

G54SIM AnyLogic BlobWorld Tutorial v02 (04/11/2011)

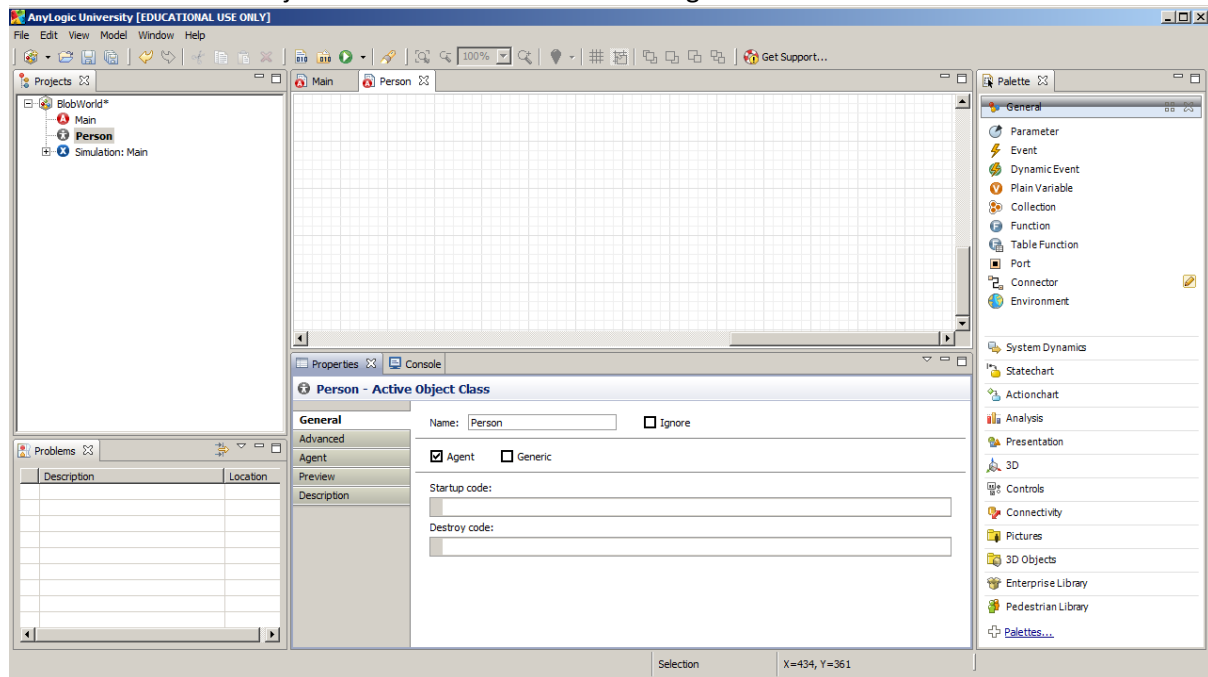
Peer-Olaf Siebers

Nottingham University

peer-olaf.siebers@nottingham.ac.uk

Create a new model **BlobWorld**

Create a new Active Object class **Person** and make it an Agent class

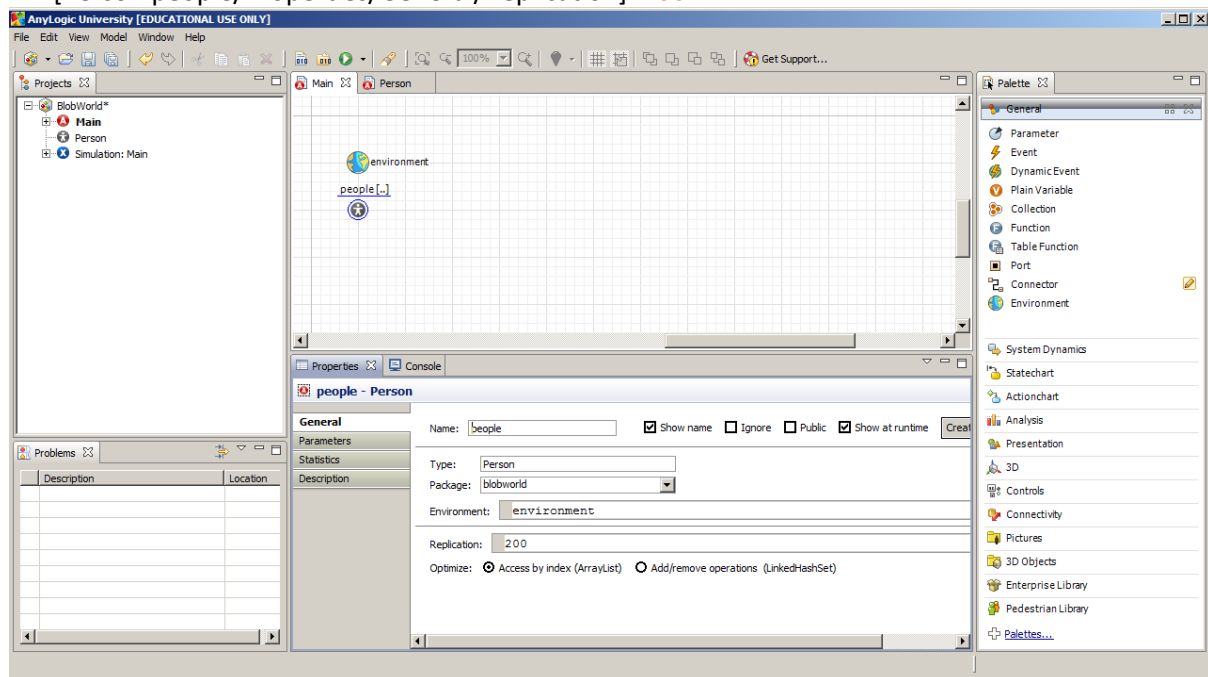


Create environment

- In the **Project window** click twice on the Main class
- From the **Palette window (General)** drag an Environment into the **Main window**

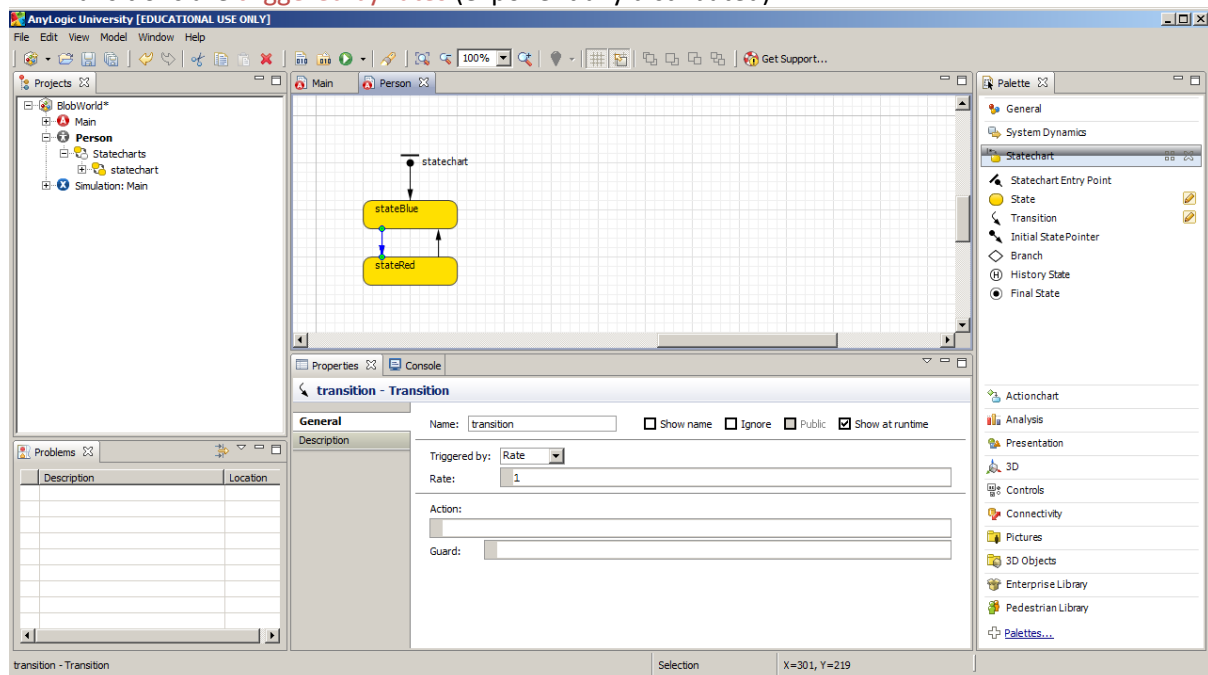
Create 200 instances of the **Person** class and add them to the **environment**

- From the Project window drag **Person** class into **Main window**
- [Person person/Properties/General/Name]: **people**
- [Person people/Properties/General/Environment]: **environment**
- [Person people/Properties/General/Replication]: **200**



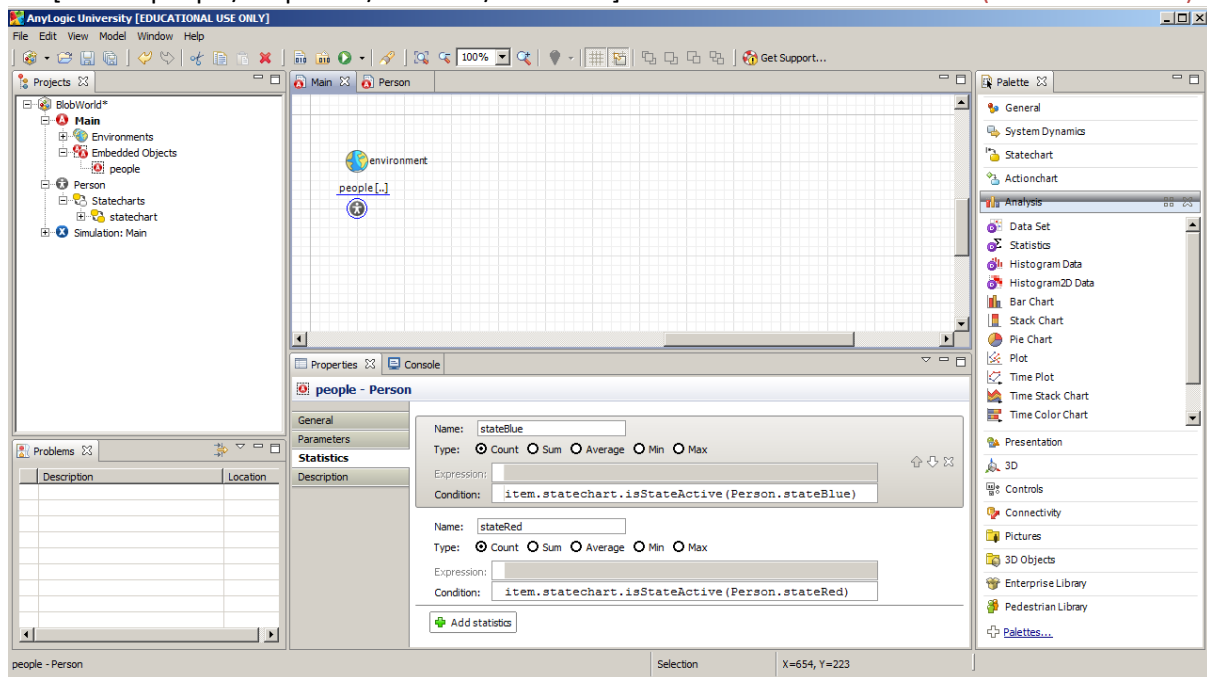
Create basic agent behaviour

- In the **Person** class create a state chart
- Transitions are **triggered by rates** (exponentially distributed)



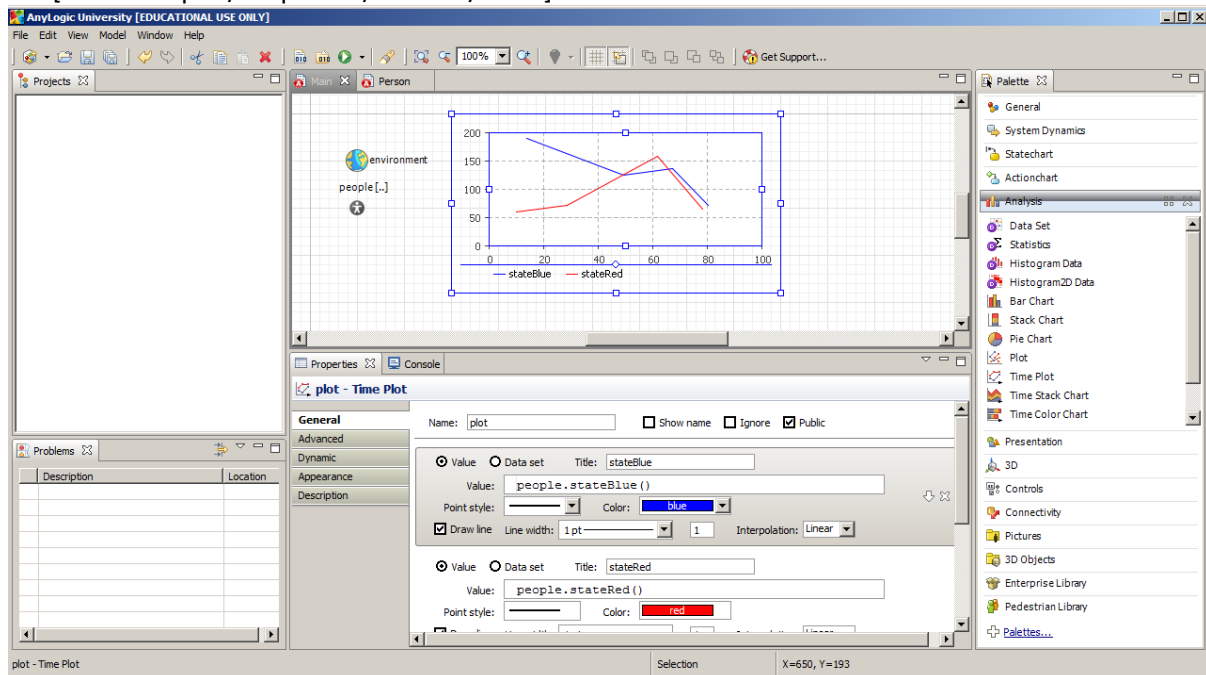
Create some statistics to display state changes

- [Person people/Properties/Statistics]: Add statistics
- [Person people/Properties/Statistics/Name]: **stateBlue**
- [Person people/Properties/Statistics/Condition]: **item.statechart.isStateActive(Person.stateBlue)**
- [Person people/Properties/Statistics]: Add statistics
- [Person people/Properties/Statistics/Name]: **stateRed**
- [Person people/Properties/Statistics/Condition]: **item.statechart.isStateActive(Person.stateRed)**

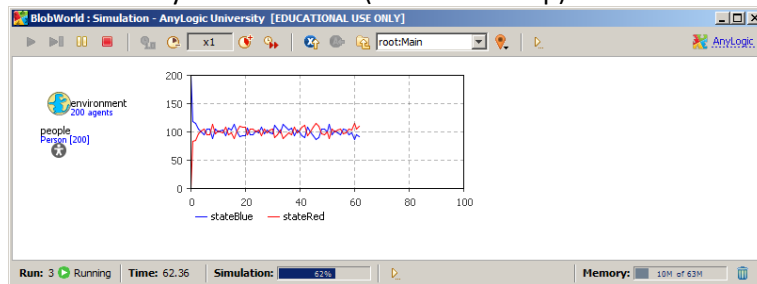


Create some statistics to display state changes (cont.)

- From the **Palette window (Analysis)** drag a **Time Plot** chart into the **Main window**
- [Time Plot plot/Properties/General/Vertical scale]: Fixed from **0** to **200**
- [Time Plot plot/Properties/General]: Add data item
- [Time Plot plot/Properties/General/Title]: **stateBlue**
- [Time Plot plot/Properties/General/Value]: **people.stateBlue()**
- [Time Plot plot/Properties/General/Color]: **blue**
- [Time Plot plot/Properties/General]: Add data item
- [Time Plot plot/Properties/General/Title]: **stateRed**
- [Time Plot plot/Properties/General/Value]: **people.stateRed()**
- [Time Plot plot/Properties/General/Color]: **red**

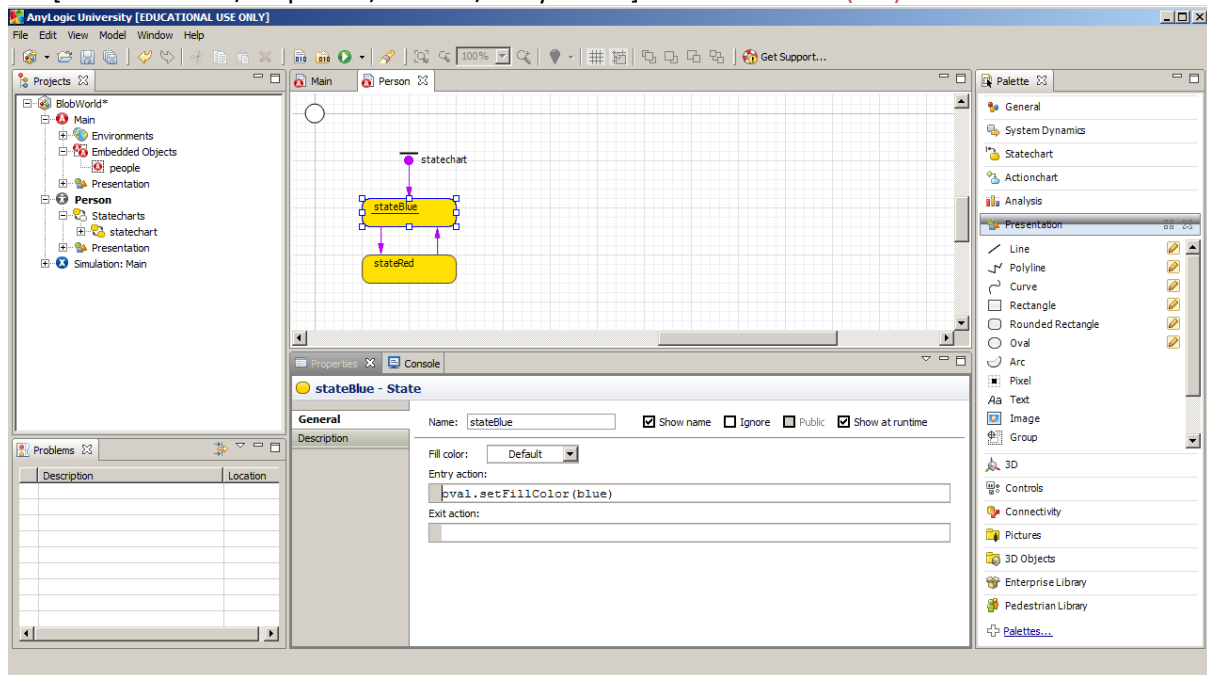


Save and test your simulation (BlobWorld1.alp)



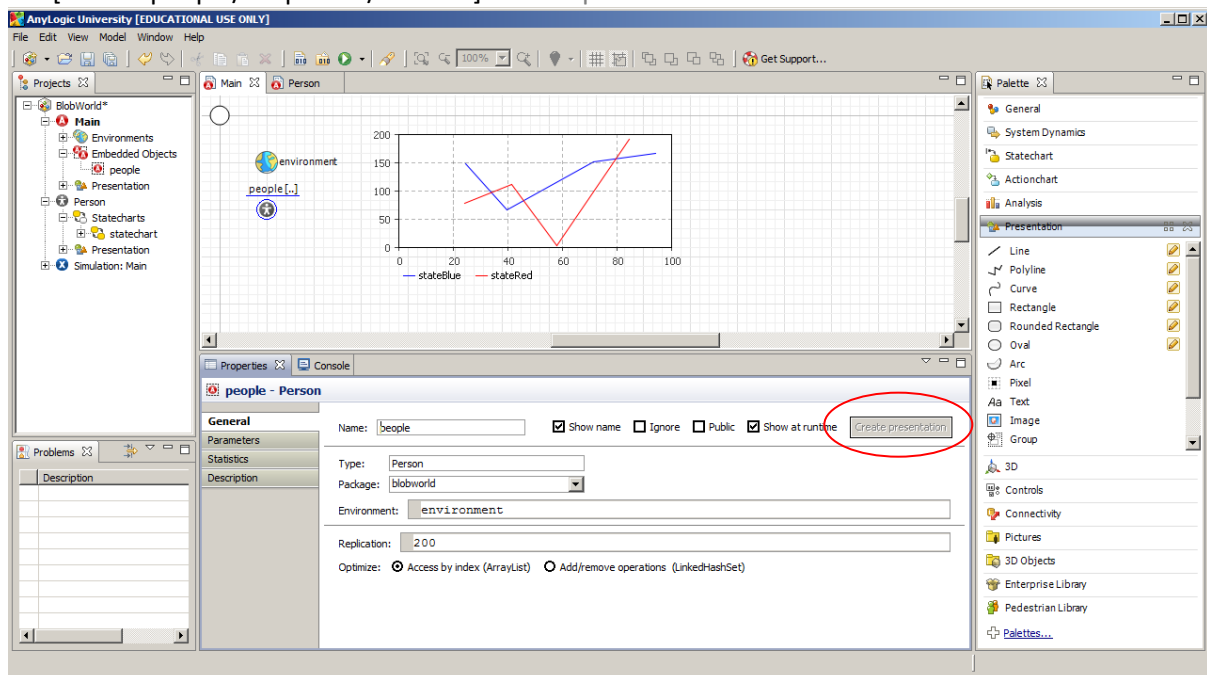
Create agent representation

- From the **Palette window (Presentation)** drag an **Oval** into the **Person** window
- [Oval oval/Properties/Advanced/Position X]: 0
- [Oval oval/Properties/Advanced/Position Y]: 0
- [Oval oval/Properties/Advanced/Radius X]: 10
- [Oval oval/Properties/Advanced/Radius Y]: 10
- [State stateBlue/Properties/General/Entry action]: `oval.setFillColor(blue)`
- [State stateRed/Properties/General/Entry action]: `oval.setFillColor(red)`



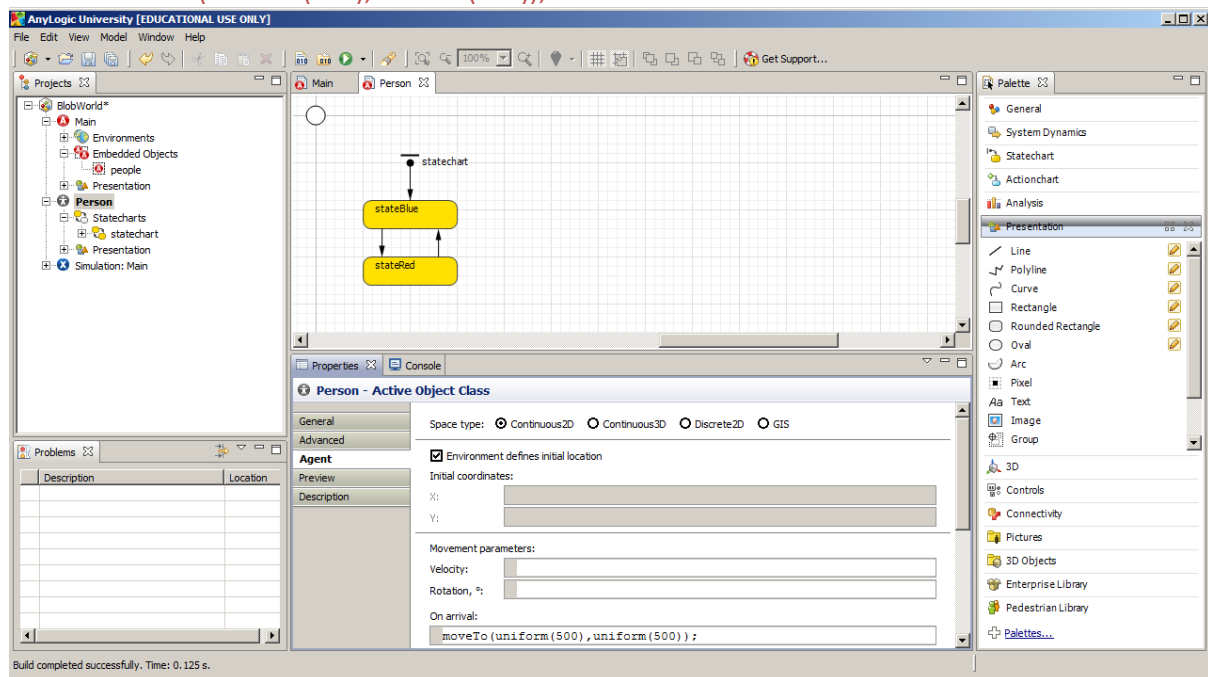
Create agent representation (cont.)

- In the main Window click on **people [..]**
- [Person people/Properties/General]: Create presentation



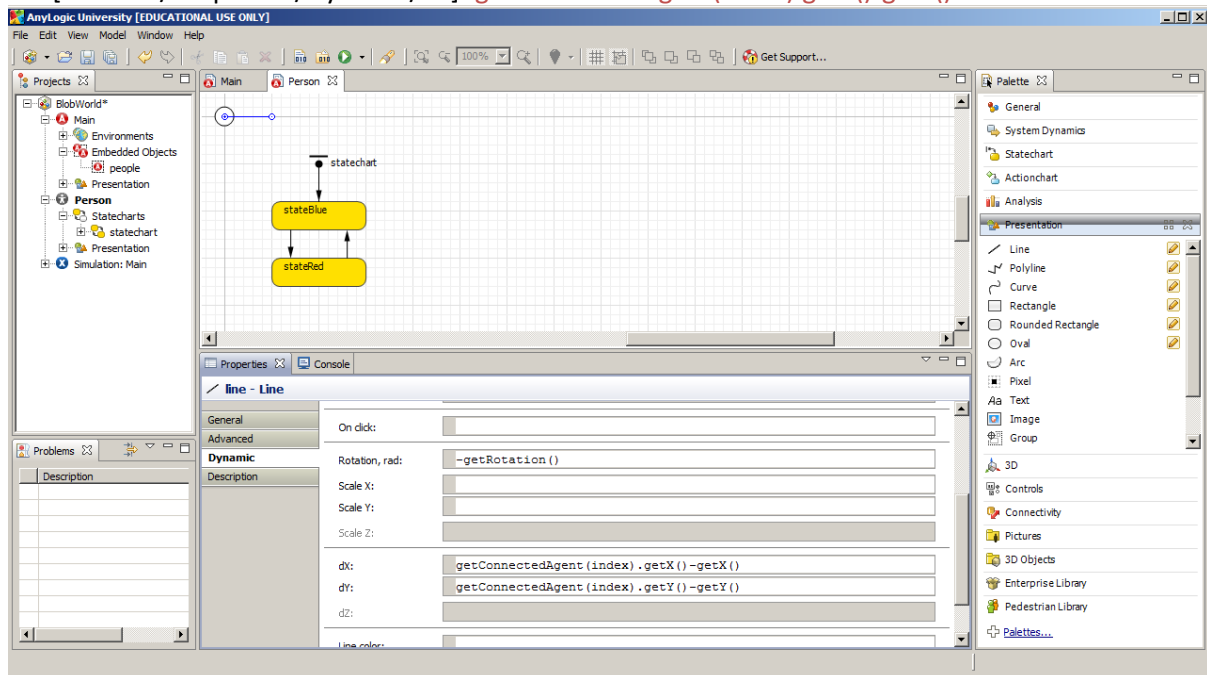
Make the agent move to random locations

- [Active Object Class Person/Properties/General/Startup code]:
`moveTo(uniform(500),uniform(500));`
- [Active Object Class Person/Properties/Agent/OnArrival]:
`moveTo(uniform(500),uniform(500));`



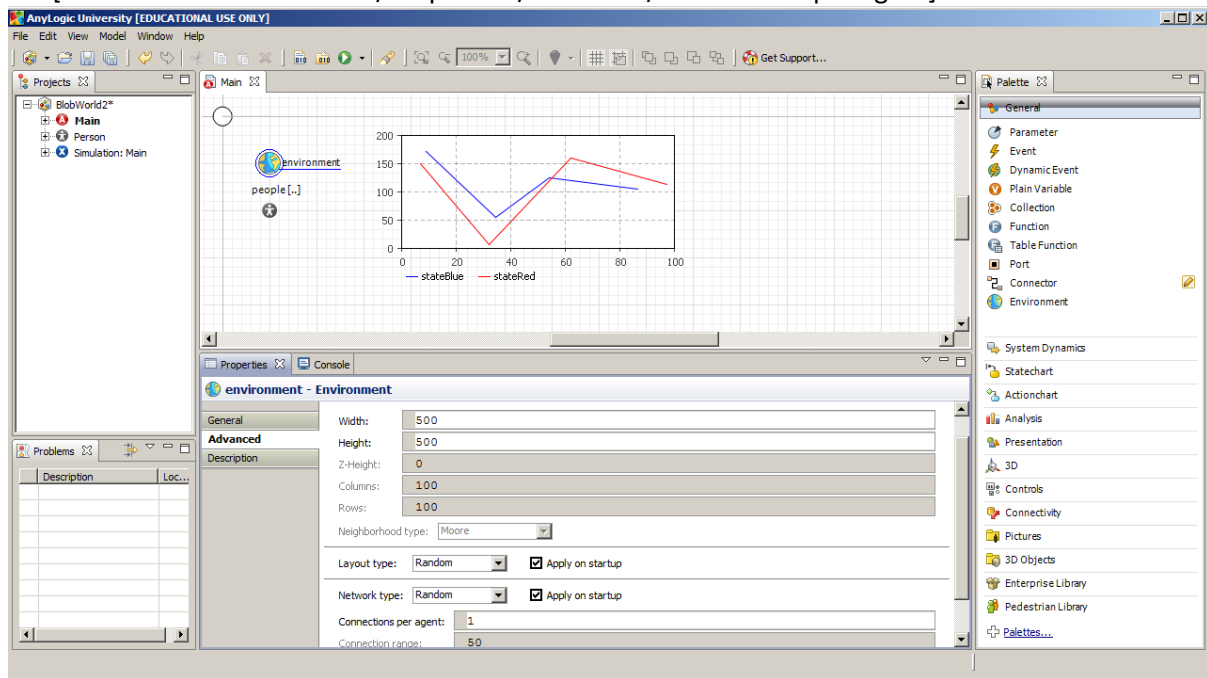
Show links between agents

- From the **Palette window (Presentation)** drag a **Line** into the **Person** window
- [Line line/Properties/General/Line color]: **silver**
- [Line line/Properties/Advanced/Position X]: **0**
- [Line line/Properties/Advanced/Position Y]: **0**
- [Line line/Properties/Advanced/Position dX]: **50**
- [Line line/Properties/Advanced/Position dY]: **0**
- [Line line/Properties/Dynamic/Replication]: **getConnectionsNumber()**
- [Line line/Properties/Dynamic/Rotation]: **-getRotation()** (note that this is a negative value!)
- [Line line/Properties/Dynamic/dX]: **getConnectedAgent(index).getX()-getX()**
- [Line line/Properties/Dynamic/dY]: **getConnectedAgent(index).getY()-getY()**

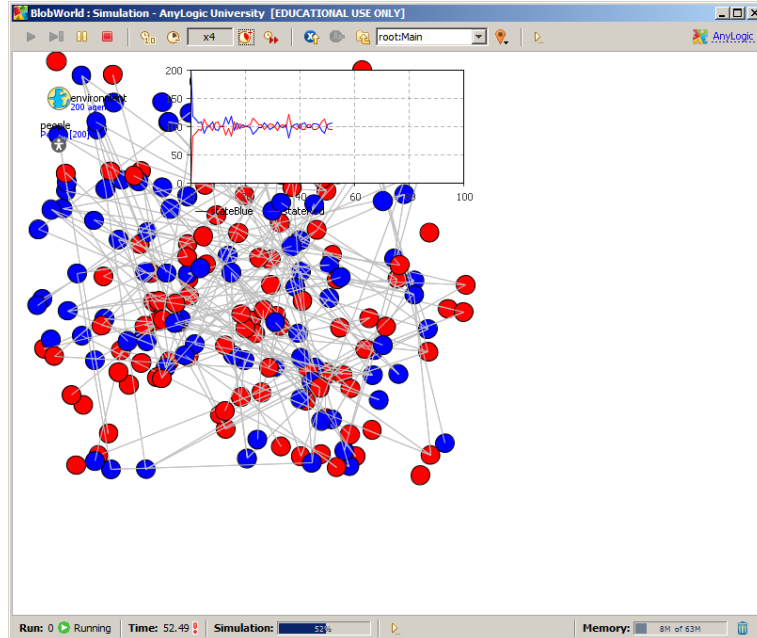


Show links between agents (cont.)

- In the **Main window** click on **environment**
- [Environment environment/Properties/Advanced/Layout type]: **Random**
- [Environment environment/Properties /Advanced/Network type]: **Random**
- [Environment environment/Properties /Advanced/Connections per agent]: **1**



Save and test your simulation (BlobWorld2.alp)

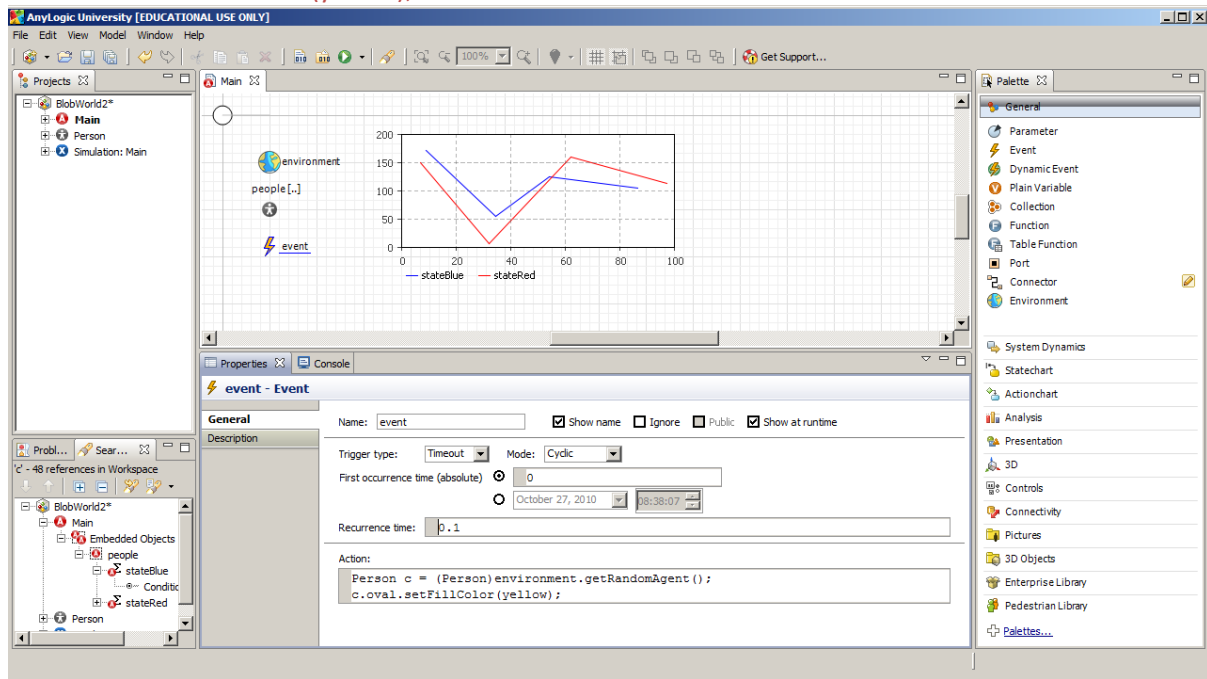


Try out different Layout and Network types (e.g. **Ring** and **Ring lattice**)!

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

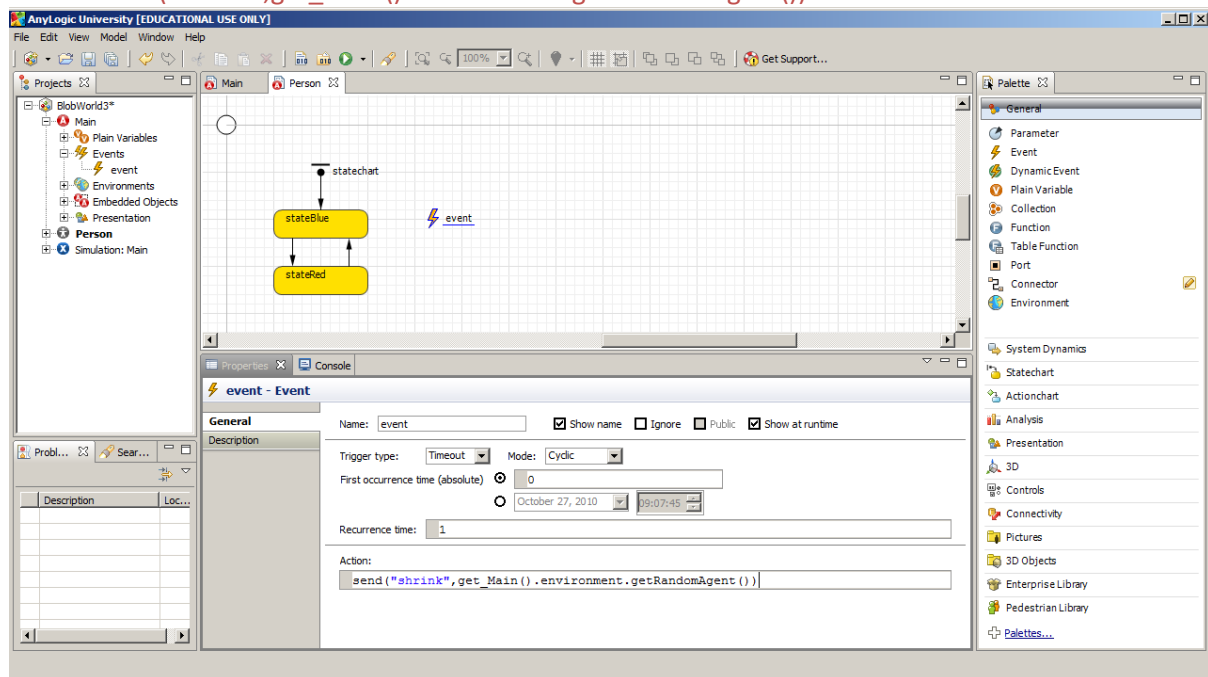
- From the **Palette window (General)** drag an **Event** into the **Main window**
- [Event event/Properties /General/Mode]: **Cyclic**
- [Event event/Properties /General/Recurrence time]: **0.1**
- [Event event/Properties /General/Action]:

```
Person c = (Person)environment.getRandomAgent();  
c.oval.setFillColor(yellow);
```



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

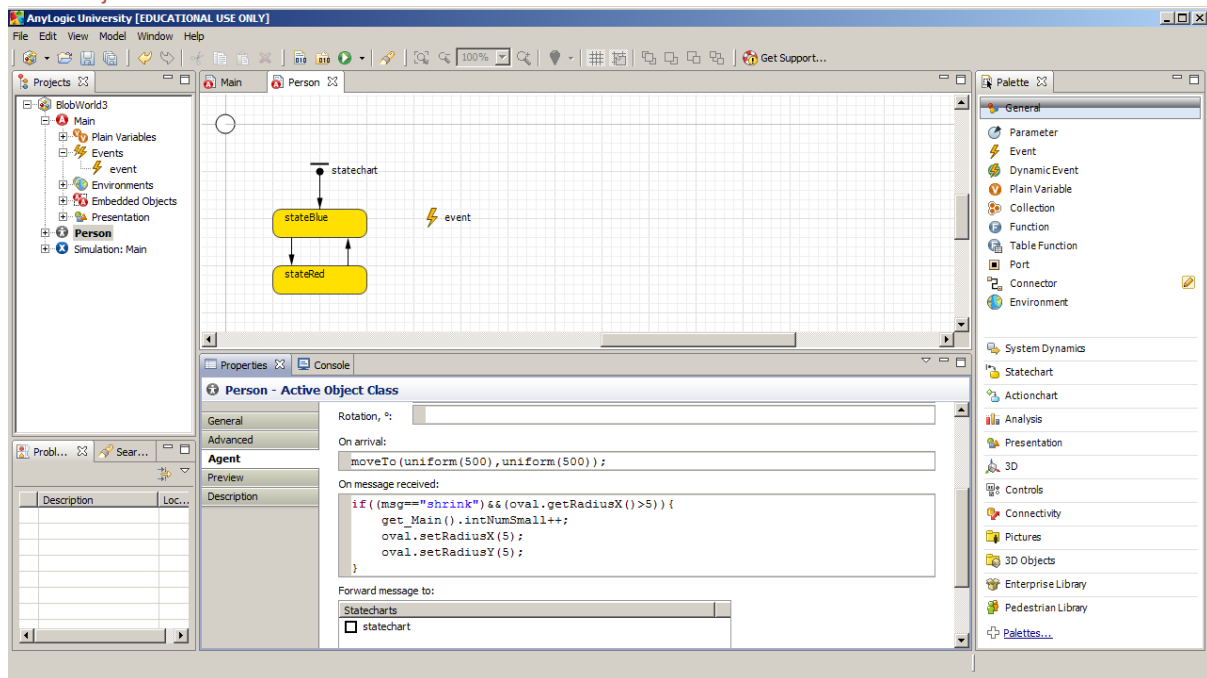
- From the **Palette window (General)** drag a **Plain Variable** into the **Main window**
- [Plain Variable plainVariable/Properties /General/Name]: intNumSmall
- [Plain Variable plainVariable/Properties /General/Type]: **int**
- [Plain Variable plainVariable/Properties /General/Initial value]: **0**
- From the **Palette window (General)** drag an **Event** into the **Person window**
- [Event event/Properties /General/Mode]: **Cyclic**
- [Event event/Properties /General/Action]:
`send("shrink",get_Main().environment.getRandomAgent())`



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

- Click on the **Person** class
- [Active Object Class Person/Properties/Agent/On message received]:

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



Save and test your simulation (BlobWorld3.alp)

