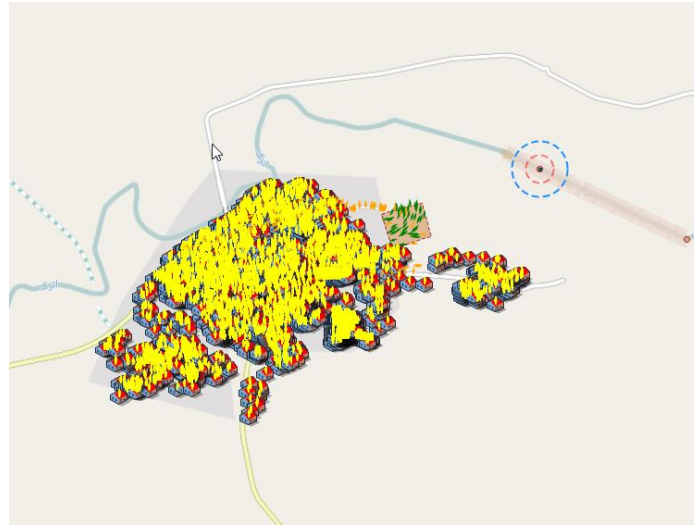


Agent-Based Simulation

**Example:
Flood simulation and evacuation
GIS Environment**



Ali Asgary
ADERSIM
York University
2017

Lesson 2: Creating the base model

- Adding single agent population
- Adding agent population using external file
- Adding agent population using internal code

Adding agent population: single agents

- In this model we have 4 agent types each with different population size. Size of **Flood** and **Sensor** will be one. Later we may want to increase their size.

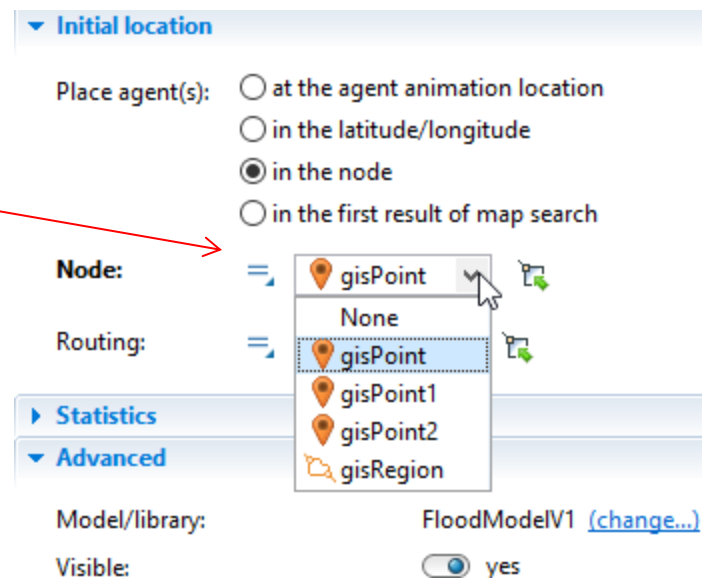
Adding agent population: Flood

- To define the population size and other attributes of a single agent click on the agent population icon to open and see its attributes. We start with the Flood first.
- Choose Single agent.
- Do not change the speed at this time.
- For the initial location of the Flood agent choose “in the node”.

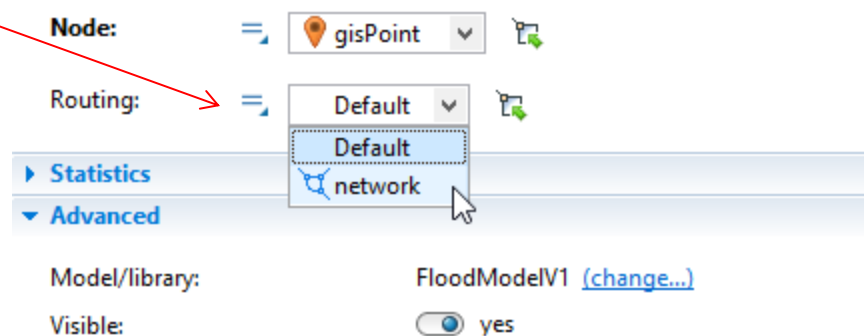
The screenshot displays the configuration interface for an agent population named "flood - Flood". The interface is divided into a map area on the left and a configuration panel on the right. The map shows a grid and a path with a red location pin. The configuration panel includes the following settings:

- Name:** flood
- Show name Ignore
- Single agent Population of agents
- Movement:** Initial speed: 10 meters per second
- Initial location:** Place agent(s):
 - at the agent animation location
 - in the latitude/longitude
 - in the node
 - in the first result of map search
- Node:** gisPoint
- Routing:** Default

- Choose gisPoint for the location of Flood agent.



- Choose the network for the Routing.



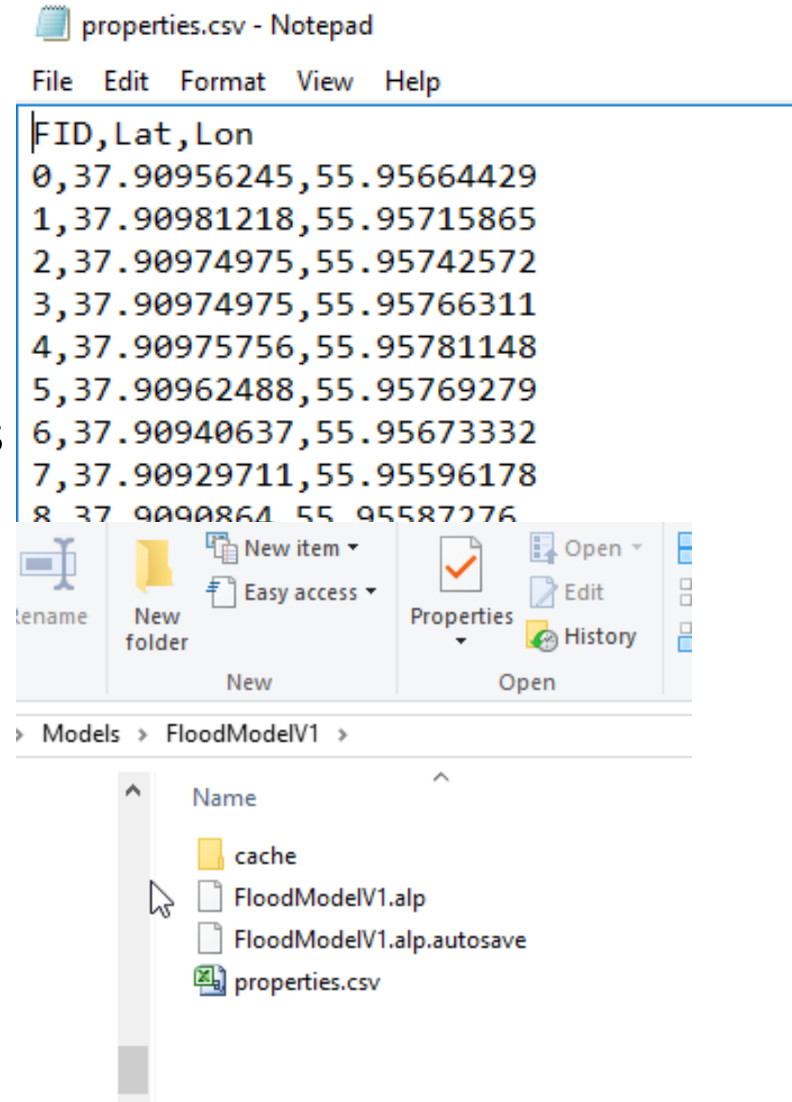
Adding agent population: sensor

- Click on the Sensor icon in the Main.
- Choose Single agent.
- Change the Initial speed to 0.
- For the initial location of the Sensor agent choose “in the latitude/longitude”.
- For Latitude insert 37.91183
- For Longitude insert 55.97919 (these are the lat/lon for the gisPoint3). This is another way of setting the initial location of your agents.

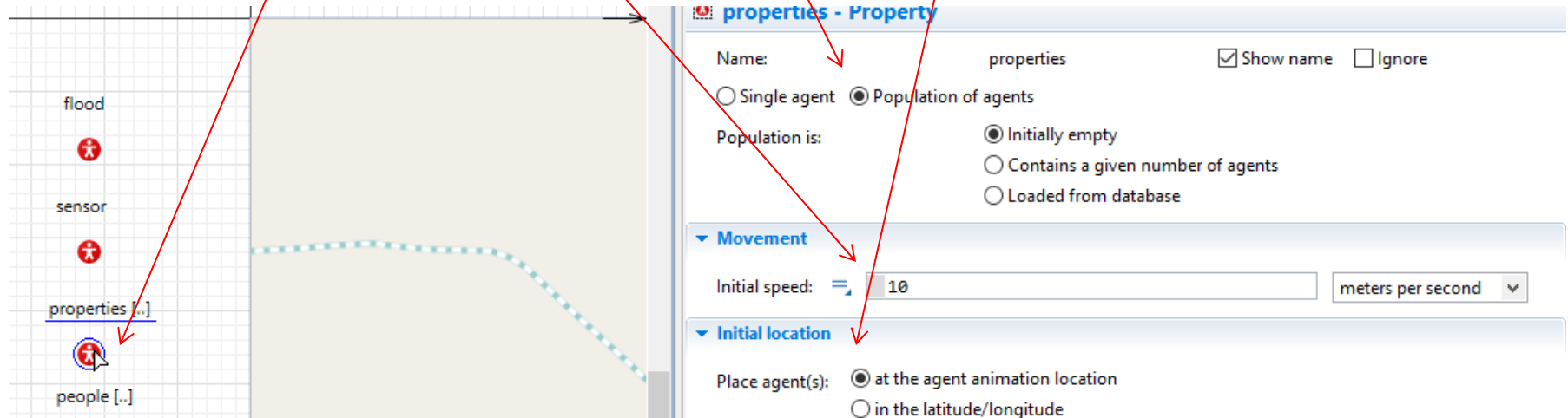
The screenshot displays the configuration window for a 'sensor' agent. The window title is 'sensor - Sensor'. It includes a 'Name' field set to 'sensor' and checkboxes for 'Show name' (checked) and 'Ignore' (unchecked). Under the 'Movement' section, the 'Initial speed' is set to 0 meters per second. The 'Initial location' section has four radio button options: 'at the agent animation location', 'in the latitude/longitude' (selected), 'in the node', and 'in the first result of map search'. The 'Latitude' is set to 37.91183 and the 'Longitude' is set to 55.97919. The 'Routing' is set to 'Default'. There are also sections for 'Statistics' and 'Advanced' which are currently collapsed.

Adding population agents for Property and person agents

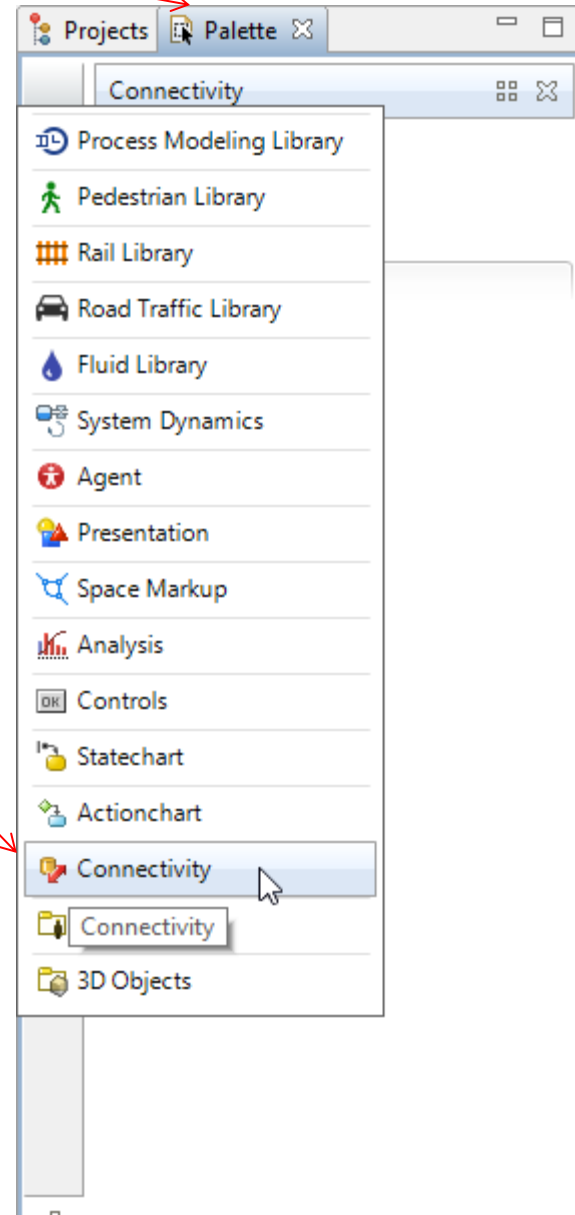
- Property (buildings) and Person agents have large population size.
- In this case we first read the information for Property from a text file.
- Our text file is a CSV file that contains three columns. FID, Lat, and Lon.
- You can create CSV files in any text editor or excel.
- These are the Latitude and Longitude of buildings in our study area.
- Download the property.csv file and add it to your project directory



- Click on the properties agent icon in the main and choose Population of agents.
- At this time we do not want to change the Movement and Initial location. So leave them at the default values



- Now click on the Palette tab.

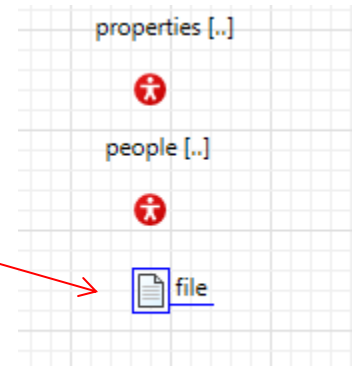
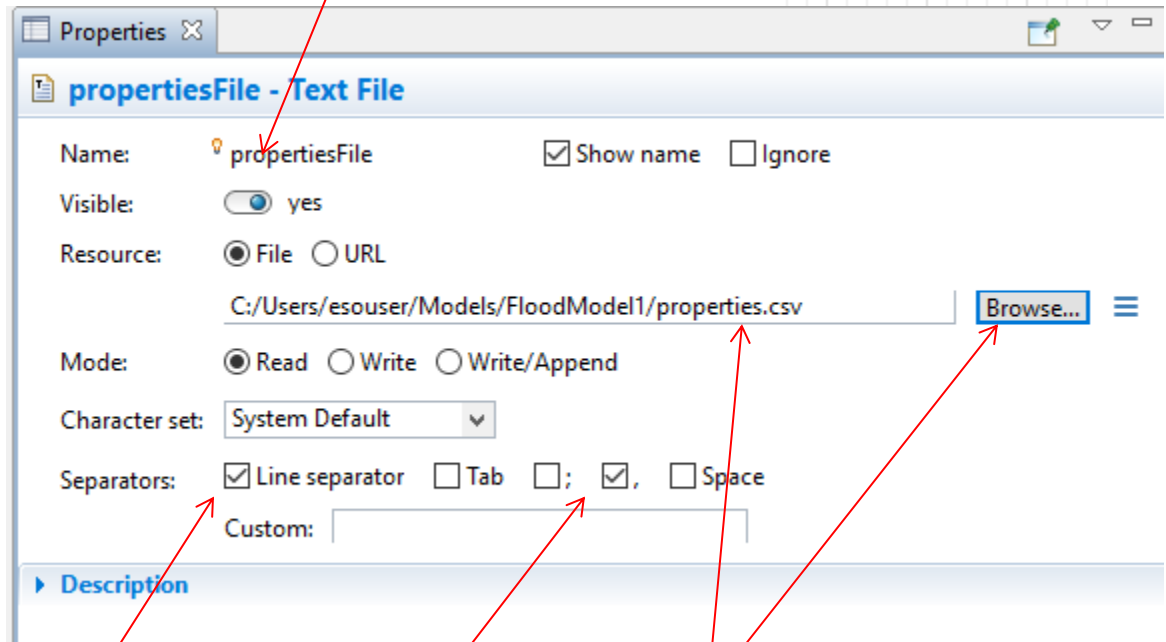


- Click on the Connectivity group.

Select and drag a text file to the Main

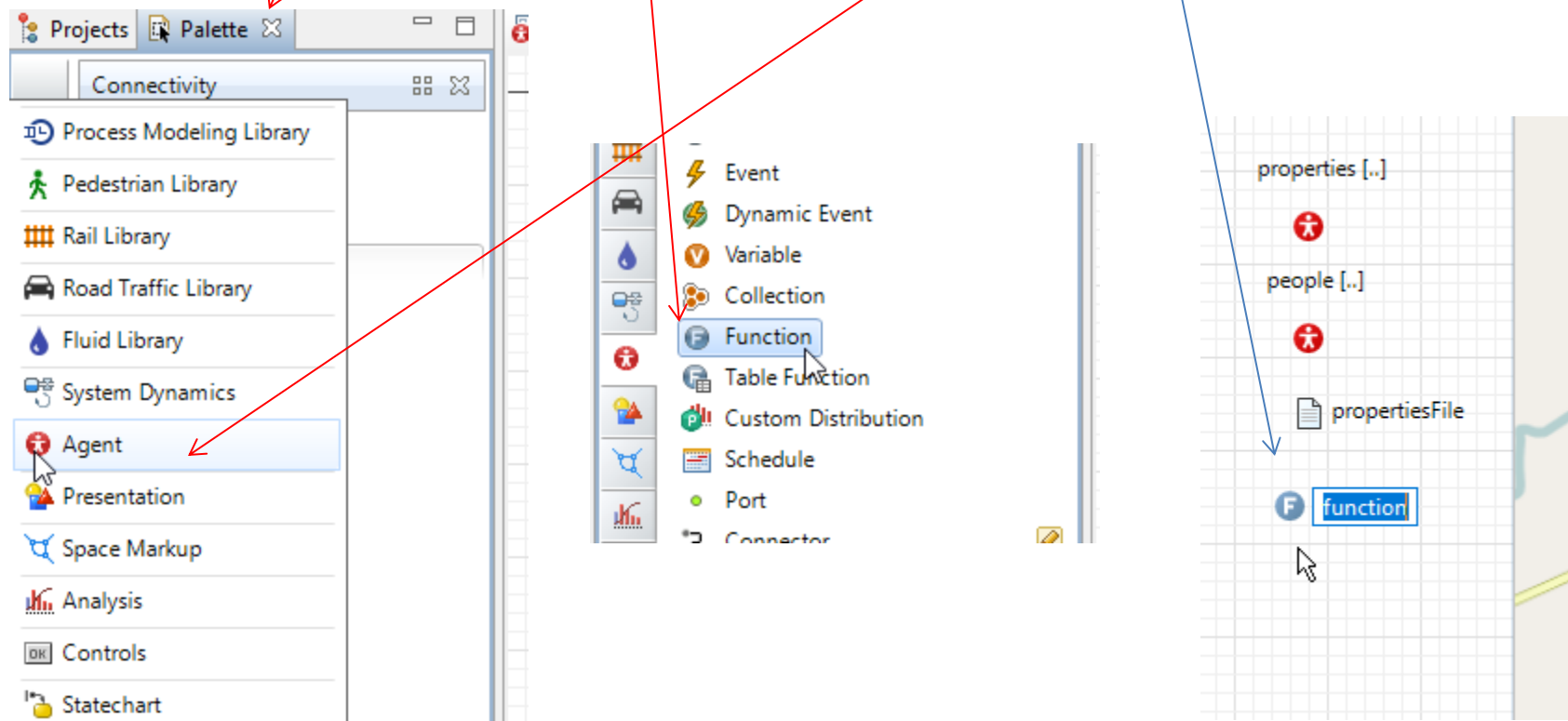


Change the name to propertiesFile

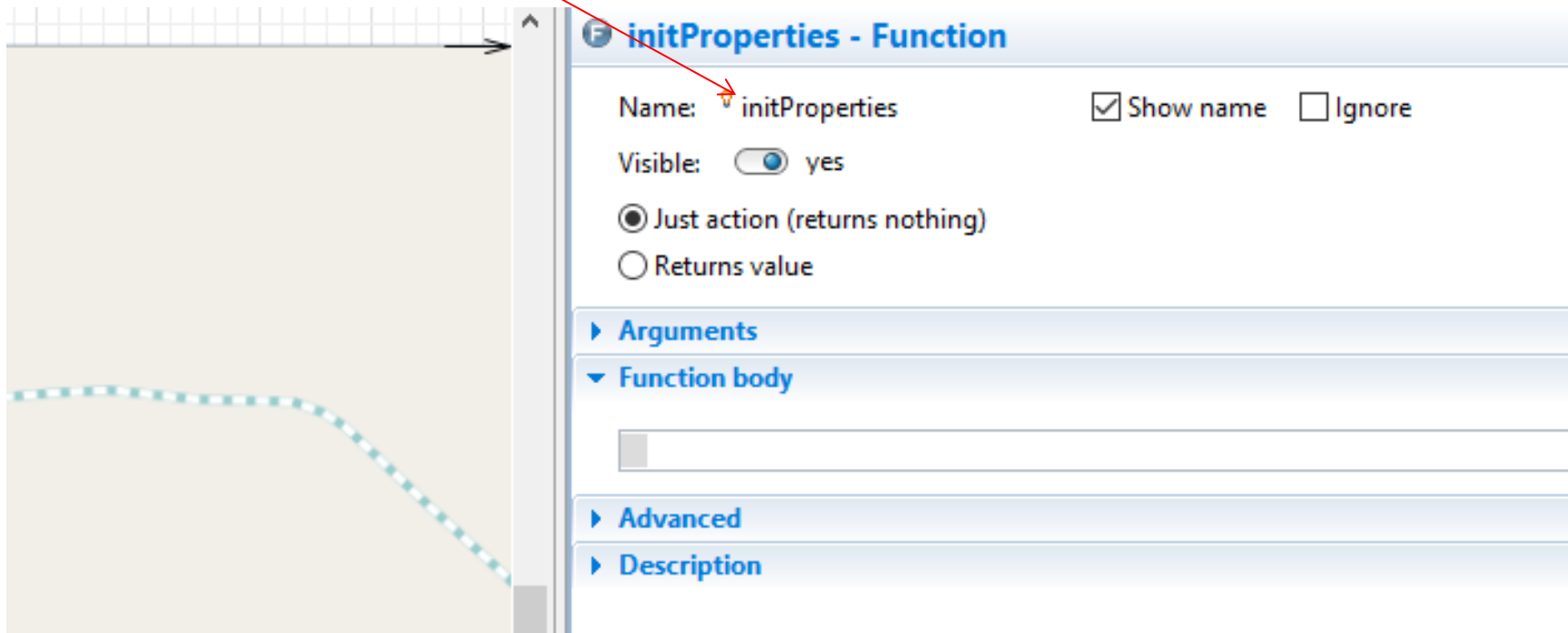


Click on the Browse button and provide the path to the text file that you downloaded and added to your project. Check the Line separator and “,” because we are reading from a comma delimited CSV file.

- Now we will create a function to read the contents of our CSV file.
- Click on the **Palette** tab and open **Agent** group.
- Drag and drop a **function** into the **Main**.



- Change the name of this function to `initProperties`

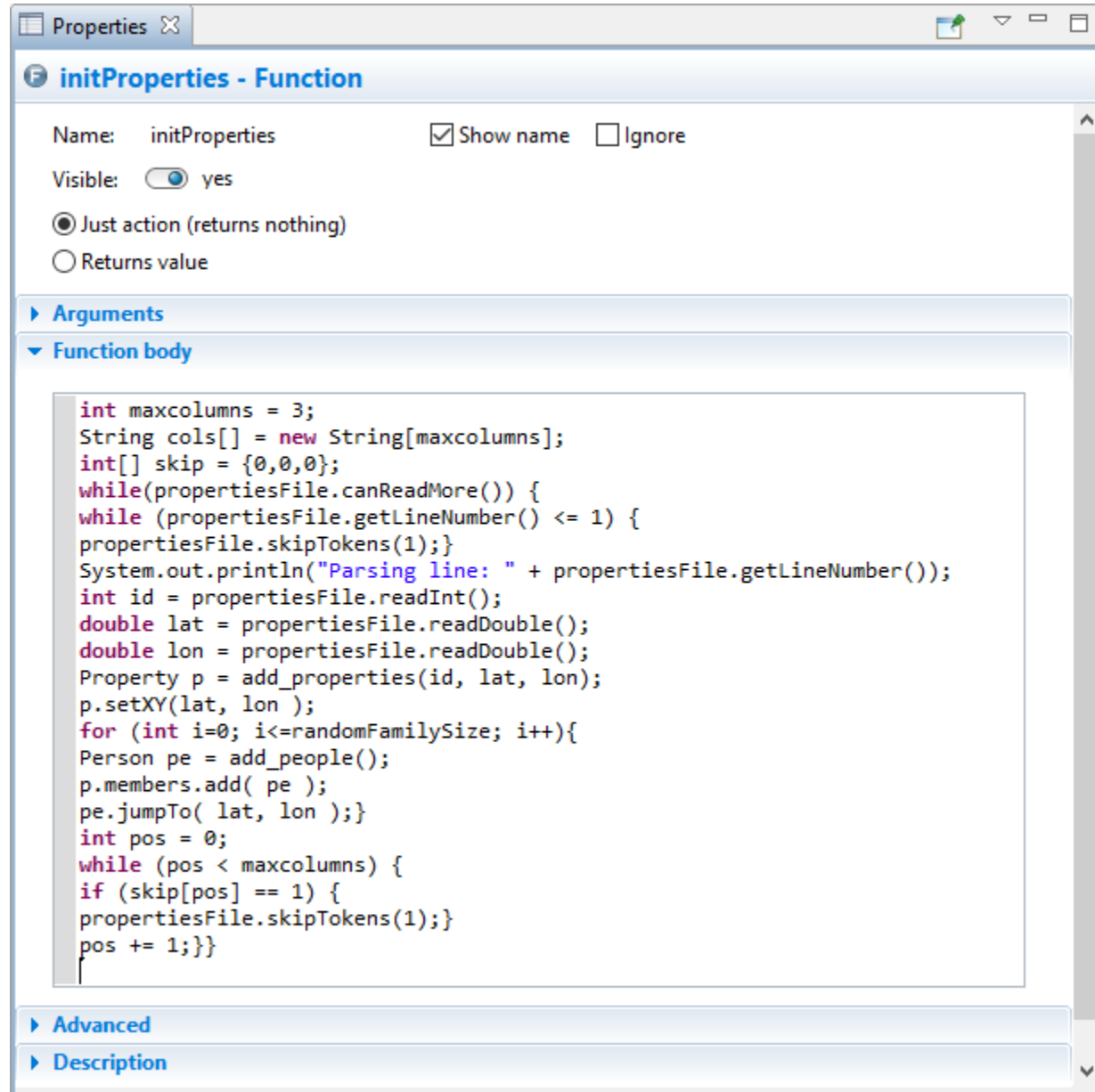


- Copy and paste the following code into the Function body section.

```
int maxcolumns = 3;
String cols[] = new String[maxcolumns];
int[] skip = {0,0,0};
while(propertiesFile.canReadMore()) {
while (propertiesFile.getLineNumber() <= 1) {
propertiesFile.skipTokens(1);}
System.out.println("Parsing line: " + propertiesFile.getLineNumber());
int id = propertiesFile.readInt();
double lat = propertiesFile.readDouble();
double lon = propertiesFile.readDouble();
Property p = add_properties(id, lat, lon);
p.setXY(lat, lon );
for (int i=0; i<=randomFamilySize; i++){
Person pe = add_people();
p.members.add( pe );
pe.jumpTo( lat, lon );}
int pos = 0;
while (pos < maxcolumns) {
if (skip[pos] == 1) {
propertiesFile.skipTokens(1);}
pos += 1;}}
```

- The `initProperties` function will look like this.

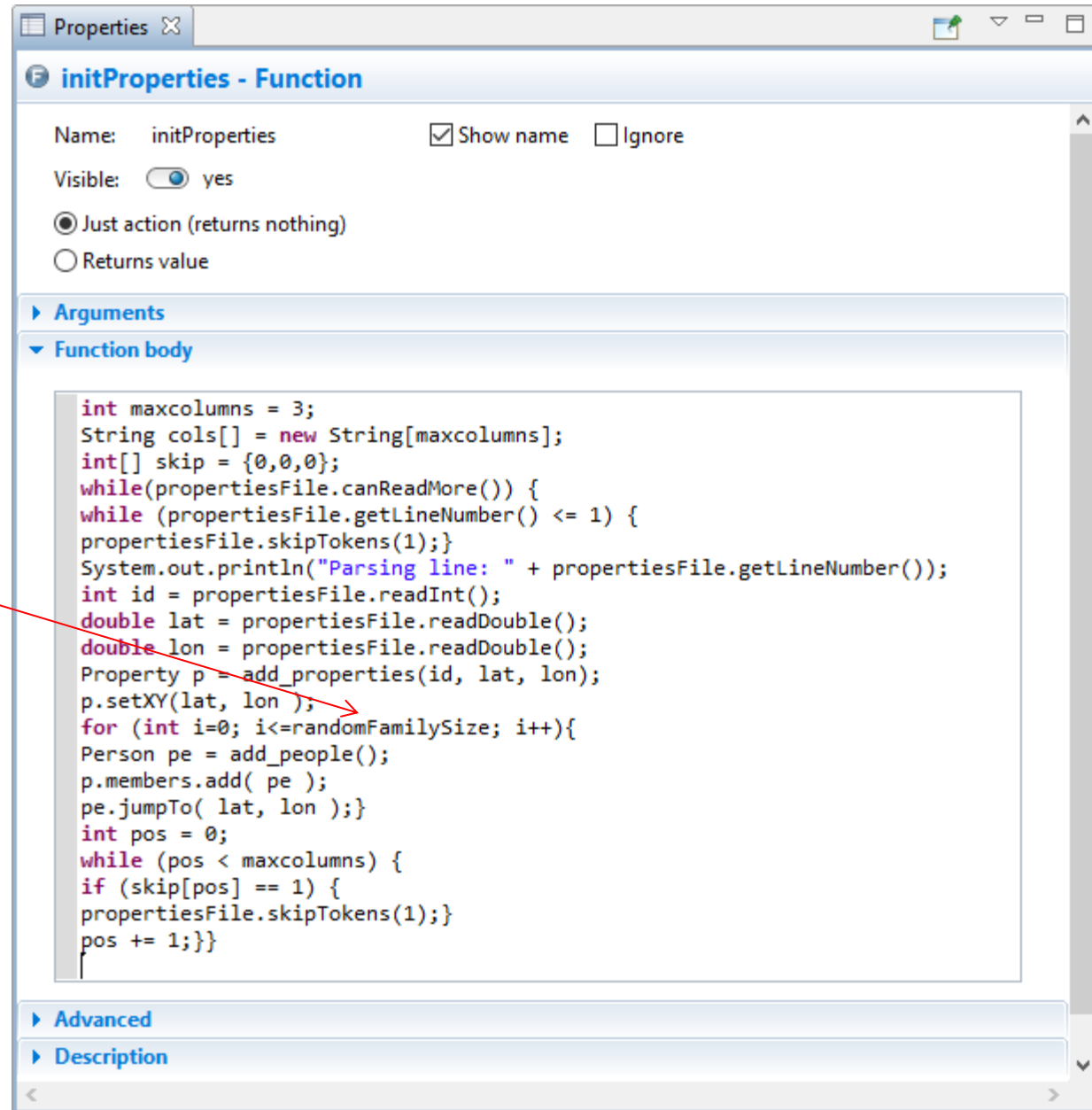
This code reads the three column from the `propertiesFile` without skipping any of them. It then defines three variables (`id`, `lat` and `lon`) to fill them by reading from the text file. It then iterates over each line of the file (except for the first line) and adds `Properties` and set their locations using the `lat` and `lon` values. Since persons live in the properties, it then adds some people to each property using random family size ranging from 1 to 7 (we will define this later).



The screenshot shows the 'initProperties - Function' window in an IDE. The function is named 'initProperties', is visible, and returns nothing. The function body is as follows:

```
int maxcolumns = 3;
String cols[] = new String[maxcolumns];
int[] skip = {0,0,0};
while(propertiesFile.canReadMore()) {
while (propertiesFile.getLineNumber() <= 1) {
propertiesFile.skipTokens(1);
System.out.println("Parsing line: " + propertiesFile.getLineNumber());
int id = propertiesFile.readInt();
double lat = propertiesFile.readDouble();
double lon = propertiesFile.readDouble();
Property p = add_properties(id, lat, lon);
p.setXY(lat, lon );
for (int i=0; i<=randomFamilySize; i++){
Person pe = add_people();
p.members.add( pe );
pe.jumpTo( lat, lon );}
int pos = 0;
while (pos < maxcolumns) {
if (skip[pos] == 1) {
propertiesFile.skipTokens(1);}
pos += 1;}}
```

- We need to define a parameter to hold the family size

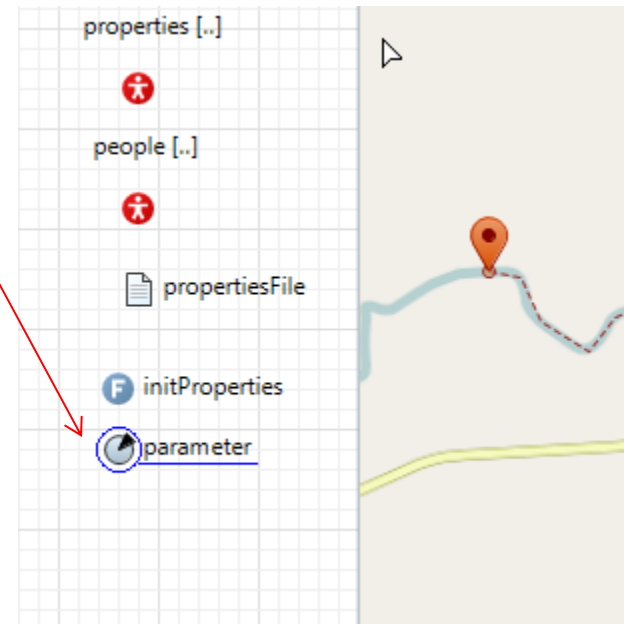
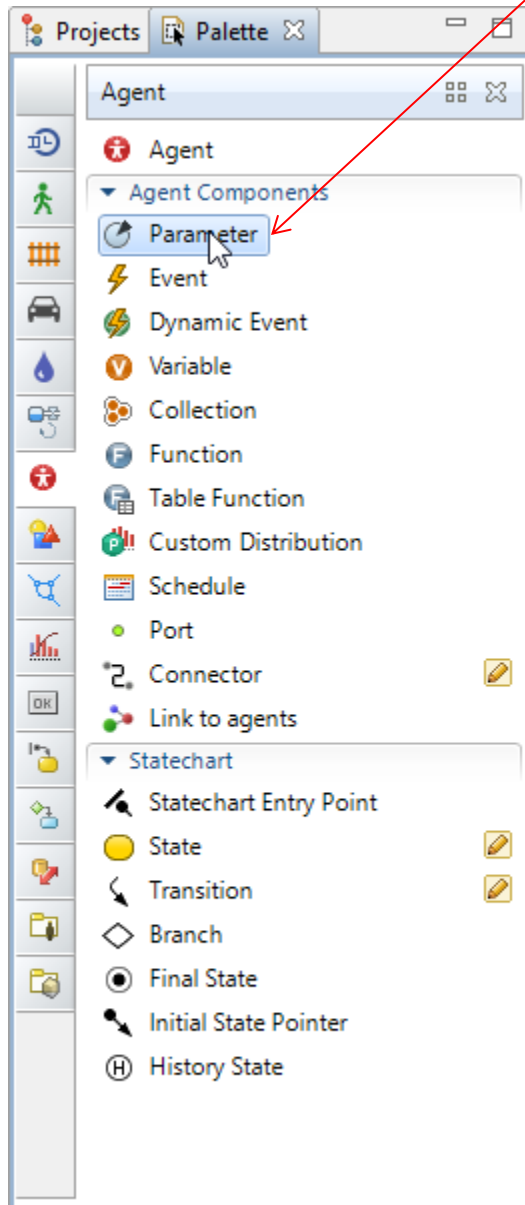


The screenshot shows the 'Properties' window for a function named 'initProperties'. The window has a title bar with 'Properties' and a close button. Below the title bar, the function name 'initProperties - Function' is displayed. There are two checkboxes: 'Show name' (checked) and 'Ignore' (unchecked). Below these, there is a 'Visible' toggle set to 'yes'. There are two radio buttons: 'Just action (returns nothing)' (selected) and 'Returns value' (unselected). The 'Arguments' section is collapsed. The 'Function body' section is expanded, showing the following Java code:

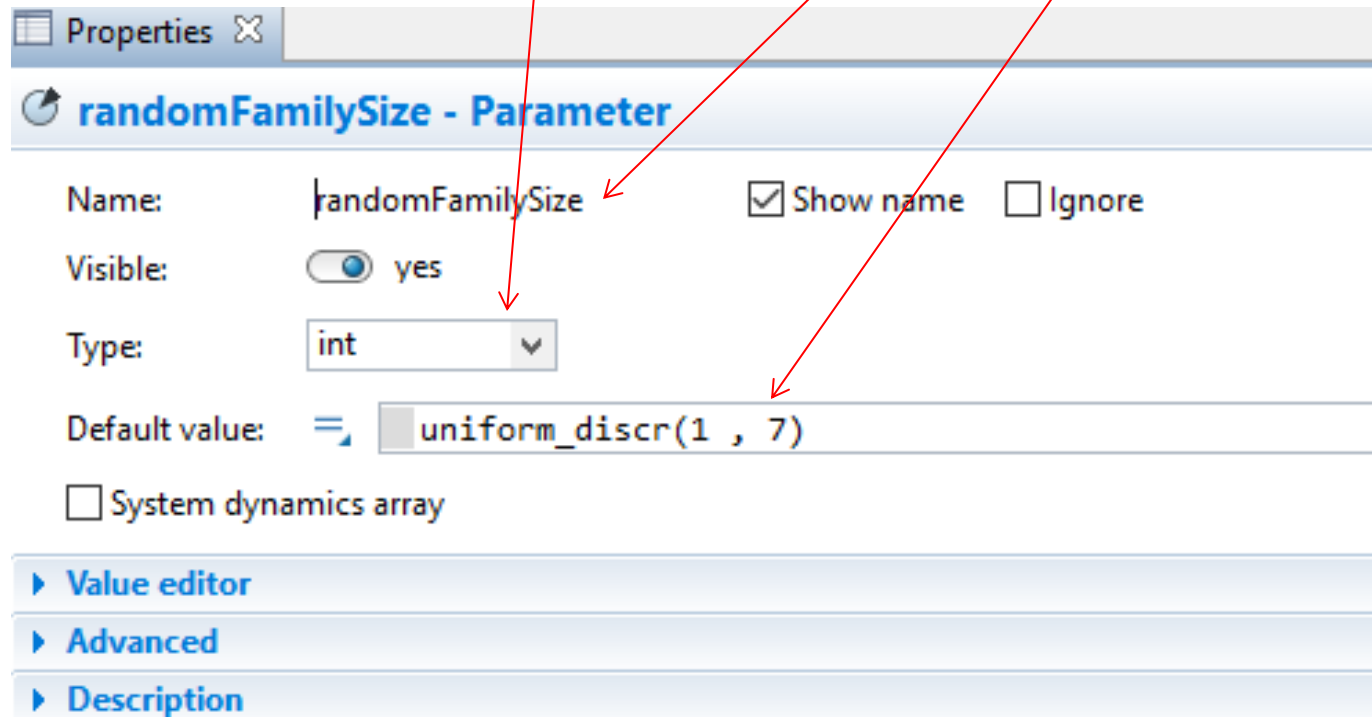
```
int maxcolumns = 3;
String cols[] = new String[maxcolumns];
int[] skip = {0,0,0};
while(propertiesFile.canReadMore()) {
while (propertiesFile.getLineNumber() <= 1) {
propertiesFile.skipTokens(1);}
System.out.println("Parsing line: " + propertiesFile.getLineNumber());
int id = propertiesFile.readInt();
double lat = propertiesFile.readDouble();
double lon = propertiesFile.readDouble();
Property p = add_properties(id, lat, lon);
p.setXY(lat, lon );
for (int i=0; i<=randomFamilySize; i++){
Person pe = add_people();
p.members.add( pe );
pe.jumpTo( lat, lon );}
int pos = 0;
while (pos < maxcolumns) {
if (skip[pos] == 1) {
propertiesFile.skipTokens(1);}
pos += 1;}}
```

The 'Advanced' and 'Description' sections are collapsed. A red arrow points from the text 'We need to define a parameter to hold the family size' to the 'randomFamilySize' variable in the code.

- To do this we drag and drop a parameter from the Agent group to the Main



- Change the name of this parameter to randomFamilySize.
- Change the Type to int.
- Use the `uniform_discr(1, 7)` for the Default value

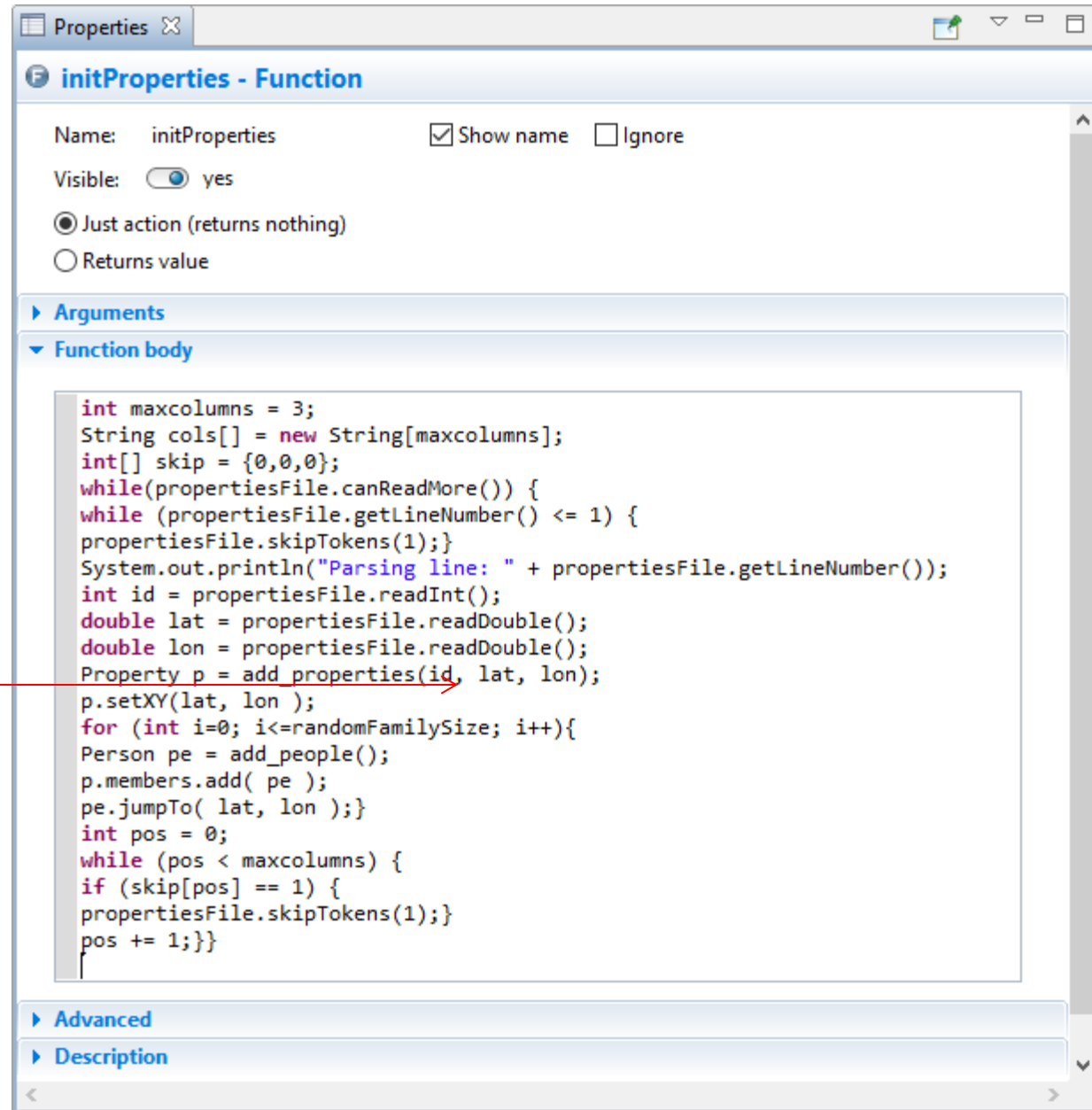


We use int for the type because it is for human count.

We use `uniform_discr` that is a uniform distribution of household family size between 1 to 7.

You can use other distributions.

- We also need to define the id, lat and lon parameters for the properties



The screenshot shows a 'Properties' window for a function named 'initProperties'. The window has a title bar with a close button and a search icon. Below the title bar, the function name 'initProperties - Function' is displayed. There are two checkboxes: 'Show name' (checked) and 'Ignore' (unchecked). Below these, there is a 'Visible' toggle set to 'yes'. There are two radio buttons: 'Just action (returns nothing)' (selected) and 'Returns value' (unselected). The 'Arguments' section is collapsed. The 'Function body' section is expanded, showing the following Java code:

```
int maxcolumns = 3;
String cols[] = new String[maxcolumns];
int[] skip = {0,0,0};
while(propertiesFile.canReadMore()) {
while (propertiesFile.getLineNumber() <= 1) {
propertiesFile.skipTokens(1);}
System.out.println("Parsing line: " + propertiesFile.getLineNumber());
int id = propertiesFile.readInt();
double lat = propertiesFile.readDouble();
double lon = propertiesFile.readDouble();
Property p = add_properties(id, lat, lon);
p.setXY(lat, lon );
for (int i=0; i<=randomFamilySize; i++){
Person pe = add_people();
p.members.add( pe );
pe.jumpTo( lat, lon );}
int pos = 0;
while (pos < maxcolumns) {
if (skip[pos] == 1) {
propertiesFile.skipTokens(1);}
pos += 1;}}
```

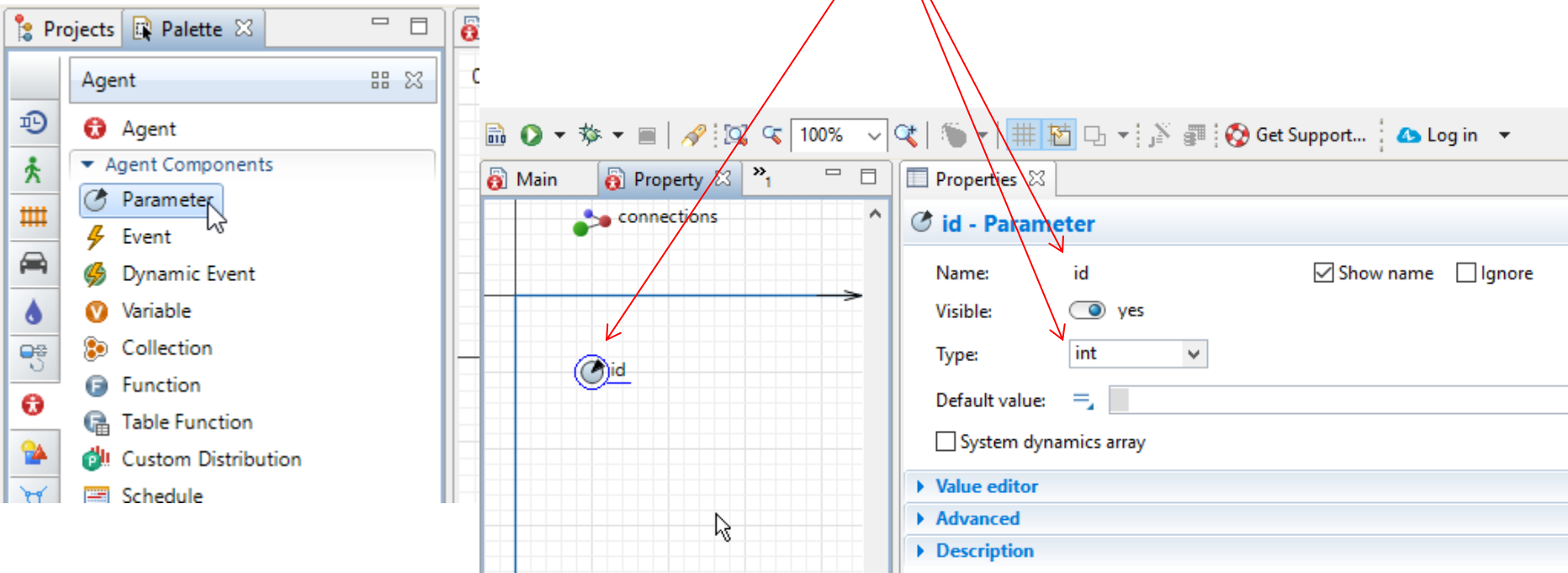
At the bottom of the window, there are two collapsed sections: 'Advanced' and 'Description'.

- To do this you need to Open the property agent page by double clicking on the Agent type Property.

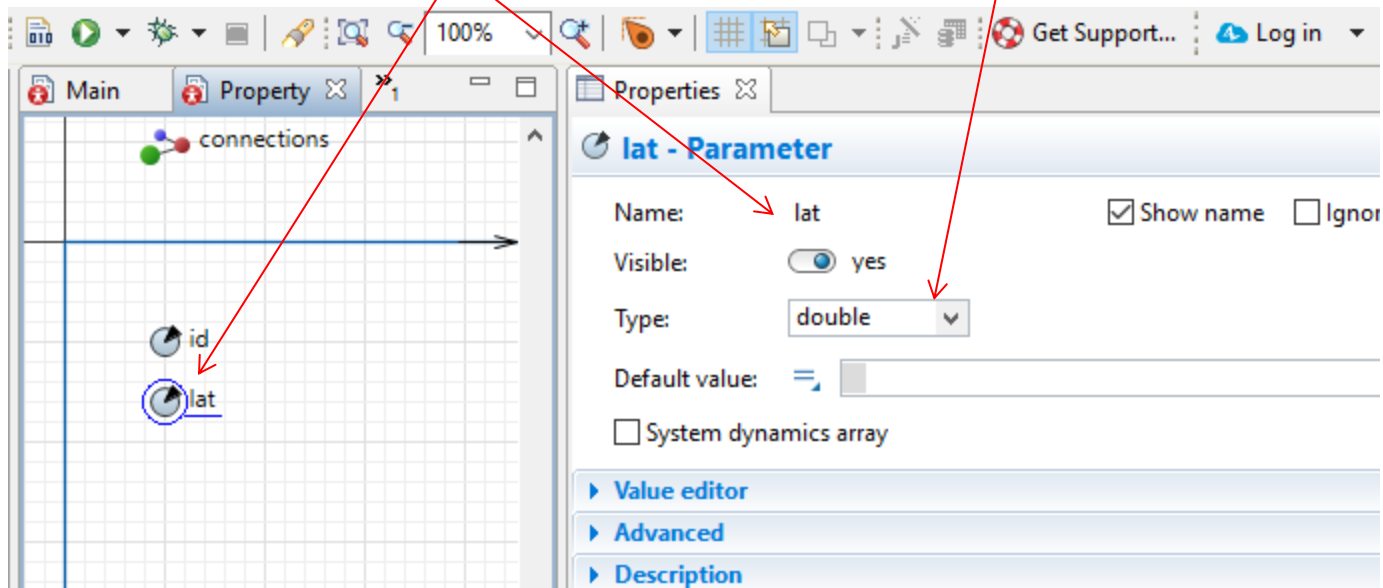
The screenshot displays a software interface with three main components:

- Left Panel (Project Tree):** A tree view under 'FloodModelV1*' containing folders for 'Flood', 'Main', 'Person', 'Property', 'Sensor', 'Simulation: Main', 'Run Configuration: Main', and 'Database'. The 'Property' folder is selected and highlighted in blue. A red arrow points from the text above to this folder.
- Central Workspace:** A grid-based workspace with a 'main' icon (a circular arrow) and a 'connections' icon (a network diagram). A blue horizontal line with an arrow at its right end is drawn across the workspace.
- Right Panel (Properties):** A 'Properties' window titled 'Property - Agent Type'. It includes a 'Name' field with the value 'Property' and an 'Ignore' checkbox. Below are sections for 'Agent actions' (with fields for 'On startup:', 'On destroy:', 'On arrival to target location:', 'On before step:', and 'On step:') and 'Agent in flowcharts' (with a dropdown menu set to 'Agent' and a field for 'On enter flowchart block:').

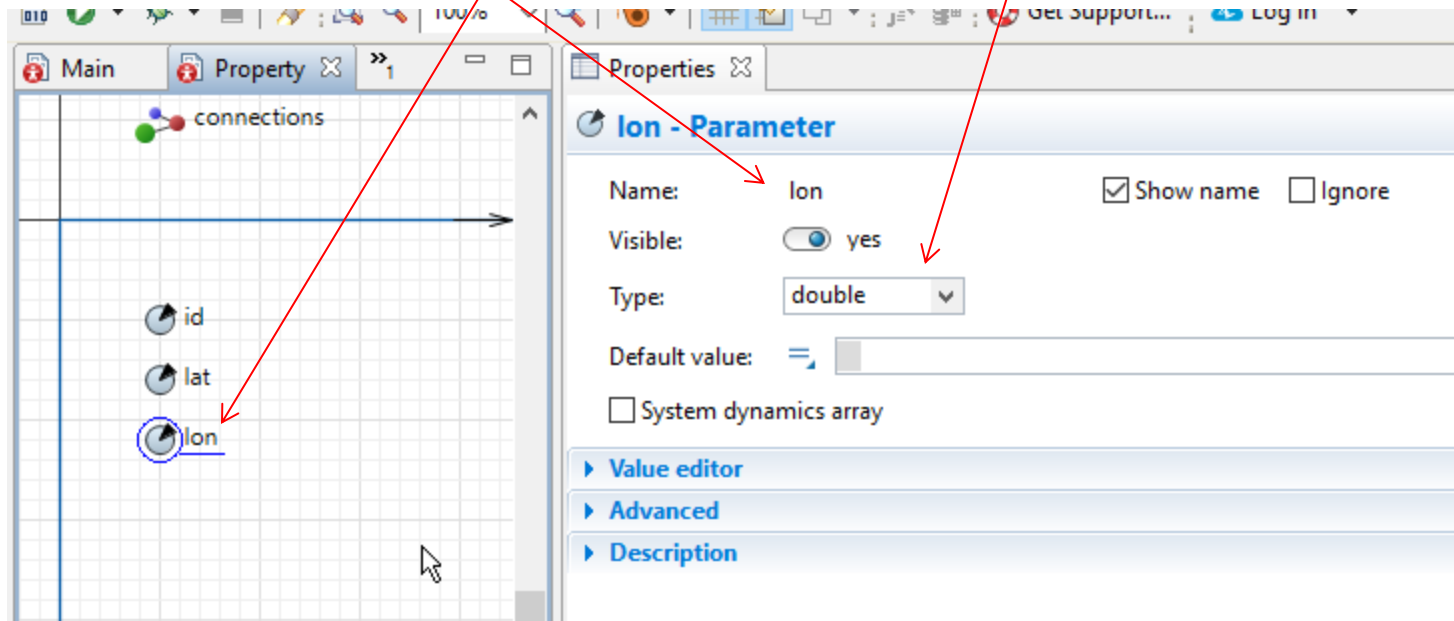
- Now drag and drop three parameters into the Property page and change their names and types accordingly.



- Again we add a parameter and change its name to lat and its type to double

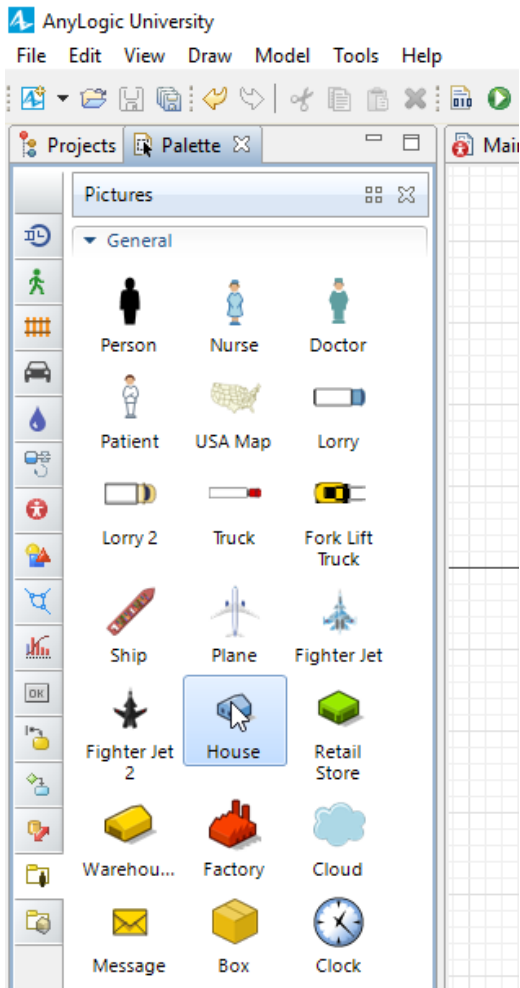


- Again we add a parameter and change its name to lon and its type to double

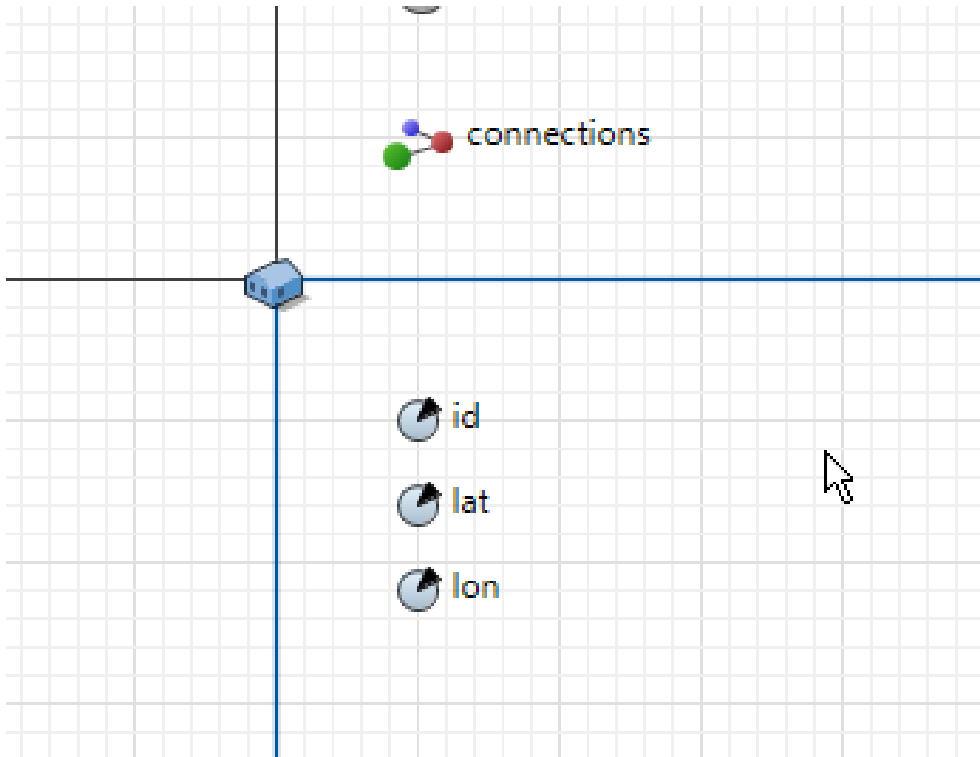


- Adding an icon for properties

From the Platte choose Picture group and select House and drag one to the Property page.



Make sure you put it in the cross point.



- At this point we also add a collection to the Property to collect the family members or people in the properties.

To do so, drag and drop a collection into the Property page.

Change its name to “members”.

Change the element class to Person

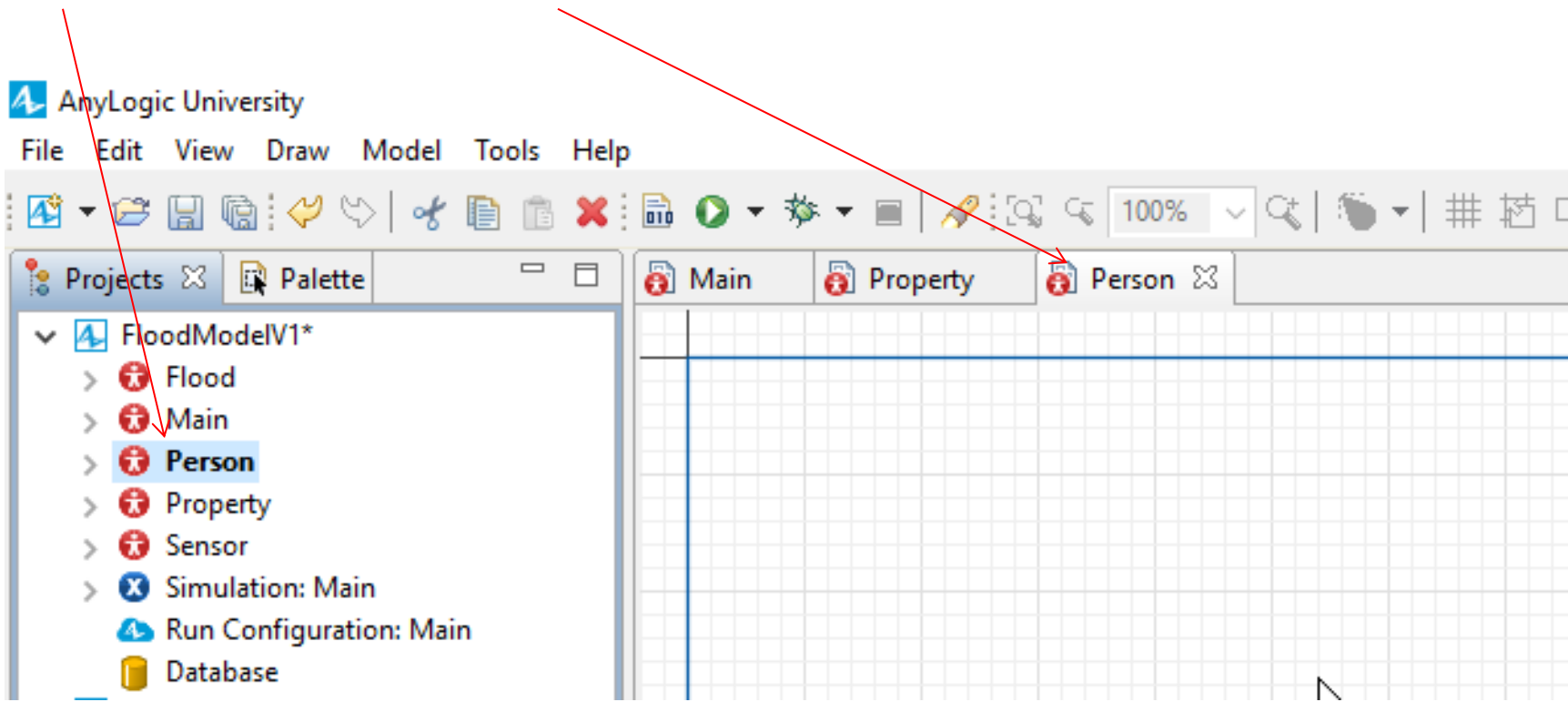
The screenshot displays the AnyLogic University software interface. The main workspace shows a statechart diagram with a 'main' state and a 'connections' state. A 'members' collection property is being added to the 'Person' state. The 'Properties' panel on the right is configured as follows:

- Name: members
- Visible: yes
- Collection class: ArrayList
- Elements class: Person

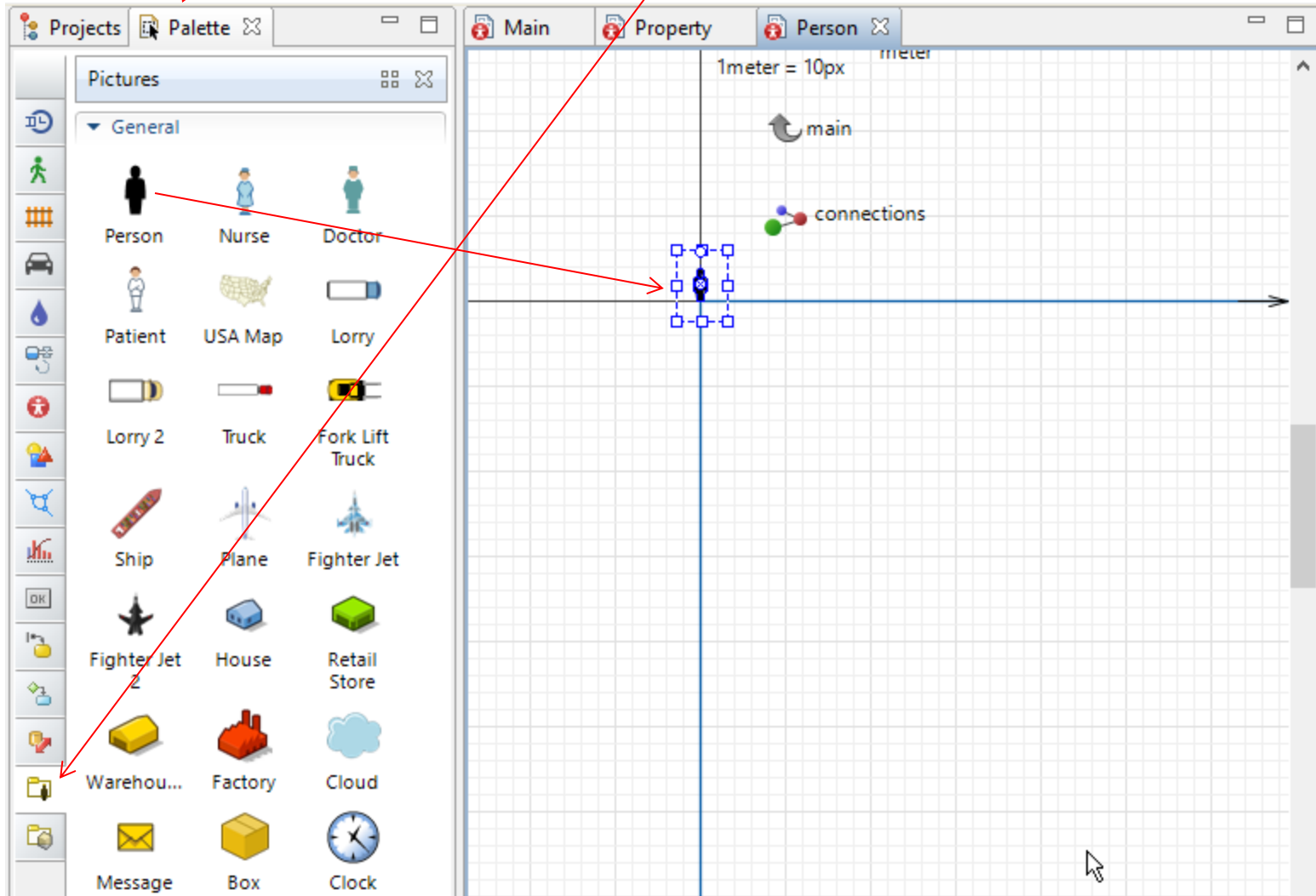
The 'Advanced' section of the Properties panel is also visible, showing an 'Initial contents' field.

- Adding an icon for Person

To do this you need to Open the Person agent page by double clicking on the Agent type Person.

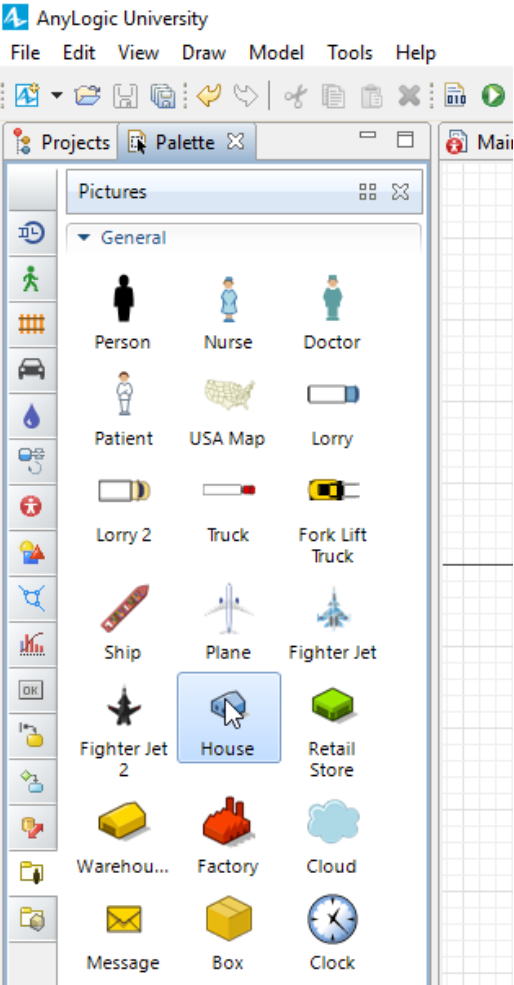


- Open the Palette and choose Picture group and drag and drop Person icon into the Person page

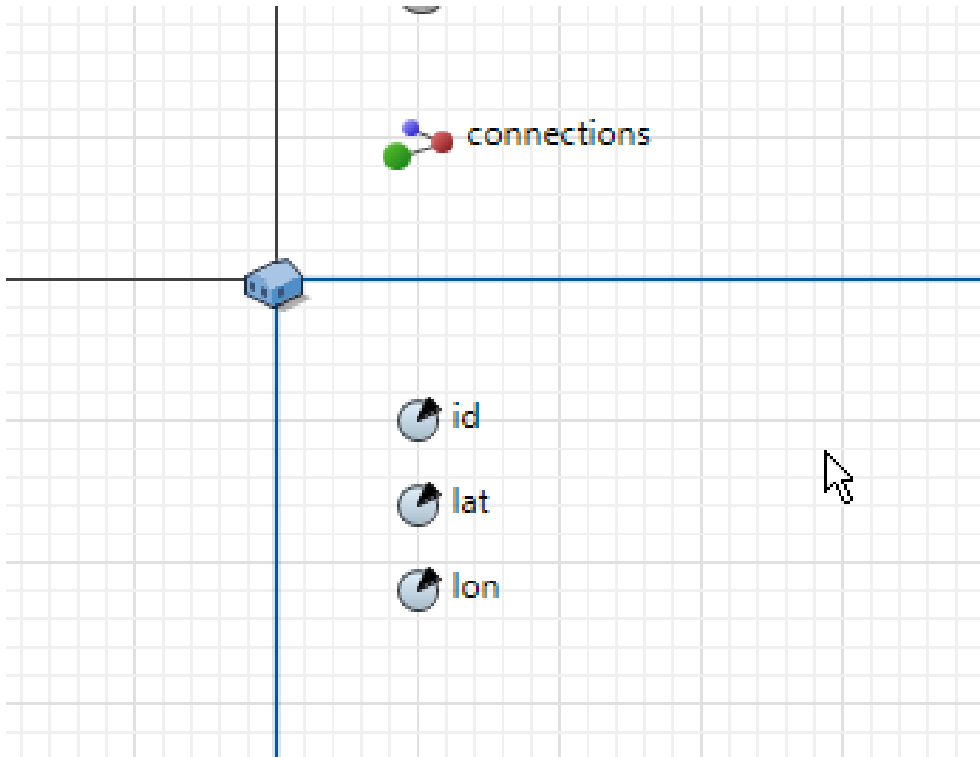


• Adding an icon for Person

To do this you need to Open the property agent page by double clicking on the Agent type Property.

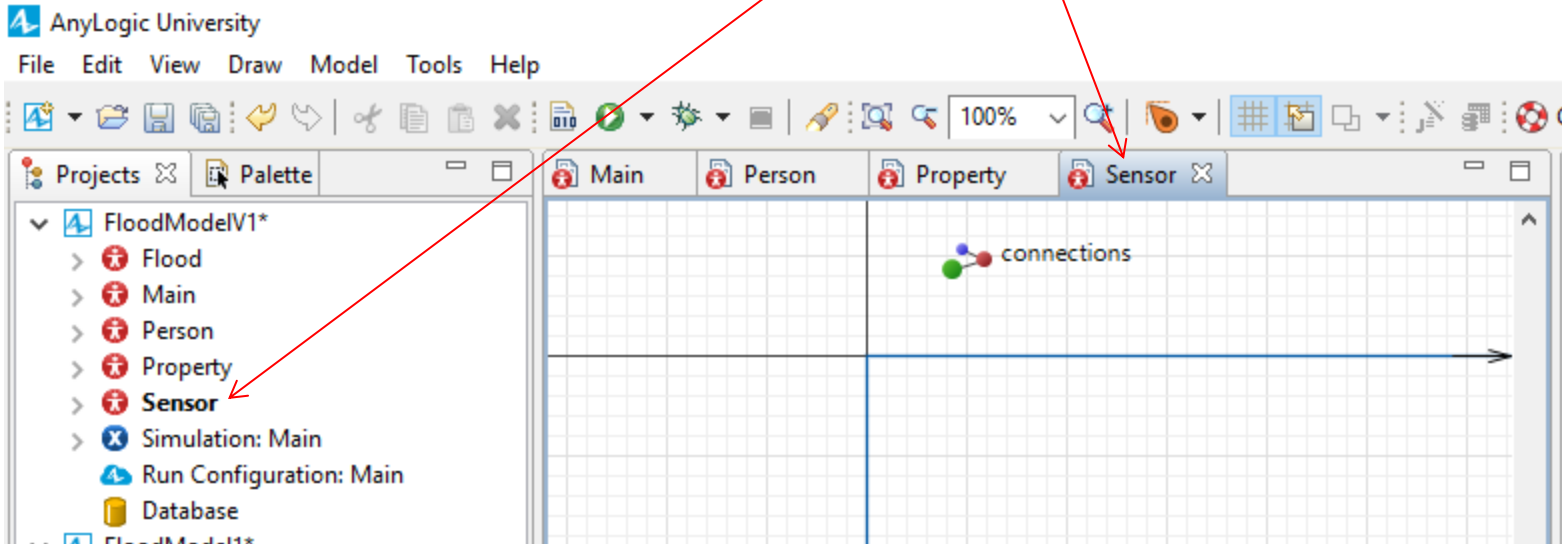


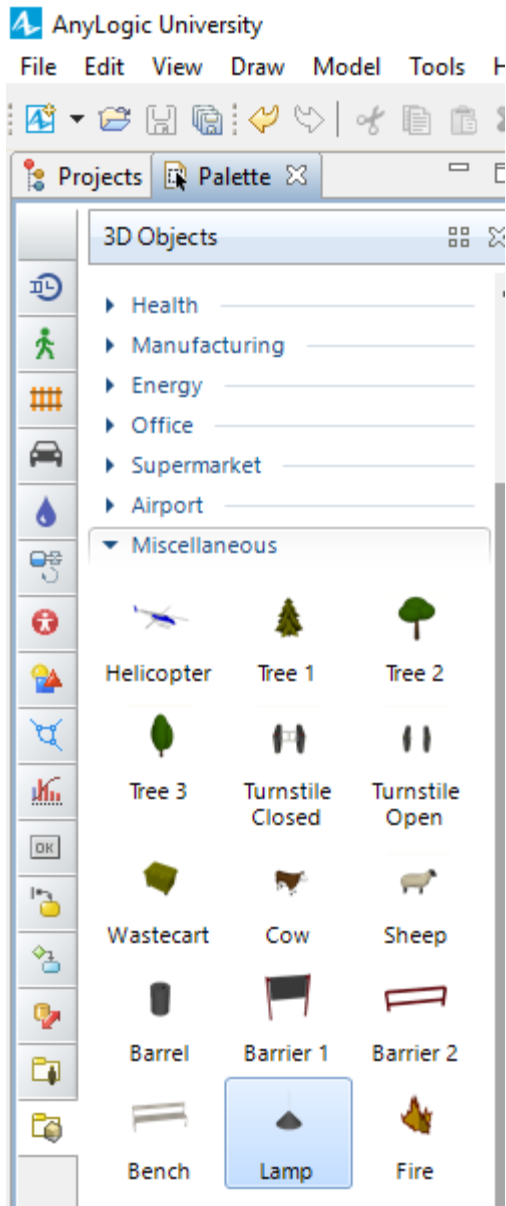
Make sure you put it in the cross point.



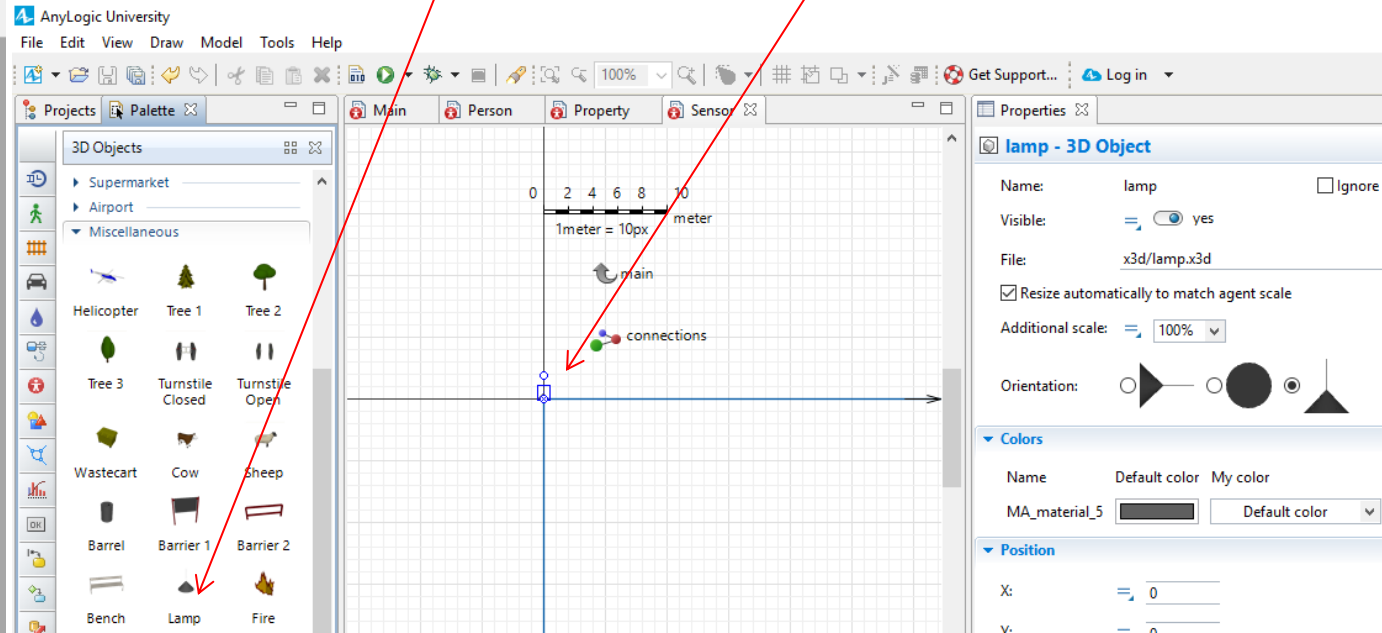
- Adding an icon for Sensor

To do this you need to Open the Sensor agent page by double clicking on the Agent type Sensor.



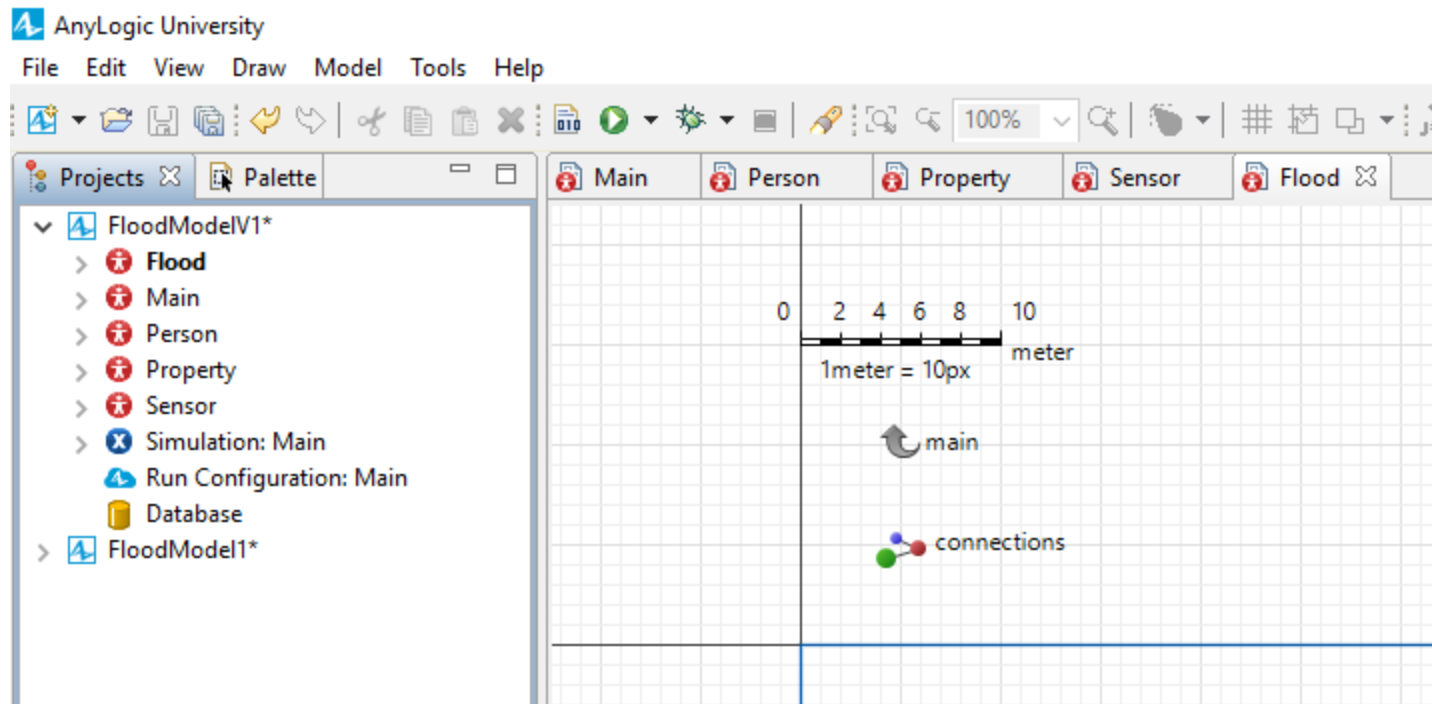


- Click on the Palette and select 3D objects and choose Miscellaneous group and then drag and drop a Lamp to the Sensor page

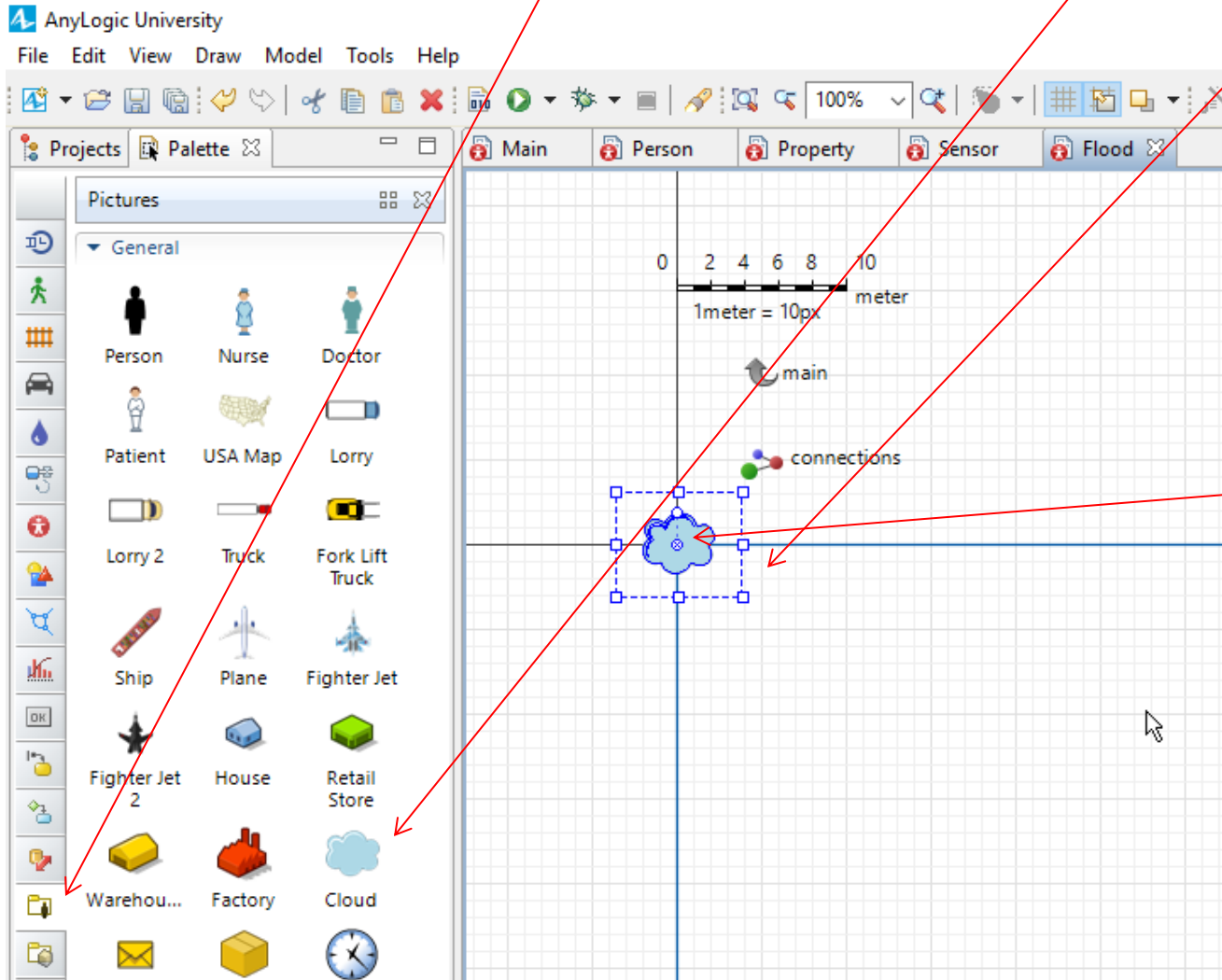


- Adding an icon for Flood

To do this you need to Open the property agent page by double clicking on the Agent type Flood.



From the Palette choose Picture group and select cloud and drag one to the Flood page.



Make sure you put it in the cross point.

Make sure that all agents icons are in Show presentation mode.

- To check this go to the Main page.
- Click on each agent's icon and in the property of agent population press the Show presentation button.

The screenshot displays the AnyLogic University software interface. On the left, a project tree shows a hierarchy of models and agents. The main workspace shows a map with a blue dashed line representing a path and a red location pin. A central panel lists agents: 'flood', 'sensor', 'properties [...]', 'people [...]', 'propertiesFile', 'initProperties', and 'randomFamilySize'. The 'flood' agent is selected, and its properties are shown in a right-hand pane. The 'Properties' pane for 'flood - Flood' includes sections for 'Movement' (Initial speed: 10), 'Initial location' (Place agent(s) options: at the agent animation location, in the latitude/longitude, in the node, in the first result of map search), 'Node' (gisPoint), 'Routing' (network), 'Statistics', 'Advanced' (Model/library: FloodModelV1, Visible: yes, Log to database checked), and 'Description'. A 'Show presentation' button is highlighted in the 'Advanced' section. Red arrows from the text above point to the 'flood' agent icon in the central panel and the 'Show presentation' button in the properties pane.

- Same for sensor.

The image shows a simulation interface with a central map area. On the left, a sidebar lists various elements: 'flood', 'sensor' (highlighted with a red circle and arrow), 'properties [...]', 'people [...]', 'propertiesFile', 'initProperties', and 'randomFamilySize'. The map displays a terrain with a blue dashed line, a solid blue line, and a yellow line. A red location pin is placed on the map. On the right, a configuration panel is visible. It includes an 'Initial speed' field set to 0. The 'Initial location' section has radio buttons for 'at the agent animation location', 'in the latitude/longitude' (selected), 'in the node', and 'in the first result of map search'. Below this are input fields for 'Latitude: 37.91183' and 'Longitude: 55.97919', and a 'Routing' dropdown set to 'Default'. The 'Advanced' section contains a 'Model/library' field set to 'FloodModelV1', a 'Visible' toggle set to 'yes', and a checked 'Log to database' checkbox with a link to 'Turn on model execution logging'. A 'Show presentation' button is located below the 'Log to database' checkbox. A red arrow points from the 'sensor' icon in the sidebar to the 'Log to database' checkbox.

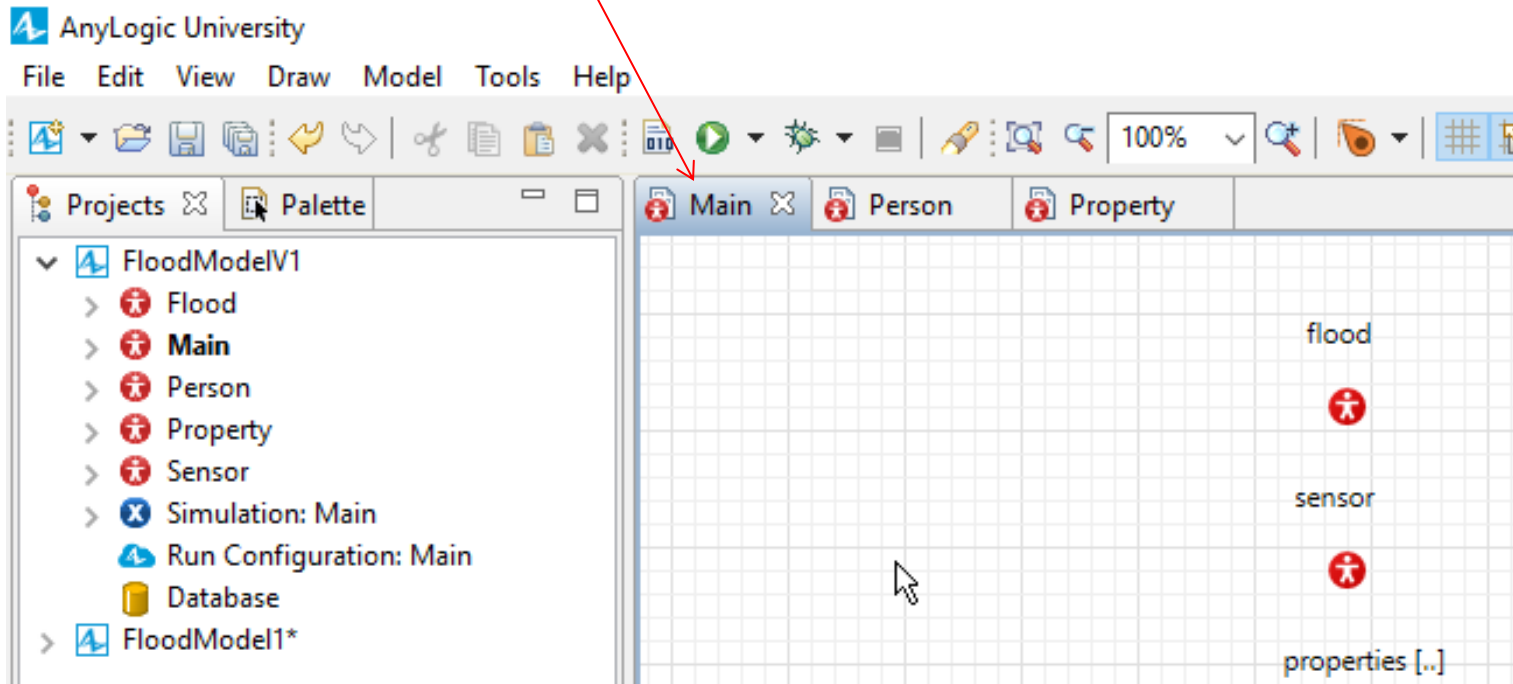
- Same for properties.

The screenshot displays the NetLogo environment. On the left, a grid-based workspace contains several objects: 'flood', 'sensor', 'properties [..]', 'people [..]', 'propertiesFile', 'initProperties', and 'randomFamilySize'. A red arrow points from the text 'Same for properties.' to the 'properties [..]' object. The main map area shows a terrain with a blue dashed line representing a path and a red location pin. On the right, the 'Properties' panel for the selected 'properties - Property' object is visible. It includes settings for 'Initial speed' (10 meters), 'Initial location' (selected as 'at the agent animation location'), 'Routing' (Default), and 'Advanced' options like 'Model/library' (FloodModelV1), 'Visible' (yes), and 'Optimize for' (Access by index (ArrayList)). A 'Show presentation' button is located at the bottom of the panel.

- Same for people.

The screenshot shows the NetLogo interface with a 'people - Person' properties window open. The left sidebar contains a list of objects: 'flood', 'sensor', 'properties [..]', 'people [..]', 'propertiesFile', 'initProperties', and 'randomFamilySize'. The 'people [..]' entry is selected and underlined. A red triangle is drawn over the 'people [..]' entry and the 'Log to database' checkbox in the properties window. The properties window shows various settings for the 'Person' object, including 'Initial speed' (10 meters), 'Initial location' (at the agent animation location), 'Routing' (Default), 'Model/library' (FloodModelV1), 'Visible' (yes), 'Optimize for' (Access by index (ArrayList)), and 'Log to database' (checked). A 'Show presentation' button is also visible.

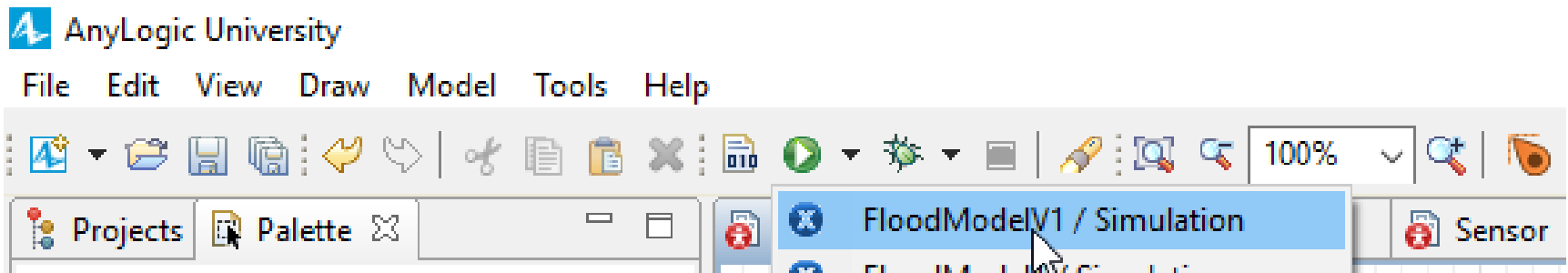
- Now we just need to call the `initProperties` function and run the model and check if it is working.
- Click on the **Main** page to view it.



- Click somewhere in the Main page.
- Call the `initProperties` function by typing the `initProperties();`
- In the On startup section of the main.

The screenshot displays the AnyLogic University software interface. The main workspace shows a grid with several elements: 'flood', 'sensor', 'properties [..]', and 'people [..]'. A 'connections' diagram is visible in the top right corner. The Properties panel on the right is titled 'Main - Agent Type' and shows the 'On startup' section with the code `initProperties();` entered. Red arrows point from the text in the list above to the 'Main' page in the workspace and the 'On startup' section in the Properties panel.

- Now you can run the model and see the results of your work up until here



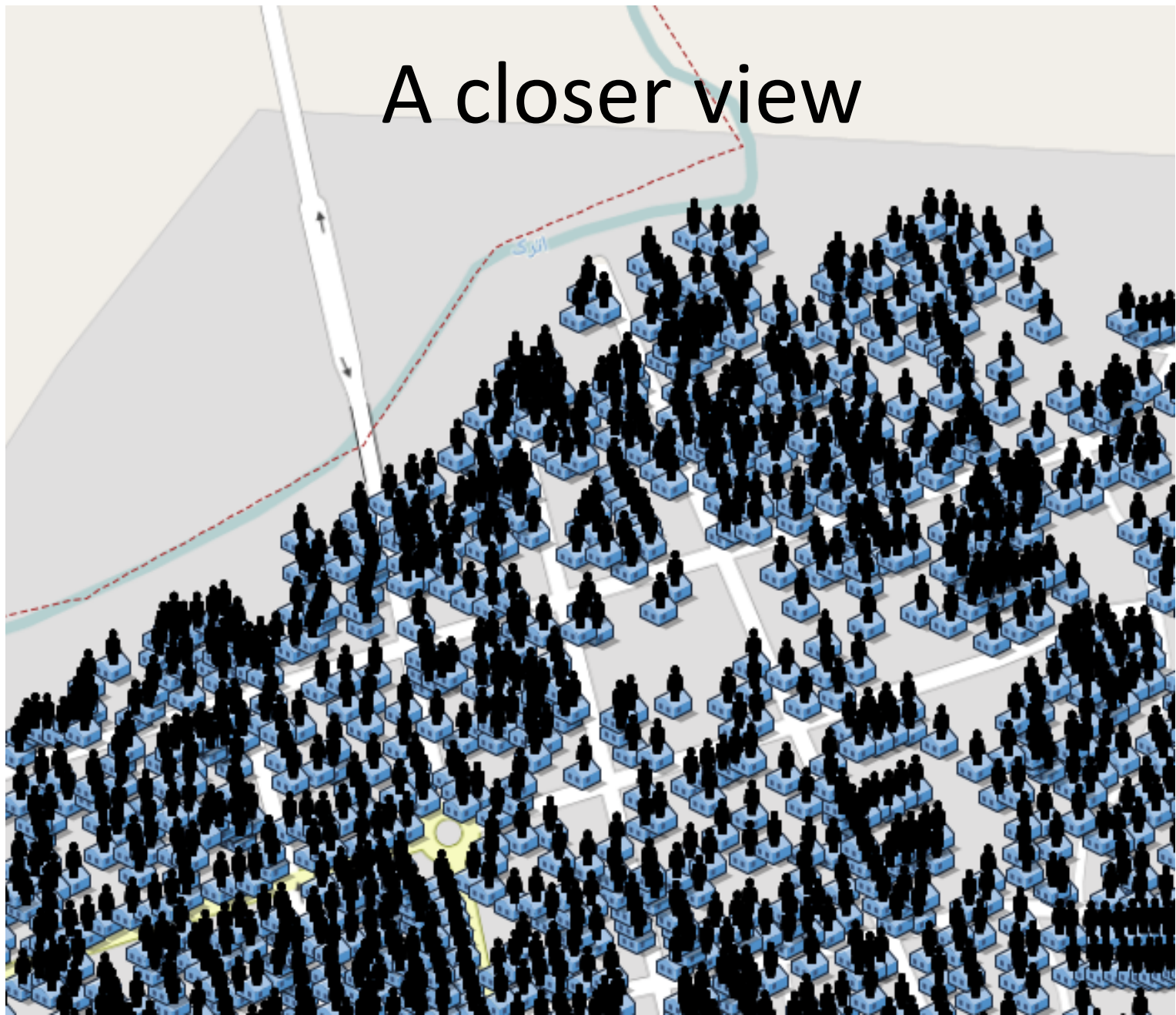
If you have no error your run should look like this.
As you can see 2259 properties and 15813 persons have been added.

Flood beginning

The screenshot displays the AnyLogic simulation environment for a flood model. The top window title is "FloodModelV1 : Simulation - AnyLogic University [PUBLIC RESEARCH USE ONLY]". The toolbar includes standard simulation controls like play, stop, and zoom. The left-hand palette lists objects: "Flood Flood", "sensor Sensor", "properties Property [2259]", "people Person [15813]", "propertiesFile TextFile. File: C:/Users/lesou...", "initProperties", and "randomFamilySize 6". The central 3D view shows a city with a river and a flood area. A sensor is positioned on the river. The bottom status bar indicates the simulation is running, with a time of 26.20 and a date of May 6, 2017. The memory usage is shown as 111M of 228M.

Sensor

A closer view



- Save your project.
- In the next lesson we define the behavior and movements of flood and people.
- Please let me know if you have any question or problem with your model.