

Building User Interfaces for Models

Nathaniel Osgood

MIT 15.879

April 25, 2012

Lecture Focus: Creating Custom User Interfaces using “Controls”

- ‘Controls’ are “widgets” that allow for obtaining user input
 - These widgets have properties that can be set at both design and run (execution, simulation) time
- By setting the properties of these controls at design time, we can
 - Establish their general logical & visual properties
 - Establish their correspondence with model variables
- These controls can be used by the user during simulation to set assumptions in the model



Hands on Model Use Ahead



Load Previously Built Model:
MinimalistSIRNetworkABM

After change, suggest saving as
“UISupportedMinimalistSIRNetworkABM”

Add a Related Parameter to *Main*

The screenshot displays the AnyLogic University software interface. The main workspace shows a diagram with a circle and a vertical line, and a list of parameters: population [...], environment, and exposureHazard. The Properties panel is open for a 'slider - Slider' component, showing the following configuration:

- Name: slider
- Show name:
- Ignore:
- Public:
- Icon:
- Orientation: Horizontal Vertical
- Link to: exposureHazard
- Minimum value: 0
- Maximum value: 10
- Enabled:

The Properties panel also includes tabs for General, Advanced, Dynamic, and Description. The Console panel is empty, and the Problems panel shows 'No problems'.

Setting the Transition to Refer to the Parameter in Main

The screenshot displays the AnyLogic software interface. On the left, a project tree shows the hierarchy: **UISupportedMinimalistSIRNetworkABM*** > **Main** > **Person** > **Simulation: Main**. The central workspace shows a statechart with three states: **Susceptible**, **Infective**, and **Recovered**, connected by arrows. A transition labeled **infectionStatechart** is shown above the **Susceptible** state. A red arrow points from the **Infective** state to the transition configuration panel at the bottom.

Note that "exposureHazard" lives in "Main". To get a reference to the "Main" object, we call "get_Main()" on ourselves.

The **pathogenExposure - Transition** configuration panel is shown below the statechart. It includes the following fields:

- Name:** pathogenExposure
- Triggered by:** Rate
- Rate:** `this.get_Main().expo`
- Action:** `this.send("Infection", RANDOM_C`
- Guard:** (empty)

A dropdown menu is open for the **Rate** field, showing a list of functions and variables. The selected item is `exposureHazard : double - Main`. Other items in the list include `exponential() : double - Utilities`, `exponential(double arg0) : double - Utilities`, `exponential(double arg0, double arg1) : double - Utilities`, `exponential(double arg0, double arg1, double arg2, double arg3) : double - Utilities`, and `exponential(double arg0, double arg1, double arg2, double arg3, double arg4) : double - Utilities`.

Resulting Expression

The screenshot displays the AnyLogic software interface. The main workspace shows a statechart for a 'Person' object, titled 'infectionStatechart'. The statechart consists of three yellow state boxes: 'Susceptible', 'Infective', and 'Recovered', connected by downward arrows. A transition arrow points from the 'Infective' state to the 'Recovered' state, with a blue dot on the arrow. A function 'colorForInfectionState' is visible in the workspace.

The 'pathogenExposure - Transition' configuration panel is open, showing the following details:

- Name:** pathogenExposure
- Triggered by:** Rate
- Rate:** `this.get_Main().exposureHazard`
- Action:** `this.send("Infection", RANDOM_CONNECTED);`
- Guard:** (empty)

The left sidebar shows a project tree for 'UISupportedMinimalistSIRNetworkABM*' with folders for Main, Parameters, Environments, Embedded Objects, Presentation, Person, Statecharts, Functions, and Simulation: Main. The right sidebar is a palette with categories like General, System Dynamics, Statechart, Actionchart, Analysis, Presentation, 3D, Controls, Connectivity, Enterprise Library, Pedestrian Library, Rail Library, Road Traffic Library - Preview, Pictures, 3D Objects, and Palettes...

Reminder: An Explicitly Specified Population Size

The screenshot displays the AnyLogic University interface. The main workspace shows a diagram with a circle representing a population, connected to a 'population [...]' component. Below the workspace, the 'population - Person' properties panel is visible, showing the following configuration:

- Name: population
- Type: Person
- Environment: environment
- Package: minimalistsimnetworkabmmodel
- Replicated:
- Initial number of objects: 100
- Optimize: Access by index (ArrayList) Add/remove operations (LinkedHashSet)

The 'Properties' panel also includes a 'General' tab with options for 'Show name', 'Ignore', 'Public', and 'Show at runtime', along with a 'Create presentation' button. The 'Problems' panel shows 'No problems'. The 'Projects' panel on the left lists various simulation models, including 'SIR Agent Based Calibration' and 'UISupportedMinimalistSIRNe'.

A Parameter Giving the Population Size

The screenshot displays the AnyLogic University interface. The main workspace shows a diagram with a circle and arrows, and a parameter named 'populationSize' is highlighted. The Properties panel at the bottom is open for 'populationSize - Parameter'.

populationSize - Parameter

General

Name: Show name Ignore Show at runtime

Array

Type: void (just action) boolean int double String Date Color Other:

Use Units Unit:

Array

Default value:

Dynamic

On change:

Selection X=-2, Y=307

Setting the Population to Use the Parameter Value

The screenshot displays the AnyLogic University interface for setting up a population parameter. The main workspace shows a diagram with a circle representing a population and a line connecting it to the 'population' object in the 'Main' environment. The 'population' object is highlighted, and its properties are shown in the 'Properties' panel below.

Properties Panel: population - Person

- General**
 - Name: Show presentation
- Parameters**
 - Type: Person
 - Environment:
 - Package:
- Replicated
 - Initial number of objects:
 - Optimize: Access by index (ArrayList) Add/remove operations (LinkedHashSet)

A tooltip window is open over the 'Initial number of objects' field, showing the variable 'populationSize : int - Main'.

The background workspace shows a diagram with a circle representing a population and a line connecting it to the 'population' object in the 'Main' environment. Other objects visible include 'environment' and 'exposureHazard'.

Reminder: The Existing Experiment

The screenshot displays the AnyLogic University software interface, which is used for building and simulating agent-based models. The main window shows a grid-based workspace with the title "MinimalistNetworkABMModel" and the subtitle "Experiment setup page". A button in the center of the workspace reads "Run the model and switch to Main view".

The interface is divided into several panels:

- Projects:** A tree view on the left showing the project structure, including "Main", "Parameters", "Environments", "Embedded Objects", "Presentation", "Person", "Statecharts", "Functions", and "Simulation: Main".
- Palette:** A panel on the right containing various UI components and libraries, such as "General", "System Dynamics", "Statechart", "Actionchart", "Analysis", "Presentation", "3D", "Controls", "Connectivity", "Enterprise Library", "Pedestrian Library", "Rail Library", "Road Traffic Library - Preview", "Pictures", "3D Objects", and "Palettes...".
- Properties:** A panel at the bottom left showing the properties of the selected simulation experiment.
- Console:** A panel at the bottom center for viewing simulation output.

The **Simulation - Simulation Experiment** properties panel is currently active and shows the following settings:

- General:** Name: Simulation; Main active object class (root): Main; Ignore:
- Advanced:** Random number generation: Random seed (unique simulation runs); Fixed seed (reproducible simulation runs) with Seed value: 1; Custom generator (subclass of Random): new Random()
- Parameters:** populationSize: 100; exposureHazard: 0.5; Paste from clipboard button.

The bottom status bar indicates "Simulation - Simulation Experiment".

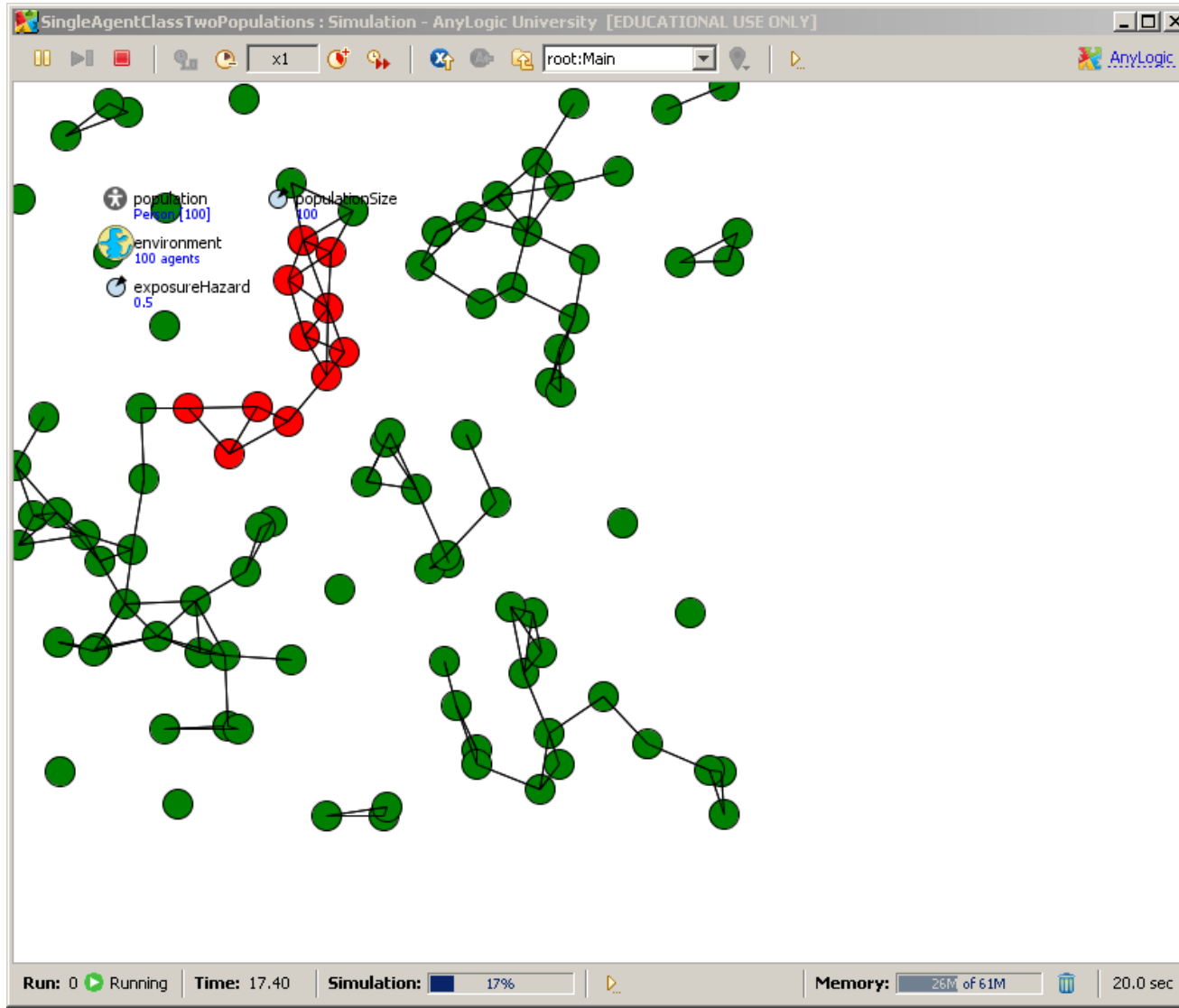
Running that Experiment

The screenshot displays the AnyLogic software interface. At the top, a toolbar contains various icons for navigation and execution, including a play button, a stop button, and a refresh button. The current experiment is identified as 'experiment: Singl...'. The main content area features the title 'MinimalistNetworkABMModel' and the subtitle 'Experiment setup page'. A prominent button labeled 'Run the model and switch to Main view' is centered on the page. The bottom status bar provides real-time information: 'Run: 0' with an idle indicator, 'Time: -', 'Simulation: Stop time not set', 'Memory: 10M of 61M', and '0.0 sec'.

Run the model and switch to Main view

Run: 0 Idle | Time: - | Simulation: Stop time not set | | Memory: 10M of 61M | | 0.0 sec

Reminder: Pushing the Button Shows the Simulation Visualization



Understanding the Button's Actions

The screenshot displays the AnyLogic University interface. The main workspace shows a simulation model titled "MinimalistNetworkABMModel" with the subtitle "Experiment setup page". A button is placed on the workspace with the text "Run the model and switch to Main view". A blue arrow points from the button to the "Main" view in the project tree on the left.

The "button - Button" properties window is open, showing the following configuration:

- Name:** button
- Label:** Run the model and switch to Ma
- Enabled:** getState() == IDLE
- Action:**

```
run();
getEngine().getPresentation().setPresentable( getEngine().getRoot() );
```

The right-hand side of the interface shows a palette of UI components, including buttons, checkboxes, and sliders.

Adding a Slider to Represent the Population Size

The screenshot displays the AnyLogic software interface for configuring a slider widget. The main workspace shows a grid with the text "MinimalistNetworkABMModel" and "Experiment setup page". A button labeled "Run the model and switch to Main view" is visible. A slider widget is positioned on the right side of the workspace. A red arrow points from the text "Fill in this information" to the slider's configuration panel.

Fill in this information

sliderPopulationSize - Slider

General

Name: sliderPopulationSize Show name Ignore Public Icon

Orientation: Horizontal Vertical Add labels...

Link to:

Minimum value: 1

Maximum value: 1000

Default value: 100

Enabled:

Action:

Setting the Simulation Parameter Values to Use the Slider Setting

The screenshot displays the AnyLogic University interface for setting simulation parameters. The main workspace shows a grid with the text "MinimalistNetworkABMModel" and "Experiment setup page". A slider is positioned in the center of the grid. A red arrow points from the slider to the "populationSize" parameter in the "Simulation - Simulation Experiment" panel. The value of the slider is being set to the expression `sliderPopulationSize.getIntValue()`.

Simulation - Simulation Experiment

General

Name: Main active object class (root): Ignore

Random number generation:

- Random seed (unique simulation runs)
- Fixed seed (reproducible simulation runs) Seed value:
- Custom generator (subclass of Random):

populationSize

exposureHazard

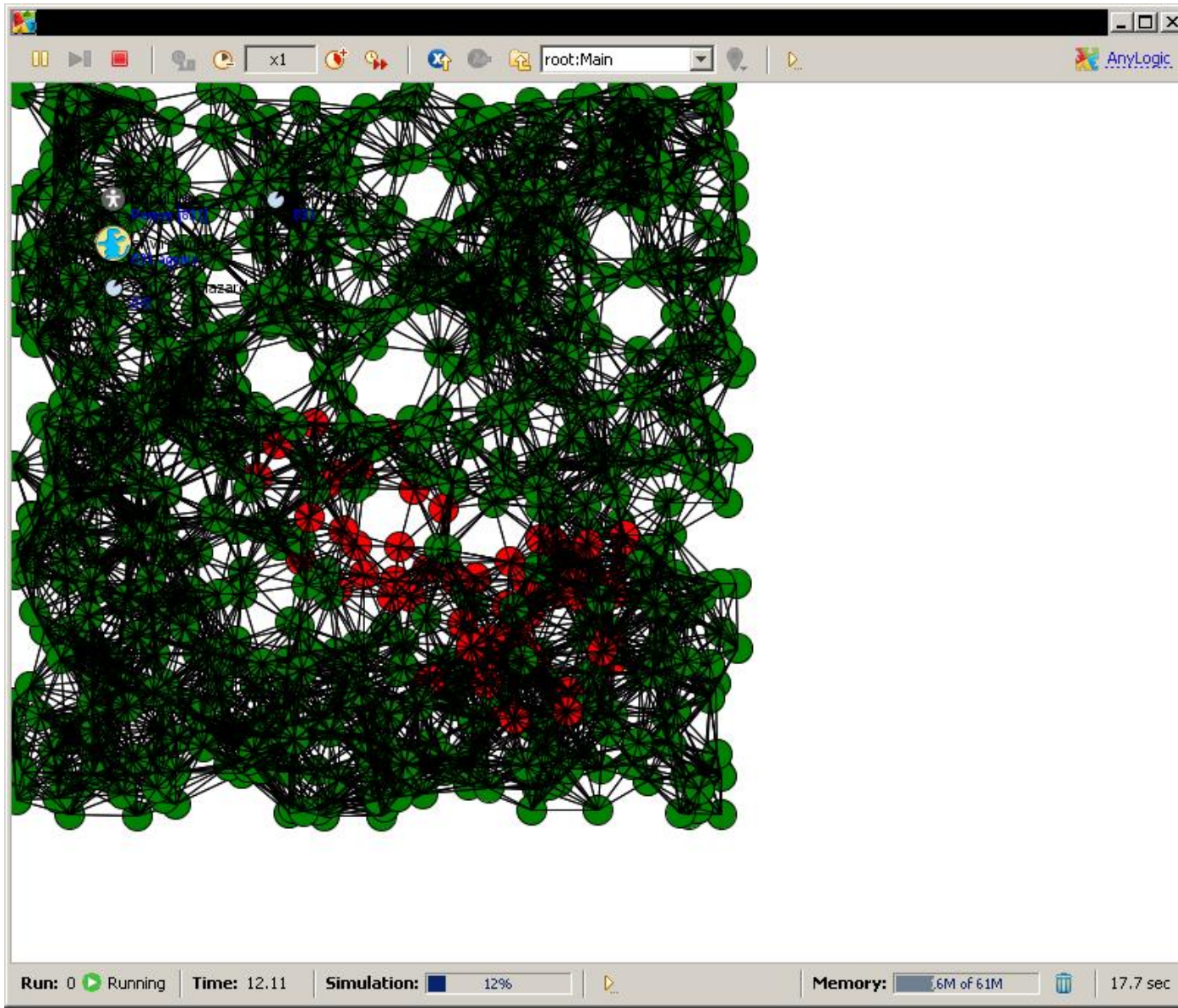
Controls

- Button
- Check Box
- Edit Box
- Radio Buttons
- Slider
- Combo Box
- List Box
- File Chooser
- Progress Bar

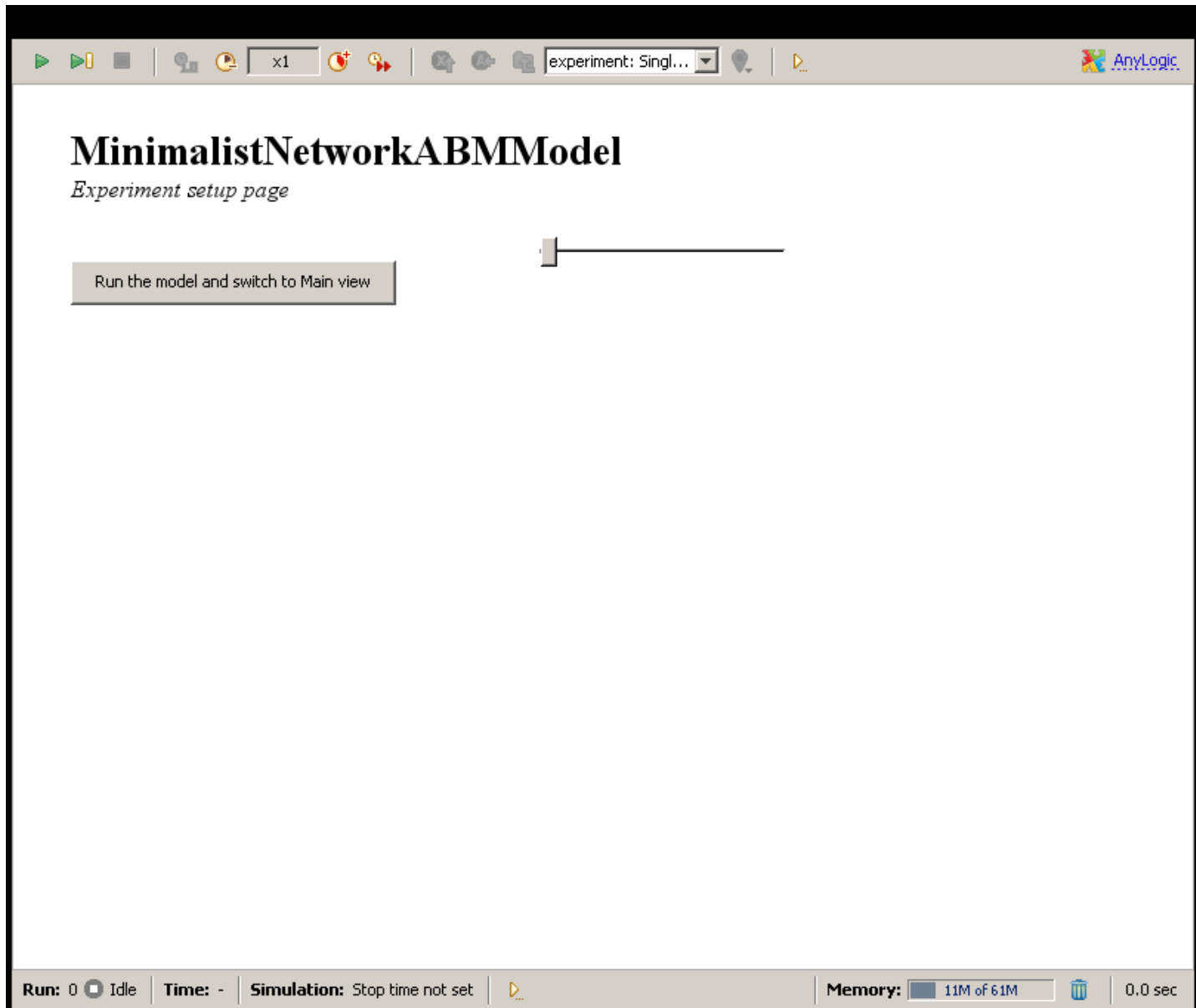
Choosing a High Value on the Slider

The screenshot shows the AnyLogic software interface for the **MinimalistNetworkABMModel**. The window title is "MinimalistNetworkABMModel" and the subtitle is "Experiment setup page". A slider control is visible, and a button labeled "Run the model and switch to Main view" is present. The status bar at the bottom shows "Run: 1 Idle", "Time: -", "Simulation: Stop time not set", "Memory: 13M of 61M", and "61.5 sec".

Resulting Network – Large Population

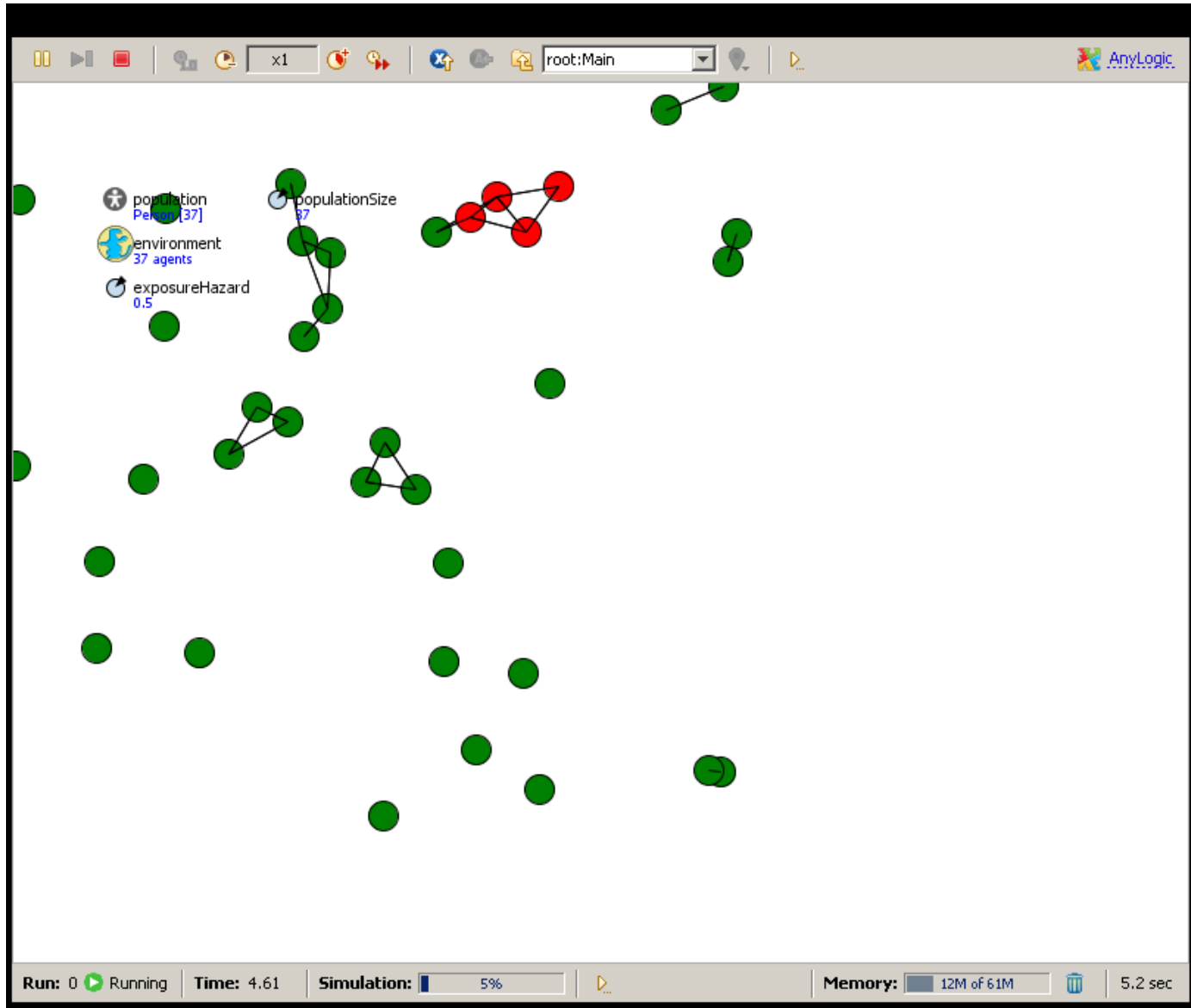


Choosing a Low Value on the Slider



The screenshot displays the AnyLogic software interface for the 'MinimalistNetworkABMModel' experiment. The title bar shows 'x1' and 'experiment: Singl...'. The main area features the title 'MinimalistNetworkABMModel' and the subtitle 'Experiment setup page'. A button labeled 'Run the model and switch to Main view' is located on the left. A horizontal slider is positioned to the right of the button, with its handle at the far left end, indicating a low value. The status bar at the bottom shows 'Run: 0', 'Idle', 'Time: -', 'Simulation: Stop time not set', 'Memory: 11M of 61M', and '0.0 sec'.

Resulting Network -- Small Population



Adding (Static) Text Labeling Slider

The screenshot displays the AnyLogic software interface. The main workspace shows a simulation model titled "MinimalistNetworkABMModel" with the subtitle "Experiment setup page". A slider control is visible, labeled "Population size". A red arrow points from the text "Fixed text – doesn't change over time" to the "Population size" label. The Properties panel at the bottom shows the configuration for the "textPopulationSize - Text" object, including its name, color (black), font (SansSerif), and size (10 pt).

Fixed text – doesn't change over time

MinimalistNetworkABMModel
Experiment setup page

Population size

Run the model and switch to Main view

textPopulationSize - Text

Name: textPopulationSize Show name Ignore Public Icon Lock Show at runtime

Color: black Alignment: Font: SansSerif 10 pt

Italic Bold

Text: Population size

textPopulationSize - Text Selection X=716, Y=169

Dynamic Properties to Report the Slider Value

MinimalistNetworkABMModel
Experiment setup page

Population size <textSliderValue>

Run the model and switch to Main view

By setting **this expression** to determine the dynamic value of the text field **“Text” property**, the string associated with this text will automatically change with the slider value

Properties Console

Aa textSliderValue - Text

General	Replication:
Advanced	Visible:
Dynamic	X:
Description	Y:
	Text: <code>sliderPopulationSize.getIntValue()</code>
	Color:
	Alignment:
	On click:
	Rotation, rad:
	Scale X:
	Scale Y:

Example Resulting Output

The screenshot displays the AnyLogic software interface for the 'MinimalistNetworkABMModel' experiment. The window title is 'experiment: Singl...'. The main content area shows the model name and the subtitle 'Experiment setup page'. A button labeled 'Run the model and switch to Main view' is positioned on the left. To its right is a slider control for 'Population size', which is currently set to 723. The bottom status bar provides simulation status: 'Run: 0' (Idle), 'Time: -', 'Simulation: Stop time not set', 'Memory: 12M of 61M', and '0.0 sec'.

MinimalistNetworkABMModel
Experiment setup page

Run the model and switch to Main view

Population size 723

Run: 0 Idle | Time: - | Simulation: Stop time not set | Memory: 12M of 61M | 0.0 sec

Reflecting on Temporal Specificity of UI Elements

- The user interface component (slider) we created thus has had its value used to set the initial state of the model (the population size)
- User interface components can also be used to vary assumptions dynamically during runtime
 - For example, vary parameter values

Example: Creating a Slider to Dynamically Vary the Infection Hazard

The screenshot displays the AnyLogic software interface. The main workspace shows a diagram with a slider widget. A blue arrow points from the text "Link so that changing the slider automatically changes the exposureHazard parameter" to the "Link to:" field in the properties panel, which is set to "exposureHazard". Another red arrow points from the text "Set these as the upper and lower bounds of the slider" to the "Minimum value:" and "Maximum value:" fields, which are set to 0 and 10 respectively.

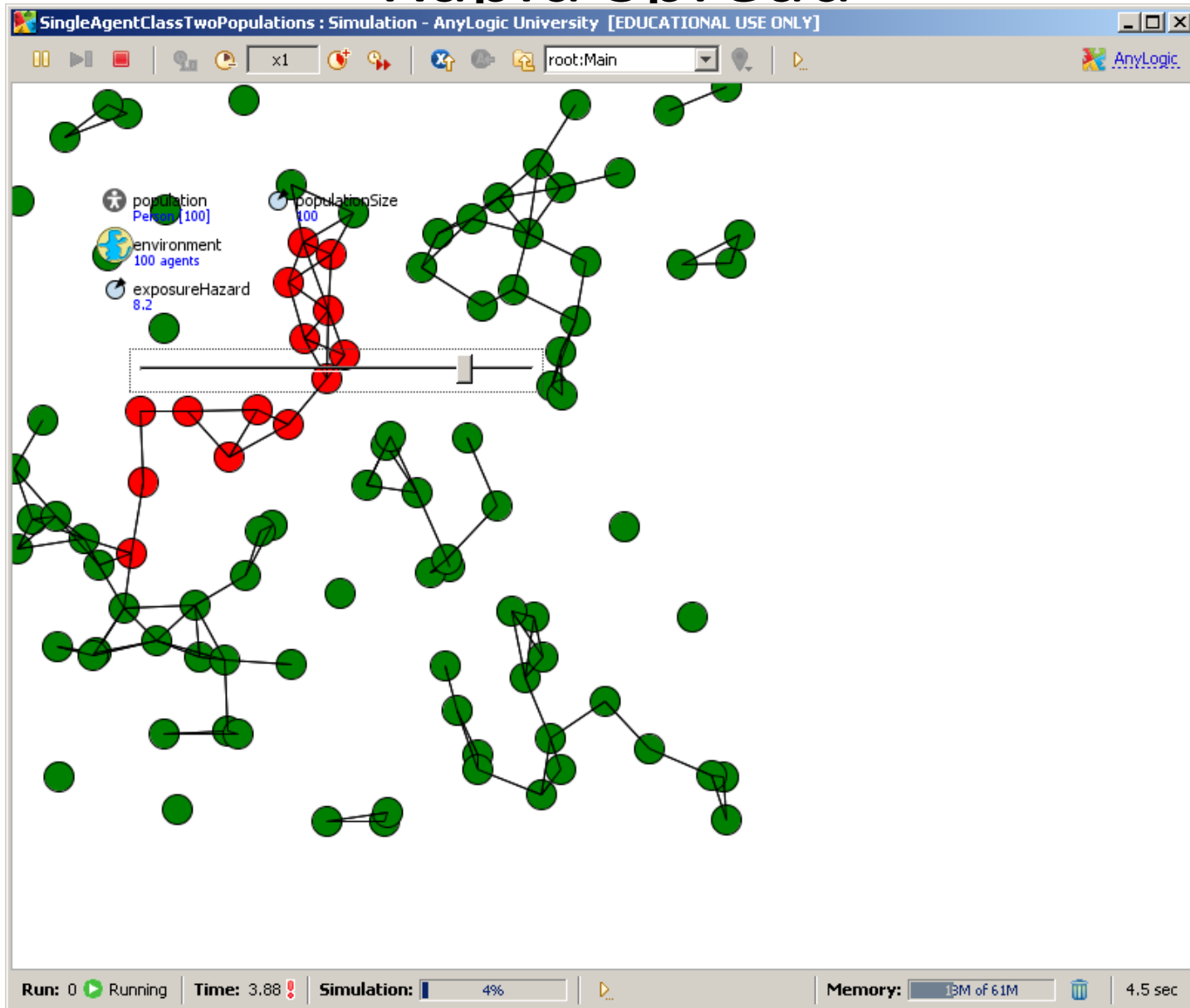
Link so that changing the slider automatically changes the exposureHazard parameter

Set these as the upper and lower bounds of the slider

Properties - slider - Slider

Category	Property	Value
General	Name	slider
General	Show name	<input type="checkbox"/>
General	Ignore	<input type="checkbox"/>
General	Public	<input checked="" type="checkbox"/>
General	Icon	<input type="checkbox"/>
Orientation	Horizontal	<input checked="" type="radio"/>
Orientation	Vertical	<input type="radio"/>
Linking	Link to:	exposureHazard
Values	Minimum value:	0
Values	Maximum value:	10
Enabled	Enabled:	<input type="checkbox"/>
Action	Action:	

A High Slider Value Leads to a More Rapid Spread



Dropping the Slider Value (Exposure Hazard) to 0 Can Stop the Spread

The screenshot displays the AnyLogic software interface for a simulation model. The main workspace shows a diagram with a central circle and arrows pointing to various components: 'population [...]', 'populationSize', 'environment', and 'exposureHazard'. A slider control is positioned below the diagram, representing the 'exposureHazard' parameter.

The left sidebar shows the project structure for 'UISupportedMinimalistSIRNetworkABM*', including 'Main', 'Parameters', 'Environments', 'environment', 'Embedded Objects', 'Presentation', 'Person', 'Statecharts', 'Functions', and 'Simulation: Main'.

The bottom console window shows the following log output:

```
anylogic config [Java Application] C:\Program Files\AnyLogic 6_7 University\jre\bin\javaw.exe (Apr 22, 2012 1:59:27 PM)
root.population[9]has been infected!
root.population[17]has been infected!
root.population[39]has been infected!
root.population[44]has been infected!
root.population[64]has been infected!
root.population[80]has been infected!
root.population[75]has been infected!
root.population[26] has recovered!
root.population[25] has recovered!
root.population[6] has recovered!
root.population[81] has recovered!
root.population[67] has recovered!
root.population[30] has recovered!
root.population[5] has recovered!
root.population[90] has recovered!
root.population[20] has recovered!
root.population[19] has recovered!
root.population[55] has recovered!
root.population[96] has recovered!
```

The right sidebar contains a palette of UI controls, including Button, Check Box, Edit Box, Radio Buttons, Slider, Combo Box, List Box, File Chooser, Progress Bar, Connectivity, Enterprise Library, Pedestrian Library, Rail Library, Road Traffic Library - Preview, Pictures, 3D Objects, and Palettes...

Recall: The Initial Infection Seed

The screenshot displays the AnyLogic University interface. The main workspace shows a diagram with a circle and arrows, representing a message delivery process. A red text overlay reads: "This delivered an infection message to a randomly selected person in the population". Below the diagram, the "Main - Active Object Class" properties window is open, showing the "Startup code" field with the following code: `environment.deliverToRandom("Infection");`. The "Properties" window also shows the "Name" field set to "Main" and the "Agent" checkbox checked. The "Palette" on the right lists various UI components like Button, Check Box, Edit Box, etc. The "Projects" window on the left shows the project structure, including "Main", "Parameters", "Environments", "Embedded Objects", "Presentation", "Person", "Statecharts", "Functions", and "Simulation: Main".

This delivered an infection message to a randomly selected person in the population

Main - Active Object Class

Name: Main Ignore

Agent Generic

Startup code:
`environment.deliverToRandom("Infection");`

Destroy code:

Cut Text from Startup Code for Main

The screenshot displays the AnyLogic software interface. The main workspace shows a diagram with a central circle and a horizontal line. A button labeled "Seed New I..." is connected to a vertical line that leads to a horizontal line at the bottom. Below this line is a slider control. The diagram includes several objects: "population [...]", "populationSize", "environment", and "exposureHazard".

The "Main - Active Object Class" configuration panel is open, showing the following details:

- Name:** Main
- Ignore
- Agent Generic
- Startup code:** [Empty text box]
- Destroy code:** [Empty text box]

The left sidebar shows a project tree with the following structure:

- UISupportedMinimalistSIRNetworkABM*
 - Main
 - Parameters
 - Environments
 - environment
 - Embedded Objects
 - Presentation
 - Person
 - Statecharts
 - Functions
 - Presentation
 - Simulation: Main

The right sidebar shows a palette of controls:

- General
- System Dynamics
- Statechart
- Actionchart
- Analysis
- Presentation
- 3D
- Controls**
 - OK Button
 - Check Box
 - ab Edit Box
 - Radio Buttons
 - Slider
 - Combo Box
 - List Box
 - File Chooser
 - Progress Bar
- Connectivity
- Enterprise Library
- Pedestrian Library
- Rail Library
- Road Traffic Library - Preview
- Pictures
- 3D Objects
- Palettes...

Setting the Button to Seed a New Infection

The screenshot displays the AnyLogic software interface. The main workspace shows a diagram with a button labeled "Seed New Infection". A red arrow points from the text "This is the action the button will perform when pushed" to the button. Below the workspace, the Properties window is open for the "buttonSeedNewInfection - Button". The "Action" field is highlighted in pink and contains the code: `environment.deliverToRandom("Infection");`. The Properties window also shows the button's name, label, and other settings.

This is the action the button will perform when pushed

buttonSeedNewInfection - Button

General

Name: Show name Ignore Public Icon

Advanced

Label:

Dynamic

Enabled:

Description

Action:

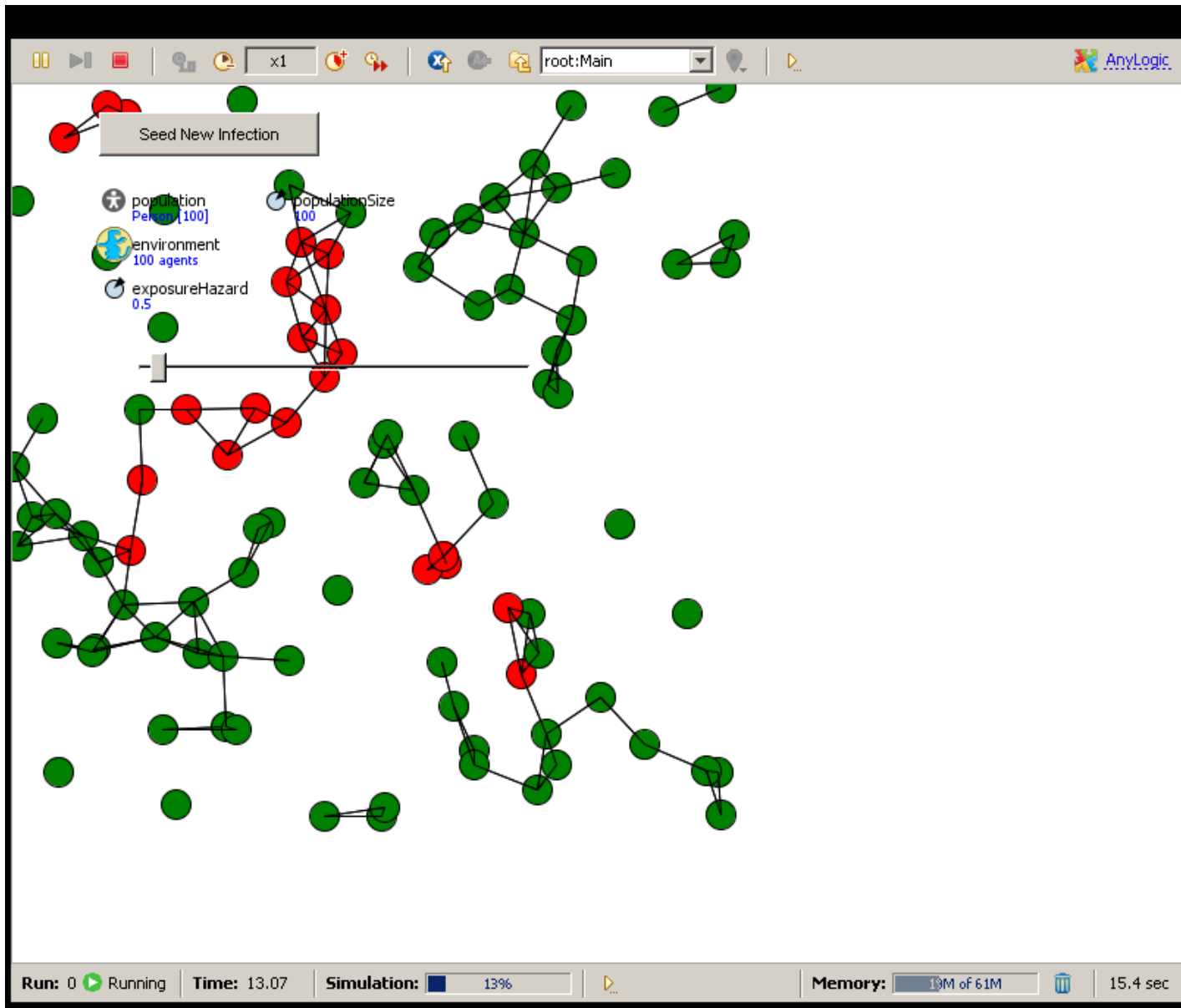
Properties window also shows:

- environment
- exposureHazard
- population [..]
- populationSize

Palette window shows:

- General
- System Dynamics
- Statechart
- Actionchart
- Analysis
- Presentation
- 3D
- Controls
 - Button
 - Check Box
 - Edit Box
 - Radio Buttons
 - Slider
 - Combo Box
 - List Box
 - File Chooser
 - Progress Bar
- Connectivity
- Enterprise Library
- Pedestrian Library
- Rail Library
- Road Traffic Library - Preview
- Pictures
- 3D Objects
- Palettes...

With Multiple Presses, Multiple “Seed” Infections



Add a Contingent Reporting Variable

The screenshot displays the AnyLogic University interface. On the left, a project tree shows the hierarchy: UI Supported Minimalist SIR Network ABM* > Main > Variables. The main workspace shows a diagram with a 'Seed New Infection' button and a variable 'isReportingEnabled' represented by a slider. A red arrow points from the text 'Initial value should be false' to the 'false' value in the variable's configuration panel.

Initial value should be false

isReportingEnabled - Variable

General

Name: `isReportingEnabled` Show name Ignore Show at runtime

Access: `public` Static Constant Save in snapshot

Type: boolean int double String Other: `boolean`

Initial value: `false`

Use Units Unit:

Contingent Infection Reporting

The screenshot displays the AnyLogic software interface. On the left, a project tree shows the hierarchy: UISupportedMinimalistSIRNetworkABM* > Main > Person > Simulation: Main. The central workspace shows a statechart for 'infectionStatechart' with three states: Susceptible, Infective, and Recovered. A transition arrow points from Susceptible to Infective. A red arrow points from a text box to the 'if' statement in the transition's action field.

This makes the reporting contingent on the value of isReportingEnabled

infection - Transition

General

Name: infection show name Ignore Show at runtime

Triggered by: Message

Message type: boolean int double String Other

Class name: Object

Fire transition: Unconditionally If message equals If expression is true

Action:

```
if (this.get_Main().isReportingEnabled)
    traceIn(this + "has been infected!")
```

Guard:

Enabling Reporting

The screenshot displays the AnyLogic software interface. The main workspace shows a simulation model with a grid background. A red arrow points from the text "Link to the 'isReportingEnabled' parameter" to the "Link to:" field in the Properties window, which is set to "isReportingEnabled".

Link to the "isReportingEnabled" parameter

The Properties window shows the following details for the "checkboxEnableReporting - Check Box":

- Name: checkboxEnableReporting
- Show name:
- Ignore:
- Public:
- Icon:
- Label: Enable Reporting
- Link to: isReportingEnabled
- Enabled:
- Action:

The simulation model includes a "Seed New Infection" block with a checked "Enable Reporting" checkbox. The model also features a slider for "isReportingEnabled" and various other parameters like "populationSize" and "exposureHazard".

Unless Reporting is Enabled (i.e. Checkbox is Checked), No Output

The screenshot displays the AnyLogic University interface for a simulation titled "SingleAgentClassTwoPopulations". The main workspace shows a network diagram with green and red nodes connected by lines. A control panel on the left includes a "Seed New Infection" button and several variables: "population" (100), "populationSize" (100), "environment" (100 agents), "exposureHazard" (0.5), and "isReportingEnabled" (false). The "isReportingEnabled" variable is currently unchecked. The bottom status bar indicates the simulation is running, with a time of 19.83 and a simulation progress of 20%. The memory usage is shown as 20M of 6.1M.

AnyLogic University [EDUCATIONAL USE ONLY]

File Edit View Draw Model Tools Help

Projects

UISupportedMinimalistSIRNetworkABM*

- Main
 - Parameters
 - Variables
 - Environments
 - environment
 - Embedded Objects
 - Presentation
- Person
 - Statecharts
 - Functions
 - Presentation
- Simulation: Main

Person Simulation Main

SingleAgentClassTwoPopulations : Simulation - AnyLogic University [EDUCATIONAL USE ONLY]

Enable Reporting

Seed New Infection

population 100

populationSize 100

environment 100 agents

exposureHazard 0.5

isReportingEnabled false

Run: 0 Running Time: 19.83 Simulation: 20% Memory: 20M of 6.1M

anylogic config [Java Application] C:\Program Files\AnyLogic 6_7 Ur

Properties Console

Problems Search

No problems

Description	Location
-------------	----------

Palettes...

Enabling Reporting Allows Output

The screenshot displays the AnyLogic software interface for a simulation titled "SingleAgentClassTwoPopulations : Simulation - AnyLogic University [EDUCATIONAL USE ONLY]". The main workspace shows a network of agents represented by green and red circles connected by lines. A control panel includes a "Seed New Infection" button and a "Enable Reporting" checkbox, which is checked. The simulation parameters are listed as follows:

- population: 100
- environment: 100 agents
- exposureHazard: 0.5
- isReportingEnabled: true

The console window at the bottom displays the following output:

```
anylogic config [Java Application] C:\Program Files\AnyLogic 6_7 Ur  
root.population[39]has been infected!  
root.population[26] has recovered!  
root.population[9]has been infected!  
root.population[25] has recovered!  
root.population[6] has recovered!  
root.population[44]has been infected!  
root.population[64]has been infected!
```

The status bar at the bottom indicates the simulation is running, with a progress bar showing 34% completion. The time taken is 33.52, and the memory usage is 15M of 6.

Cleaning Up by Separating the Network Display Space from Other Model Components

The screenshot shows the AnyLogic University software interface. The main workspace displays a model with a grid background. A red box highlights a stick figure agent icon, with a red arrow pointing to it from the text below. The interface includes a 'Projects' pane on the left, a 'Palette' on the right, and a 'Properties' pane at the bottom. The 'Properties' pane is titled 'population_presentation - Embedded Object Presentation' and shows settings for the selected agent.

population_presentation - Embedded Object Presentation

General: Name: Ignore Public Show at runtime

Advanced: Show in 3D scene

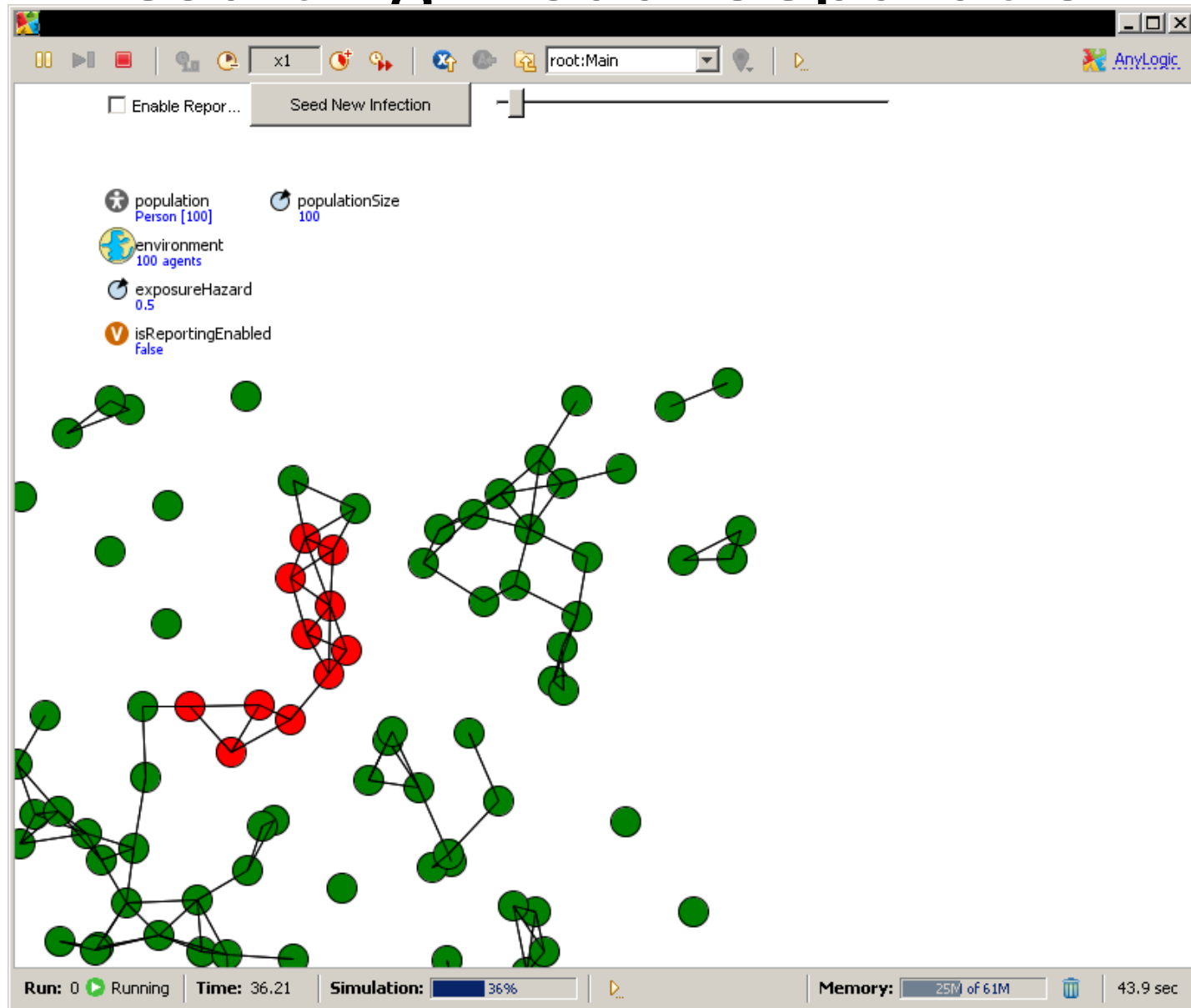
Dynamic: [empty]

Description: [empty]

This is the display “origin” for the agents. Positive coordinates for the agents will yield locations visually to the right and below this

population_presentation - Embedded Object Presentation | Selection | X=-36, Y=222

Resulting Visual Separation





Hands on Model Use Ahead



Load Example Model:

HardcodedMinimalistNetworkABMModelWithFileDrivenNetworkStructure

Recall: “Hardcoded” File Names

The screenshot displays the AnyLogic University software interface. On the left, a project tree shows a hierarchy of objects including 'Main', 'Parameters', 'Variables', 'Environments', 'Person', and 'Simulation: Main'. The central workspace contains a diagram with a 'population' object and an 'environment' object. Below the workspace, the 'Main - Active Object Class' properties panel is visible. The 'Startup code' field contains the following code:

```
//establishNetworkTransitionsAndPopulationsFromConnectivityMatrixFile("C:\\Usask\\Classes\\  
establishNetworkTransitionsAndPopulationsFromPajekNetworkFile("C:\\Usask\\Classes\\15879 Sp  
environment.applyLayout(); // now that established connectivity, perform layout
```

A red text annotation in the center of the workspace reads: "This currently 'hardcodes' that we are opening a particular Pajek file". A red arrow points from this text to the startup code in the properties panel.

Creating a Parameter to Communicate the Network File Name & Location (“Path”)

The screenshot displays the AnyLogic University interface. The main workspace shows a simulation diagram with a parameter named `networkFilePathAndName`. A red arrow points from a text annotation to the `String` type selection in the parameter's properties panel.

Indicate that this parameter holds a (reference to a) String

networkFilePathAndName - Parameter

General

Name: `networkFilePathAndName` Show name Ignore Show at runtime

Type: void (just action) boolean int double String Date Color Other: `String`

Use Units Unit:

Array

Default value:

Dynamic

On change:

Creating an "Enum" to Encode the Possible Types of the Specified File

The screenshot displays the AnyLogic University interface. The main workspace shows a simulation model with a circle and a line pointing to a list of objects. A red arrow points from the text "Specifies legal types of files" to the "Additional class code" field in the "Main - Active Object Class" properties panel.

Specifies legal types of files

Main - Active Object Class

Imports section:
`import java.io.*;`

Extends (single ActiveObject or Agent subclass):

Implements (comma-separated list of interfaces):

Additional class code:
`enum NetworkFileType { Pajek, ConnectivityMatrix };`

Persistent top-level presentation group
 Persistent top-level icon group
 Auto-create datasets for dynamic variables

Recurrence time:
Keep up to latest samples

Creating a Parameter to Encode the Network File Type

The screenshot displays the AnyLogic development environment. In the center, a parameter named `networkFileType` is being defined. A red text box with an arrow pointing to the 'Other' type field contains the following text: **Specifies that this parameter encodes the legal types of files (as specified by the NetworkFileType enumeration)**. The 'Other' field in the parameter properties is set to `Main.NetworkFileType`.

networkFileType - Parameter

General

Name: `networkFileType` Show name Ignore Show at runtime

Type: void (just action) boolean int double String Date Color Other: `Main.NetworkFileType`

Use Units Unit:

Array

Default value:

Dynamic

On change:

Projects

- HardcodedMinimalistNetworkABMModelWithFi
 - Main
 - Person
 - Simulation: Main

Main Simulation

HardcodedMinimalistNetworkABMModelWithFileDrivenNetworkStructure

Experiment setup page

Run the model and switch to Main view

Palette

- General
- System Dynamics
- Statechart
- Actionchart
- Analysis
- Presentation
- 3D
- Controls**
 - Button
 - Check Box
 - Edit Box
 - Radio Buttons
 - Slider
 - Combo Box
 - List Box
 - File Chooser
 - Progress Bar
- Connectivity
- Enterprise Library
- Pedestrian Library
- Rail Library
- Road Traffic Library - Preview
- Pictures
- 3D Objects
- Palettes...

Problems

No problems

Description	Location

Properties Console

HardcodedMinimalistNetworkABMModelWithFileDrivenNetworkStructure - Model

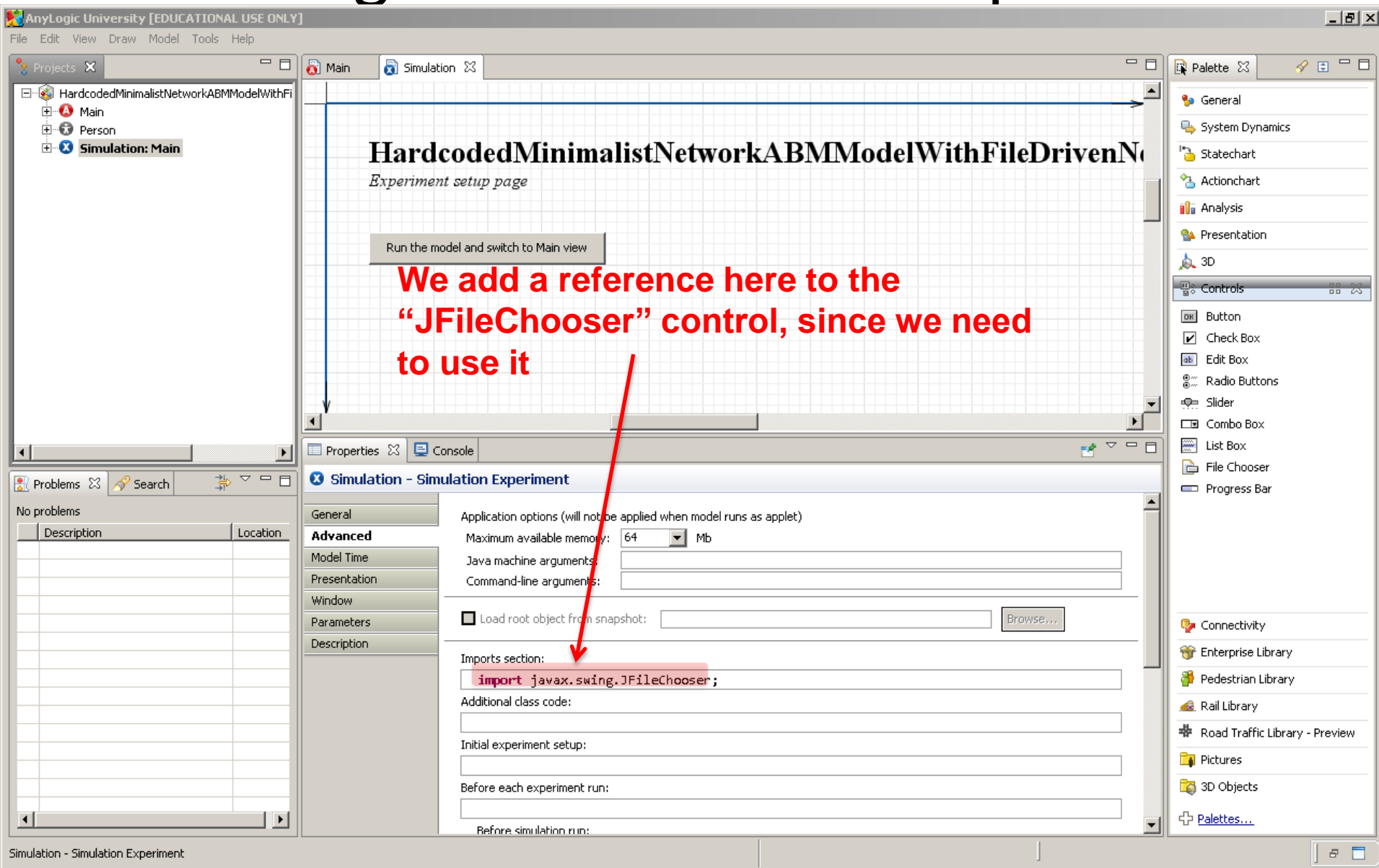
AnyLogic libraries required to build the model:

Name	Version	Location

Jar files and class folders required to build the model:

Location

Adding a Reference to the Java “Swing” File Chooser Component



Add an EditText

editboxNetworkFilePathAndName

The screenshot displays the AnyLogic University interface. The main workspace shows a simulation setup page titled "HardcodedMinimalistNetworkABMModelWithFileDrivenNetworkStructure" with the subtitle "Experiment setup page". The page contains a "Run the model and switch to Main view" button, a "Select File" button, and a file selection dialog with radio buttons for "Pajek File" and "Connectivity Matrix File".

The Properties panel at the bottom shows the configuration for the "editboxNetworkFilePathAndName - Edit Box".

editboxNetworkFilePathAndName - Edit Box

General

Name: Show name Ignore Public Icon

Link to:

Minimum value:

Maximum value:

Default value:

Enabled:

Action:

Adding a Label for the Filename

The screenshot displays the AnyLogic University interface. The main workspace shows a simulation setup page titled "HardcodedMinimalistNetworkABMModelWithFileDrivenNetworkStructure" with the subtitle "Experiment setup page". A red arrow points from the text "This is static text" to the "Network Input File:" label in the "textFileName - Text" properties panel.

Properties Panel: textFileName - Text

- Name: textFileName
- Color: black
- Font: SansSerif, 10 pt
- Text: Network Input File:

Simulation Page Elements:

- Buttons: "Run the model and switch to Main view", "Select File"
- Form: "Network Input File:"
- Radio Buttons: "Pajek File", "Connectivity Matrix File"

Logic to Set the File Name

Here, we open the Dialog box, which returns a value indicating the result.

HardcodedMinimalistNetworkABMModelWithFile
Experiment setup page

Run the model and switch to Main view

Select File

Network Input File:

If this return value indicates that the user wants to go ahead with the chosen file...

buttonSelectFile - Button

General

Name: buttonSelectFile Show name Ignore Public Icon

Label: Select File

Enabled:

Action:

```
JFileChooser c = new JFileChooser();  
// Demonstrate "Open" dialog:  
int iDialogResult = c.showOpenDialog(null);  
if (iDialogResult == JFileChooser.APPROVE_OPTION)  
    editboxNetworkFilePathAndName.setText(c.getCurrentDirectory().toString()+"\\"+c.getSelectedFile().getName());
```

...set the text in the Editbox to The "fully Qualified" filename (including path)

buttonSelectFile - Button Selection X=-83, Y=331

AnyLogic University [EDUCATIONAL USE ONLY]
File Edit View Draw Model Tools Help

Projects Main Simulation

HardcodedMinimalistNetwork
Main
Person
Simulation: Main
Presentation
frame
text: Hardcod
text1: Exper
textFileName:
button
buttonSelectF
editboxNetwo
radioButtonsF

HardcodedMinimalistNetworkABMModelWithFile

Experiment setup page

Run the model and switch to Main view

Select File

Network Input File:

Pajek File
Connectivity Matrix File

**Add these values.
Note that the order is
important – it must be
in the same order as
in the NetworkFileType
Enum (since we'll just use
the position to select the
Appropriate enum value)**

radioButtonsFileTypes - Radio Buttons

Name: radioButtonsFileTypes Show name Ignore Public Icon

Radio Buttons:

Pajek File	▲
Connectivity Matrix File	▼
<add new element>	✕

Orientation: Vertical Horizontal

Link to: _____

Default value: _____

Enabled: _____

Action: _____

General
Advanced
Dynamic
Description

No problems

Description

General
System Dy...
Statechart
Actionchart
Analysis
Presentation
3D
C...
Button
Check Box
Edit Box
Radio But...
Slider
Combo Box
List Box
File Chooser
Progress ...
Connectivity
Enterprise...
Pedestrian...
Rail Library
Road Traf...
Pictures
3D Objects
Palettes...

The screenshot displays the AnyLogic University interface. On the left, a project tree shows a simulation named 'Simulation: Main' with various components like 'Presentation', 'frame', and 'radioButtonsF'. The main workspace shows a diagram with components such as 'networkFilePathAndName', 'population [...]', and 'environment'. Below the workspace, the 'Simulation - Simulation Experiment' window is open, showing parameter settings for 'networkFilePathAndName' and 'networkFileType'. The 'networkFilePathAndName' parameter is set to `editboxNetworkFilePathAndName.getText()`, and the 'networkFileType' parameter is set to `Main.NetworkFileType.values()[radioButtonsFileTypes.getValue()]`. A red arrow points from the text 'For the network file path & name parameter, we just take the value from the editbox' to the first parameter value. A blue arrow points from the text 'For the network file type parameter (which requires an enum value), we just pass the enum value at the index given by the "Radio button" (the first enum value [i.e. at index 0] if the first radio button is selected, the second if the second radio button is selected)' to the second parameter value.

For the network file path & name parameter, we just take the value from the editbox

For the network file type parameter (which requires an enum value), we just pass the enum value at the index given by the "Radio button" (the first enum value [i.e. at index 0] if the first radio button is selected, the second if the second radio button is selected)

For the network file path & name parameter, we just take the value from the editbox

For the network file type parameter (which requires an enum value), we just pass the enum value at the index given by the "Radio button" (the first enum value [i.e. at index 0] if the first radio button is selected, the second if the second radio button is selected)

Startup Code for Main

The screenshot displays the AnyLogic software interface. On the left, a project tree shows the hierarchy: HardcodedMinimalistNetw > Main > Person > Simulation: Main > Presentation > frame > text: Hardcod, text1: Experir, textFileName:, button, buttonSelectF, editboxNetwo, and radioButtonsF. The main workspace contains a red text box with the following text: "For the startup code for Main, we call the appropriate method to process the specified file, where the identity of that method is indicated by the specified NetworkFileType". A red arrow points from this text box to the "Startup code:" field in the "Main - Active Object Class" properties window. The "Startup code:" field contains the following code:

```
switch (networkFileType)
{
    case Pajek:
        establishNetworkTransitionsAndPopulationsFromPajekNetworkFile(networkFilePathAndName);
        break;
    case ConnectivityMatrix:
        establishNetworkTransitionsAndPopulationsFromConnectivityMatrixFile(networkFilePathAndName);
        break;
    default:
        throw new RuntimeException("Unexpected networkFileType " + networkFileType);
}
environment.applyLayout(); // now that established connectivity, perform layout
```

The "Main - Active Object Class" properties window also shows the "Name" field set to "Main" and the "Agent" and "Generic" checkboxes unchecked. The "Destroy code:" field is empty.