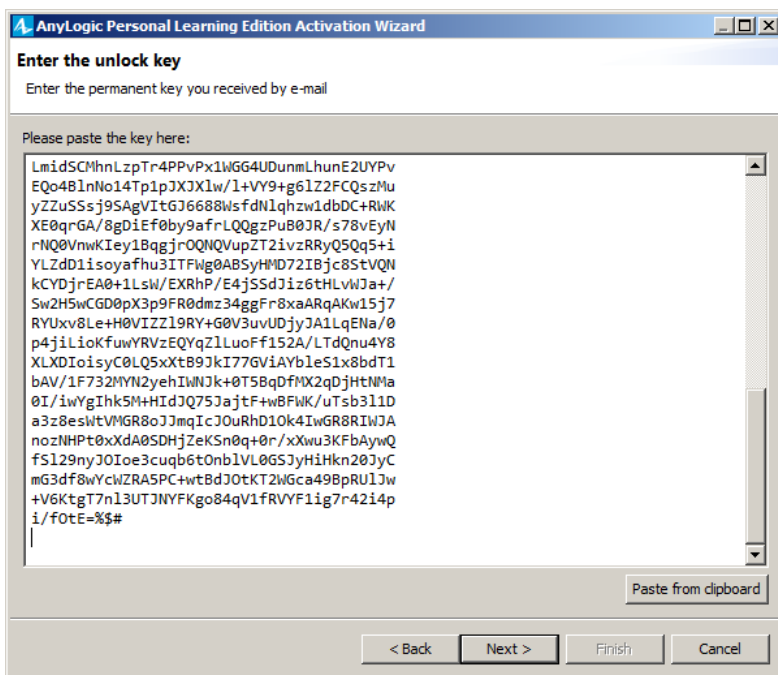
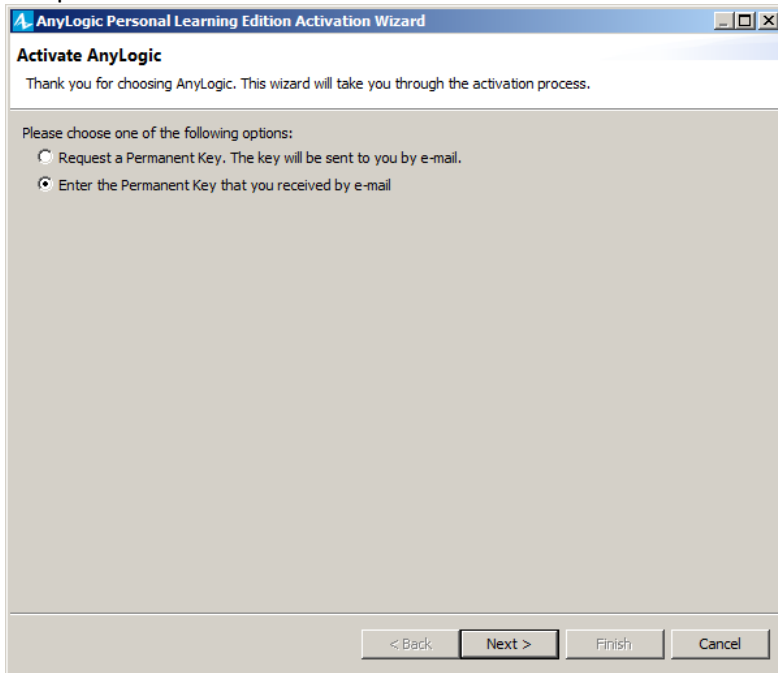


AnyLogic BlobWorld Tutorial (v22/02/2016)

Developed by Peer-Olaf Siebers and Tuong Vu
Nottingham University

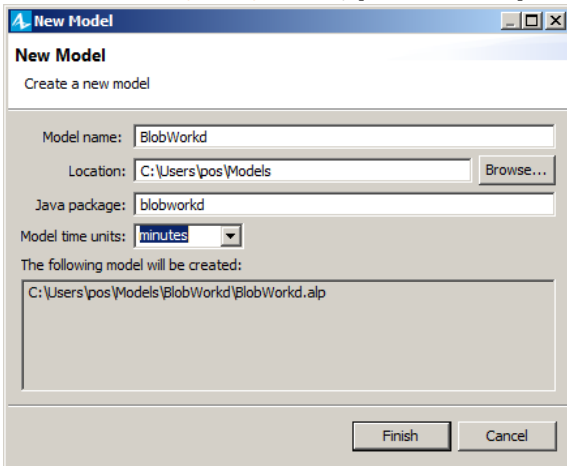
For this tutorial we use **AnyLogic v7.2.0 PLE**. You can download it for free and install it on your own computer or use the virtual desktop where it is pre-installed under "UoN Applications/_School of Computer Science". In both cases you will have to apply for a permanent key via email, which arrives very fast. Once you received the key just copy it and paste it when asked.



Once you have registered your copy you can start with the modelling process.

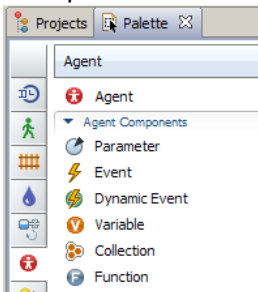
Create a new model

- Choose **File/New/Model**, [Model name] **BlobWorld**; {do not type the ;} [Model time units] **minutes**

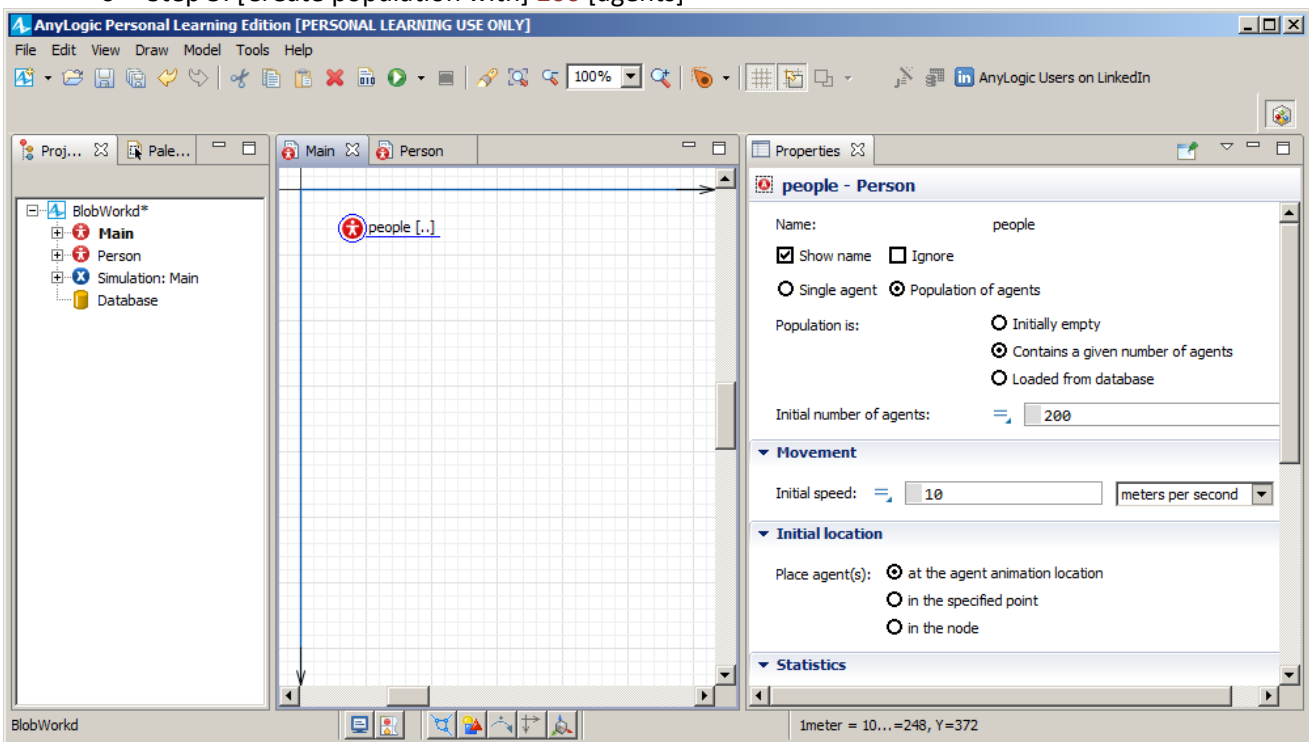


Create a new agent type **Person** and a population of 200 agents

- Choose the **Project window** and double click on **Main**
- Choose the **Palette window** and then the **Agent palette**; drag an **Agent** to the **Main window**. A wizard will pop up.

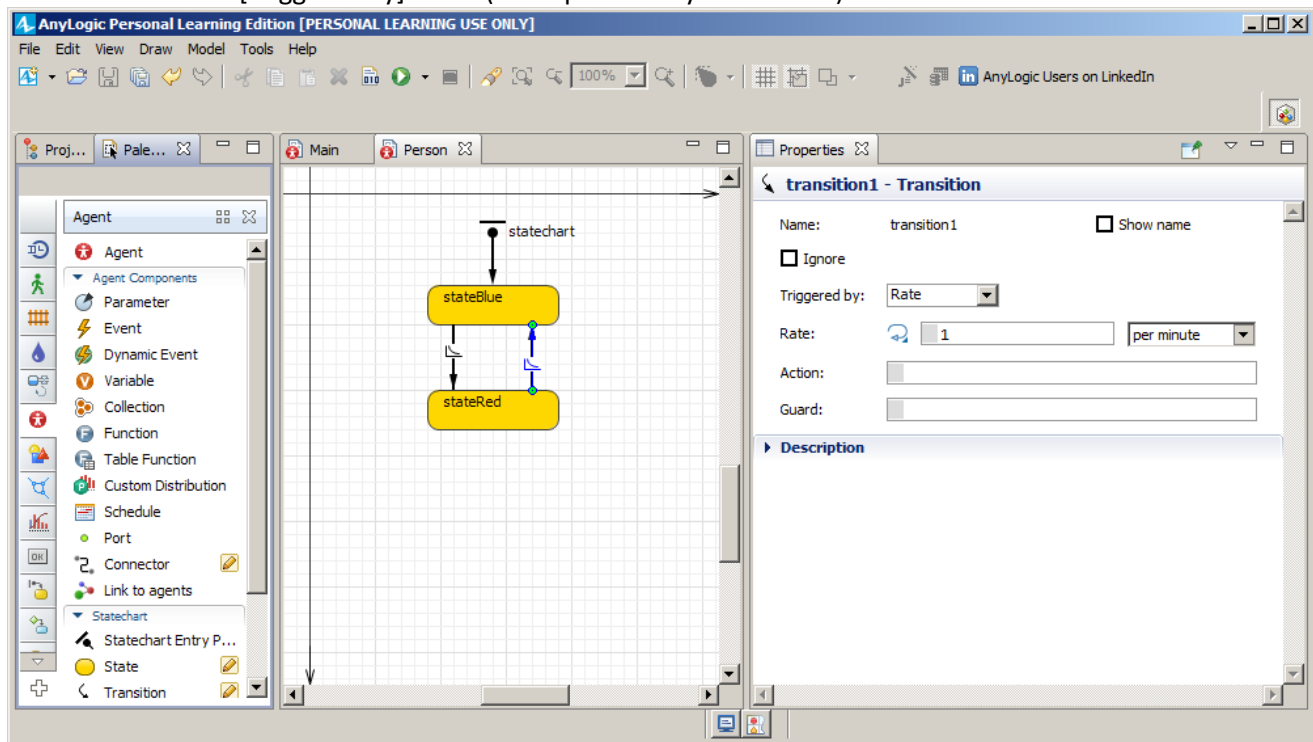


- Step 1: Choose **Population of agents**
- Step 2: [Agent type name] **Person**; [Agent population name] **people**
- Step 3: [Choose animation] **None**
- Step 4: /
- Step 5: [Create population with] **200** [agents]



Create basic agent behaviour

- Choose the **Project window** and double click on **Person**
- Choose the **Palette window** and then the **Agent palette**; create a state chart as shown below
- Transitions are [Triggered by] **Rates** (i.e. exponentially distributed)



Create some statistics to display state changes

- Choose **Project window** and double click on **Main**; in the **Main window** choose **people[..]**
- In the **Properties window** go to the **Statistics** section and choose [+] to add new data items **stateBlue** and **stateRed**

The screenshot shows the 'Statistics' section in the Properties window. It contains two statistics: 'stateBlue' and 'stateRed'. Both are set to 'Count' type and have a condition 'item.statechart.isStateActive(Person.stateBlue)' and 'item.statechart.isStateActive(Person.stateRed)' respectively.

- Choose the **Palette window** and then the **Analysis palette**; drag a **Time Plot** chart into the **Main window**
- In the **Properties window** go to the **Data** section and choose [+] to add new data items **stateBlue** and **stateRed**

Data

☒ Value ☐ Data set

Title: stateBlue

Value: people.stateBlue()

Point style: [dropdown]

Line width: [dropdown] 1 pt

Color: blue

☒ Value ☐ Data set

Title: stateRed

Value: people.stateRed()

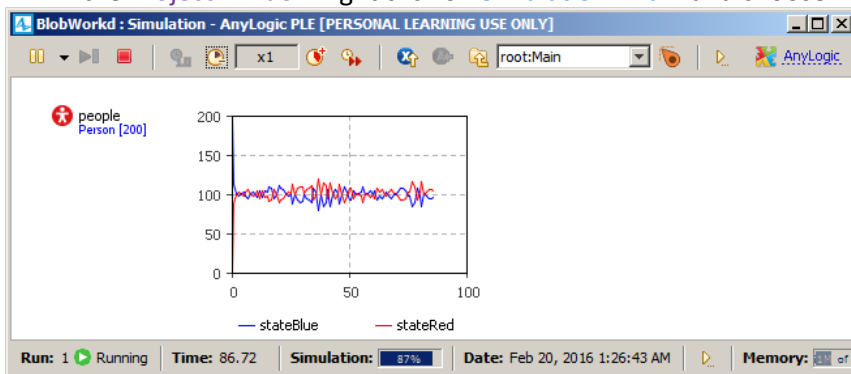
Point style: [dropdown]

Line width: [dropdown] 1 pt

Color: red

[+][x][up][down]

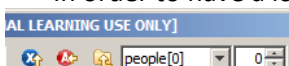
- In the **Properties window** go to the **Scale section** and choose [Vertical scale] **Fixed**; [From] 0 [to] 200
- Prepare simulation for first run and run it
- In the **Project window** double click on **Simulation: Main**
- In the **Properties window** go to the **Model time section** and choose [Stop] Stop at specific time
- In the **Project window** right click on **Simulation: Main** and choose **Run** from the pop-up menu



- In order to speed up / slow down the simulation, use the following buttons



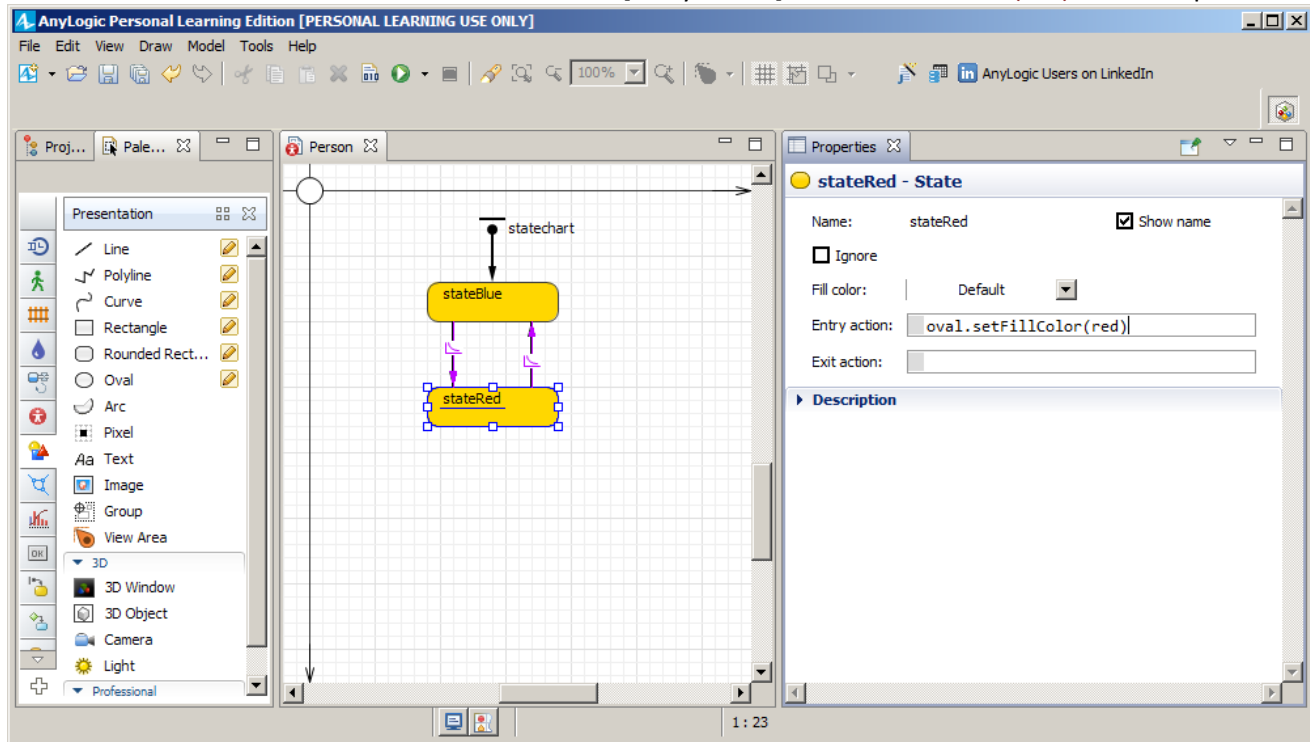
- In order to have a look at individual agent states choose an agent from the agent array



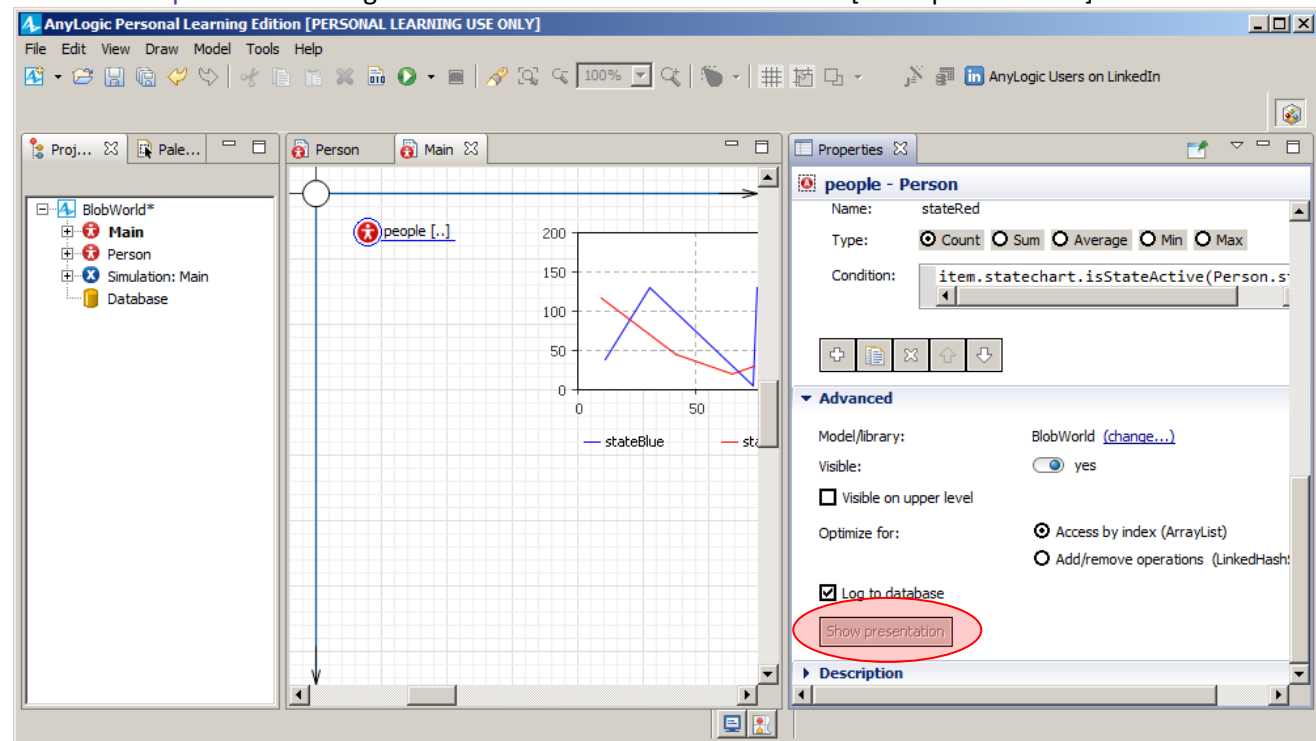
Your model corresponds to **BlobWorld1.alp** that comes with this tutorial

Create agent representation

- Choose the **Palette window** and then the **Presentation palette**; drag an **Oval** into the **Person window**
- In the **Properties window** go to the **Position and size** section and choose [x] 0; [Y] 0; [Radius] 10
- Choose **stateBlue** in the **Person window** and set [Entry action] to **oval.setFillColor(blue)** in the Properties window
- Choose **stateRed** in the **Person window** and set [Entry action] to **oval.setFillColor(red)** in the Properties window



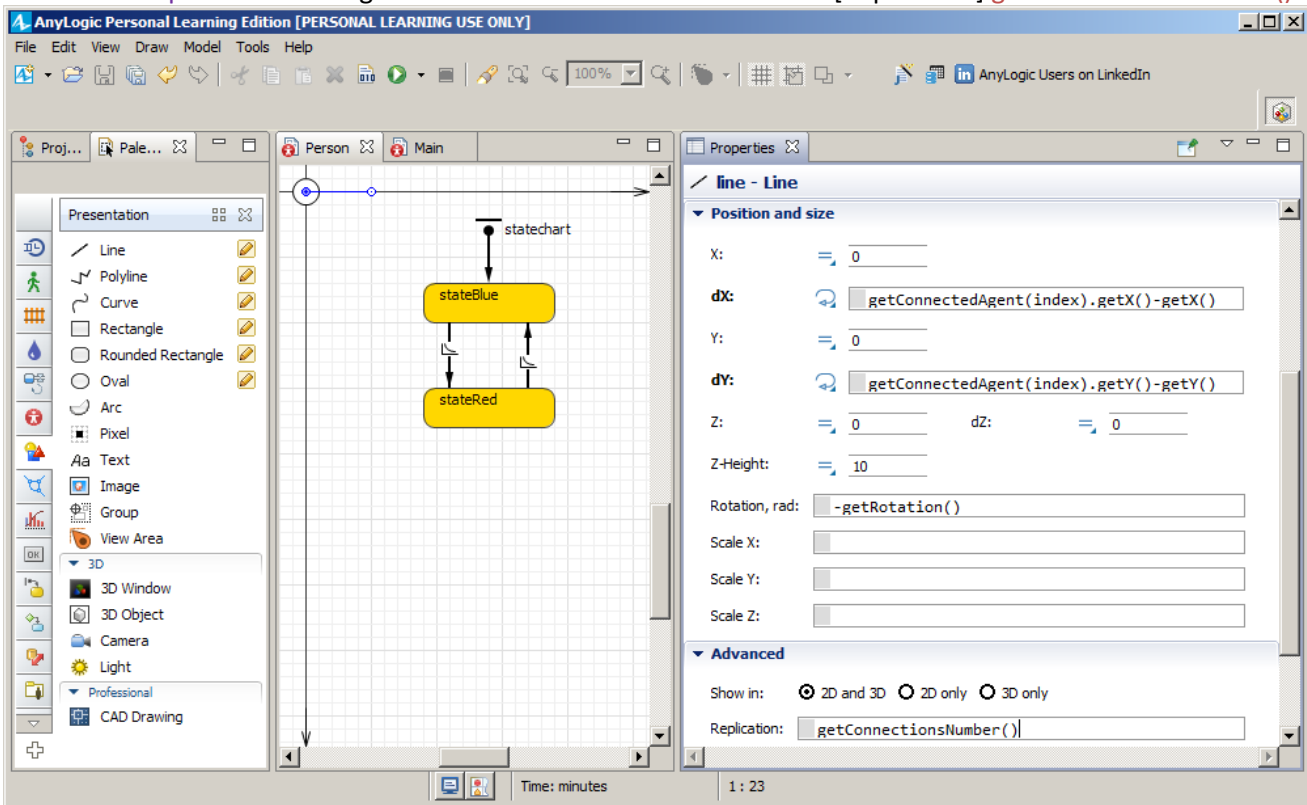
- Choose **Project window** and double click on **Main**; in the **Main window** choose **people[.]**
- In the **Properties window** go to the **Advanced** section and click the [Show presentation] button



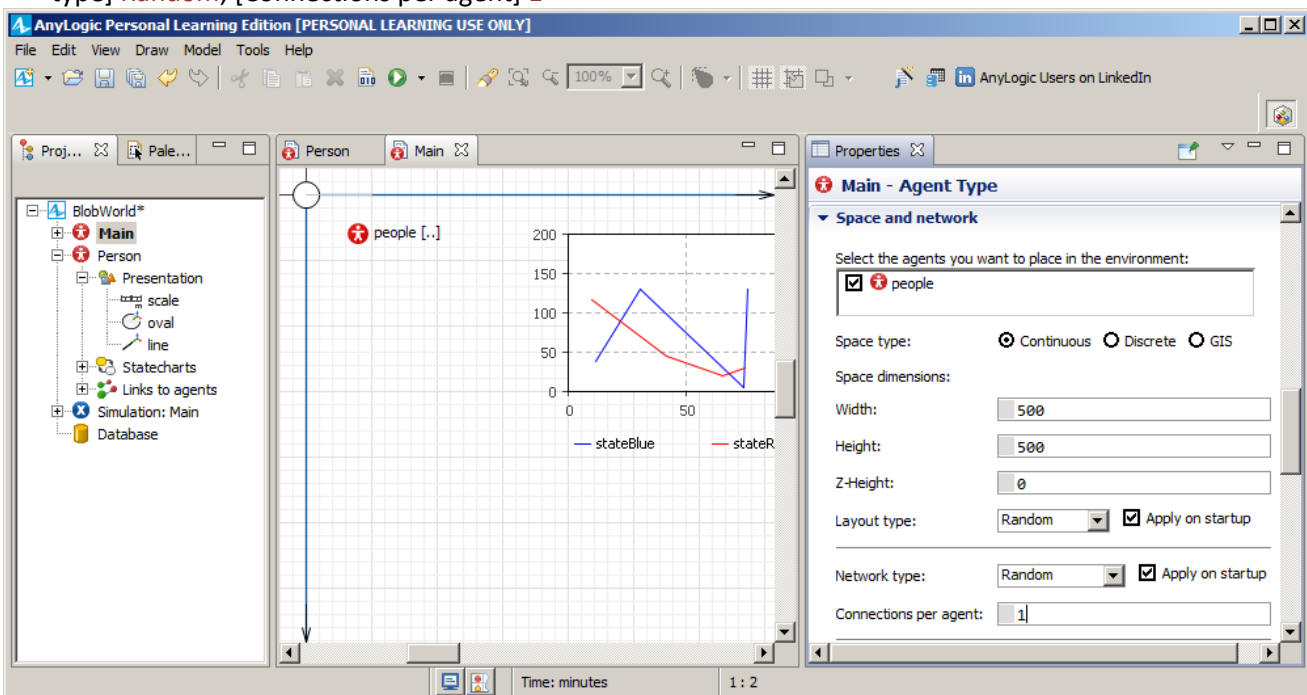
Make the agent move to random locations

- Choose the **Project window** and double click on **Person**
- In the **Properties window** go to the **Agent actions** section and choose [Startup code]
`moveTo(uniform(500),uniform(500));` [Java code and therefore requires the ;]
[On arrival at target location]
`moveTo(uniform(500),uniform(500));` [Java code and therefore requires the ;]

- In the **Properties window** go to the **Movement** section and choose [Initial speed] **0.1 kilometers/hour**
- Show links between agents
- Choose the **Palette window**; drag a **Line** into the **Person window**
- In the **Properties window** go to the **Appearance** section and choose [Line color] **silver**
- In the **Properties window** go to the **Position and size** section and choose [X] **0**; [Y] **0**; [dX (click on the = sign to change to dynamic values)] **getConnectedAgent(index).getX()-getX()**; [dY (click on the = sign to change to dynamic values)] **getConnectedAgent(index).getY()-getY()**; [Rotation, rad] **-getRotation()**
- In the **Properties window** go to the **Advanced** section and choose [Replication] **getConnectionsNumber()**

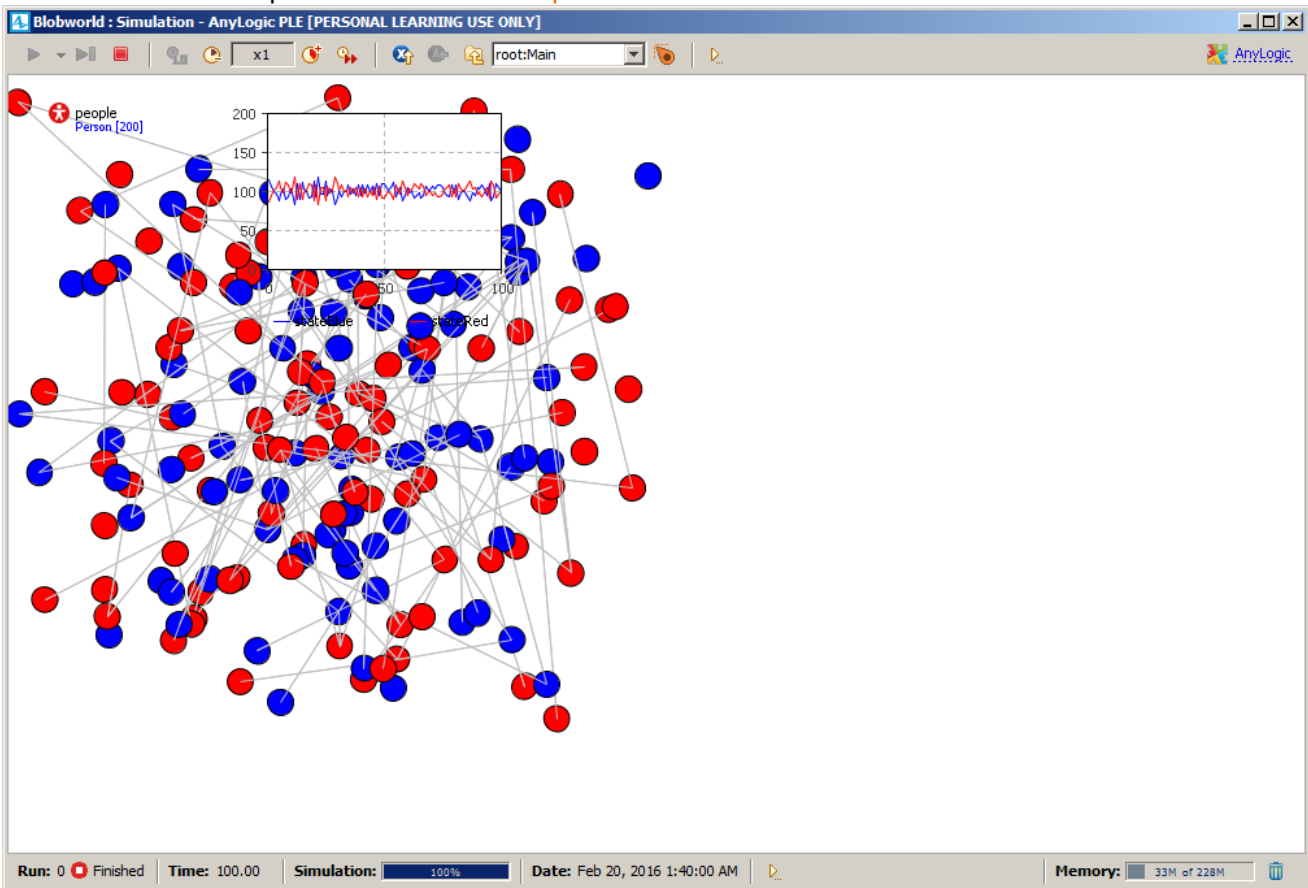


- Choose the **Project window** and double click on **Main**
- In the **Properties window** go to the **Space and network** section and choose [Layout type] **Random**; [Network type] **Random**; [Connections per agent] **1**



Run and test your model

- In the **Project window** right click on **Simulation: Main** and choose **Run** from the pop-up menu
- Your model corresponds to **BlobWorld2.alp** that comes with this tutorial

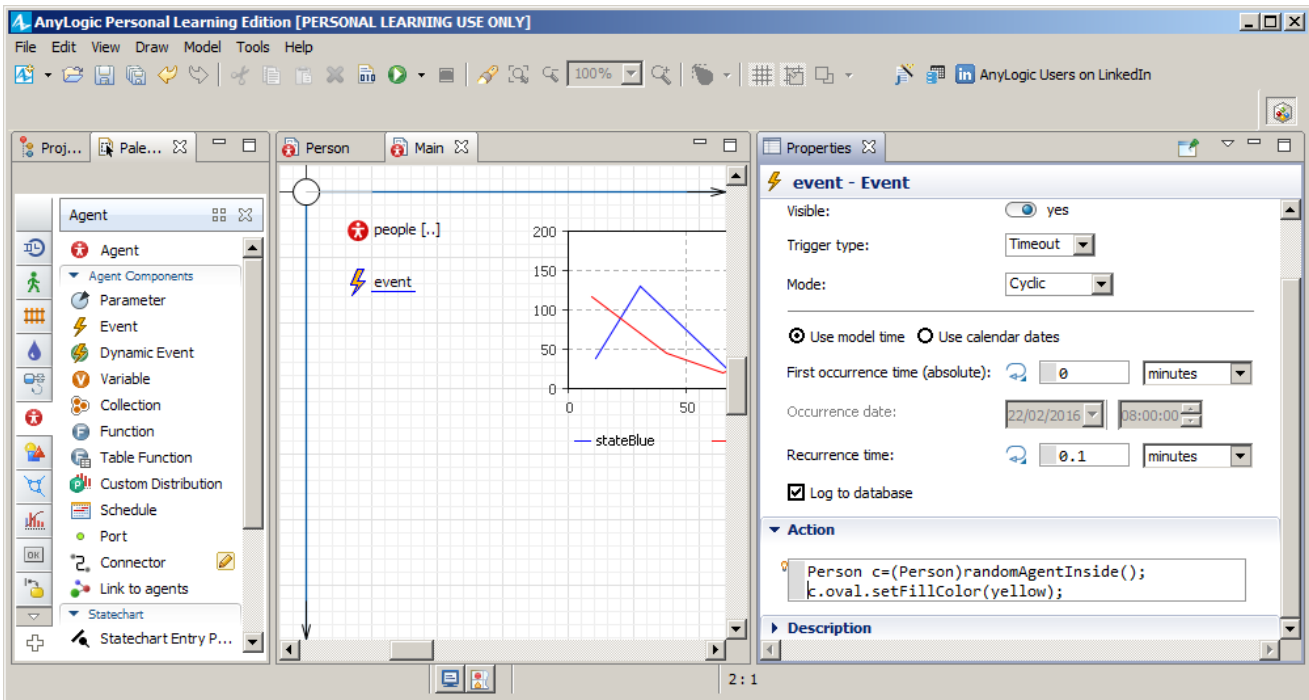


Play with the model

- Try out different Layout and Network types (e.g. **Ring** and **Ring lattice**)!

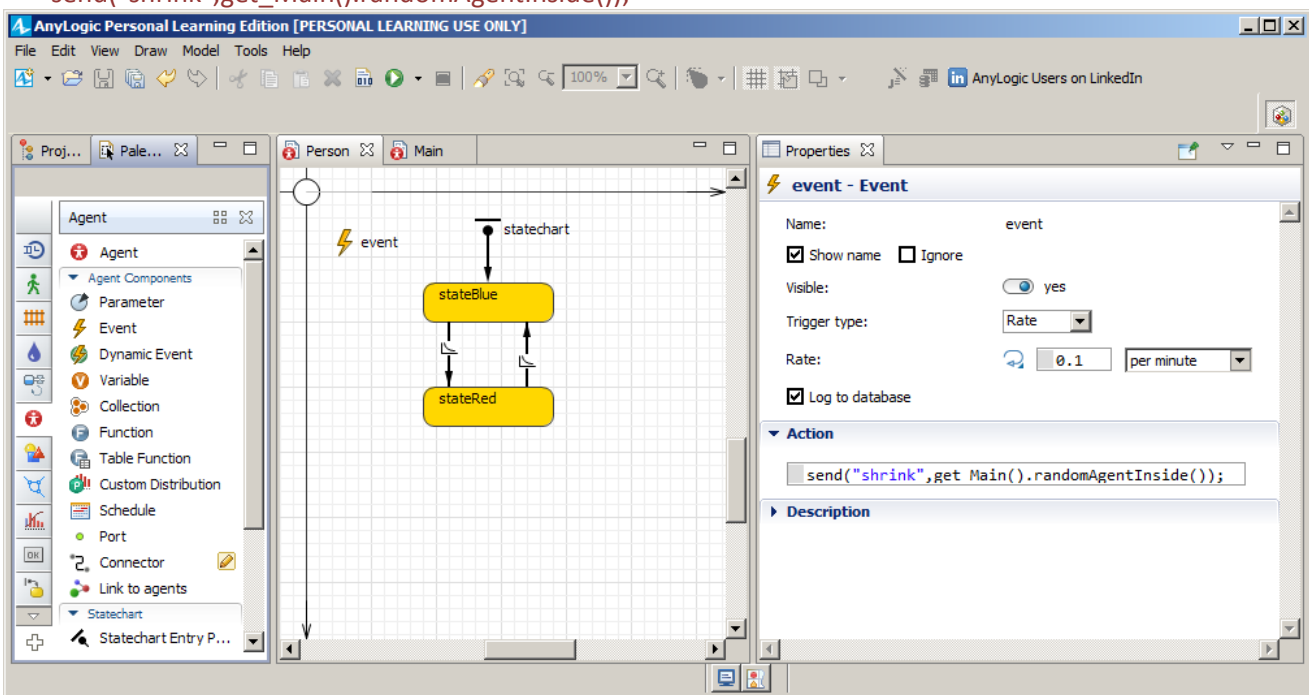
Address individual agents (change colour of random agent to yellow)

- Choose the **Project window** and double click on **Main**
- Choose the **Palette window** and then the **Agent palette**; drag an **Event** to the **Main window**
- In the **Properties window** go to the **General section** and choose [Mode] **Cyclic**; [Recurrence time] **0.1**; [Action]
`Person c=(Person)randomAgentInside(); c.oval.setFillColor(yellow);`

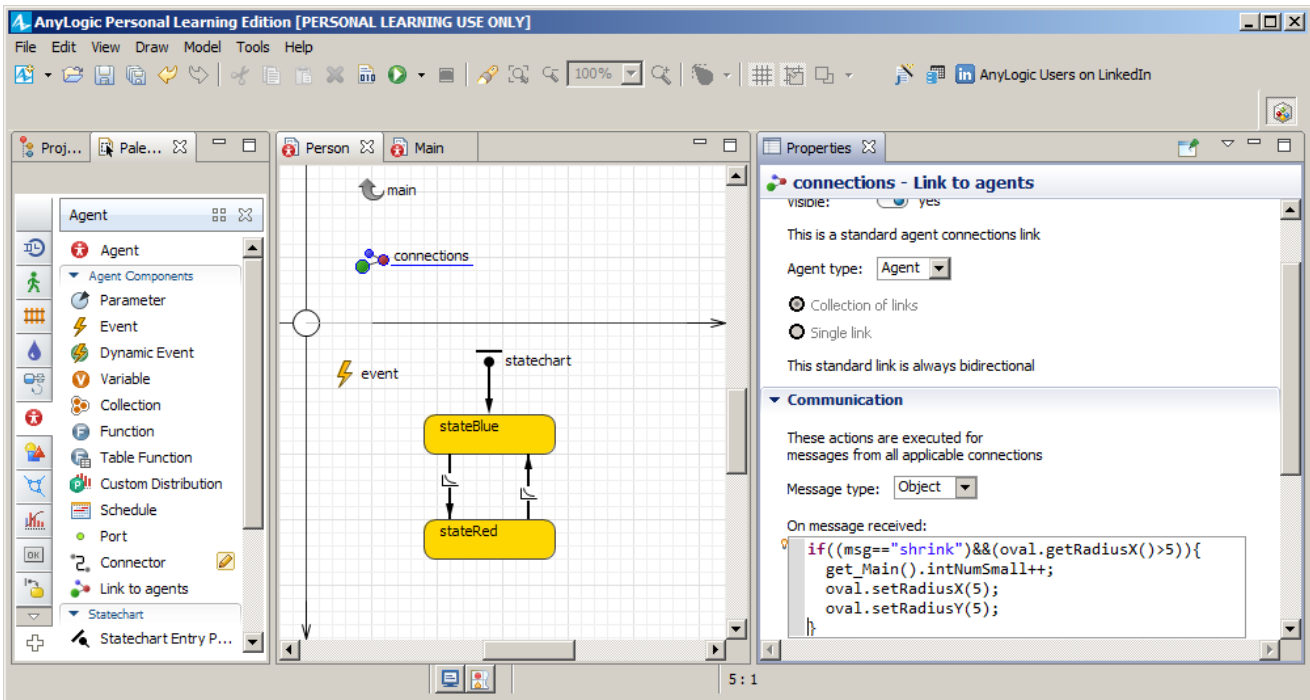


Send messages between agents (send and receive commands to self-shrink)

- Choose the **Palette window** and then the **Agent palette**; drag an **Variable** to the **Main window**
- In the **Properties window** go to the **General** section and choose [Name] `intNumSmall`; [Type] `int`; [Initial value] `0`
- Choose the **Project window** and double click on **Person**
- Choose the **Palette window** and then the **Agent palette**; drag an **Event** to the **Person window**
- In the **Properties window** go to the **General** section and choose [Mode] `Rate`; [Rate] `0.1 per minute`; [Action] `send("shrink",get_Main().randomAgentInside());`

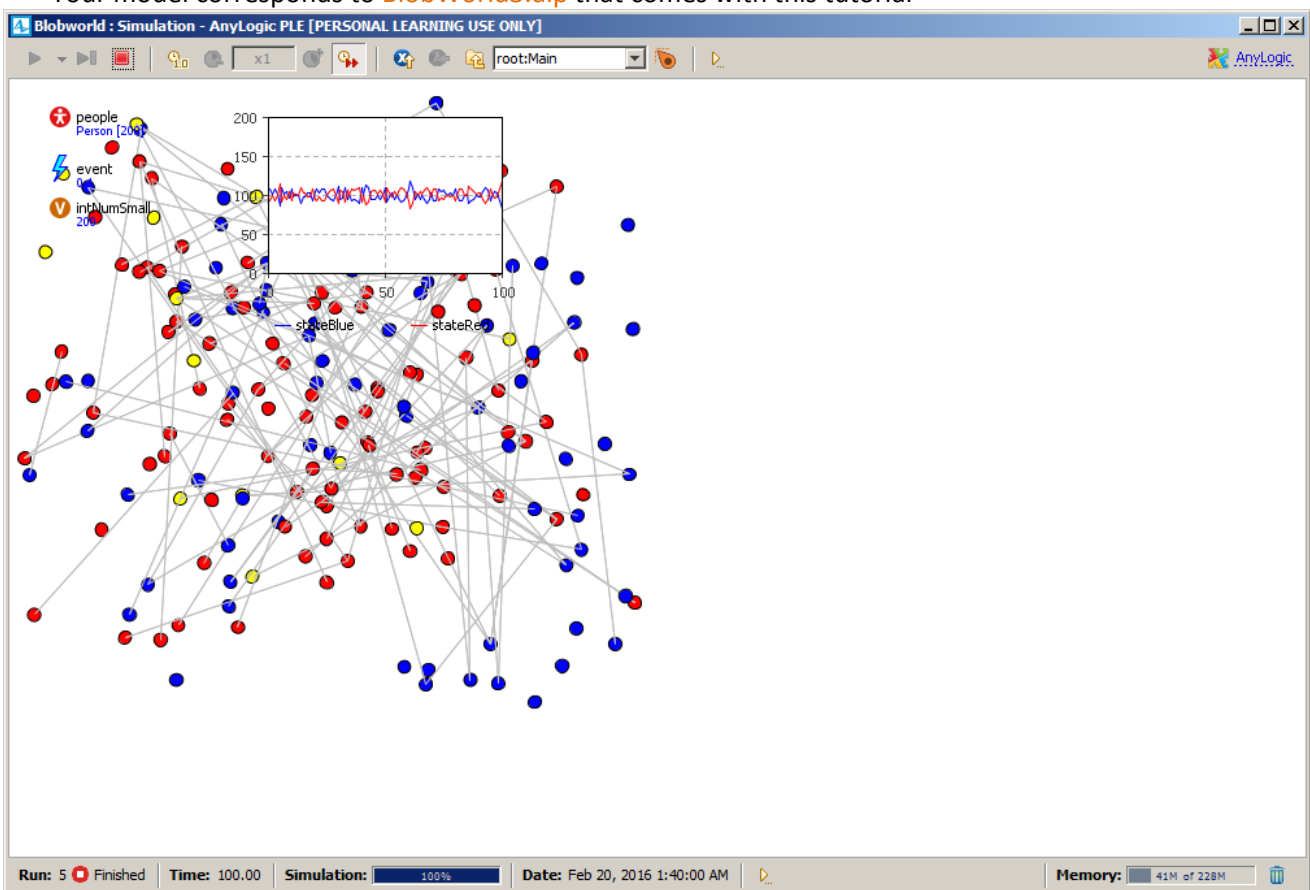


- Scroll the **Person window** so that you can see the **connection** symbol and click on it
- In the **Properties window** go to the **Communication** section and choose [On message received] `if((msg=="shrink")&&(oval.getRadiusX()>5)){get_Main().intNumSmall++; oval.setRadiusX(5); oval.setRadiusY(5);}`



Run and test your model

- In the **Project window** right click on **Simulation: Main** and choose **Run** from the pop-up menu
- Your model corresponds to **BlobWorld3.alp** that comes with this tutorial



Play with the model

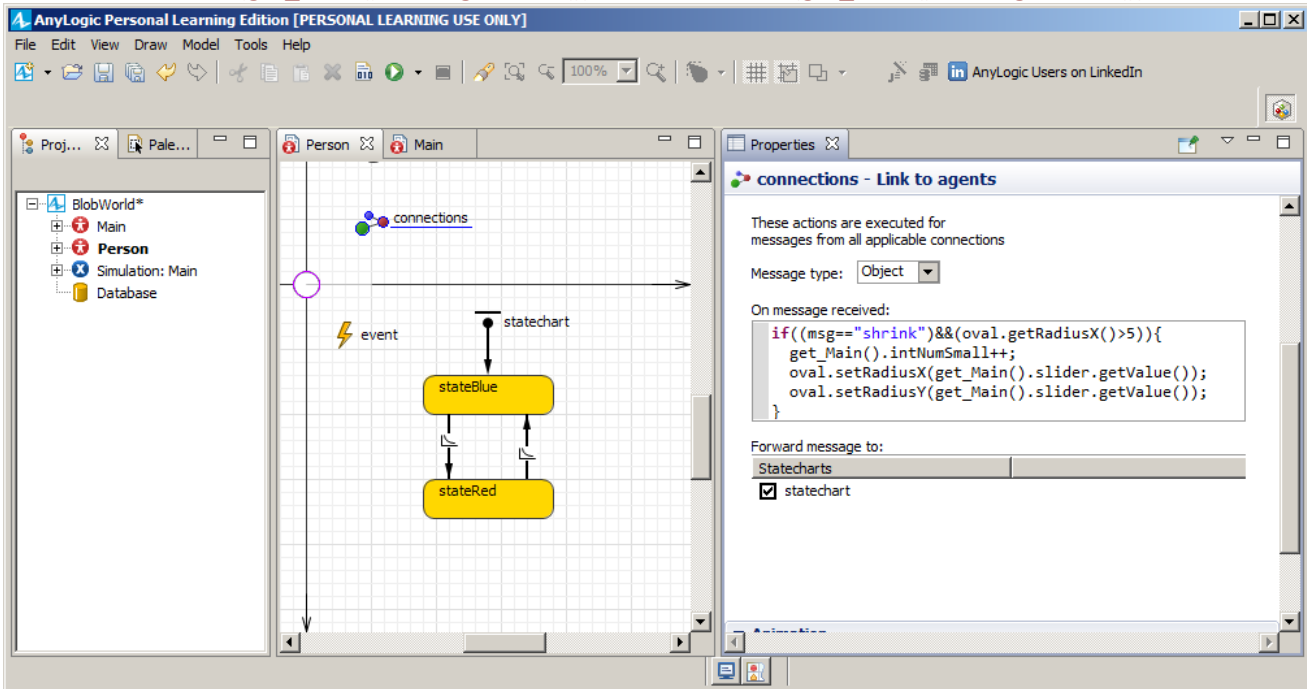
- Make the blobs shrink and grow (come up with a story why they would shrink and grow)

Add controls

- Choose the **Project window** and double click on **Main**
- Choose the **Palette window** and then the **Controls palette**; drag an **Slider** to the **Main window**

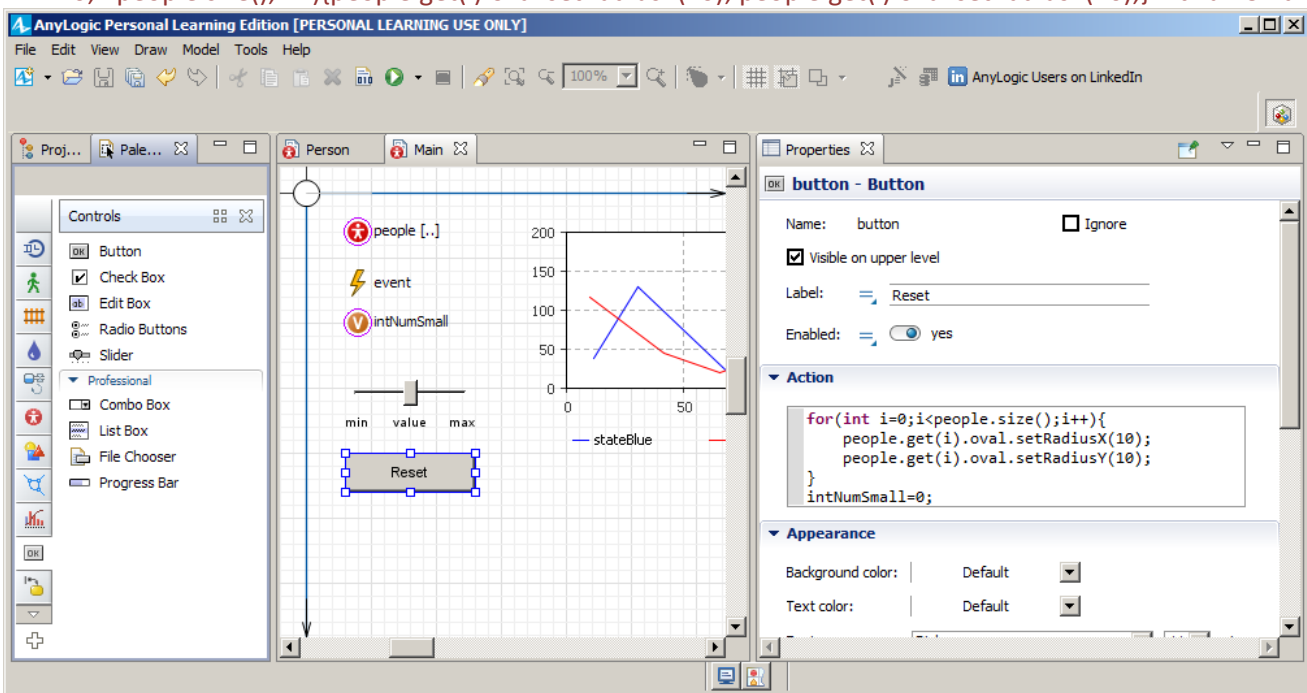
- In the **Properties window** go to the **General section** and choose [Minimum value] 2; [Maximum value] 5; [Default value] 5; click the [Add labels] button
- Choose the **Project window** and double click on **Person**
- Scroll the **Person window** so that you can see the **connection** symbol and click on it
- In the **Properties window** go to the **Communication section** and replace [On message received]

```
if((msg=="shrink")&&(oval.getRadiusX(>5)){get_Main().intNumSmall++;
oval.setRadiusX(get_Main().slider.getValue()); oval.setRadiusY(get_Main().slider.getValue());}
```



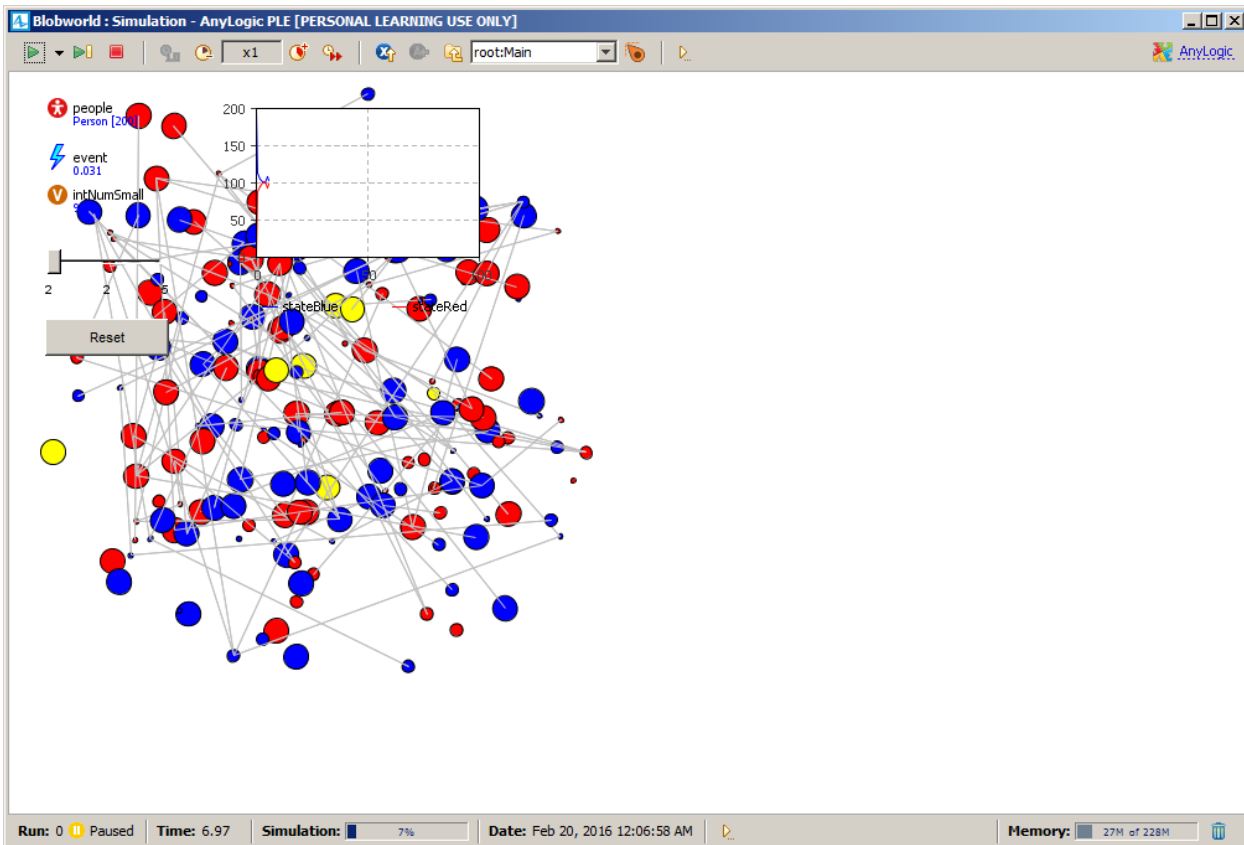
- Choose the **Project window** and double click on **Main**
- Choose the **Palette window** and then the **Controls palette**; drag an **Button** to the **Main window**
- In the **Properties window** go to the **General section** and choose [Label] **Reset**
- In the **Properties window** go to the **Action section** and add code for(int

```
i=0;i<people.size();i++){people.get(i).oval.setRadiusX(10); people.get(i).oval.setRadiusY(10);} intNumSmall=0;
```



Run and test your model

- In the **Project window** right click on **Simulation: Main** and choose **Run** from the pop-up menu
- Your model corresponds to **BlobWorld4.alp** that comes with this tutorial



Challenge

- Think about a serious application for this abstract model (e.g. this could represent an employment model where red represents "unemployed", blue represents "employed", yellow represents "in training", and messages represent different marketing strategies). If you found an interesting application of this abstract model please email me your ideas (pos@cs.nott.ac.uk).
- Try to expand this model (e.g. collecting more statistics; redefining rules for transitions; ...)