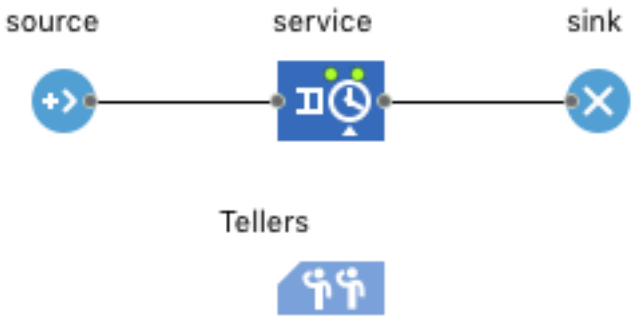
A2. Modeling with Anylogic

We will use different small scenarios to see how various modules in Anylogic is useful. All the modules can be found in Process modeling Palette, unless otherwise mentioned.

Consider a bank, where there are 4 teller counters each operated by a clerk working from 8am to 5pm. Customers arrive with an exponentially distributed inter-arrival time of 5 minutes. The service times are triangular (10, 15, 30) minutes. Simulate system for 1 working day.

- Create a new blank model in Anylogic, with the name as *Model2*
- Click on Process Modeling Library. Click, drag, place and connect SOURCE, SERVICE and SINK *in* the workspace.
  - SERVICE block is a combination of QUEUE and DELAY.
- In the Properties view of SOURCE, define the arrivals as per above specs
- From Process Modeling Library, Drag and drop RESOURCE POOL below the Service.
- In Properties view of RESOURCE POOL, do:
  - Give 'Name' as *Tellers*
  - Set 'Resource Type' as *Static*
  - Set 'Capacity' to 4
    - → indicates there are 4 clerks for use
  - Under Advanced: Check 'Force Statistics Collection'



- In Properties view of SERVICE, do:
  - Give appropriate 'Delay time'
  - Select 'Seize' option as *units of the same pool*
  - In 'Resource pool' select, *Tellers*
  - Set 'Number of units' to *1* → Though there are 4 agents, each customer goes to only one ticket counter
  - Set 'Queue Capacity' to *100*
  - Under Advanced: Check 'Force Statistics Collection'
- In the Projects view (left side panel), click on 'Simulation:Main'. Now, in the Properties view, click 'Model Time'. In that:

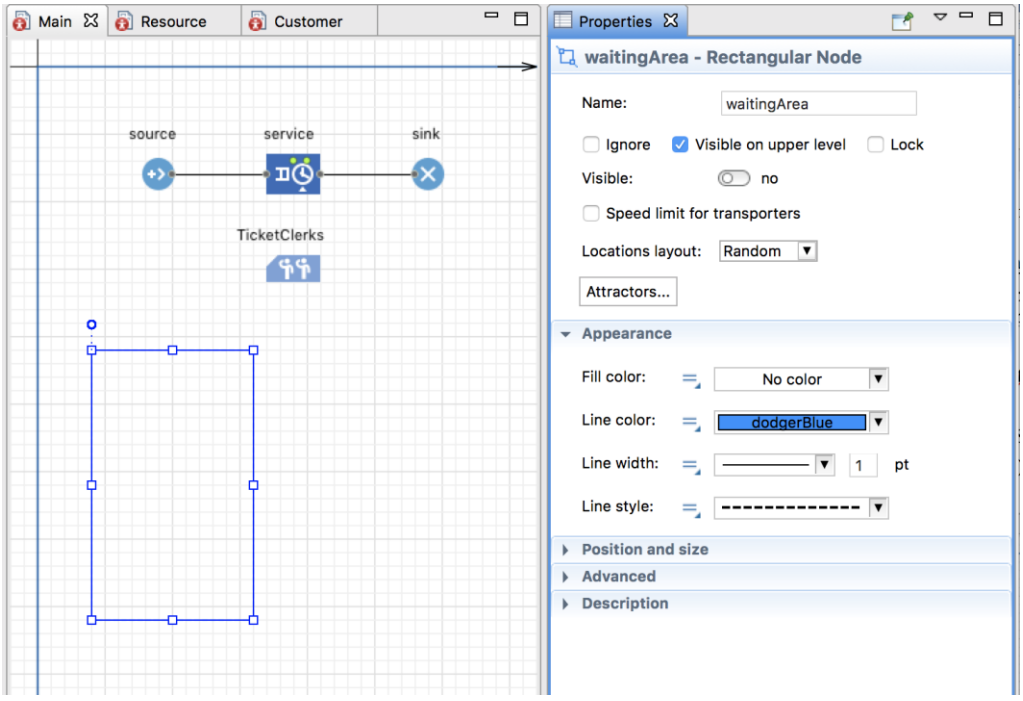
- Select *Real time with scale* as *10*
- Choose 'Stop' at *Stop at specified date*
- You can define the working day as 8am to 5pm.
- Under Randomness: Select 'Random Seed'

- **Save, Compile and Run the model. Observe results.**

ANIMATION (this is not essential for analysis, but useful for model verification and communication)

Setup a *WaitingArea* where customers can wait when they arrive.

- From Space Markup pallet, drag, place and resize, '**Rectangular node**' in the graphical editor window, as shown in figure below
  - In properties of the node,
    - 'Name' it as *waitingArea*
    - Set 'Visible' to *No*
- Click SERVICE block, and go to SERVICE properties window
  - Set 'Agent Location (queue)' to *waitingArea*
- Note: Arriving customer will randomly wait in this are we have defined.

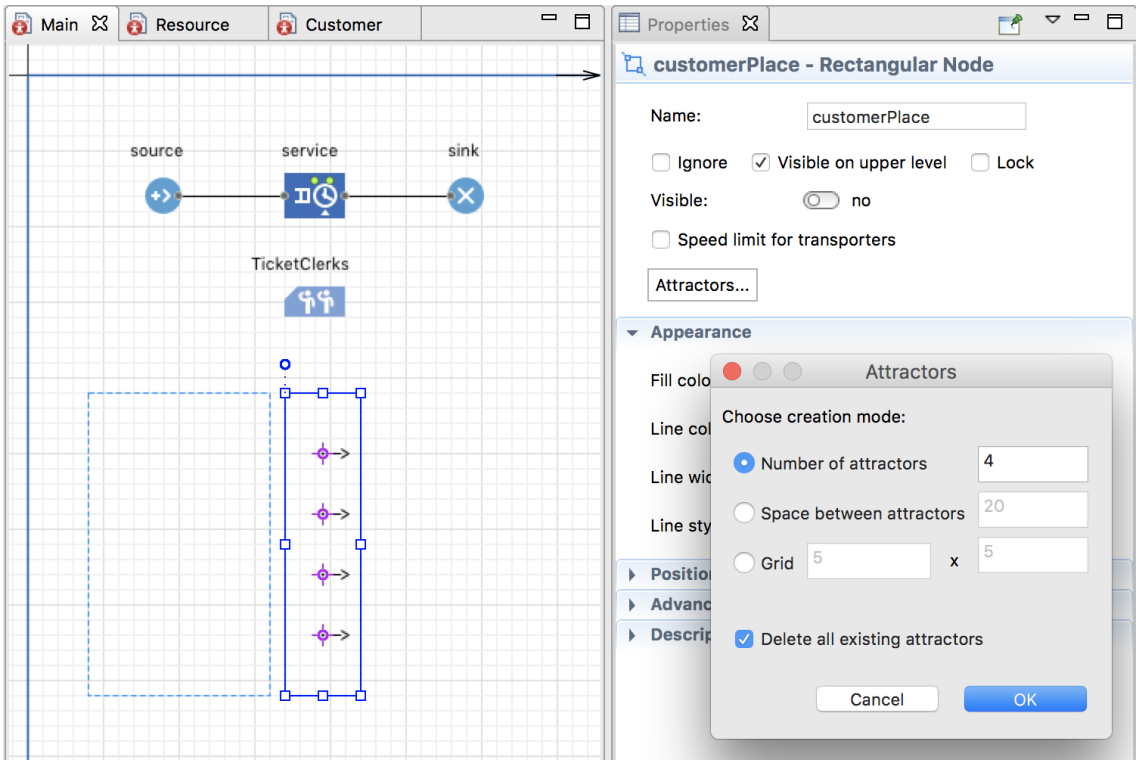


Setup a *CustomerPlace* where customers stand while getting service from tellers

- From Space Markup pallet, drag, place and resize, another '**Rectangular node**' in the editor window, as shown in figure below
- In properties of the node,

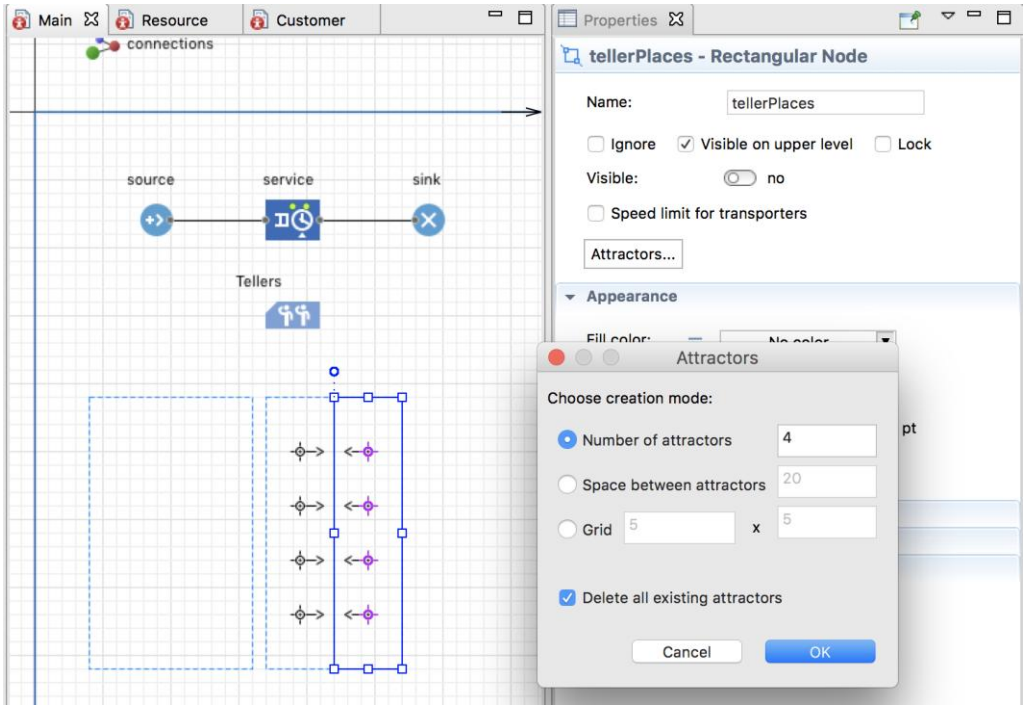
- ‘Name’ it as *customerPlace*
- Set ‘Visible’ to *No*
- Click ‘Attractors’ and set ‘Number of attractors’ to 4. Note: Attractors define the place where customers stand when they talk to the tellers. The arrow direction shows the way the customer faces when talking to teller. And the target circle indicates the place where customer will stand.

- Click SERVICE block, and go to SERVICE properties window
  - Set ‘Agent Location (delay)’ to *customerPlace*



<u>Setup a *TellerPlace* where tellers stand while giving service to customer</u>

- From Space Markup pallet, drag, place and resize, another ‘**Rectangular node**’ in the editor window, as shown in figure below
- In properties of the node,
  - ‘Name’ it as *tellerPlaces*
  - Set ‘Visible’ to *No*
  - Click ‘Attractors’ and set ‘Number of attractors’ to 4. Note: Attractors define the place where tellers stand when they talk to the customers.
- You will see that the arrows of your *tellerPlaces* face the wrong direction.
  - Click each of the circle in ‘attractor’, and set its ‘Orientation’ to +180.
- Click RESOURCE POOL, Teller block, and go to properties window
  - Set ‘Home Location (nodes)’ to *tellerPlaces* by clicking on the green plus



ADD 3D Objects

We will add three 3-D objects, one for customers, one for tellers, and a table to indicate the physical counter.

Create a 3D customer

From Process Modelling Library, drag and drop 'AGENT TYPE'

- In the pop-up window that comes,
    - give name as *Customer*, click option 'Create Agent type from scratch'. Click **Next**.
    - Choose animation 3D, and select 'Person'. Click **Finish**
- The new Customer diagram will open. You can find the *Person* 3D figure in the axis origin. Switch back to the Main diagram.
- On the Main diagram, select the block *source* in the graphical editor.
    - In SOURCE Properties, Choose *Customer* in **New agent** drop-down list.

Create a 3D Teller

From Process Modelling Library, drag and drop 'RESOURCE TYPE'

- In the pop-up window that comes,
    - give name as *Teller*, click option 'Create Agent type from scratch'. Click **Next**.
    - Choose animation 3D, and select 'Office Worker'. Click **Finish**

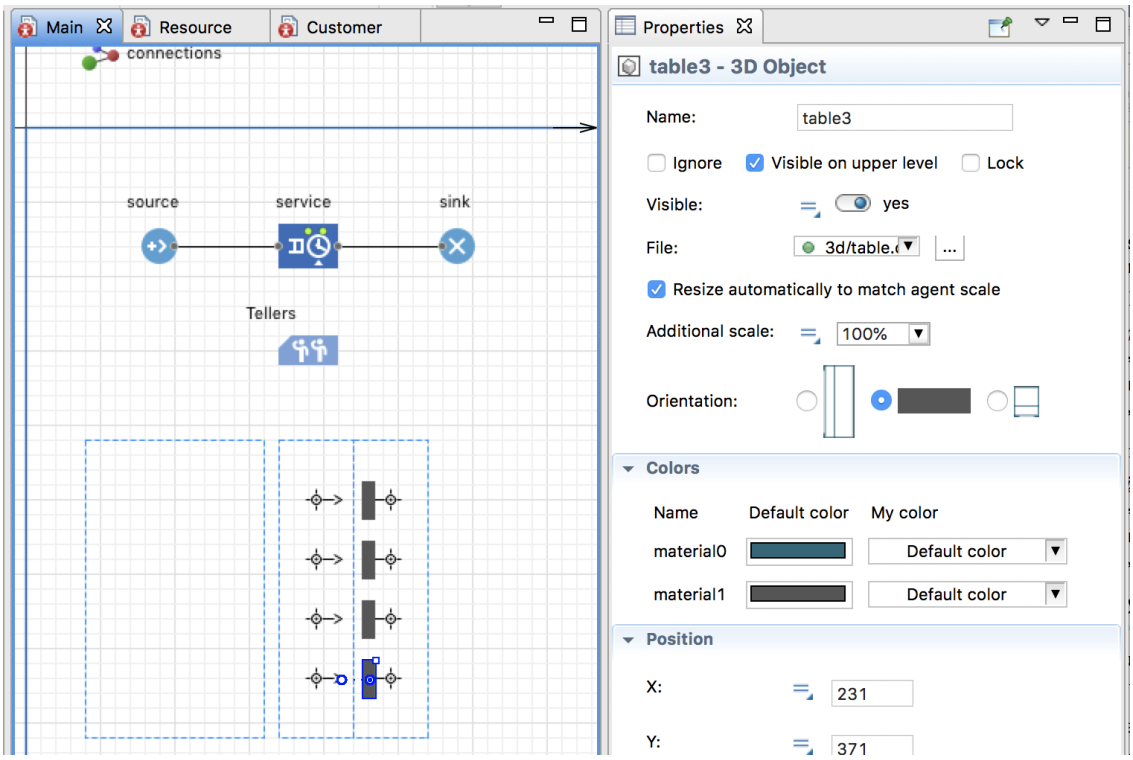
- The new Teller diagram will open. You can find the *Person* 3D figure in the axis origin. Switch back to the Main diagram.
- On the Main diagram, select the resource pool block *TELLERS* in the graphical editor.
    - In TELLERS Properties, Choose *Teller* in the **New resource unit** drop-down list.

Create a 3D Table

From 3D Objects Palette, Office Section, identify the TABLE object.

- Drag and drop the TABLE on arrow head of the resource
- In the pop-up window that asks about to resize the object, click YES.
- Rotate the table by 90 degrees so that it looks like the figure below.

Repeat above steps to create 3 more tables, one for each of the other counters.



A 3D window where we can view the 3D animation

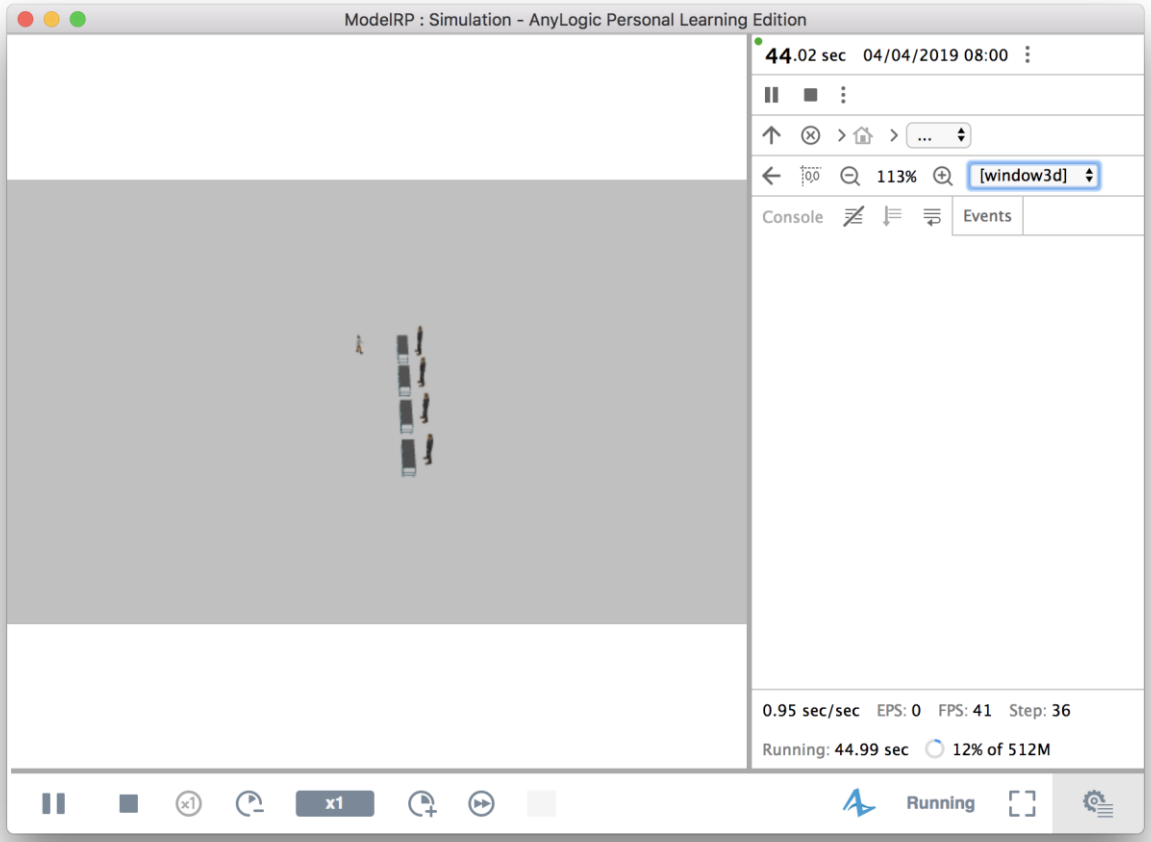
From Presentation Palette, drag and drop a **3D Window Object** in the editor window below the 2D rectangles and below the thick horizontal line. This is the place where 3D animation will be seen during run time.

- **Save, Compile and Run the model.**

In the **run window**, you will observe some animation dynamics in 2D in the area where we drew rectangles. These are nothing but **top view** the bank!

TO view the 3D more clearly, click on the Gear icon on the bottom right corner to open the Developer Pane on the right side.

In that click and select the area as **[window3d]** from that drop down box.



TO ZOOM and PANE:

- To Zoom in and out: Use the Magnifying glass in the Developer Pane to zoom; or use mouse wheel.
- To Rotate the Scene:
  - Press **Alt** key (Mac OS: Option key) and hold it pressed.
  - Click in the 3D scene window and, while holding Alt and the left mouse button down, Move the mouse in the required rotation direction
- TO move scene:
  - Press the left mouse button in the 3D view and hold the mouse button pressed.
  - Move the mouse in the required direction.

Include elements in the model to measure/ display the following (some may be collected & shown by default; we can use that too):

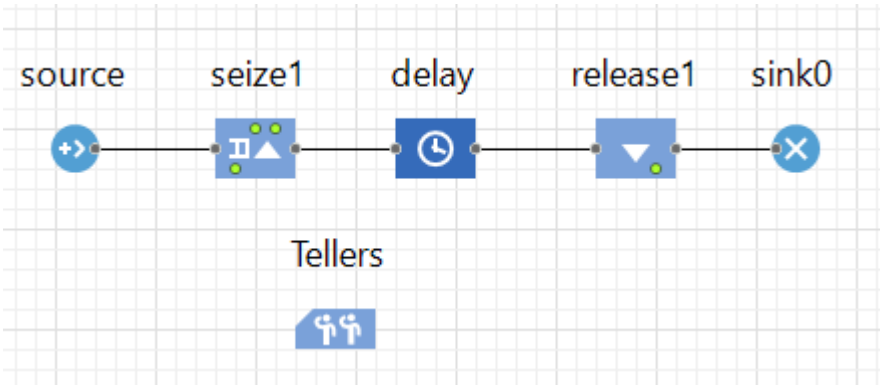
- Fraction of customers who left the system.
- Average time spent by customers in the system.
- Average number of customers in waiting area.
- Operator utilization

Alternative Approach

SEIZE-DELAY-RELEASE and ResourcePool (Create a New Model)

Consider a bank, where there are 4 teller counters each operated by a clerk working from 8am to 5pm. Customers arrive with an exponentially distributed inter-arrival time of 5 minutes. The service times are triangular (10, 15, 30) minutes. Simulate system for 1 working day.

- Create a new blank model in Anylogic, with the name as *Model2*. Time units as *Minutes*
- Click on Process Modeling Library. Click, drag, place and connect SOURCE, SEIZE, DELAY, RELEASE and SINK and a ResourcePool *Modules* in the workspace as shown.



- In the Properties view of SOURCE, define the arrivals as per above scenario.
- In Properties view of RESOURCE POOL, do:
  - Give 'Name' as *Tellers*
  - Set 'Resource Type' as *Static*
  - Set 'Capacity' to 4
    - → indicates there are 4 clerks for use
  - Under Advanced: Check 'Force Statistics Collection'
- In Properties view of SEIZE, do:

  - Select 'Seize' option as *units of the same pool*
  - In 'Resource pool' select, *Tellers*
  - Set 'Number of units' to *1* → Though there are 4 agents, each customer goes to only one ticket counter
  - Set 'Queue Capacity' to *100*
  - Under Advanced: Check 'Force Statistics Collection'
- In Properties view of DELAY, do:
  - Give appropriate 'Delay time'
  - Tick the 'Max Capacity' (Since we already seized the resource)
- In Properties view of RELEASE, do:
  - Set 'Release' to *All Resources Seized at given Seize block(s)*
  - In 'Seize blocks' use the + button to select the seize block.
- In the Projects view (left side panel), click on 'Simulation:Main'. Now, in the Properties view, click 'Model Time'. In that:
  - Select *Real time with scale* as *10*
  - Choose 'Stop' at *Stop at specified date*
  - You can define the working day as 8am to 5pm.
  - Under Randomness: Select 'Random Seed'
- **Save, Compile and Run the model. Observe results.**

Notes:

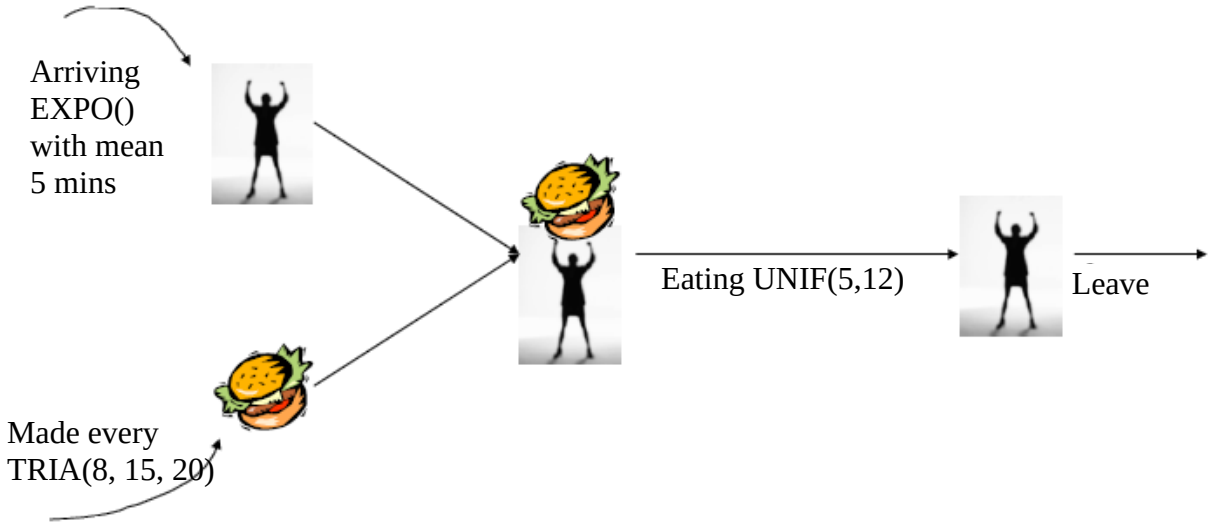
- SERVICE block is equivalent to a sequence Seize, Delay, Release and can be used if the agent does not need to do anything but execute a delay between seize and release → We can try it in the above model, by replacing SEIZE-DELAY-RELEASE with a SERVICE block (ensure you fill the info of the SEIZE and DELAY into the SERVICE block)

MATCH & BATCH

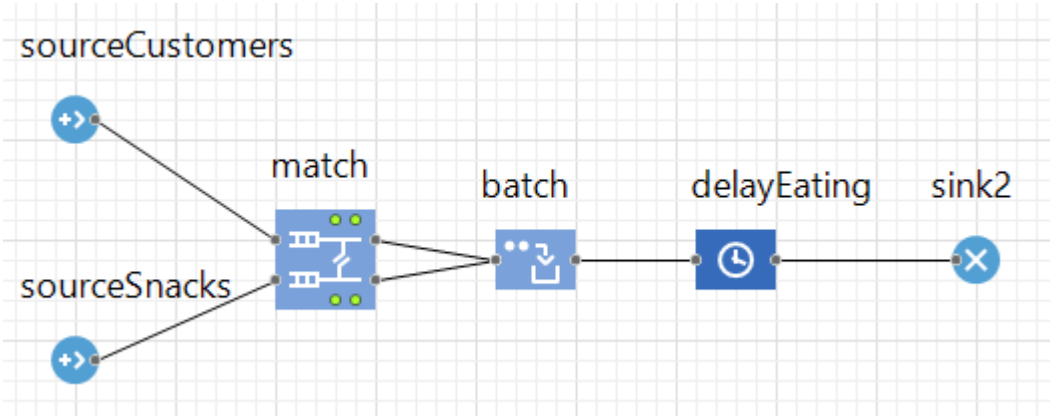
CREATE a new model

Create a new model in Anylogic. Keep default model time units in **Minutes.** All the exercises shown can be built in the model itself. Keep run length as 1000 minutes.

Customers arrive at a fast food joint, picks up a snack, eats it there (with eating time is Uniformly distributed), and leaves. If snack is not readily available, the customer waits. The snacks are being continuously made and delivered in batches of 4, irrespective of whether customer is there or not. Assume only one type of snack, and each customer picks up only one snack. How 'fresh' is the snack when a customer eats it?



We can model the above scneario in Anylogic as follows. Drag and drop the corresponding modules from Process Modeling Library.



- *sourceCustomers* and *sourceSnacks* blocks: Enter appropriate arrival pattern as given. Note "Snacks delivered in batches of 4" is modeled, using *Multiple agents per Arrival*.

- *Match* module: This synchronizes two streams of agents by matching pairs according to a given criteria. Agents waiting separate queues until a match is found. If the match is found, both agents exit the Match object at the same time.
    - You can view the properties of Match modules. Leave the values at the default values.
- *Batch* module: Converts a number of agents into one agent (batch).
    - In our case the customers and snacks are to be batched. The Match module ensure that one customer is matched with one snack. We need to batch these two.
    - In batch Properties, set *BatchSize = 2*.
- *Delay* module
    - Specify *DelayTime* as given in scenario.
    - Make capacity as *Max Capacity* (Think why this is correct, and not capacity of 1)

How 'fresh' is the snack when a customer eats it?

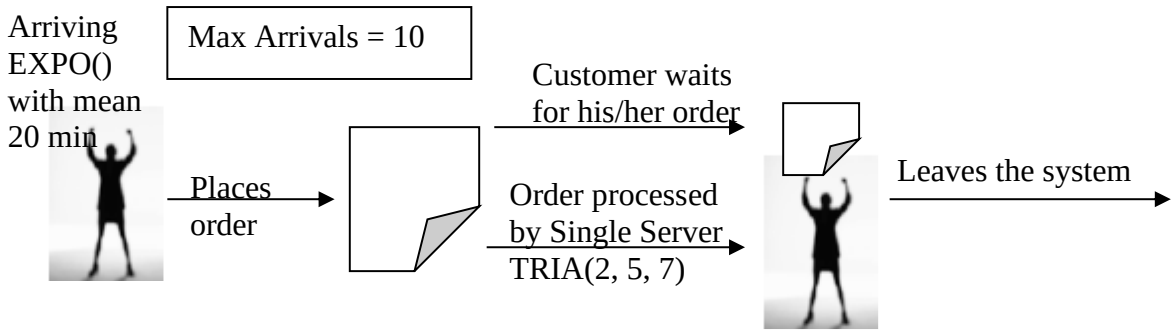
We can answer this question by seeing how long a snack has been 'waiting in queue' for a customer. We assume that the customer picks the snacks in FIFO order, and not the freshest. To compute this, insert a *TimeMeasureStart* between *sourceSnacks* and *match*; and a *TimeMeasureEnd* between *match* and *batch* bottom link.

Run the model.

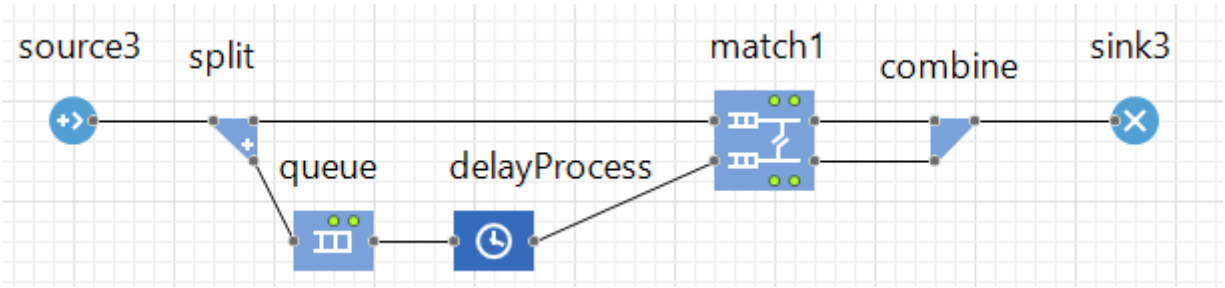
- **Additional points to Ponder:**
    - If we use only BATCH and **no** MATCH, what will happen?
    - If Delay capacity is 1 instead of Max capacity, what will happen?
    - Suppose the snacks are always ordered 'freshest first', how to model this?

SPLIT & COMBINE

Model the following system. Find the average time customers spend waiting for his/her order to arrive.



Anylogic Model is as follows:



- *source3* module*:* Enter appropriate arrival pattern as given
- *split* module*:* For each incoming agent ("original") creates one or several other agents and outputs them via 'outCopy' port. Leave the default values as is.
- *Queue*, and *delayProcess* modules: models a simple server. Keep *delayProcess* capacity to 1.
- *Match1* module: You can leave the default values as is.
- *Combine* module: Waits for the two agents to arrive (in arbitrary order) at ports in1and in2, produces a new agent and outputs it. You can leave the default values as is.

Average time customers spend waiting for his/her order to arrive
To model is, insert a *TimeMeasureStart* between *split1* and *match* (direct link); and a *TimeMeasureEnd* between *match1* and *combine* bottom link

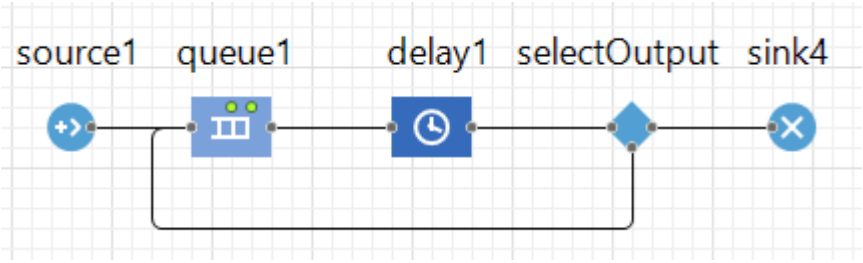
Run the model.

Point to ponder: Instead of *Combine*, can we use *batch* module?

MODELING DECISION LOGIC

Scenario: In a particular workshop, parts arrive with an inter-arrival time exponentially distributed with mean 10 minutes. Parts are processed on single capacity machine in UNIF(6, 12) minutes. After processing, the parts are inspected (ignore the inspection time as it is very small). Upon inspection, 90% of the parts are usually found to be good, and hence leave the shop. The remaining 10% of the parts are sent back to the machine for rework, where they are treated like a new part only. Simulate model for 100 minutes.

We will use the *SelectOutput* module to model the above scenario, as follows:



selectOutput module: Write the probability value as 0.9.

Implement the above in Anylogic, and run your simulation. Observe behavior