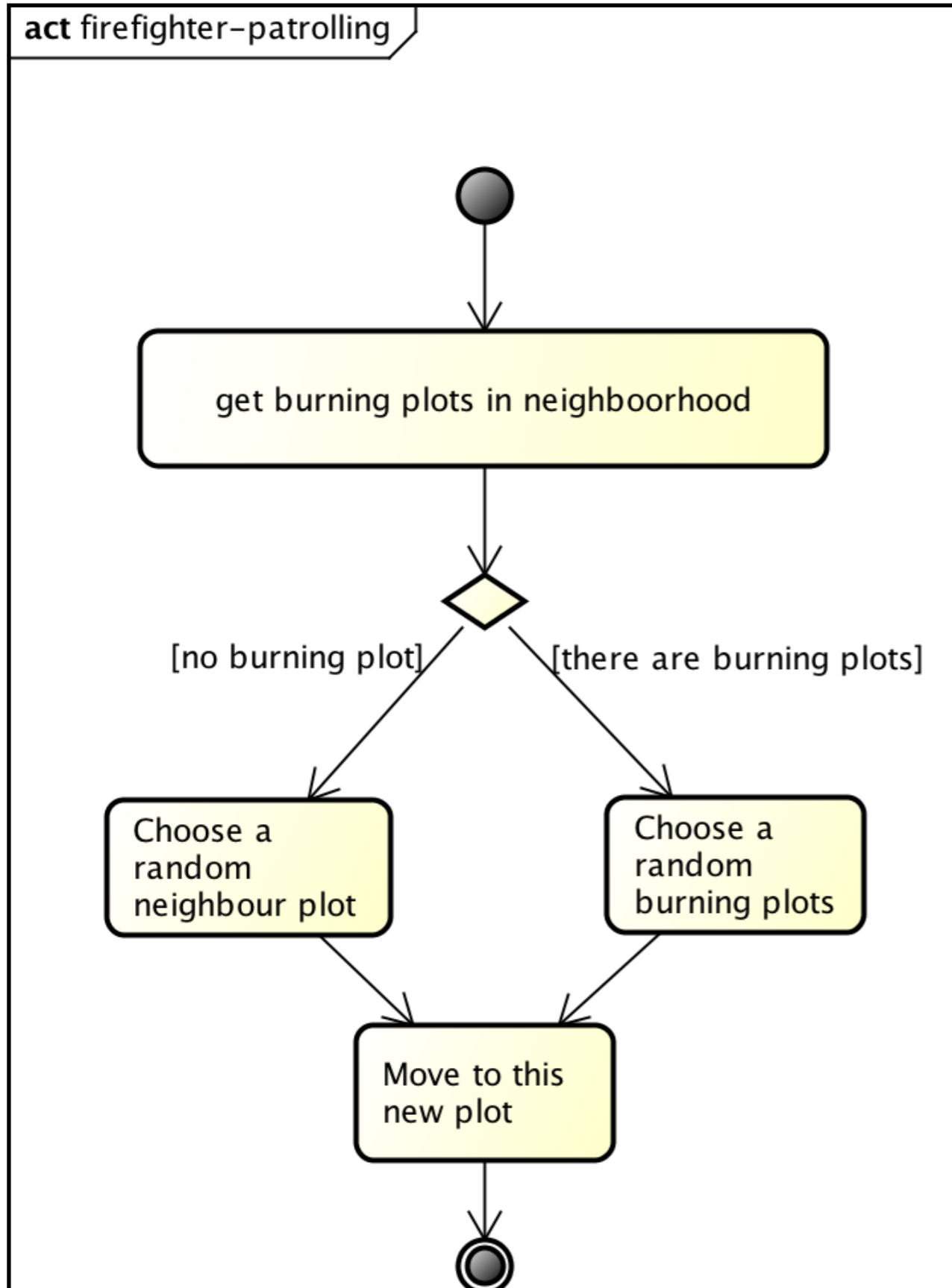
-> Change the scheduling of the plots (schedules facet) to choose at the beginning of the step only plots which are on fire.

# Dynamics of firefighters

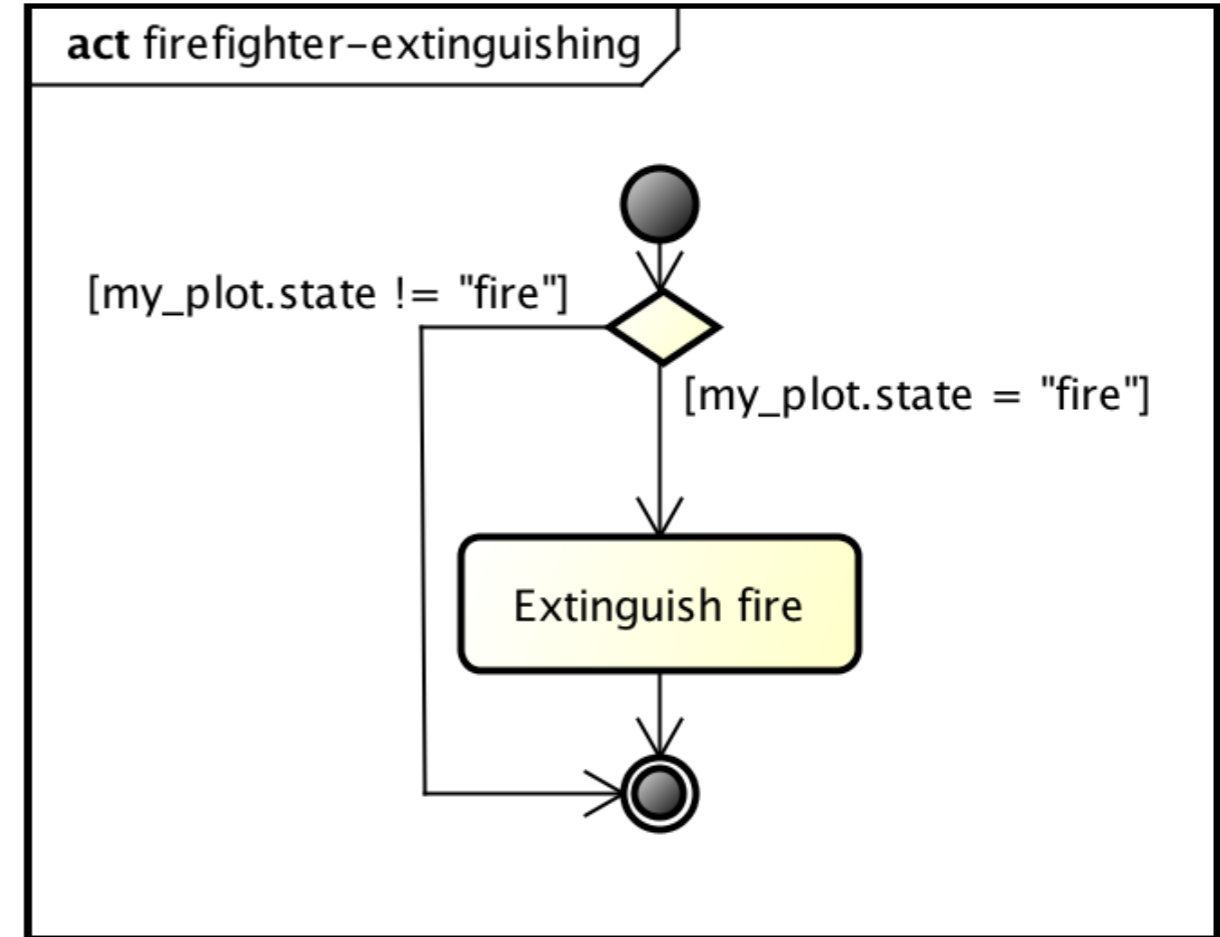
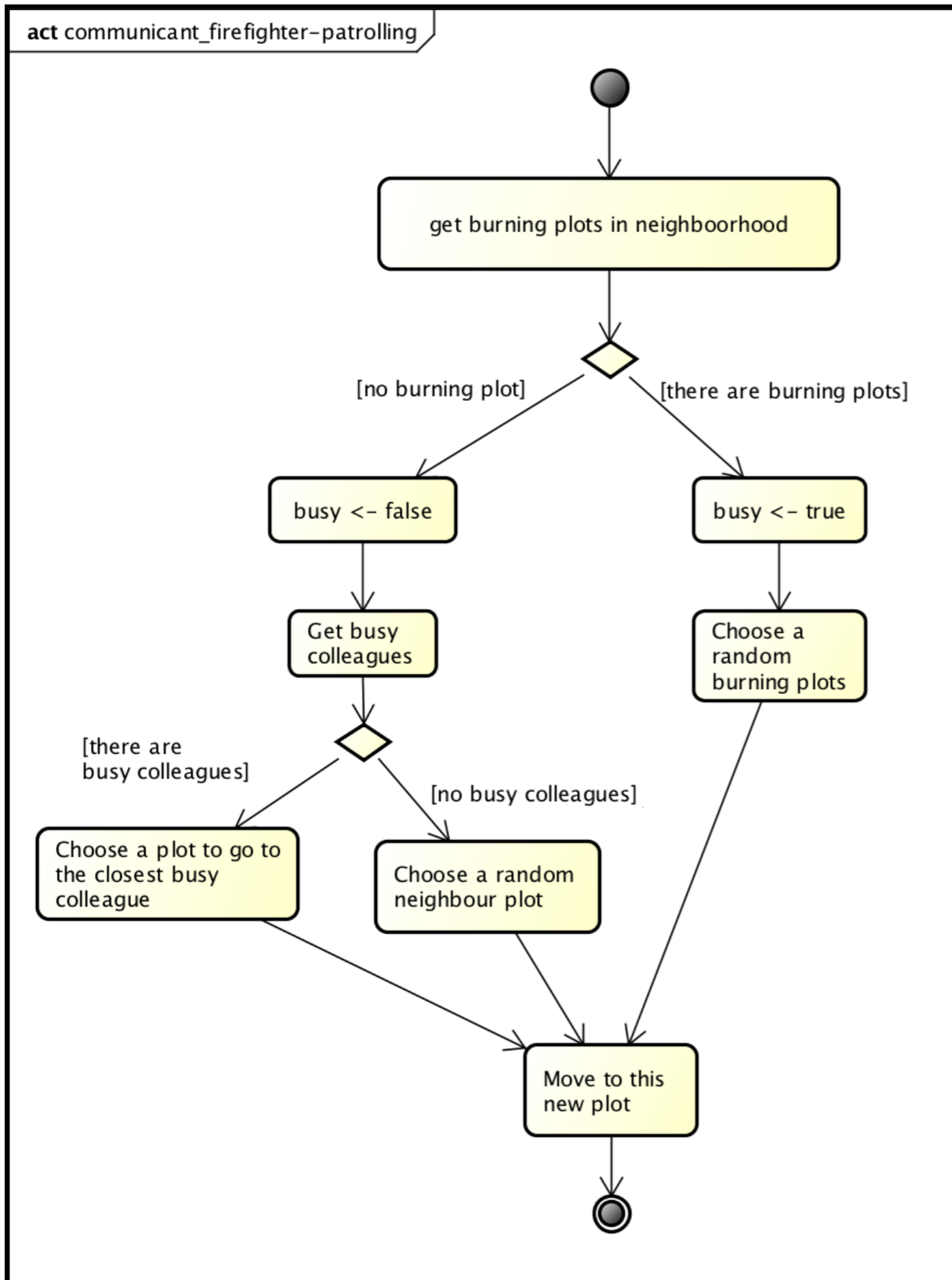
- ▶ Implement the following diagram of the firefighters' behaviours.
- ▶ You can implement patrol and extinguish fire as 2 reflexes (see next slide).



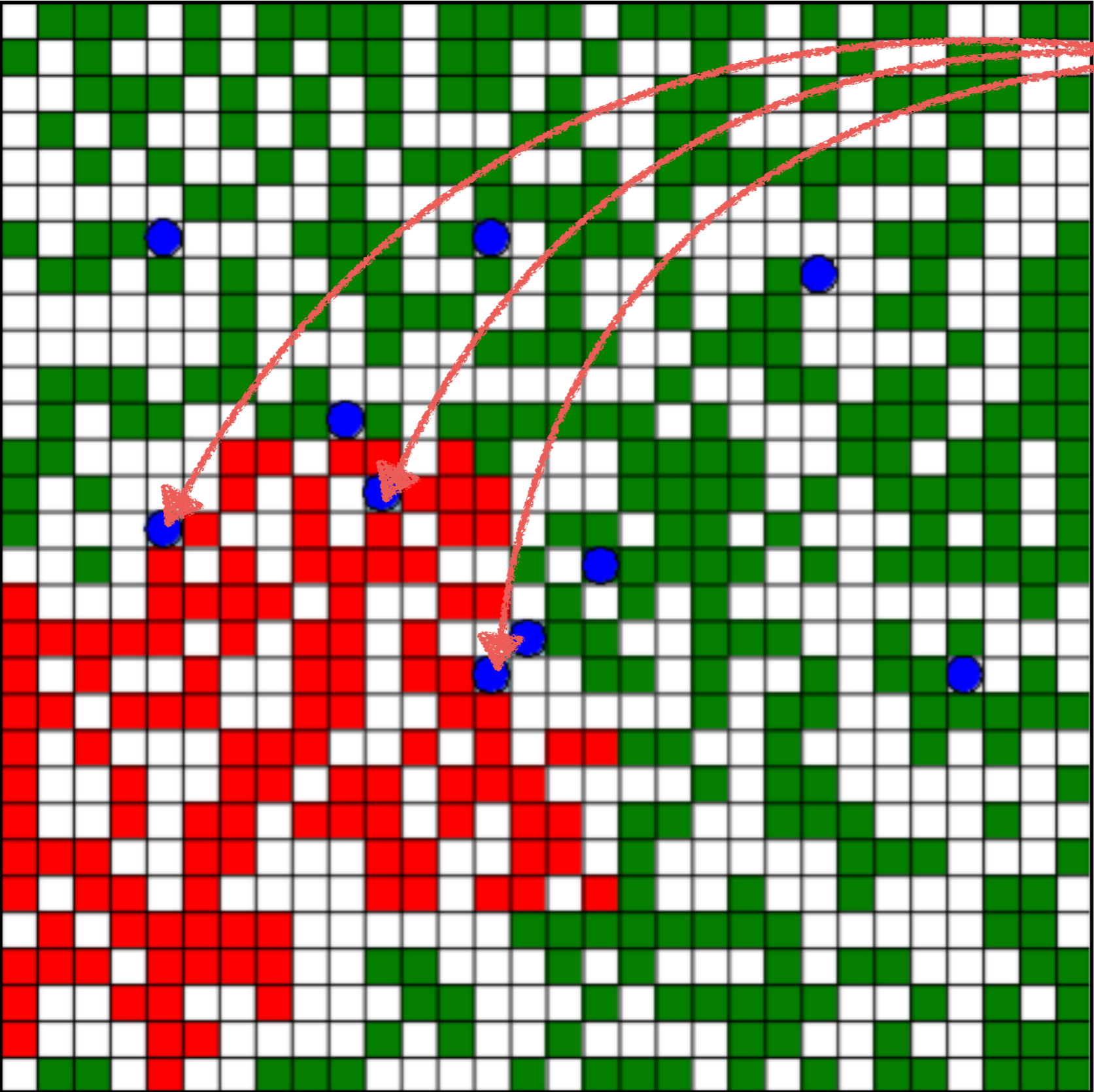
# Dynamics of firefighters



# Dynamics of communicant firefighters



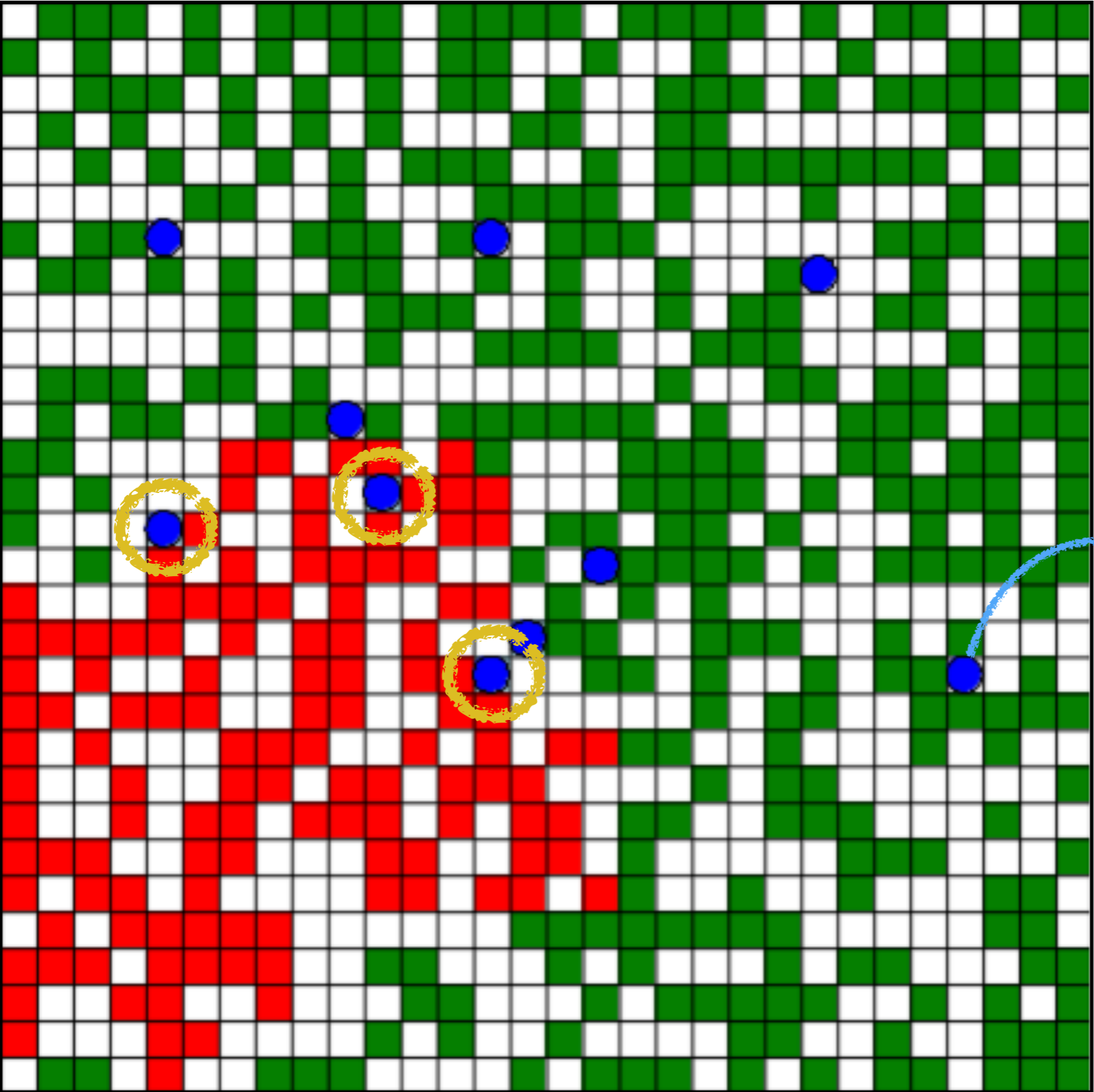
# Choose a plot toward the closest busy firefighter



Busy firefighters

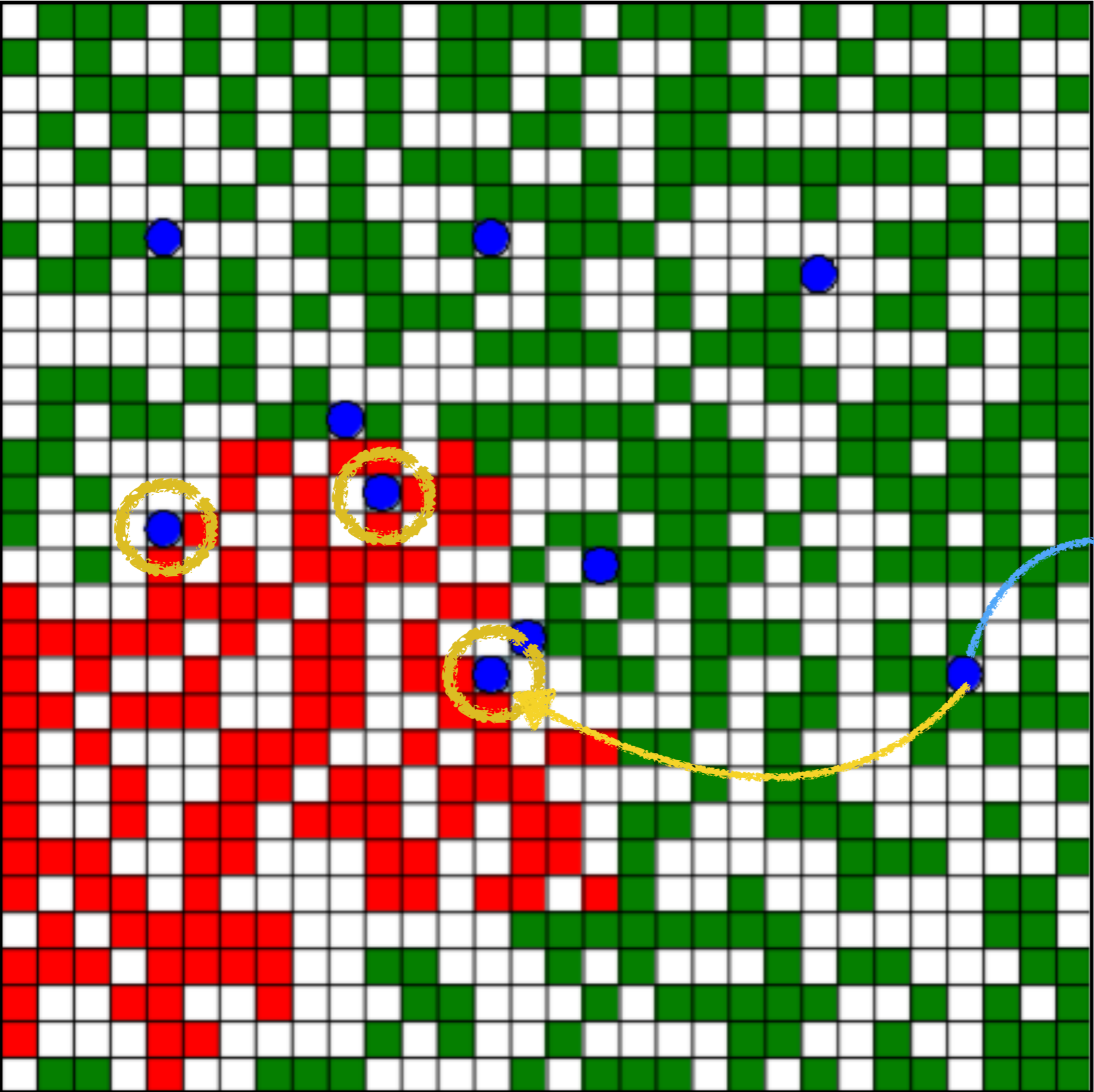


# Choose a plot toward the closest busy firefighter



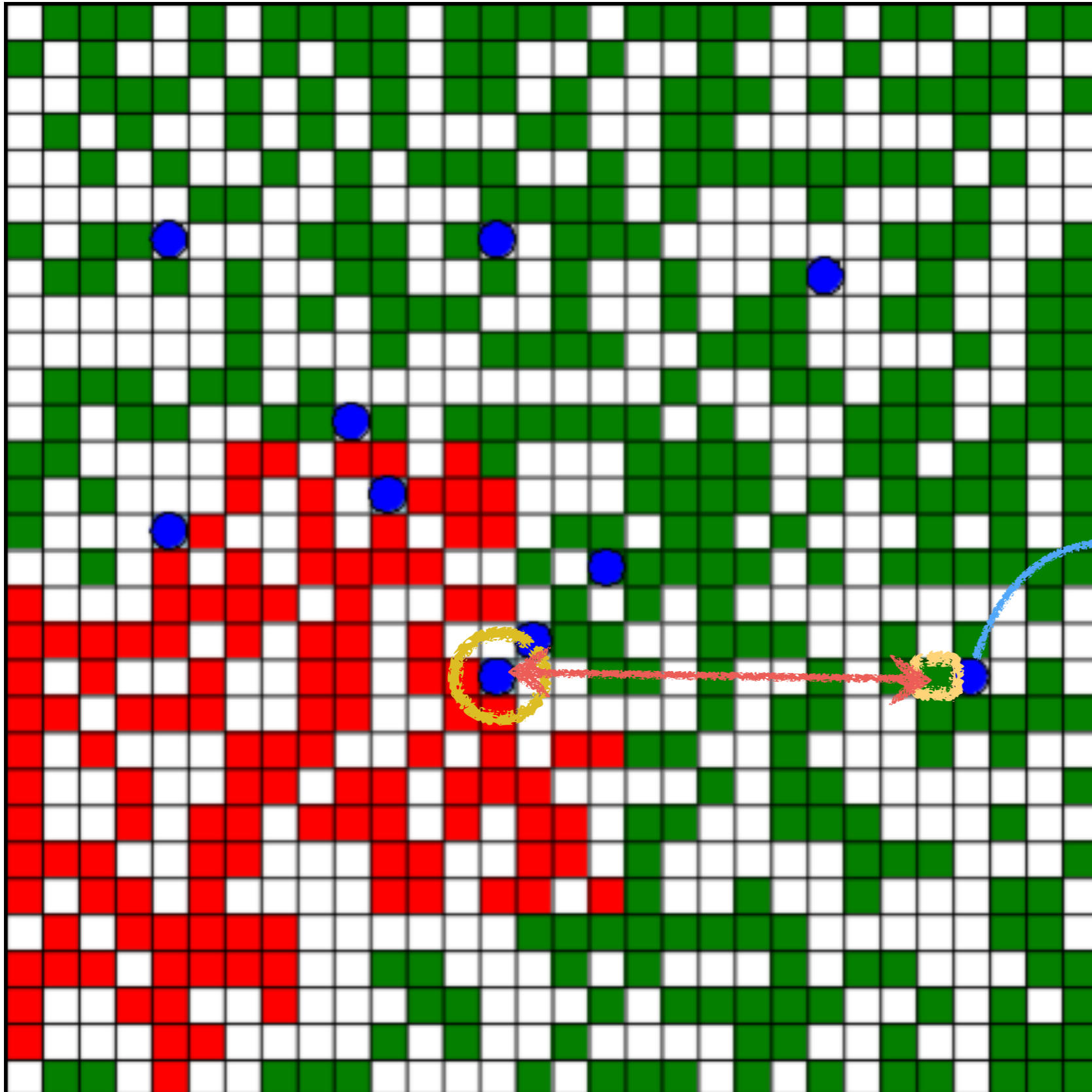
How will it choose its next location ?

# Choose a plot toward the closest busy firefighter



How will it choose its next location ?  
- Find the closest busy firefighter

# Choose a plot toward the closest busy firefighter

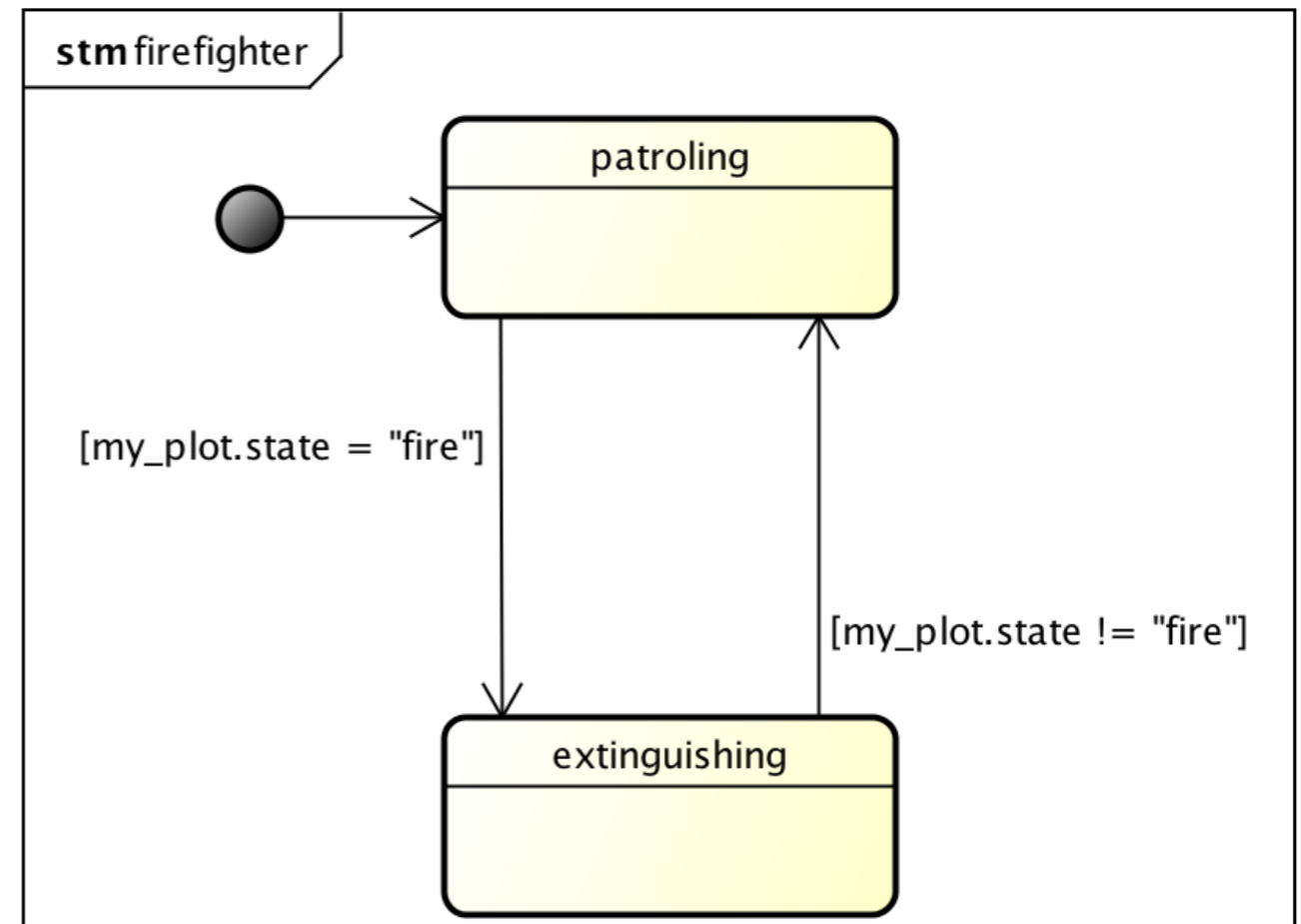


How will it choose its next location ?

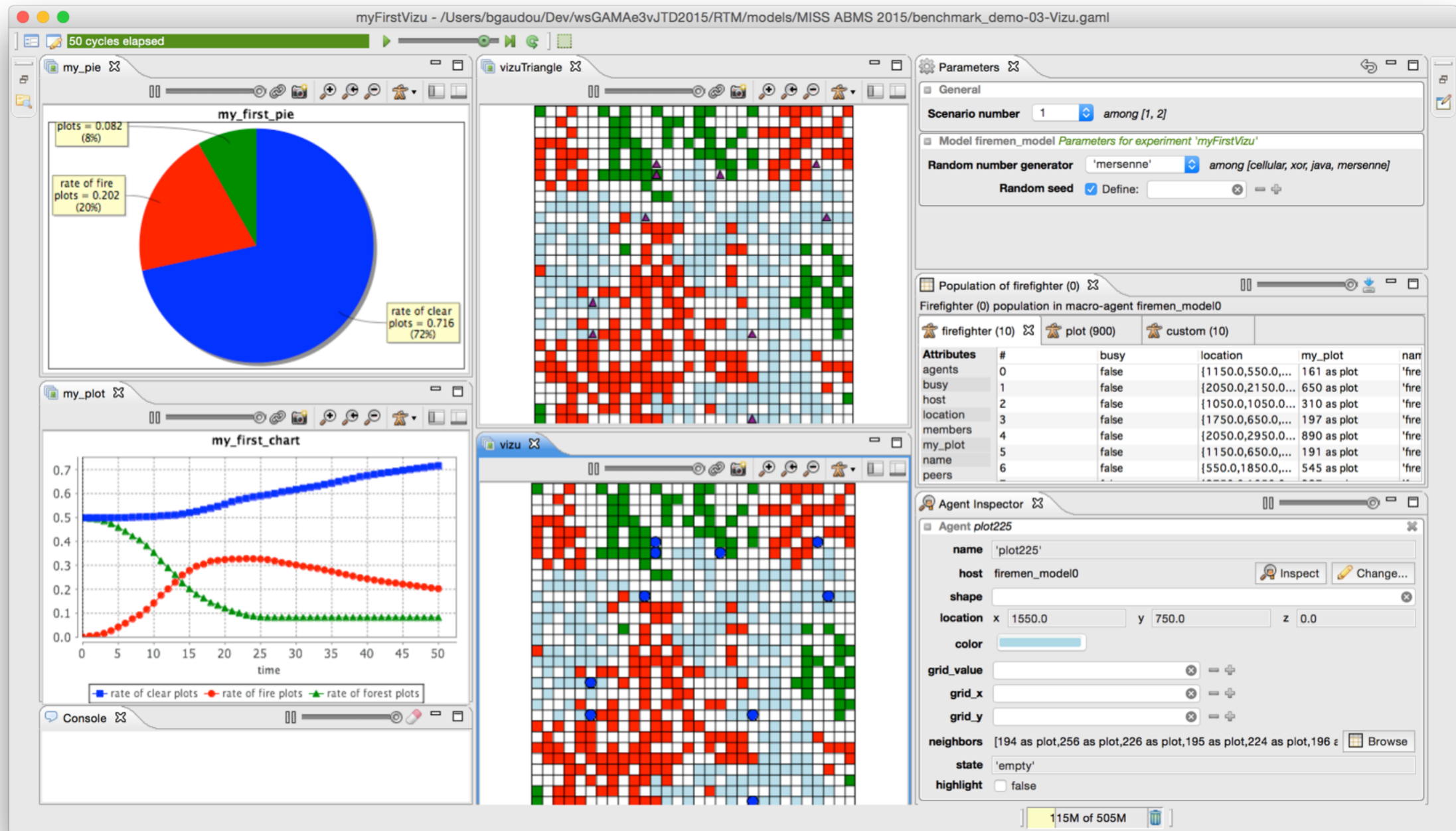
- Find the closest busy firefighter
- Find the closest cell to this firefighter

# Extensions

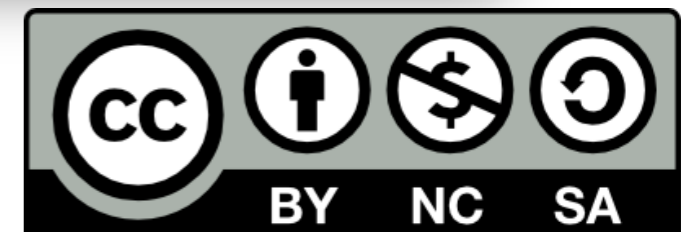
- ▶ Add a action `do_fire` to plot
- ▶ Stop simulation when no more fire
- ▶ Finite State Machine architecture: patrolling and extinguishing are now 2 states (and not 2 reflexes anymore)



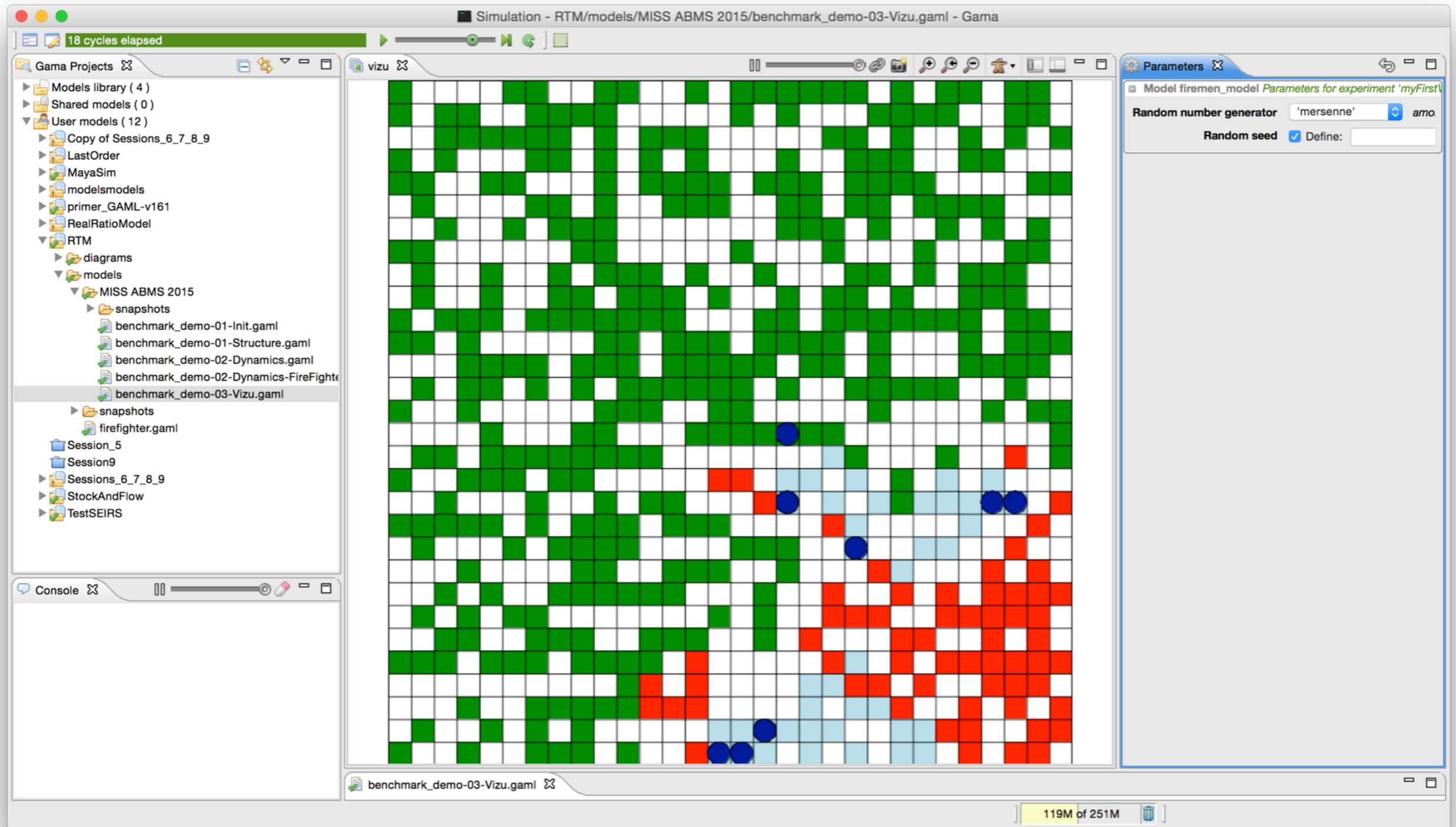
# Firefighter model: Visualisation, monitoring and model exploration



Benoit Gaudou (Univ. Toulouse 1)  
Patrick Taillandier (INRAE)



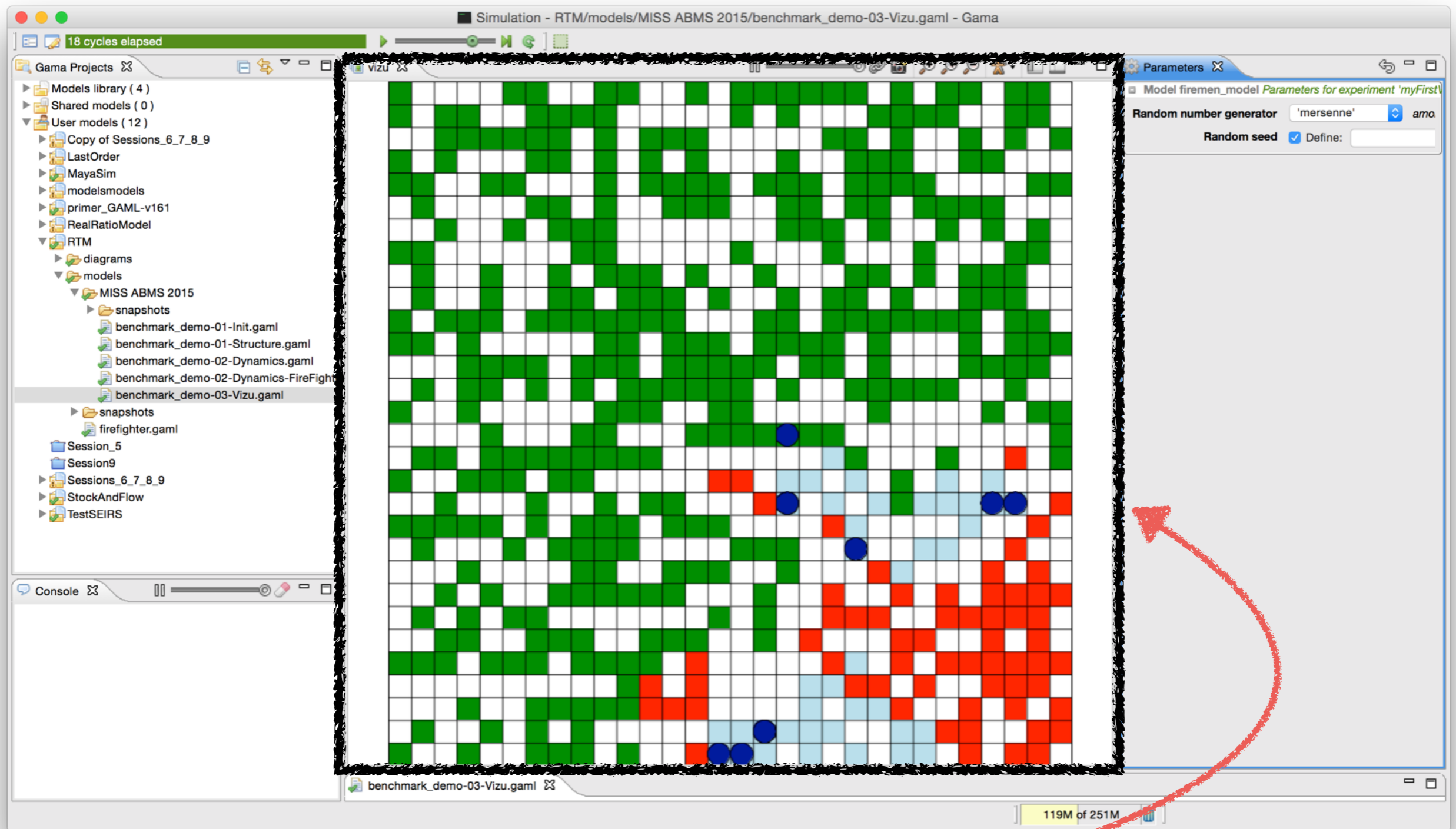
# Benchmark model - First visualisation



In GAMA, there are 2 kinds of experiments:

- gui
- batch

# Displays in GAMA



- ▶ In GAMA, we can define as many displays as needed, each of them represent a point of view on the simulation.

# Let's build our virtual laboratory

myFirstVizu - /Users/bgaudou/Dev/wsGAMAE3vJTD2015/RTM/models/MISS ABMS 2015/benchmark\_demo-03-Vizu.gaml

50 cycles elapsed

my\_pie

plots = 0.082 (8%)

rate of fire plots = 0.202 (20%)

rate of clear plots = 0.716 (72%)

my\_plot

my\_first\_chart

rate of clear plots

rate of fire plots

rate of forest plots

Parameters

Scenario number 1 among [1, 2]

Model firemen\_model Parameters for experiment 'myFirstVizu'

Random number generator 'mersenne' among [cellular, xor, java, mersenne]

Random seed Define: [ ]

Population of firefighter (0)

Firefighter (0) population in macro-agent firemen\_model0

Attributes	#	busy	location	my_plot	name
agents	0	false	{1150.0,550.0,...}	161 as plot	'fire'
busy	1	false	{2050.0,2150.0...}	650 as plot	'fire'
host	2	false	{1050.0,1050.0...}	310 as plot	'fire'
location	3	false	{1750.0,650.0,...}	197 as plot	'fire'
members	4	false	{2050.0,2950.0...}	890 as plot	'fire'
my_plot	5	false	{1150.0,650.0,...}	191 as plot	'fire'
name	6	false	{550.0,1850.0,...}	545 as plot	'fire'
peers	-	-	-	-	-

Agent Inspector

Agent plot225

name 'plot225'

host firemen\_model0

shape [ ]

location x 1550.0 y 750.0 z 0.0

color [ ]

grid\_value [ ]

grid\_x [ ]

grid\_y [ ]

neighbors [194 as plot,256 as plot,226 as plot,195 as plot,224 as plot,196 as plot] Browse

state 'empty'

highlight false

115M of 505M



# Let's build our virtual laboratory

The image shows a screenshot of a virtual laboratory interface with several panels and annotations:

- my\_first\_pie**: A pie chart showing the distribution of plot states. Annotations include:
  - plots = 0.082 (8%)
  - rate of fire plots = 0.202 (20%)
  - rate of clear plots = 0.716 (72%)
- my\_first\_chart**: A line graph showing the rate of clear plots (blue), rate of fire plots (red), and rate of forest plots (green) over time (0 to 50 cycles).
- Console**: A text area at the bottom left for logging output.
- Display agents with circle**: A grid view showing agents represented by small blue circles.
- Display agents with triangle**: A grid view showing agents represented by small purple triangles.
- Parameters**: A panel for configuring the simulation, including:
  - Scenario number: 1
  - Random number generator: mersenne
  - Random seed: Define
- Browse a population**: A table showing the population of firefighters and plots.

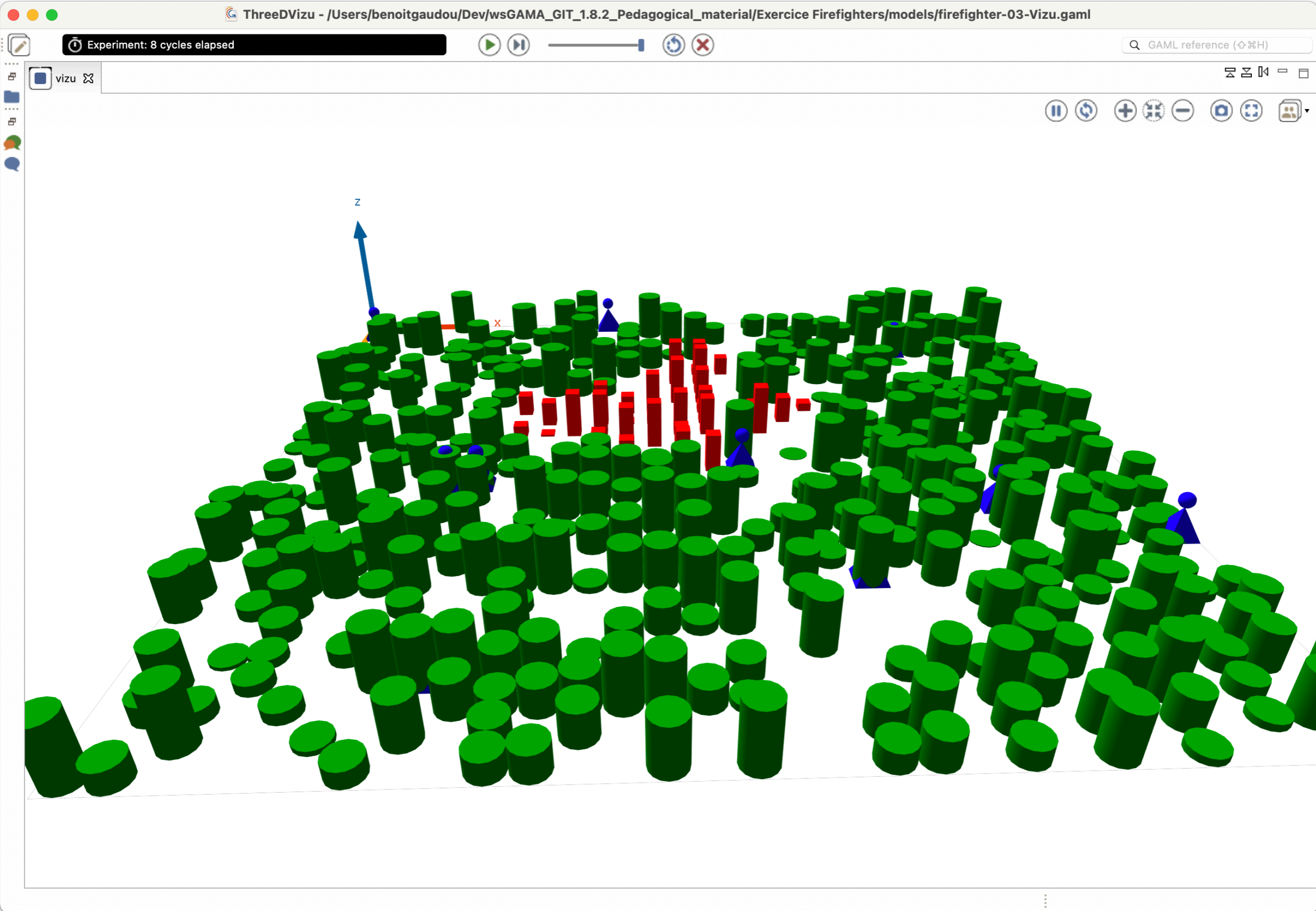
Attributes	#	busy	location	my_plot
agents	0	false	{1150.0,550.0,...}	161 as p
busy	1	false	{2050.0,2150.0...	650 as p
host	2	false	{1050.0,1050.0...	310 as p
location	3	false	{1750.0,650.0,...}	197 as p
members	4	false	{2050.0,2950.0...	890 as p
my_plot	5	false	{1150.0,650.0,...}	191 as plot
name	6	false	{550.0,1850.0,...}	545 as plot
peers				
- Inspect 1 agent**: A detailed view of a specific agent (plot225), showing its name, host, location (x: 1550.0, y: 750.0, z: 0.0), and state ('empty').

# Update the experiment to get the following displays

- ▶ 1 display with firefighters displayed with a **circle aspect**
- ▶ 1 display with firefighters displayed with a **triangle aspect**
  - ▶ Add corresponding aspects to the species
- ▶ A display plotting (as a **time series**):
  - ▶ The number of forest plots
  - ▶ The number of on fire plots
  - ▶ The number of empty plots.
- ▶ A display plotting the same information as a **pie**



# Create a new experiment to have a 3D display



# Exploration of the model

## ▶ Create a batch experiment to explore the model

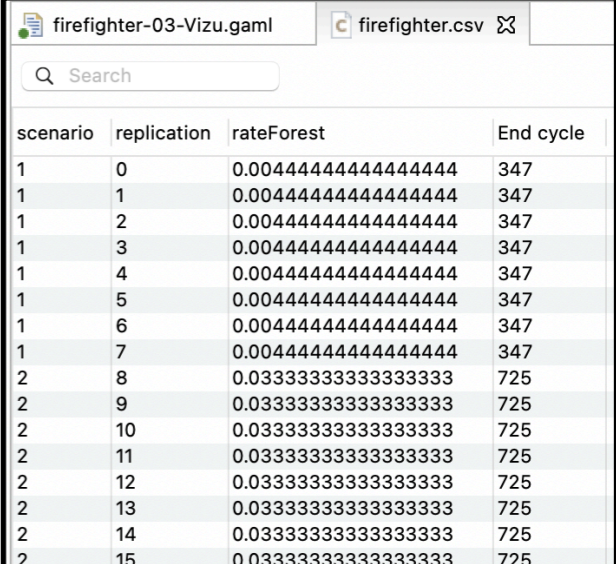
- ▶ Parameter to explore: the scenario
- ▶ Stop condition: when there is no more plot on fire
- ▶ Replication: 10
- ▶ Exploration method: exhaustive method

## ▶ Outputs - Displays (cf. permanent statement)

- ▶ The rate of remaining forest plots (at the end of each bunch of replications)
- ▶ The end cycle (at the end of each bunch of replications)

## ▶ Outputs - Save in file

- ▶ Save the scenario number, the id of the replication, and the 2 indicators (remaining forests and end cycle).



scenario	replication	rateForest	End cycle
1	0	0.0044444444444444	347
1	1	0.0044444444444444	347
1	2	0.0044444444444444	347
1	3	0.0044444444444444	347
1	4	0.0044444444444444	347
1	5	0.0044444444444444	347
1	6	0.0044444444444444	347
1	7	0.0044444444444444	347
2	8	0.0333333333333333	725
2	9	0.0333333333333333	725
2	10	0.0333333333333333	725
2	11	0.0333333333333333	725
2	12	0.0333333333333333	725
2	13	0.0333333333333333	725
2	14	0.0333333333333333	725
2	15	0.0333333333333333	725

# Exploration of the model

wsGAMA\_GIT\_1.8.2\_Pedagogical\_material - Exercice Firefighters/models/firefighter.csv - Gama (runtime)

Batch over. 2 runs, 16 simulations.

Model firemen / Experiment explo

Parameters to explore

scenario 2

Exploration method

Stop condition empty(plot where (each.state = 'fire'))

Best parameter set found {scenario=2}

Best fitness 0.1423611111111113

Last parameter set tested {scenario=2}

Last fitness 0.1423611111111113

Parameter space scenario (2) = 2

Interactive console Console

forest

rate of forest plots

step

firefighter-03-Vizu.gaml firefighter.csv

Search

scenario	replication	rateForest	End cycle
1	0	0.004444444444444444	347
1	1	0.004444444444444444	347
1	2	0.004444444444444444	347
1	3	0.004444444444444444	347
1	4	0.004444444444444444	347
1	5	0.004444444444444444	347
1	6	0.004444444444444444	347
1	7	0.004444444444444444	347
2	8	0.03333333333333333	725
2	9	0.03333333333333333	725
2	10	0.03333333333333333	725
2	11	0.03333333333333333	725
2	12	0.03333333333333333	725
2	13	0.03333333333333333	725
2	14	0.03333333333333333	725
2	15	0.03333333333333333	725