

GAMA, modeling made easy



GAMA PLATFORM

Que es GAMA?

GAMA (GIS Agent-based Modeling Architecture) es un entorno de desarrollo de modelado para la construcción de simulaciones basadas en agentes.

Es un software de código abierto que sigue creciendo, su versión actual es 1.7.

GAMA se desarrolla por varios equipos de la IRD / UPMC unidad de investigación internacional UMMISCO :

MSI Equipo de Investigación de la Universidad Nacional de Vietnam, Hanoi , Vietnam (2007 - 2015)

UMR 6228 IDEES , CNRS / Universidad de Rouen , Francia (2010-2015)

UMR 5505 IRIT , CNRS / Universidad de Toulouse 1 , Francia (2010-2015)

SUEÑO Equipo de Investigación de la Universidad de Can Tho , Vietnam (2011 - 2015)

UMR 8623 LRI , CNRS / Université Paris- Sud, Francia (2011 - 2015)

Que es GAMA?

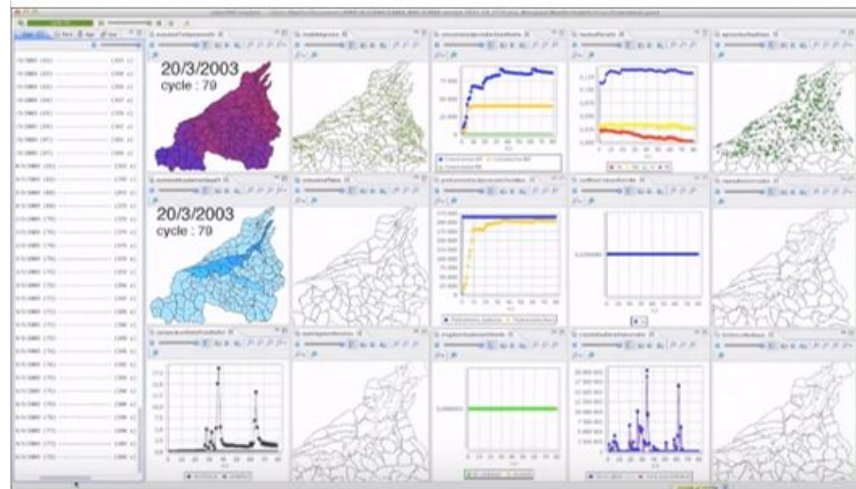
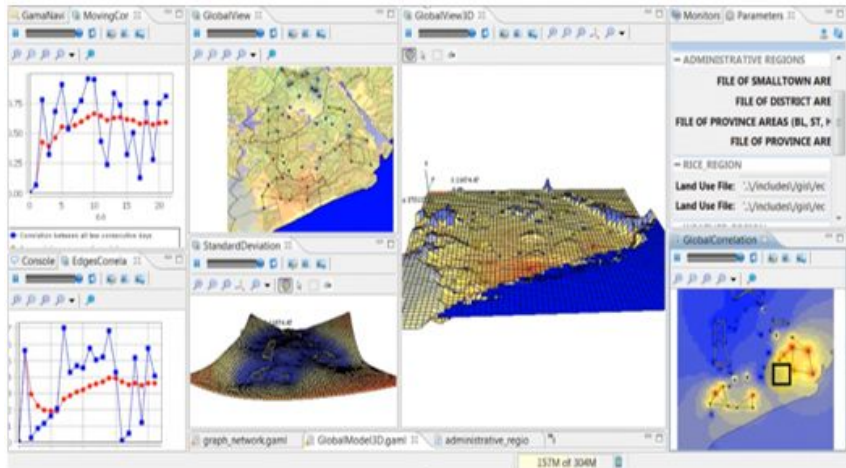
- **Aplicable a múltiples entornos:** puedes usar GAMA para crear aplicaciones en el entorno que desees.
- **Un lenguaje intuitivo y de alto nivel:** Puedes escribir tus modelos fácilmente utilizando GAML.
- **SIG y modelos basados en los datos:** Puedes crear agentes para cualquier grupo de datos, incluyendo SIG (Sistema de Información Geográfica) y ejecutar simulaciones a gran escala.
- **Interfaz de usuario:** Declara interfaces para inspeccionar los agentes , paneles de acciones controlados por el usuario, control sobre 2D / 3D y aspectos de agentes.

Aplicable a múltiples dominios

GAMA se ha desarrollado con un enfoque muy general, y se puede aplicar a múltiples campos . Han sido desarrollados algunos “plugins” para adaptarse a las necesidades más particulares .

Ejemplo de entornos en los que Gama está presente sobre todo :

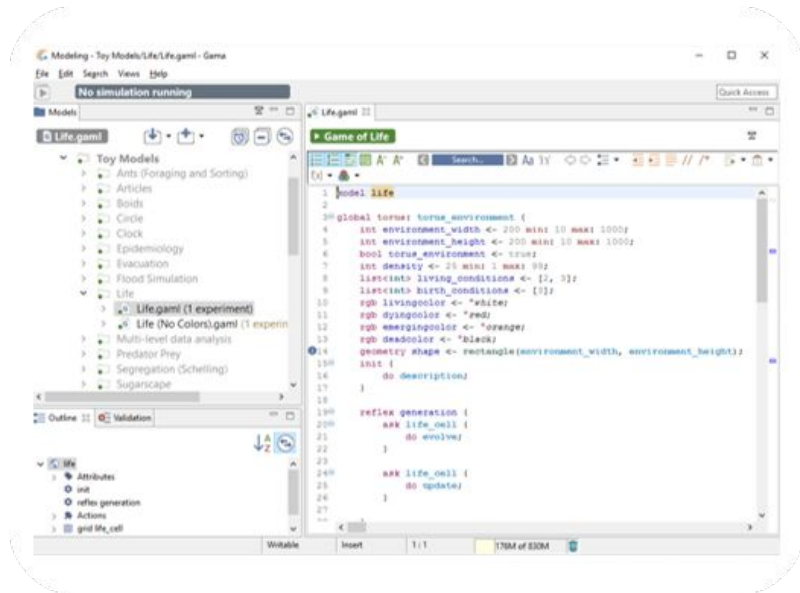
- Transporte
- La planificación urbana
- Epidemiología
- Ambiente



Un lenguaje intuitivo y de alto nivel

Gracias a su alto nivel y el lenguaje intuitivo, GAMA puede ser usado fácilmente por gente sin conocimientos informáticos. Se puede declarar especies, dándoles algunos comportamientos especiales, crearlos en su mundo, y mostrarlos en menos de 10 minutos .

GAML es el idioma utilizado en Gama , codificado en Java . Es un lenguaje basado en agentes , que le proporciona la posibilidad de construir su modelo con varios paradigmas de modelado. Una vez que el modelo está listo, le permite explorar y calibrar algunas características, utilizando los parámetros que ha definido como entrada de la simulación.



TUTORIALS REFERENCES COMMUNITY DOWNLOAD



Learn GAML Step by Step

This large progressive tutorial has been designed to be (in English). It will cover the main part of the possibilities and some more.

- Programming basis
- Global species
- Regular species

How to proceed to learn better

As you will progress in the tutorial, you will see several other parts. You can click on them if you want to learn more. We encourage you to do this, because you can get easily lost but we encourage you to follow this tutorial "step by step". All in all, our advice is to use the "learning graph" interface to find an area of interest, and a learning path will be automatically generated for you. Good luck with your reading, and please use our [mailing list](#) if you have a question/suggestion!

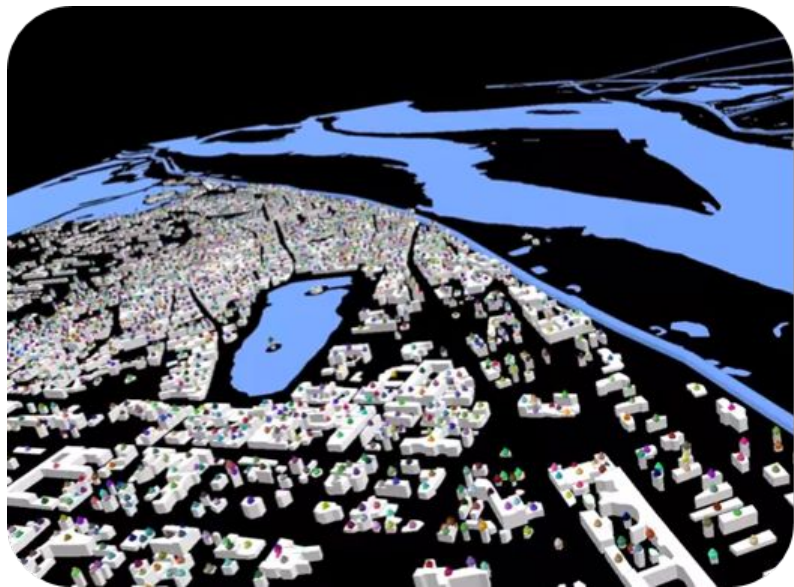
GIS y modelos basados en los datos

GAMA ofrece, desde su creación, la posibilidad de cargar fácilmente GIS (Sistema de Información Geográfica) .

Puede importar un gran número de tipos de datos, tales como texto, archivos, CSV, archivo de formas, OSM (Open Street Mapa-data), rejilla, imágenes SVG, también archivos 3D, como 3DS o OBJ.

Algunas de las características avanzadas que ofrece GAMA es la posibilidad de conectarse a sus bases de datos, y también la de utilizar potentes herramientas estadísticas tales como R.

GAMA se ha utilizado en proyectos a gran escala, el uso de un gran número de agentes (hasta millones de agentes) .

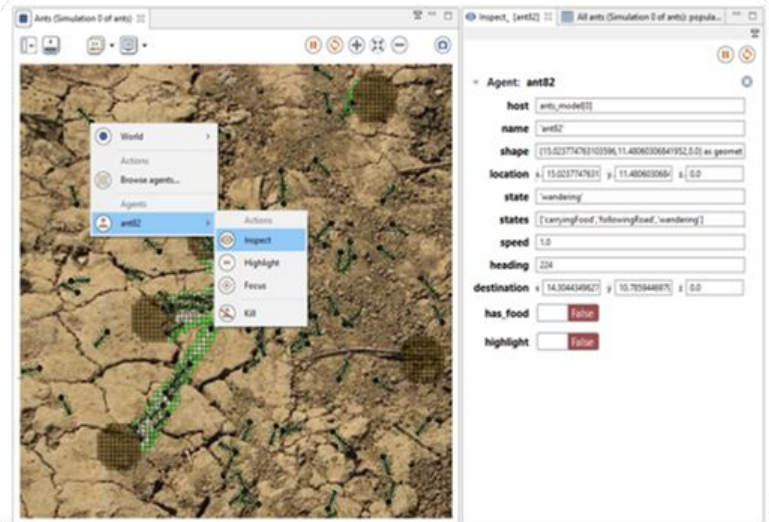
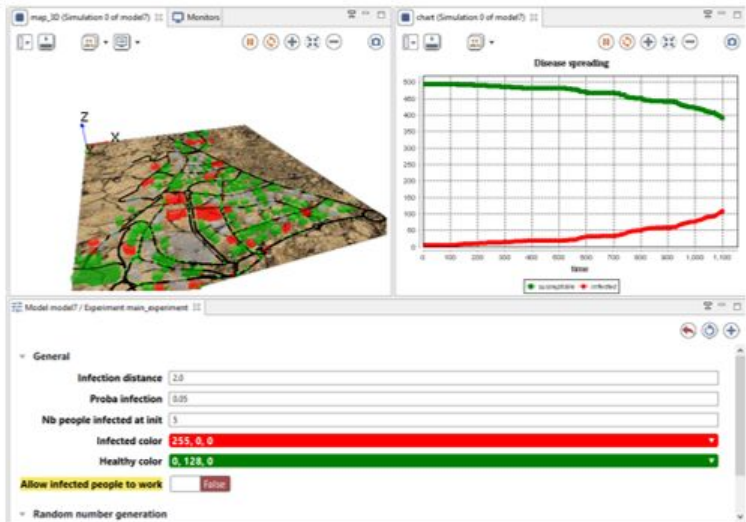


Interfaz de usuario declarativa

GAMA proporciona la posibilidad de tener varias representaciones para el mismo modelo. Se pueden añadir hasta representaciones visuales para el modelo, con el fin de recalcar un cierto aspecto de la simulación.

Se proporcionan opciones avanzadas para 3D: se pueden controlar luces, cámaras y también añadir texturas a tus objetos 3D. Por otro lado, permite definir fácilmente diferentes gráficos.











Durante las simulaciones, algunas de las características avanzadas están disponibles para inspeccionar la población de los agentes . Para continuar con un modelo interactivo, se puede agregar fácilmente algunos paneles de acción controlados por el usuario o eventos de ratón.

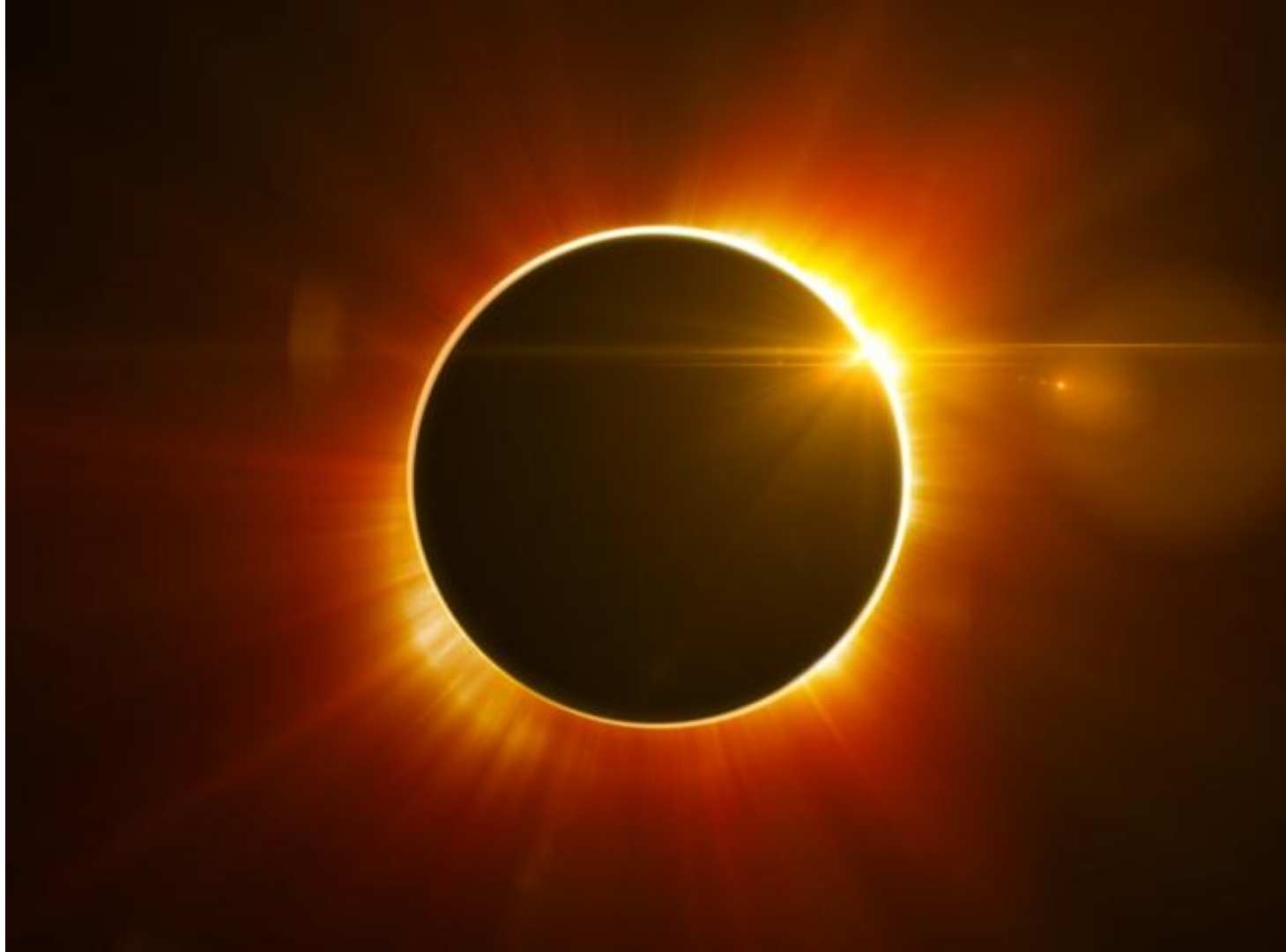




Pros

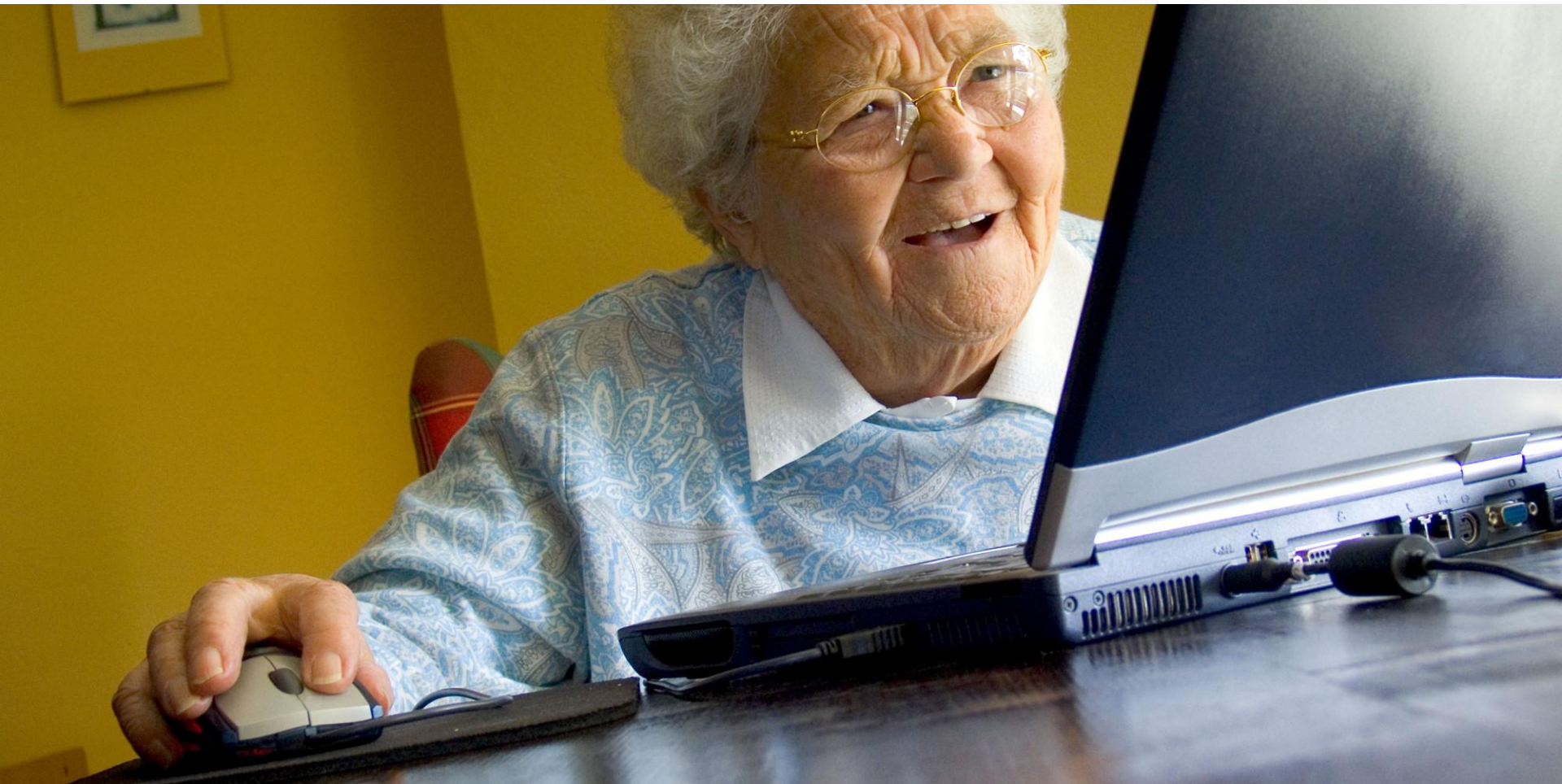


Name	Date
 configuration	25/06
 features	22/06
 headless	22/06
 p2	22/06
 plugins	22/06
 readme	22/06
 artifacts.xml	22/06
 eclipssec.exe	22/06
 Gama.exe	22/06
 Gama.ini	22/06









3316 G

118 Combo!!





News

26 7 2015



There are no healthy people left in the world X

The last healthy person on the planet recently became infected with Randomia



DNA

89



Cure

0%

Disease



Infected

127450459

Japan



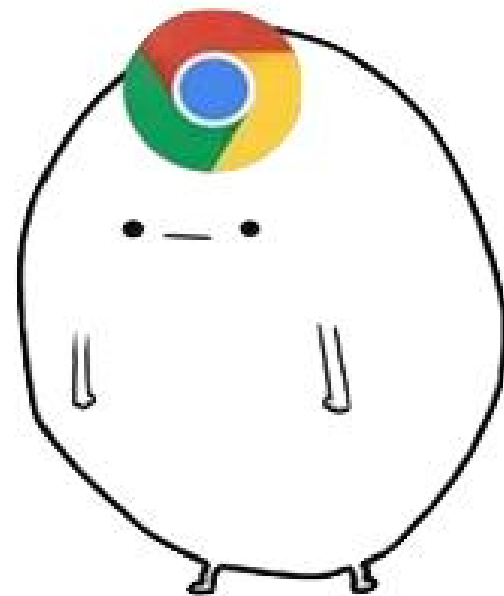
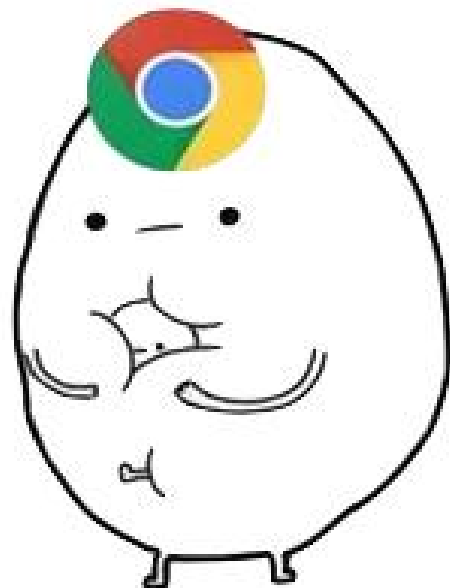
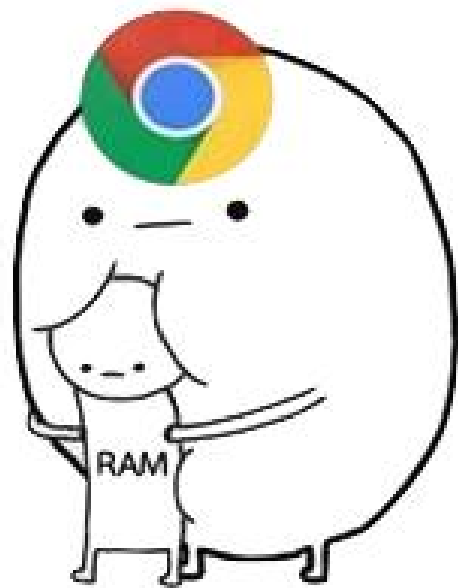
Dead

0

World

Contras





Task Manager

File Options View

Processes Performance App history Startup Users Details Services

Name	79% CPU	83% Memory	5% Disk	0% Network
> Java(TM) Platform SE binary	70,3%	363,8 MB	0 MB/s	0 Mbps
System and compressed memory	1,0%	308,3 MB	0,1 MB/s	0 Mbps
Google Chrome (32 bit)	1,8%	128,2 MB	0 MB/s	0 Mbps
Google Chrome (32 bit)	0%	98,4 MB	0 MB/s	0 Mbps
Google Chrome (32 bit)	0,2%	71,6 MB	0 MB/s	0 Mbps

Java(TM) Platform SE binary



Java(TM) Platform SE binary is not responding

If you close the program, you might lose information.

→ Close the program

→ Wait for the program to respond



```
species people{
```

```
    bool is_infected <- false;
```

```
}
```

```
species people{  
  bool is_infected <- false;  
  aspect base{  
    draw circle(2) color: (is_infected) ? #red : #green;  
  }  
}
```

```
species people skills:[moving]{  
  bool is_infected <- false;  
  aspect base{  
    draw circle(2) color: (is_infected) ? #red : #green;  
  }  
  reflex moving{  
    do wander;  
  }  
}
```

```
species rats skills:[moving]{  
  bool is_infected <- flip(0.5);  
  int attack_range <- 5;  
  ...  
  reflex attack when:people at_distance attack_range{  
    ...  
  }  
}
```

my_experiment - F:\Gama\Workspace\WorkspaceOnPlay2\ten_min_tuto\models\my_model.gaml

File Edit Search Experiment Agents Views Help

147 cycles elapsed

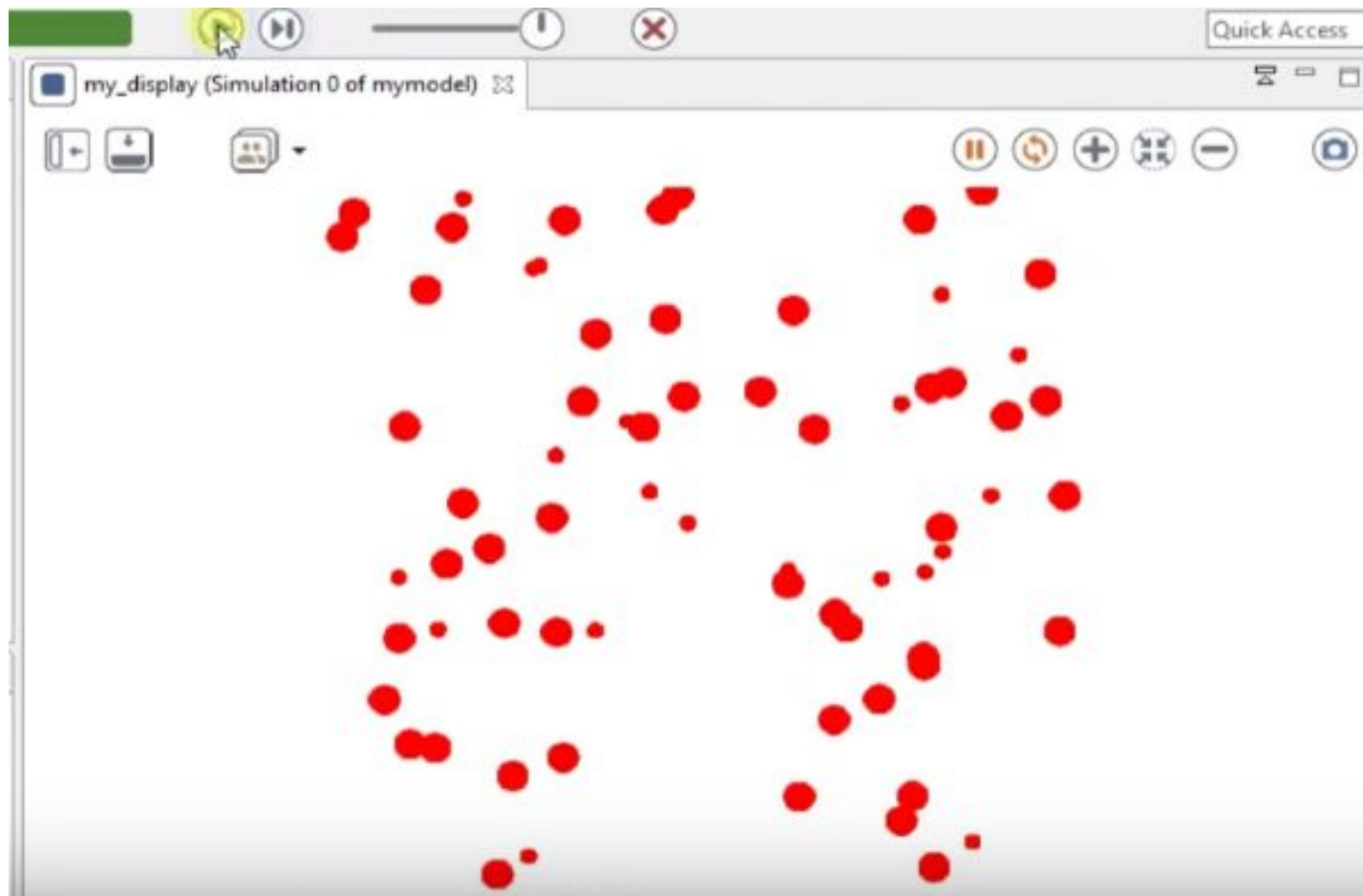
Model mymodel / Exp... Models my_display (Simulation 0 of mymodel)

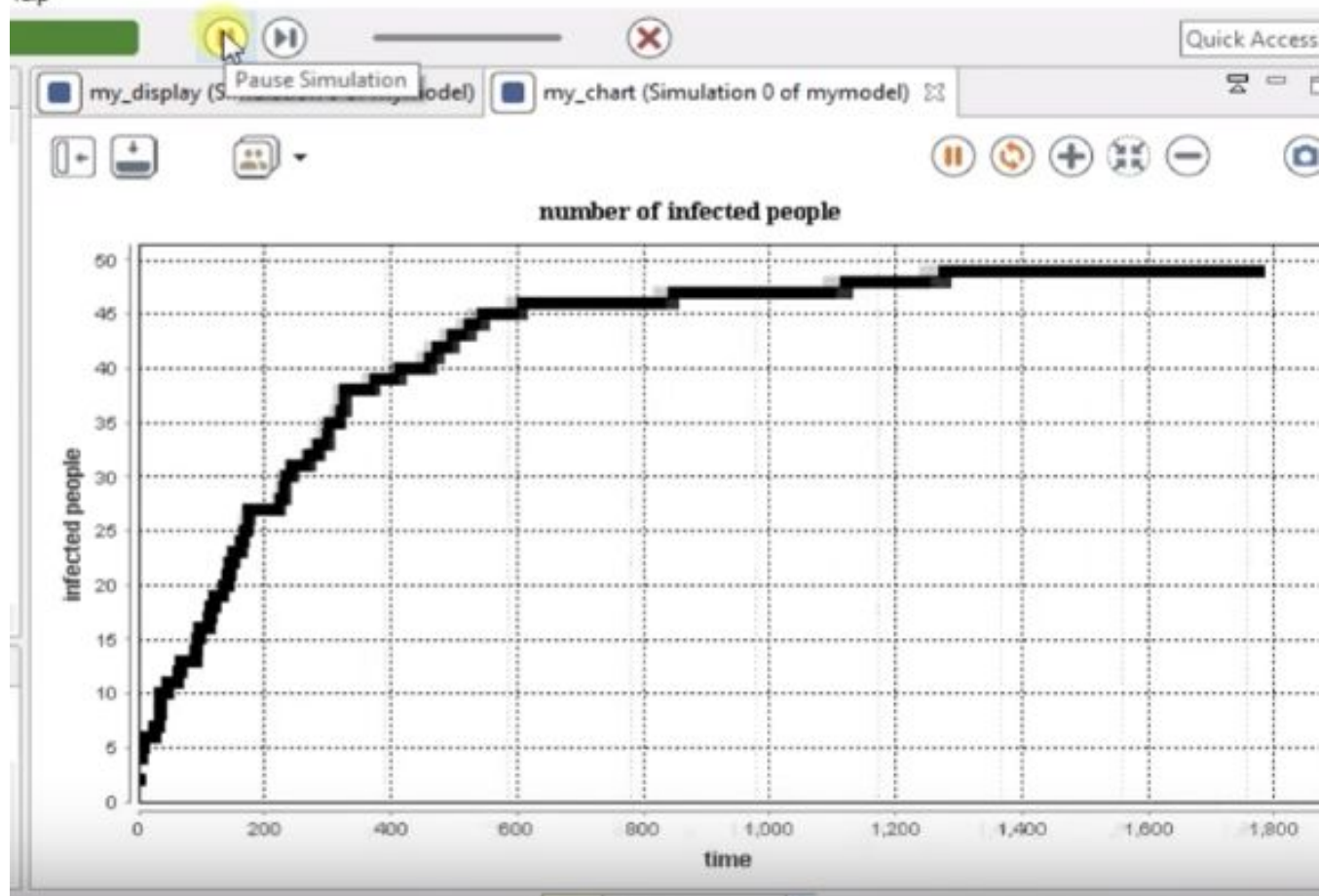
ten...uto

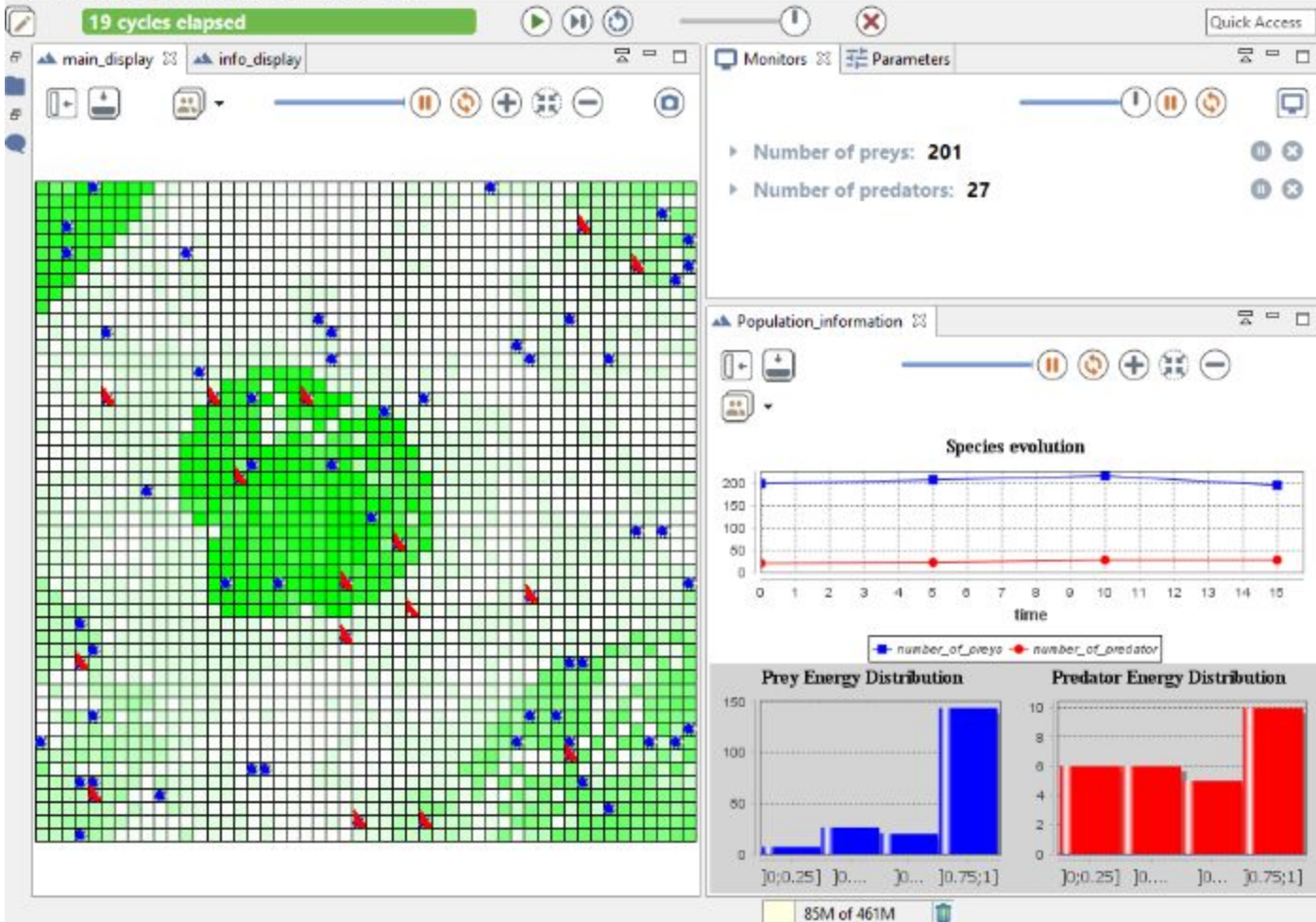
- Library models
- Plugin models
- User models
 - ten_min_tuto

Console

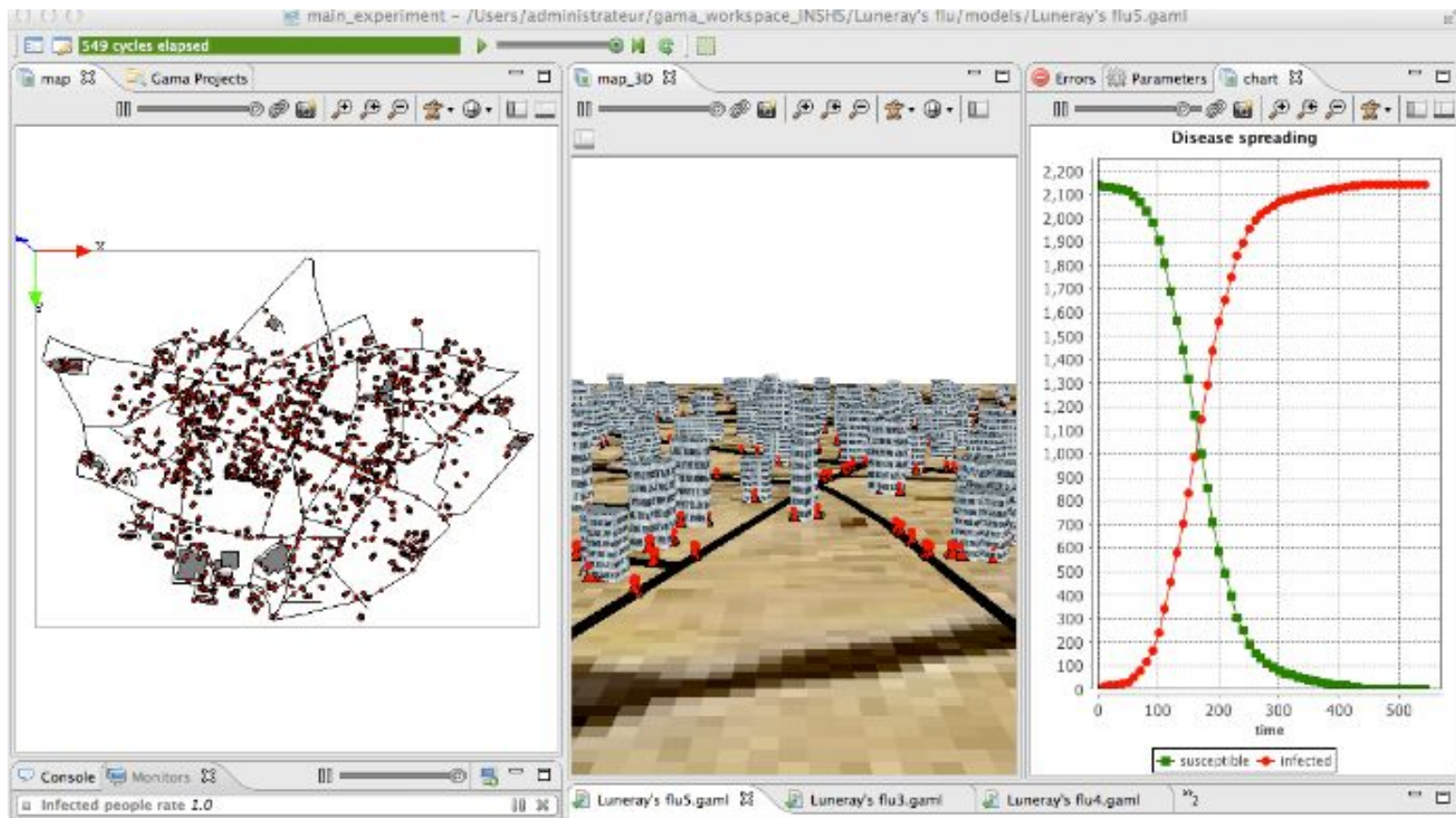
The screenshot displays the GAMA (Geometric Agent Modeling Architecture) software interface. The main window shows a simulation titled "my_display (Simulation 0 of mymodel)". The simulation area contains numerous green and red circular agents scattered across a white background. The top toolbar includes a progress bar indicating "147 cycles elapsed" and various control icons such as play, pause, and zoom. The left sidebar shows a "Models" browser with a tree view containing "Library models", "Plugin models", and "User models", with "ten_min_tuto" selected under the user models. The bottom panel is a "Console" window with several icons for text and execution control. A mouse cursor is positioned over a yellow circle in the top toolbar.











NOW I WILL START SLOW CLAP....

