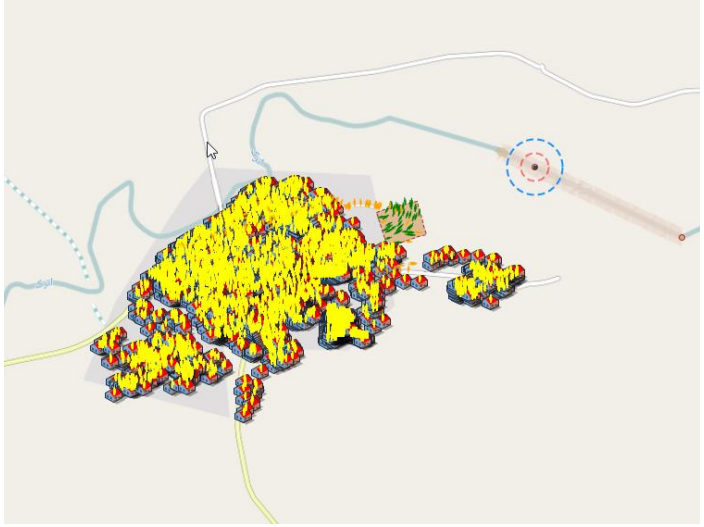
Disaster Simulation: An AnyLogic Agent-Based Approach

Example:
Flood simulation and evacuation
GIS Environment



Ali Asgary
ADERSIM
York University
2017

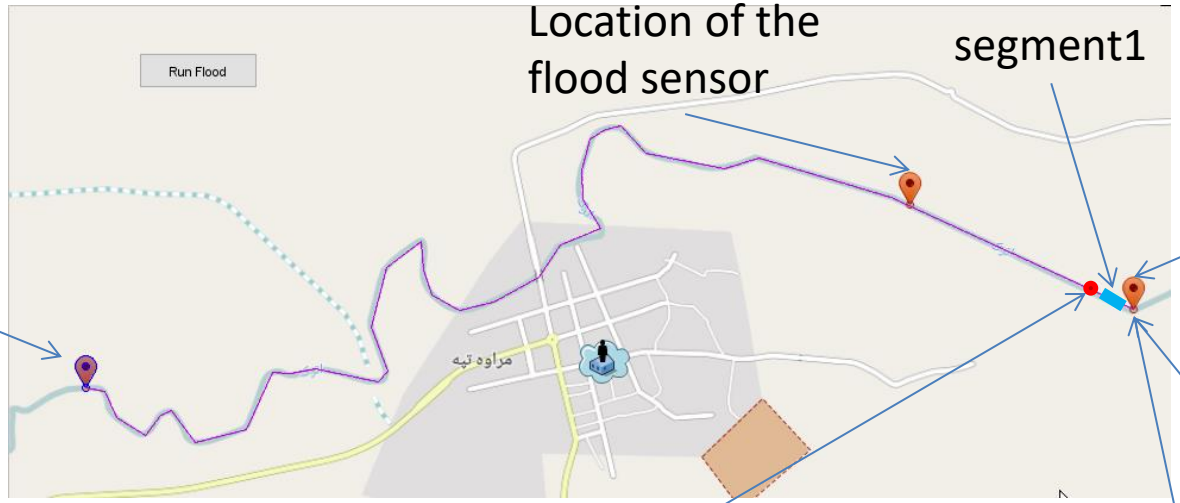
Lesson 3: Defining Agents Behavior-Flood

- Defining GISPoints
- Defining GISSegments and Routs
- Defining agents (flood) move on a path

Defining a simple event for flood agent

We want flood agent to move in the river path.

To do this we need to move flood agent by shifting it from one point (point1) to another point (point2) on the river path and then connect each two points (segment) and creating flood path.



Location of the flood sensor

segment1

gisPoint

Start of flood

gisPoint1

End of flood

point1

Second point of flood (point2)- after 1 second move

First point of flood simulation (point1)

Define some variables

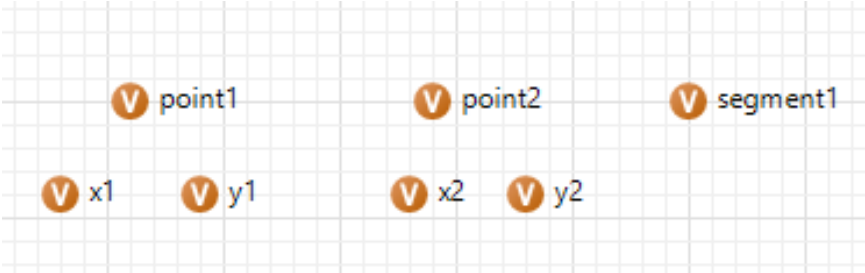
We need to define two points (GISPoint):

point1 and point2

We need to define one segment (GISMarkupSegmentLine): segment1

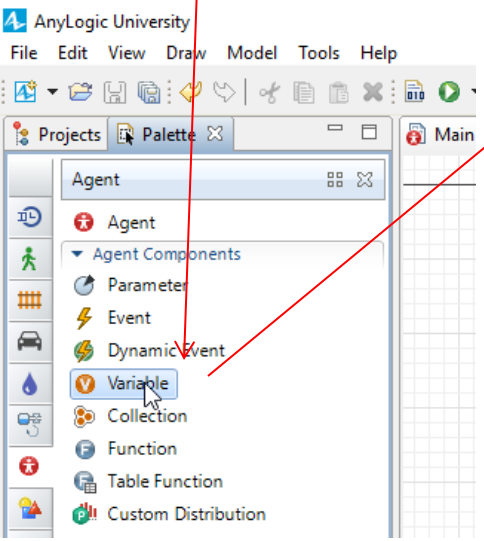
We need to define x (longitude) and y (latitude) for each of the above points:

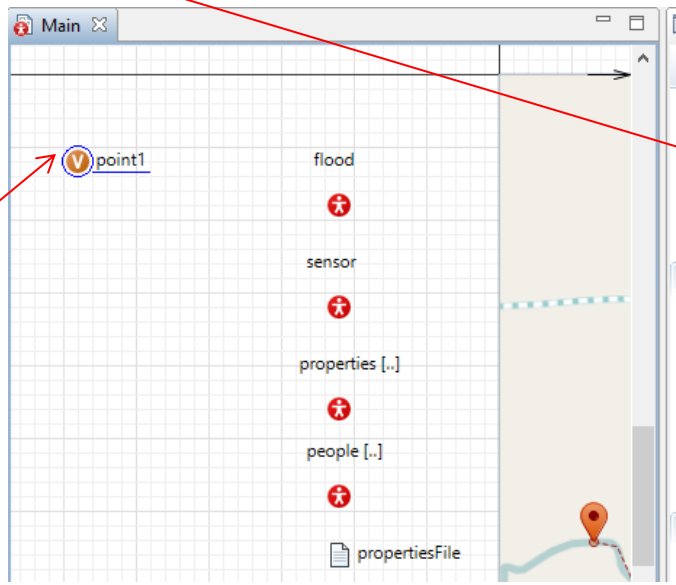
x1, y1 for point1 and x2 and y2 for point2

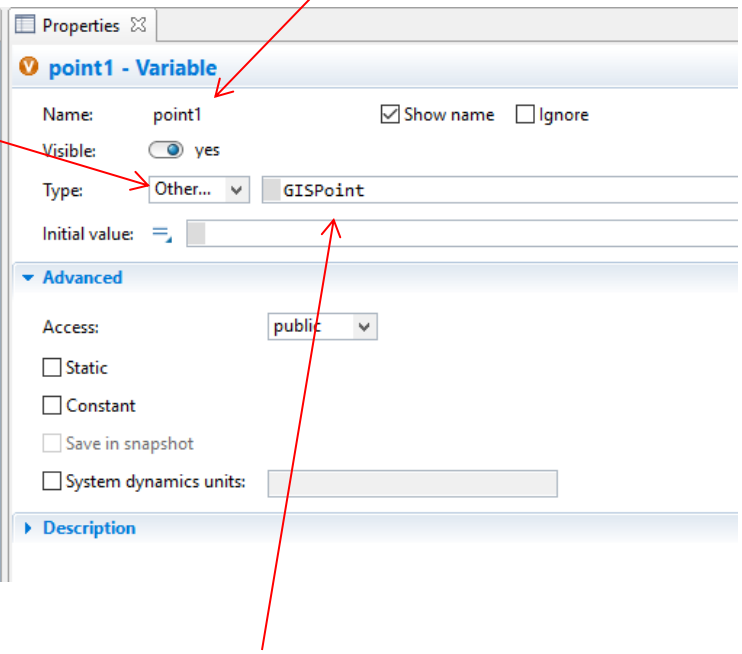


Define point1

Drag and drop a variable into the Main and define its properties. Change its name to point1. Choose Other for the type.



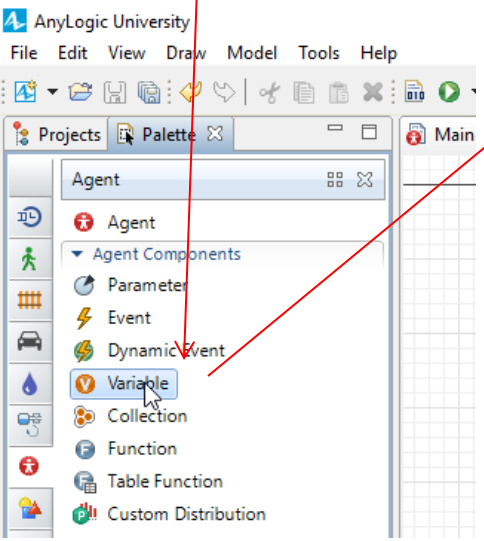


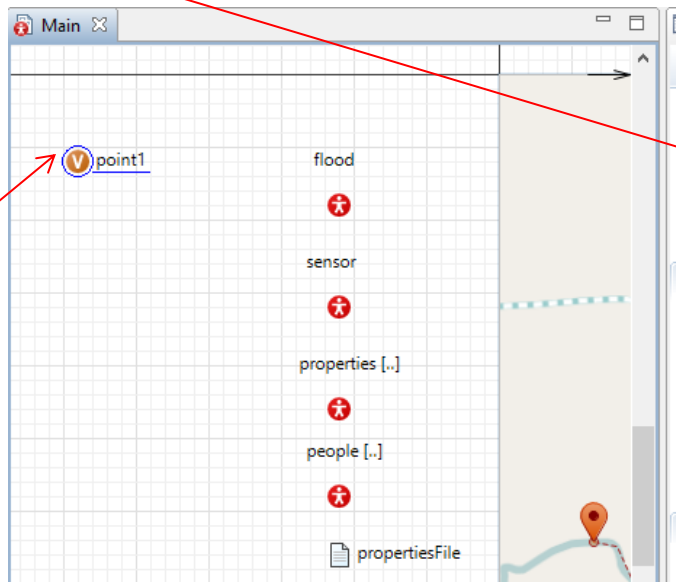


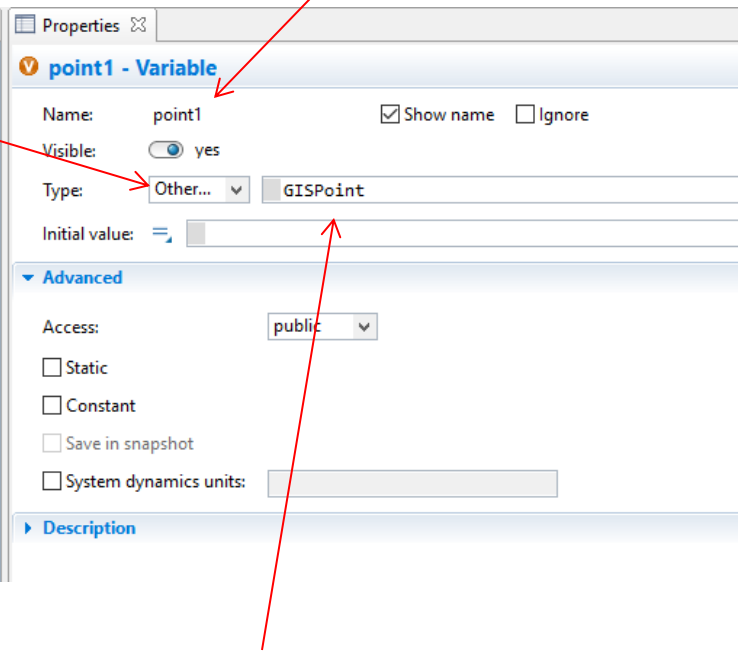
Change Double to GISPoint for the Type

Define point2

Drag and drop a variable into the Main and define its properties. Change its name to point2. Choose Other for the type.



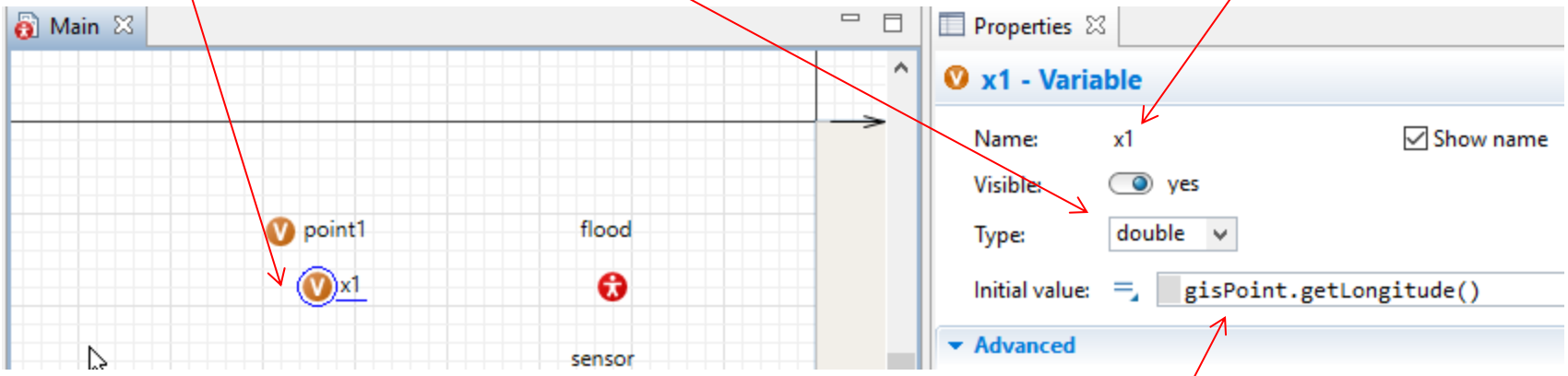




Change Double to GISPoint for the Type

Define x1 for point1

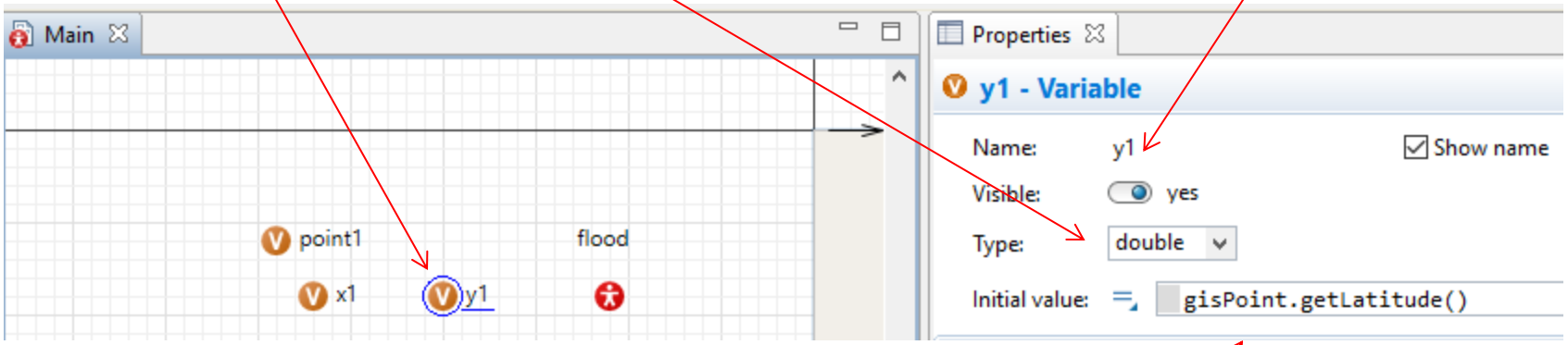
Drag and drop a variable into the Main and change its properties. Change its name to x1. Choose double for the type.



write the initial value as: gisPoint.getLongitude()

Define y1 for point1

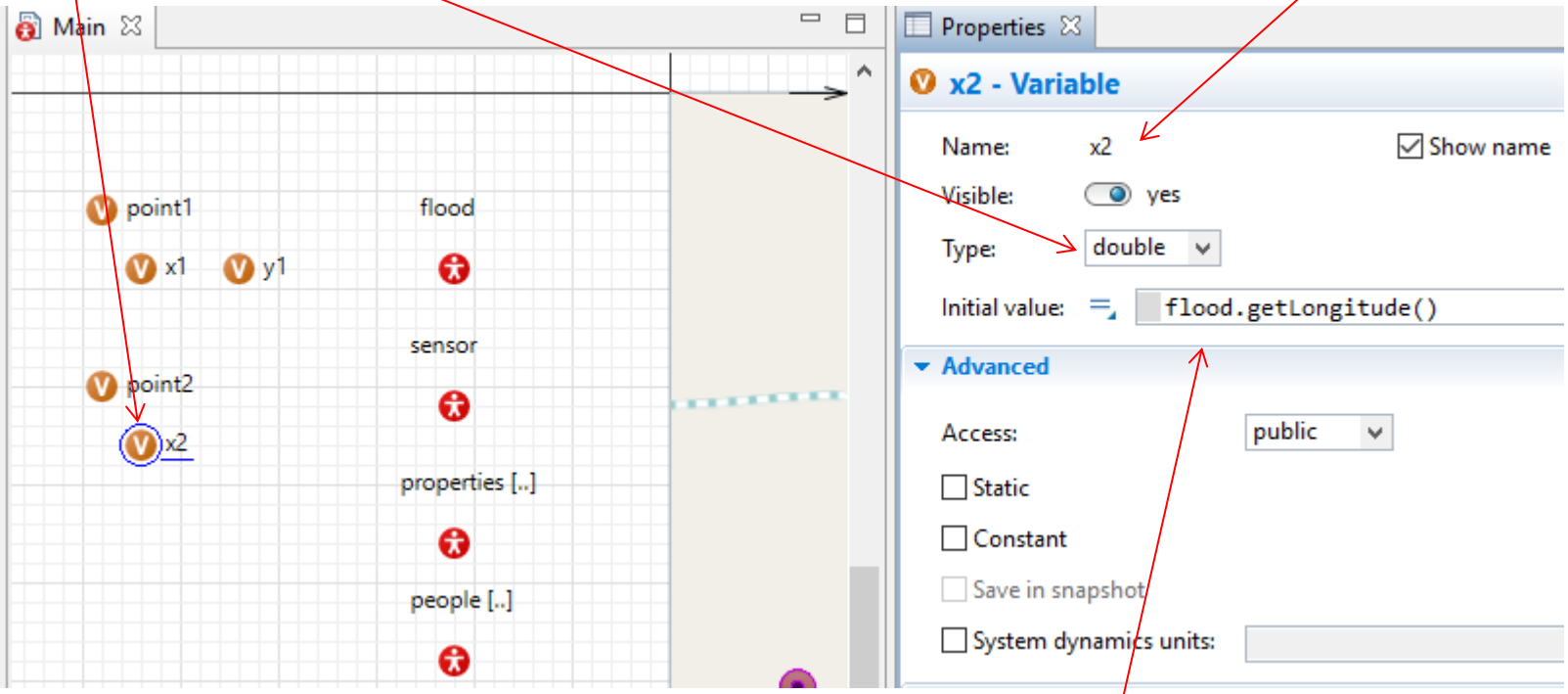
Drag and drop a variable into the Main and change its properties. Change its name to y1. Choose double for the type.



write the initial value as: gisPoint.getLatitude()

Define x2 for point2

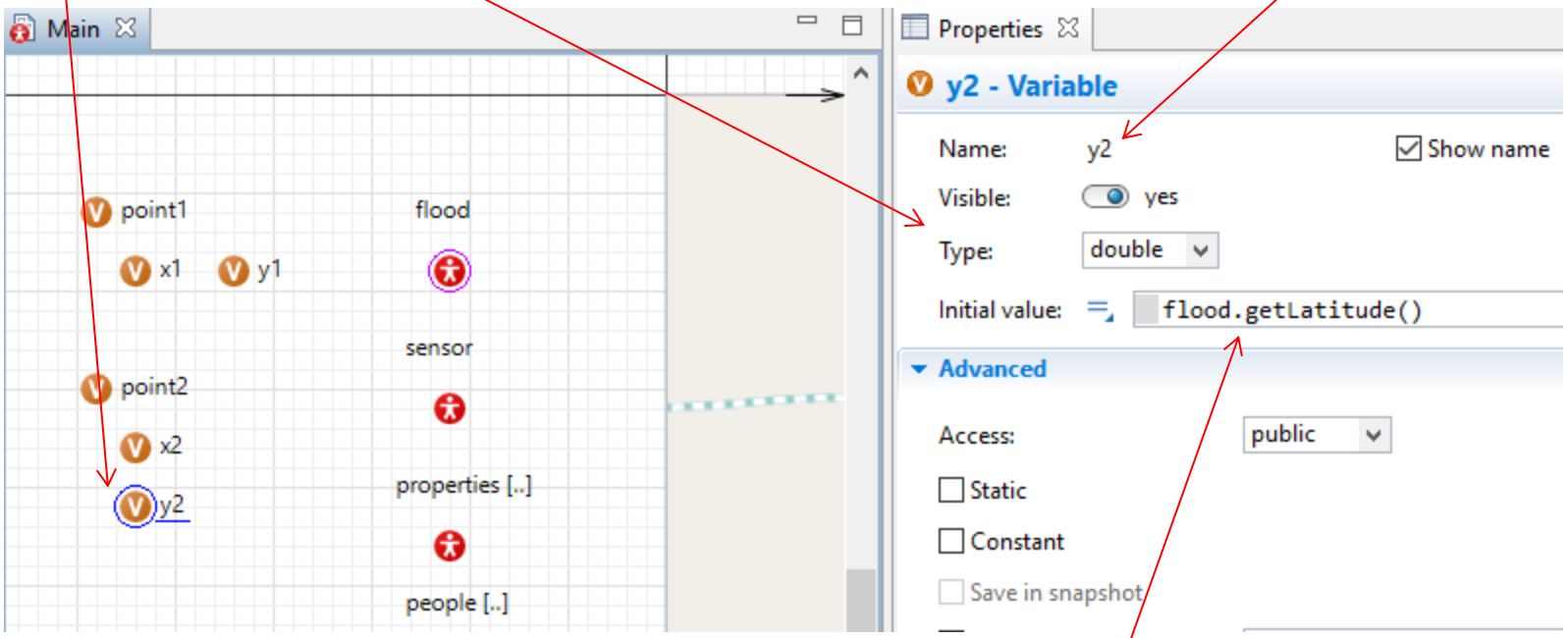
Drag and drop a variable into the Main and change its properties. Change its name to x2. Choose double for the type.



write the initial value as: flood.getLongitude()

Define y2 for point2

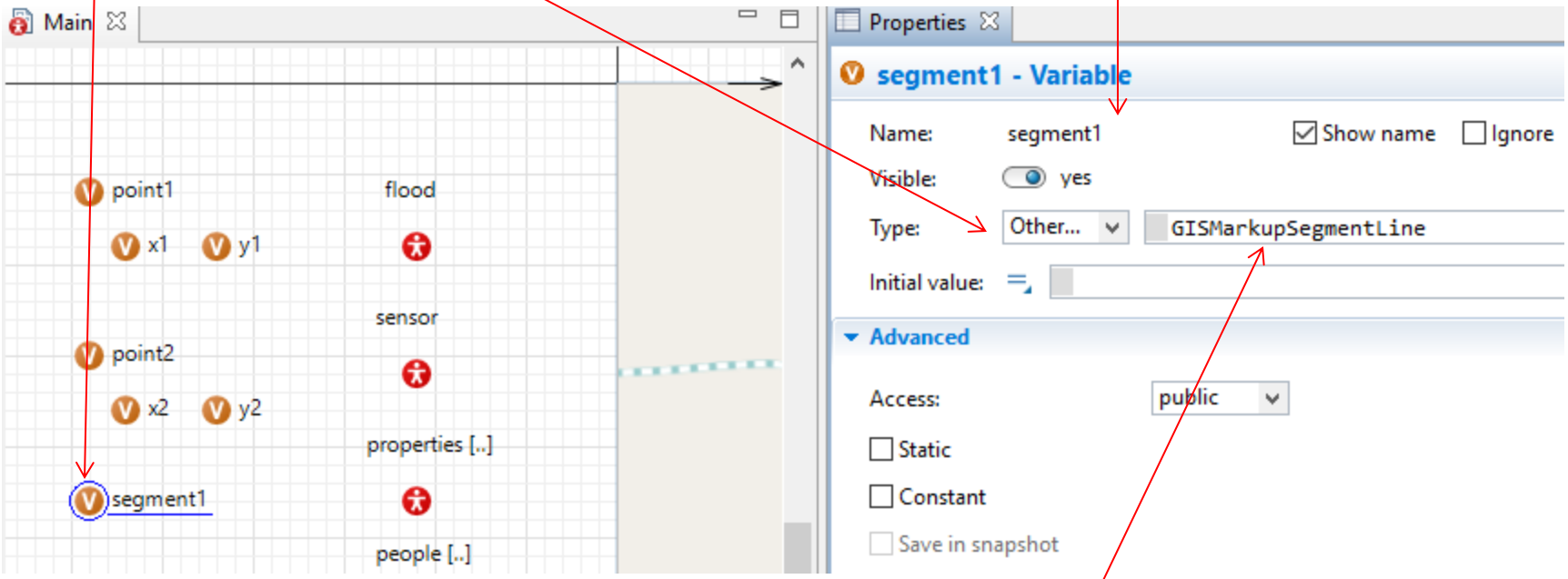
Drag and drop a variable into the Main and change its properties. Change its name to y2. Choose double for the type.



write the initial value as: flood.getLatitude()

Define segment1

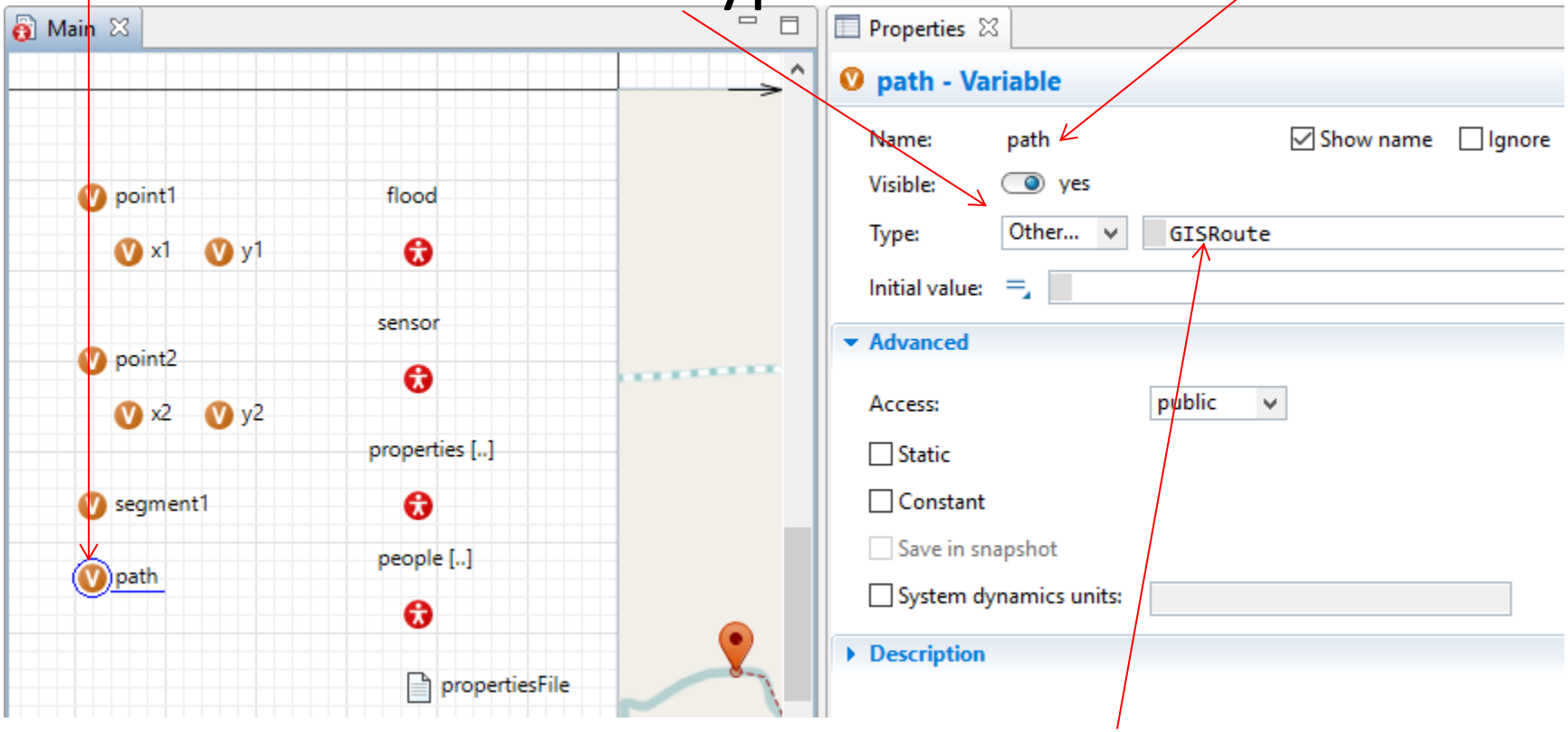
Drag and drop a variable into the Main and change its properties. Change its name to segment1. Choose other for the type.



Change the type to : GISMarkupSegmentLine

Define path

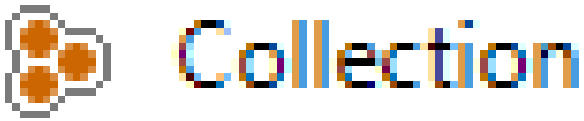
Drag and drop a variable into the Main and change its properties. Change its name to path. Choose other for the type.

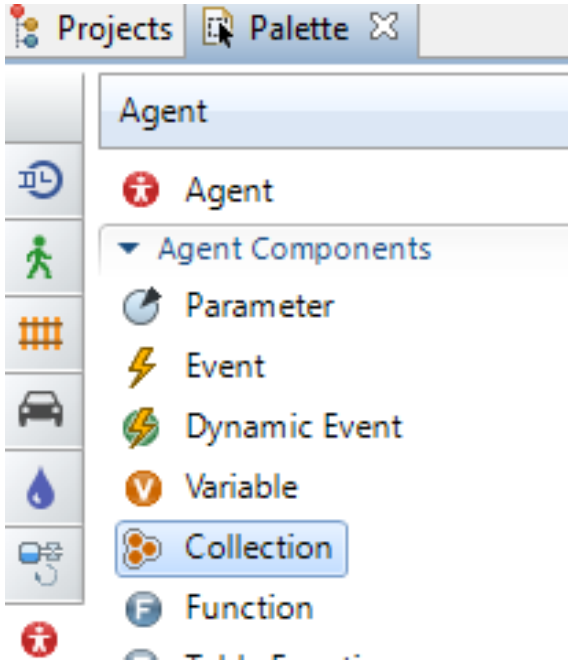


Change the type to : GISRoute

Define two collections to collect points and paths

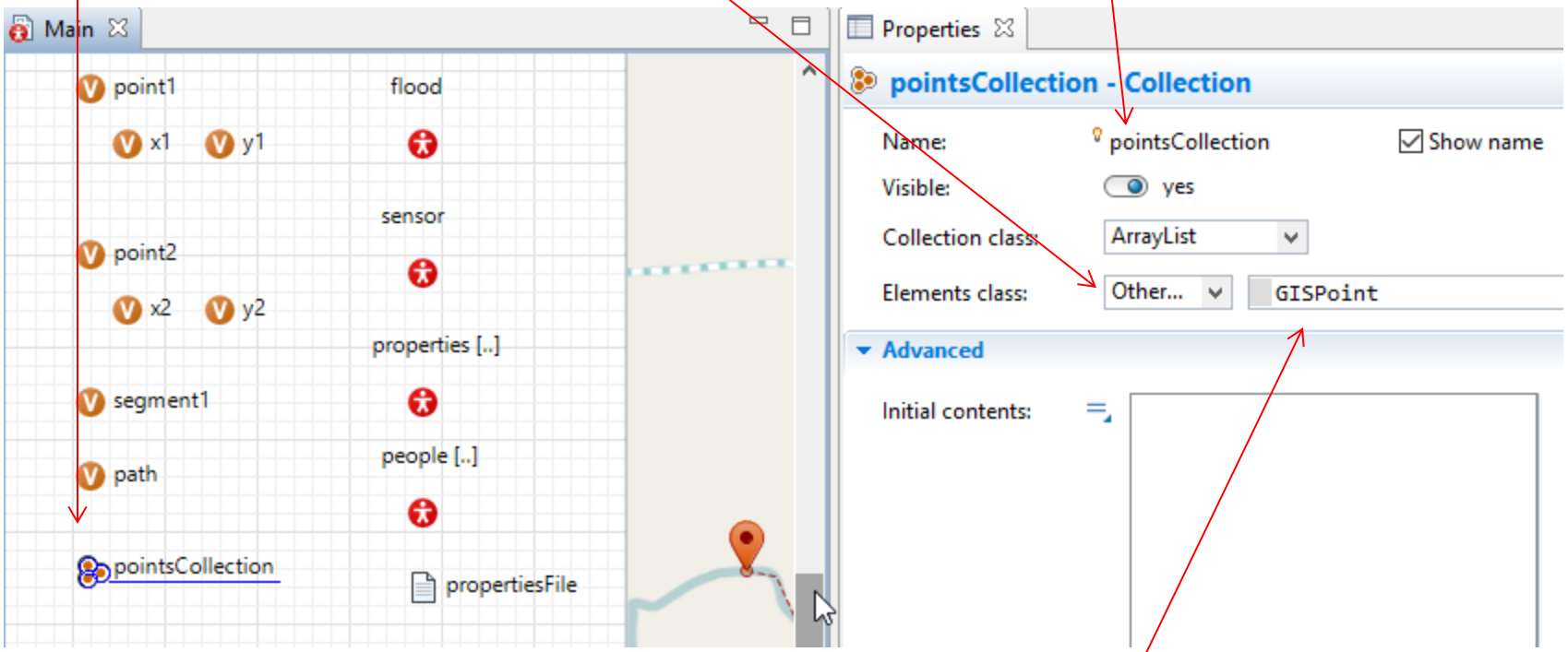
- In order to connect all the points and paths along the river, we create two collections to hold all these point and paths.





Define pointsCollection

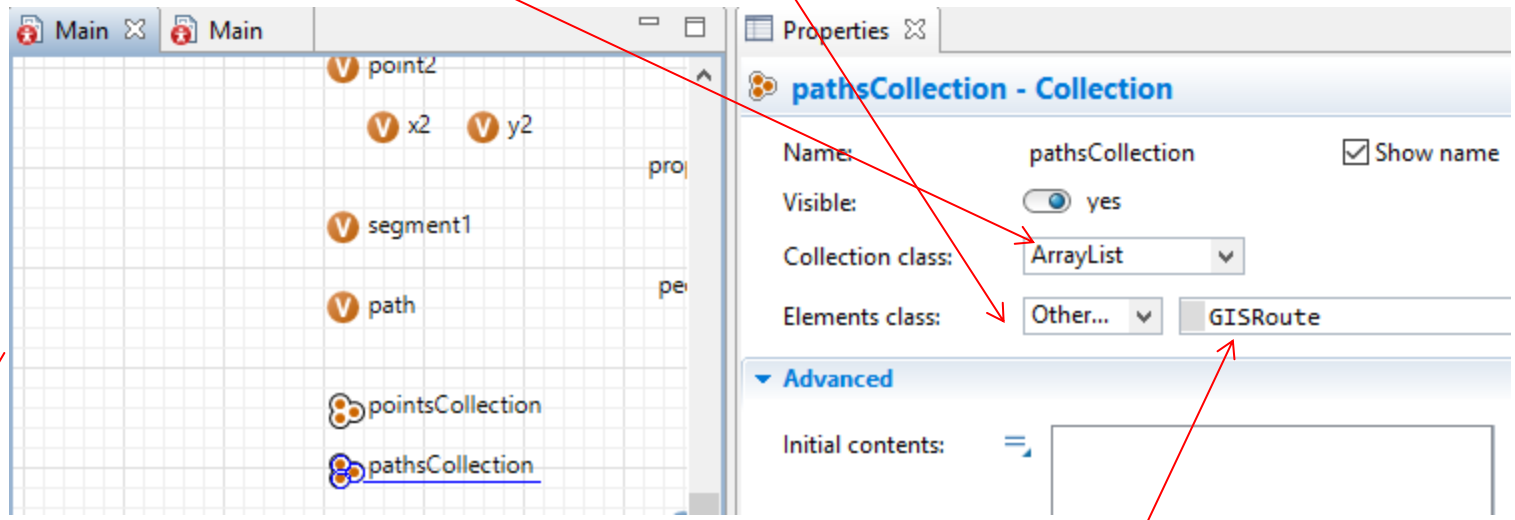
Drag and drop a collection into the Main and change its properties. Change its name to pointsCollection. Choose other for the type.



Change the type to : GISPoint

Define pathsCollection

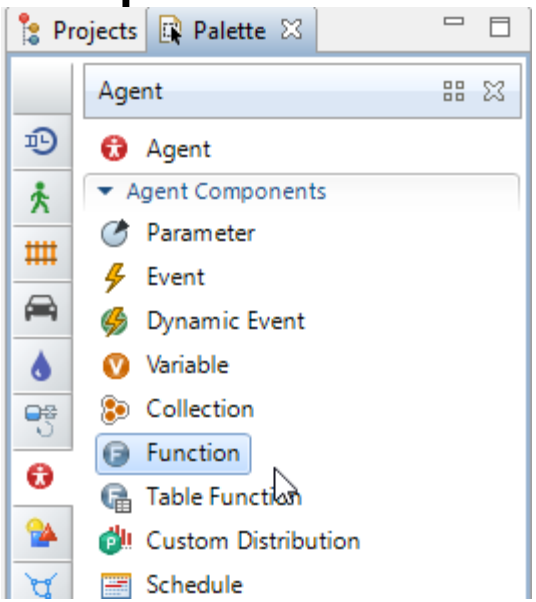
Drag and drop a collection into the Main and change its properties. Change its name to pathsCollection. Make sure that collection class is: ArrayList. Choose other for the type.



Change the type to : GISRoute

Create Flood Behavior Function

Now we use a function to define flood water movement in the river using the variables that were defined in the previous slides.



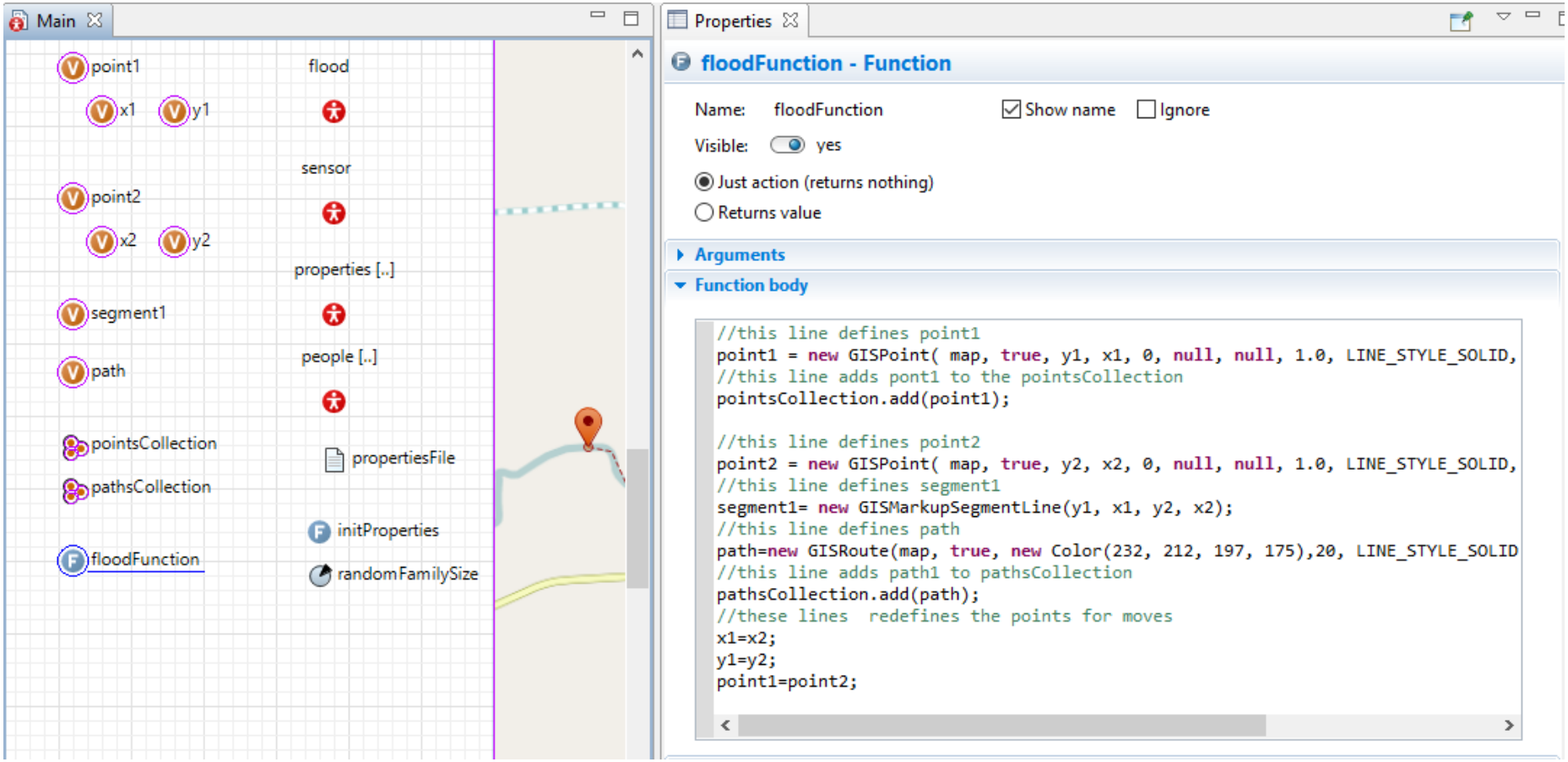
Creating flood function

Drag and drop a function into the Main and change its properties. Change its name to floodFunction. Fill its function body with the following lines of codes:

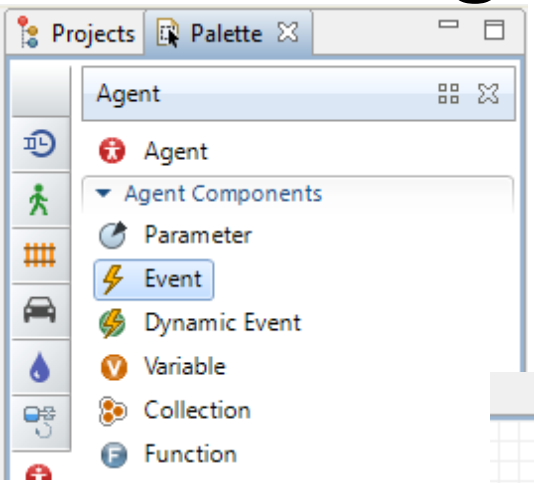
```
//this line defines point1
point1 = new GISPoint( map, true, y1, x1, 0, null, null, 1.0, LINE_STYLE_SOLID, "" );
//this line adds pont1 to the pointsCollection
pointsCollection.add(point1);

//this line defines point2
point2 = new GISPoint( map, true, y2, x2, 0, null, null, 1.0, LINE_STYLE_SOLID, "" );
//this line defines segment1
segment1= new GISMarkupSegmentLine(y1, x1, y2, x2);
//this line defines the flood path
path=new GISRoute(map, true, new Color(232, 212, 197, 175),20, LINE_STYLE_SOLID ,false,point1,
point2, segment1);
//this line adds path1 to pathsCollection
pathsCollection.add(path);
//these lines  redefines the points for moves
x1=x2;
y1=y2;
point1=point2;
```

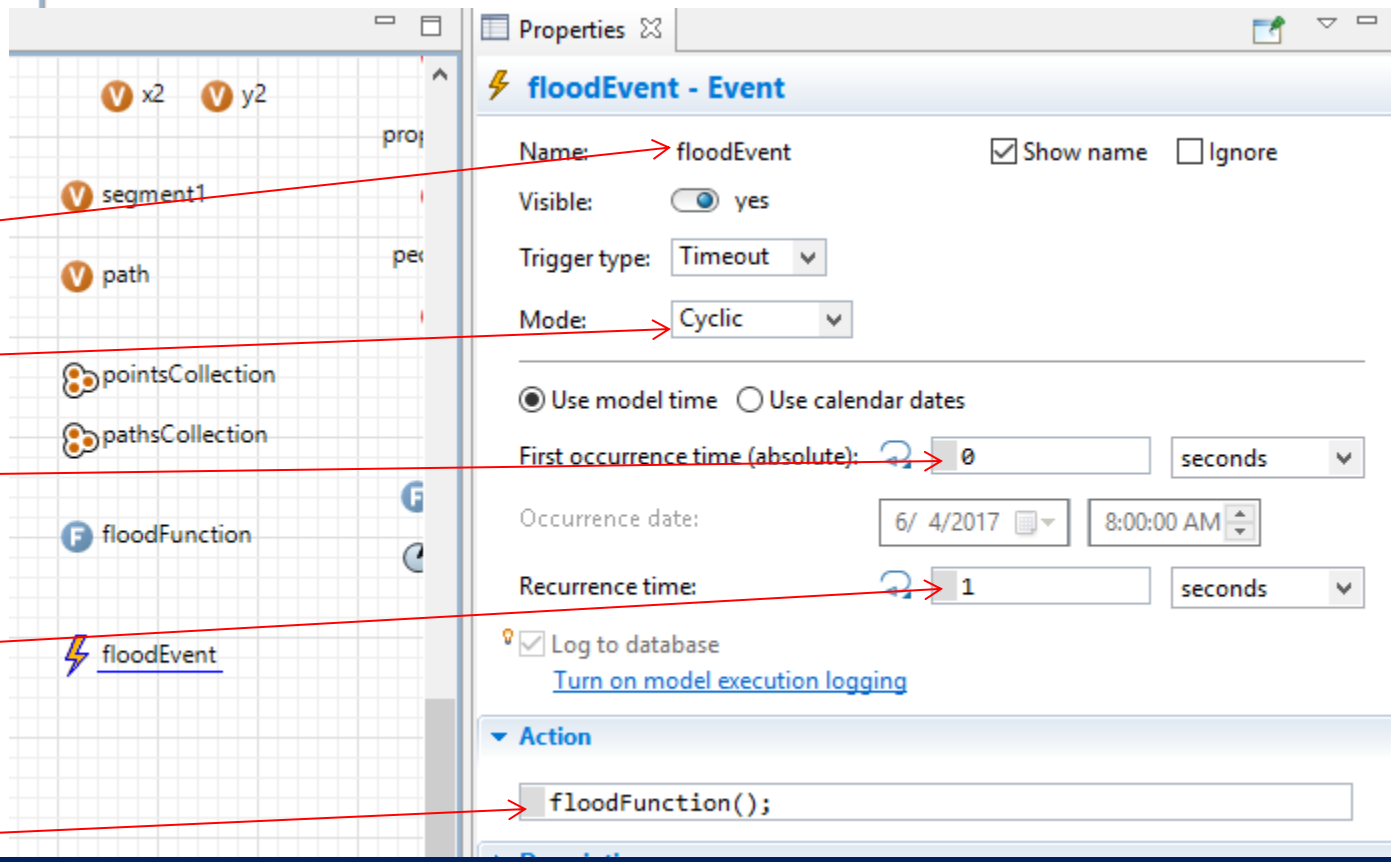
Your function should look like this



Defining an event for flood function



- Drag and drop an Event to the Main and change its properties



Change its name to : floodEvent

Change its Mode: Cyclic

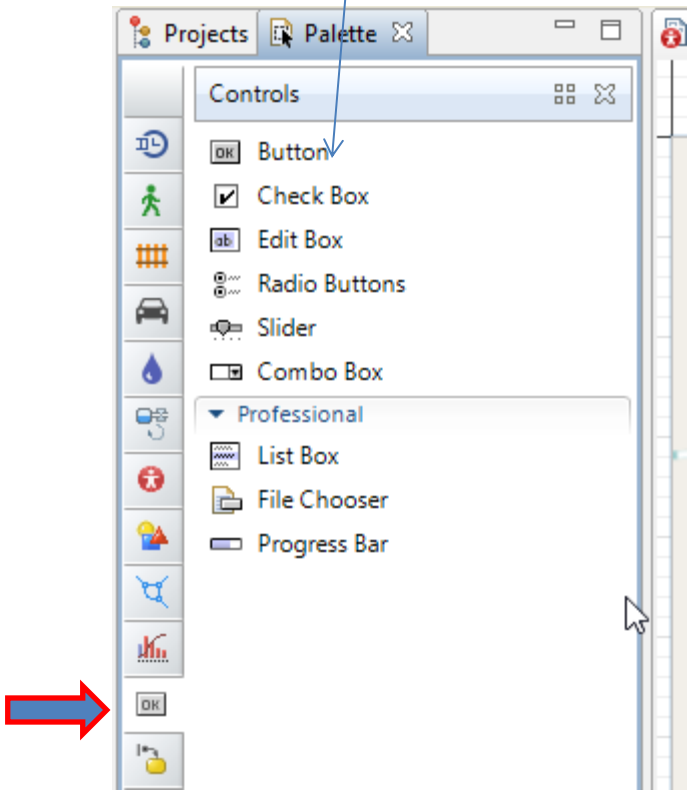
Change the first occurrence time to 0 seconds

Change the recurrence time to 1 seconds

Add the following code to the Action section: floodFunction();

Move the flood agent and activate the floodEvent using control button

- We now add a control button to our simulation to start the flood

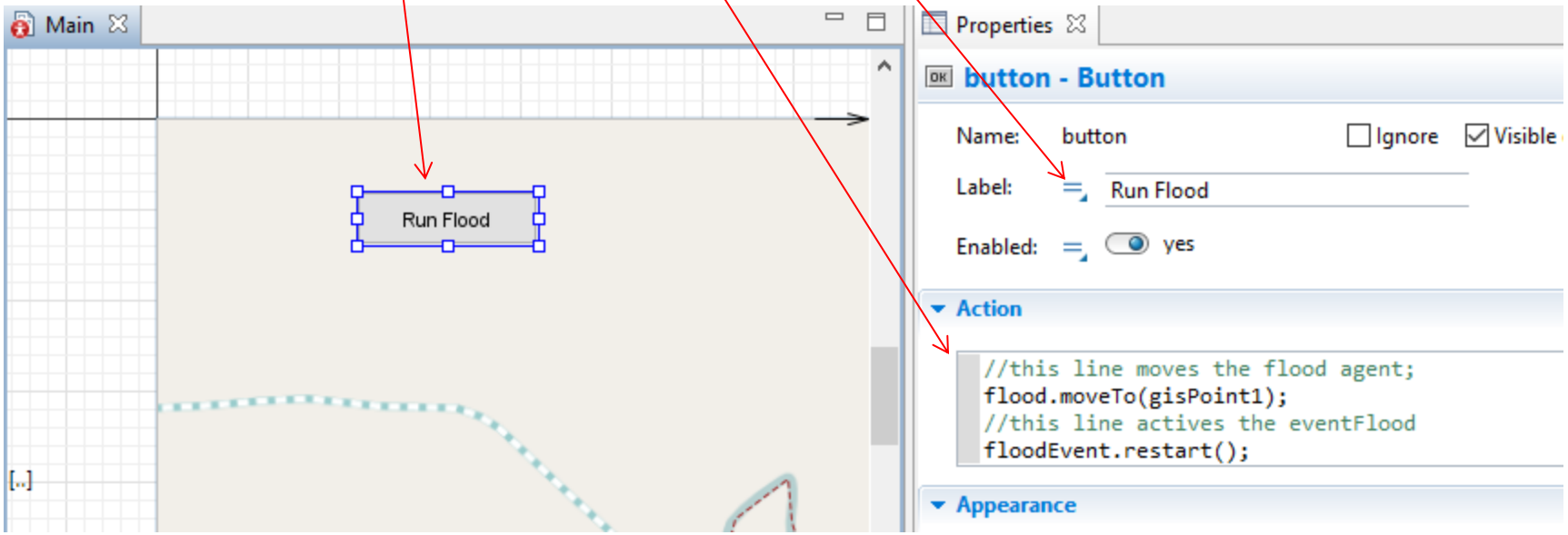


Add a button to the Main and set its properties

Set the Label as: Run Flood

Add the following codes to the Action section:

```
//this line moves the flood agent;
flood.moveTo(gisPoint1);
//this line actives the floodEvent
floodEvent.restart();
```



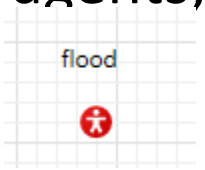
Some additional note about the **move function**.

- You are using a special anyLogic function for moving the agents in this case.

flood.moveTo(gisPoint1);

In the above function flood refers to your agent population (flood in this case).

moveTo() is a function that allows you to move the agents, in this case the flood.

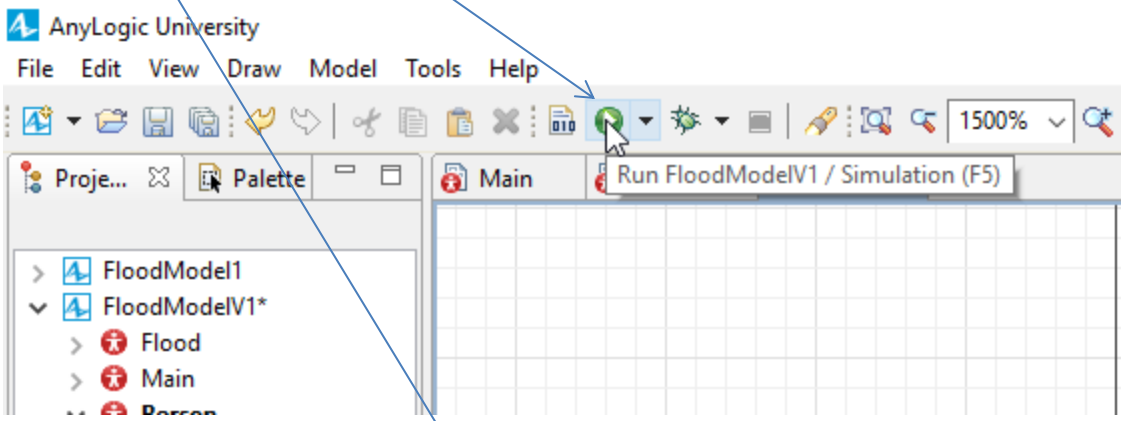


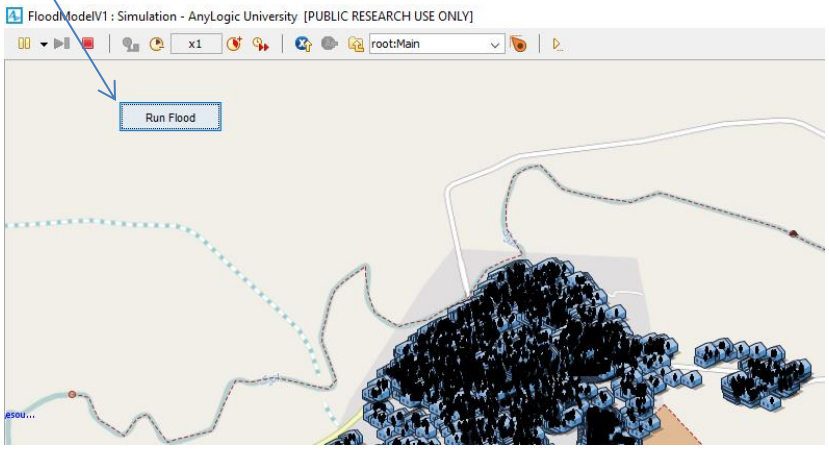
gisPoint1 is the target to which you want the flood agent to move to.

Inside the moveTo() function you can add a gisPoint or latitude and longitude of your target location in form of moveTo(x1,y1).

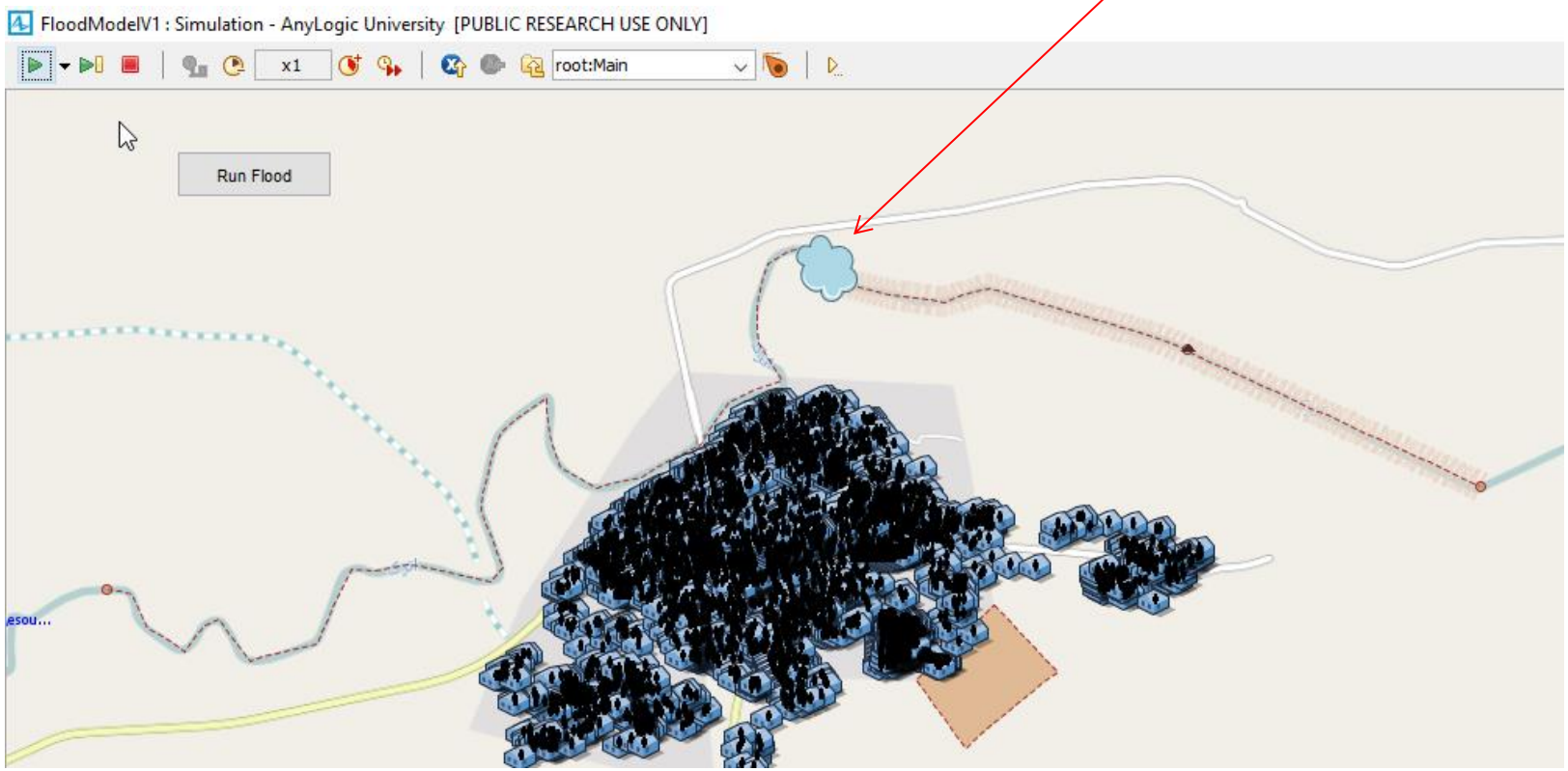
In this case moveTo(gisPoint1) would be similar to moveTo(gisPoint1.getLatitude(), gisPoint1.getLongitude()).

- Run the model by pressing on the Run Icon
- Click on the Run button on the Simulation page.
- Click on the Run Flood Button on the Main.





- If there is no error you should see that flood is moving over the river and creates its on path over the river.



- This is a simple way of showing flood moving along the river.
- We will make this model more complicated in the future lessons.
- We will explain how to define flood sensor behavior in lesson 4.
- Save your project.