Disaster Simulation: An AnyLogic Agent-Based Approach

Example: Flood simulation and evacuation GIS Environment



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.



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:



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.



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.

👸 Main	X						Properties 2	3			
					\rightarrow	^	🛛 segment	1 - Variable			
						\checkmark	Name:	v segment1		Show name	I gnore
C C) po	oint1		flood			Visible:	yes			
	C) x1	🕐 y1	3			Туре:	Other v	GISMarkup	SegmentLine	
							Initial value:	=			
•) po	oint2		sensor			▼ Advanced				
		N v2	M v2	1			Access:		public 🗸	1	
	,		V)	properties []			Static				
Ó	se	egment	:1	0			Constant		/		
				people []			Save in sr	napshot	<i>,</i>		

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.

👩 Mair	א <u>ו</u>			Properties 🛛	
				🛛 path - Variable	
				Name: path K	Show name 🗌 Ignore
	point1	flood		Visible: 🕥 yes	s
	💙 x1 🔍 y1	6		Type: Other	GISRoute
				Initial value: 📑	
	opint2	sensor		▼ Advanced	
	0 x ² 0 y ²	•		Access:	public 🗸
	• ~ • y-	properties []		Static	
	🧿 segment1	0		Constant	
X		people []		Save in snapshot	
•	path	6		System dynamics u	nits:
			•	Description	
		propertiesFile			

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.

Main 🖾		₹ 🗖	🔲 Properties 🛛		
V point1	flood	×	pointsCollect	tion - Collection	
💙 x1 🔍 y1	0		Name:	♥ pointsCollection	Show name
noint?	sensor		Visible: Collection class	✓ yesArrayList	
♥ x2 ♥ y2	•		Elements class:	Other V GISPoin	t
	properties []		▼ Advanced		
V segment I	people []		Initial contents:	=	
path	0				
pointsCollection	propertiesFile				
			I		I

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.

👸 Main 🔀 👸 Main			Properties 🖾	
	V point2	~	pathsCollection - Collection	
	💟 x2 🔍 y2	proj	Name: pathsCollection 🗹 Show	name
	🕐 segment1		Visible: ves Collection class: ArrayList	
	🕐 path	pe	Elements class: V Other V GISRoute	
	pointsCollection		✓ Advanced	
	pathsCollection		Initial contents:	

Change the type to : GISRoute

Ali Asgary, ADERSIM, York University, 2017

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

👸 Main 🛛		🔲 Properties 🕱 📃 🛃 🥂 🖓 🖓					
V point1 flood	^	floodFunction - Function					
⊘ x1 ⊘ y1 ⊗		Name: floodFunction Show name Ignore					
sensor		Visible: Ves Just action (returns nothing) Returns value					
properties []		Arguments Function body					
V segment 1 C people []		<pre>//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);</pre>					
pointsCollection propertiesFile pathsCollection propertiesFile pr		<pre>//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 path</pre>					
floodFunction improperties randomFamilySize		<pre>>//this line defines path path=new GISRoute(map, true, new Color(232, 212, 197, 175),20, LINE_STYLE_SOLID //this line adds path1 to pathsCollection pathsCollection.add(path); //these lines redefines the points for moves x1=x2; y1=y2; point1=point2; </pre>					

Defining an event for flood function

🎦 Projects 🙀 Palette 🔀

🔂 Agent

Agent Components

Dynamic Event

Parameter

Event

🙆 Variabla

Agent

Ð

Å

###

A

- 6

88 🖾

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

Collection	·		Properties			
G Function	💙 x2 🛛 y2	^	🗲 floodEvent - Event			
		proj	Name: floodEvent	Show name	🗌 lgnore	
Change its name to :	V segment1	1	Visible: 💿 yes			
floodEvent	nath (per	Trigger type: Timeout 🗸			
Change its Mode:	V patr		Mode: Cyclic v			
Cyclic	pointsCollection					
Change the first	pathsCollection		Use model time Use calenda	ar dates		
occurrence time to 0		G	First occurrence time (absolute):	→ 0	seconds	Υ.
seconds	FloodFunction	J	Occurrence date:	6/ 4/2017 🔍 - 8:00:00	AM ≑	
Change the recurrence		e	Recurrence time:	2>1	seconds	*
time to 1 seconds	4 floodEvent		⁰ 🗹 Log to database			
Add the following code to	/		Turn on model execution loggin	<u>19</u>		
the Action costion:			▼ Action			
the Action Section:			floodFunction();			
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:

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.

0

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.

			<	
		File Edit View Draw Medel Teels (Hate	
		File Edit View Draw Model Tools F		
		🛯 🕶 🖼 🕼 💘 🏷 🛛 👘 💼	渊 🗟 🗨 🔻 🗮 🔗 🌠 🔇 1500% 🗸 🔇	
		🐤 Proje 💥 🛱 Palette 😐 🗖 🔊 N	Main Run FloodModelV1 / Simulation (F5)	1
				f
		> 🗛 FloodModel1		
		✓ ▲ FloodModelV1*		
		> 🖸 Flood		-
		> 🕤 Main		
				I
		_ \		
			odelV1 : Simulation - AnyLogic University [PUBLIC RESEARCH USE ONLY]	
į	I Fland Made N/1 - Circulation - And ania University (DUDUC DESE			
1			Run Flood	
		eriment: Flood V		
ſ				
	FloodModelV1			Concession of the local division of the loca
	V			
	Run			
		esou		
- 18				

 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.