

Model: BeerDistributionGame (Solicitado)

Name	Value
General	
Java Package Name	beerdistributiongame
File Name	C:\Users\Alejandro Uribe\Documents\PROFESIONAL\PROYECTO DE GRADO\PROYECTO DE GRADO\Modelo en Anvlogia\BeerDistributionGame (Solicitado).aln
Model Time	
Model Time Units	Day

Active Object Class: Main

Name	Value
General	
Destroy Code	players.clear();
Advanced	
Auto-create Datasets	true
Recurrence	1
Dataset Samples To Keep	100



Parameter: sRetailer

Name	Value
General	
Type	int
Default Value	20
Editor	
Editor Control	TEXT_BOX

Parameter: iRetailer

Name	Value
General	
Type	int
Default Value	40
Editor	
Editor Control	TEXT_BOX

Parameter: sWholesaler

Name	Value
General	
Type	int
Default Value	20
Editor	
Editor Control	TEXT_BOX

Parameter: iWholesaler

Name	Value
General	
Type	int
Default Value	40
Editor	
Editor Control	TEXT_BOX

Parameter: sFactory

Name	Value
General	
Type	int
Default Value	20
Editor	
Editor Control	TEXT_BOX

Parameter: iFactory

Name	Value
General	
Type	int
Default Value	40
Editor	
Editor Control	TEXT_BOX

Parameter: sDistributor

Name	Value
General	
Type	int
Default Value	20
Editor	
Editor Control	TEXT_BOX

Parameter: iDistributor

Name	Value
General	
Type	int
Default Value	40
Editor	

Name	Value
Editor Control	TEXT_BOX

Parameter: players

Name	Value
General	
Type	ArrayList<String>
Default Value	new ArrayList<String>()
Editor	
Editor Control	TEXT_BOX

Parameter: timeFacOrder

Description: Задержка в доставке между Market & Retailer

Name	Value
General	
Type	double
Default Value	2*day()
Editor	
Editor Control	TEXT_BOX

Parameter: timeWhoOrder

Description: Задержка в доставке между Market & Retailer

Name	Value
General	
Type	double
Default Value	2*day()
Editor	
Editor Control	TEXT_BOX

Parameter: timeDisOrder

Description: Задержка в доставке между Market & Retailer

Name	Value
General	
Type	double
Default Value	2*day()
Editor	
Editor Control	TEXT_BOX

Parameter: timeWhoSend

Description: Задержка в доставке между Market & Retailer

Name	Value
General	
Type	double
Default Value	2*day()

Name	Value
Editor	
Editor Control	TEXT_BOX

Parameter: timeDisSend

Description: Задержка в доставке между Market & Retailer

Name	Value
General	
Type	double
Default Value	2*day()
Editor	
Editor Control	TEXT_BOX

Parameter: timeFacSend

Description: Задержка в доставке между Market & Retailer

Name	Value
General	
Type	double
Default Value	2*day()
Editor	
Editor Control	TEXT_BOX

Parameter: ordersMemoryDepth

Name	Value
General	
Type	int
Default Value	5
Editor	
Editor Control	TEXT_BOX

Parameter: storageCost

Name	Value
General	
Type	double
Default Value	0.5
Editor	
Editor Control	TEXT_BOX

Parameter: backlogCost

Name	Value
General	
Type	double
Default Value	1
Editor	
Editor Control	TEXT_BOX

Parameter: SRetailer

Name	Value
General	
Type	int
Default Value	20
Editor	
Editor Control	TEXT_BOX

Parameter: SWholesaler

Name	Value
General	
Type	int
Default Value	20

Editor	
Editor Control	TEXT_BOX

Parameter: SFactory

Name	Value
General	
Type	int
Default Value	20
Editor	
Editor Control	TEXT_BOX

Parameter: SDistributor

Name	Value
General	
Type	int
Default Value	20
Editor	
Editor Control	TEXT_BOX

Parameter: facOrdersPerDay

Name	Value
General	
Show At Runtime	false
Type	double
Default Value	2*day()
Editor	
Editor Control	TEXT_BOX

Function: initNewDay

Name	Value
General	

Name	Value
Return Type	void
Code	

Body	<pre> retCurSend = retailer.curSend; retCurReceive = retailer.curReceive; retailer.curSend =0; retailer.curReceive =0; //retailer.ordered = 0; whoCurSend = wholesaler.curSend; whoCurReceive = wholesaler.curReceive; wholesaler.curSend =0; wholesaler.curReceive =0; //wholesaler.ordered = 0; disCurSend = distributor.curSend; disCurReceive = distributor.curReceive; distributor.curSend =0; distributor.curReceive =0; //distributor.ordered = 0; facCurSend = factory.curSend; facCurReceive = factory.curReceive; factory.curSend =0; </pre>
------	---

Function: nextStepInitialization

Name	Value
General	
Show At Runtime	false
Return Type	void
Code	
Body	<pre> distributor.curReceive=0; distributor.curSend=0; retailer.ordering(); wholesaler.ordering(); distributor.ordering(); factory.ordering(); retailer.ordered = 0; wholesaler.ordered = 0; distributor.ordered = 0; factory.ordered = 0; </pre>

Table Function: placingOrder

Name	Value
General	
Public	false
Interpolation	SPLINE
Out Of Range Behaviour	ERROR

Table Data:

Argument	Value
0.0	10.0
10.0	20.0
15.0	18.0
20.0	13.0

Table Data:

Argument	Value
40.0	4.0
50.0	10.0
60.0	8.0
70.0	12.0
80.0	4.0
90.0	6.0
100.0	8.0

Event: orderGenerator

Name	Value
General	
Show At Runtime	false
Trigger Type	timeout
Mode	cyclic
Recurrence	day()
Occurence Time	day()*0.99
Action	//place a new order to the retailer orders //size is drawn from a table of orders distribution int orderingProduct = (int)placingOrder.get(numDay);

Event: newDay

Name	Value
General	
Trigger Type	timeout
Mode	cyclic
Recurrence	day()
Occurence Time	day()

Action	<pre> numDay++; retailer.newDayUpdate(); wholesaler.newDayUpdate(); distributor.newDayUpdate(); factory.newDayUpdate(); initNewDay(); if (players.size() == 0){ nextStepInitialization(); } else { //pause for manual mode pauseSimulation(); } storageCostTotal = retailer.storageCostTotal + wholesaler.storageCostTotal + distributor.storageCostTotal + factory.storageCostTotal; backlogCostTotal = retailer.backlogCostTotal + wholesaler.backlogCostTotal + distributor.backlogCostTotal + factory.backlogCostTotal; </pre>
--------	--

Variable: retCurReceive

Name	Value
General	
Show At Runtime	false
Type	int

Variable: retCurSend

Name	Value
General	
Show At Runtime	false
Type	int

Variable: whoCurReceive

Name	Value
General	
Show At Runtime	false
Type	int

Variable: whoCurSend

Name	Value
General	
Show At Runtime	false
Type	int

Variable: disCurReceive

Name	Value
General	
Show At Runtime	false
Type	int

Variable: disCurSend

Name	Value
General	
Show At Runtime	false
Type	int

Variable: facCurReceive

Name	Value
General	
Show At Runtime	false
Type	int

Variable: facCurSend

Name	Value
General	
Show At Runtime	false
Type	int

Variable: storageCostTotal

Name	Value
General	
Type	double

Variable: backlogCostTotal

Name	Value
General	
Type	double

Variable: numDay

Name	Value
General	
Type	int
Initial Value	0

Variable: RETAILER

Name	Value
General	
Static	true
Constant	true
Type	int
Initial Value	2

Variable: WHOLESALE

Name	Value
General	
Static	true
Constant	true
Type	int
Initial Value	3

Variable: DISTRIBUTOR

Name	Value
General	
Static	true
Constant	true
Type	int
Initial Value	4

Variable: FACTORY

Name	Value
General	

Static	true
Constant	true
Type	int
Initial Value	5

Variable: PLAYFORALL

Name	Value
General	
Static	true
Constant	true
Type	int
Initial Value	1

Variable: COMPUTERFORALL

Name	Value
General	
Static	true
Constant	true
Type	int
Initial Value	0

Port: marketPort

Player: distributor

Name	Value
General	
Type	Player
Java Package Name	beerdistributiongame
Embedded Object Collection Type	ARRAY_LIST_BASED

Embedded Object Parameters:

Name	Value
initInventory	iDistributor
minStock	sDistributor
backlogCost	backlogCost
inventoryCost	storageCost
players	players
timeSend	timeDisSend
timeService	timeDisOrder
memDepth	ordersMemoryDepth
maxStock	SDistributor
playerIndex	DISTRIBUTOR
facOrdersPerDay	5

Player: retailer

Name	Value
General	
Type	Player
Java Package Name	beerdistributiongame
Embedded Object Collection Type	ARRAY_LIST_BASED

Embedded Object Parameters:

Name	Value
initInventory	iRetailer
minStock	sRetailer
backlogCost	backlogCost
inventoryCost	storageCost
players	players
timeSend	0.01
timeService	0.01
memDepth	ordersMemoryDepth
maxStock	SRetailer
playerIndex	RETAILER
facOrdersPerDay	5

Player: wholesaler

Name	Value
General	
Type	Player
Java Package Name	beerdistributiongame
Embedded Object Collection Type	ARRAY_LIST_BASED

Embedded Object Parameters:

Name	Value
initInventory	iWholesaler
minStock	sWholesaler
backlogCost	backlogCost
inventoryCost	storageCost
players	players
timeSend	timeWhoSend
timeService	timeWhoOrder
memDepth	ordersMemoryDepth
maxStock	SWholesaler
playerIndex	WHOLESALE
facOrdersPerDay	5

Player: factory

Name	Value
General	
Type	Player
Java Package Name	beerdistributiongame

Name	Value
Embedded Object Collection Type	ARRAY_LIST_BASED

Embedded Object Parameters:

Name	Value
initInventory	iFactory
minStock	sFactory
backlogCost	backlogCost
inventoryCost	storageCost
players	players
timeSend	timeFacSend
timeService	timeFacOrder
memDepth	ordersMemoryDepth
maxStock	SFactory
playerIndex	5
facOrdersPerDay	facOrdersPerDay

Stack Chart: chart2

Name	Value
General	
Scale Type	HUNDRED_PERCENTS
Analysis Auto Update	true
Recurrence	1
Advanced	
x	2410
y	-350
Width	230
Height	200
Appearance	
Chart Area: Background Color	whiteSmoke
Show Legend	true
Legend Place	SOUTH
Bars Direction	UP
Bars Relative Width	1.0

Chart Items:

Title	Color	Value
-------	-------	-------

Storage cost	lime	retailer.storageCostTotal
Backlog cost	green	retailer.backlogCostTotal

Stack Chart: chart3

Name	Value
General	
Scale Type	HUNDRED_PERCENTS
Analysis Auto Update	true
Recurrence	1
Advanced	
x	3070

Name	Value
y	-350
Width	220
Height	200
Appearance	
Chart Area: Background Color	whiteSmoke
Show Legend	true
Legend Place	SOUTH
Bars Direction	UP
Bars Relative Width	1.0

Chart Items:

Title	Color	Value
Storage cost	red	factory.storageCostTotal
Backlog cost	fireBrick	factory.backlogCostTotal

Stack Chart: chart4

Name	Value
General	
Scale Type	HUNDRED_PERCENTS
Analysis Auto Update	true
Recurrence	1
Advanced	
x	2640
y	-350
Width	220
Height	200
Appearance	

Chart Area: Background Color	whiteSmoke
Show Legend	true
Legend Place	SOUTH
Bars Direction	UP
Bars Relative Width	1.0

Chart Items:

Title	Color	Value
Storage cost	orange	wholesaler.storageCostTotal
Backlog cost	goldenRod	wholesaler.backlogCostTotal

Stack Chart: chart5

Name	Value
General	
Scale Type	HUNDRED_PERCENTS
Analysis Auto Update	true
Recurrence	1
Advanced	
x	2860
y	-350

Name	Value
Width	210
Height	200
Appearance	
Chart Area: Background Color	whiteSmoke
Show Legend	true
Legend Place	SOUTH
Bars Direction	UP
Bars Relative Width	1.0

Chart Items:

Title	Color	Value
Storage cost	cyan	distributor.storageCostTotal
Backlog cost	blue	distributor.backlogCostTotal

Time Plot: inventoryStatisticsPlot

Name	Value
General	
Time Window	100

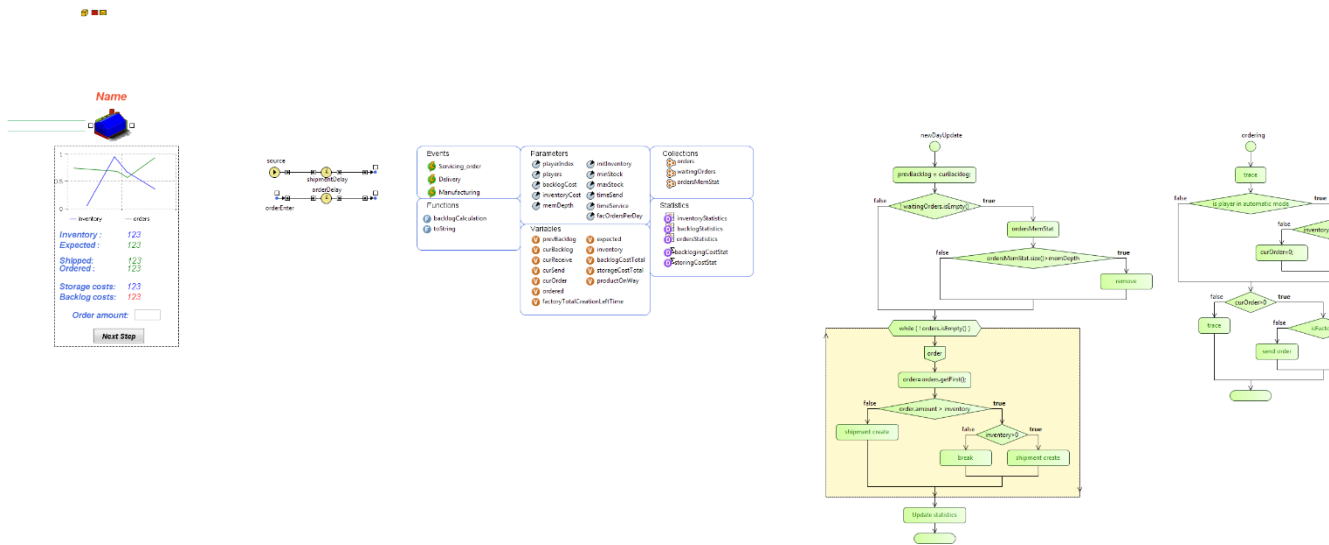
Vertical Scale	AUTO
Analysis Auto Update	true
Recurrence	1
Dataset Samples To Keep	10000
Advanced	
x	2410
y	80
Width	495
Height	220
Appearance	
Show Legend	true
Legend Place	SOUTH
Label Format	MODEL_TIME_UNITS

Plot Items:

Title	Type	Dataset / Value	Point Style	Color	Line	Width	Interpolation
Retailer	dataset	retailer.inventoryStatistics	NONE	lawnGreen	true	1	LINEAR
Wholesaler	dataset	wholesaler.inventoryStatistics	NONE	darkOrange	true	1	LINEAR
Distributor	dataset	distributor.inventoryStatistics	NONE	royalBlue	true	1	LINEAR
Factory	dataset	factory.inventoryStatistics	NONE	orangeRed	true	1	LINEAR

Active Object Class: Player

Name	Value
Advanced	
Auto-create Datasets	true
Recurrence	1
Dataset Samples To Keep	100



Parameter: initInventory

Name	Value
General	
Type	int
Default Value	20
Editor	
Editor Control	TEXT_BOX

Parameter: minStock

Name	Value
General	
Type	int
Editor	
Editor Control	TEXT_BOX

Parameter: backlogCost

Name	Value
General	
Type	double
Editor	
Editor Control	TEXT_BOX

Parameter: inventoryCost

Name	Value
General	
Type	double

Editor	
Editor Control	TEXT_BOX

Parameter: players

Name	Value
General	
Type	ArrayList<String>
Default Value	new ArrayList<String>()
Editor	
Editor Control	TEXT_BOX

Parameter: timeSend

Name	Value
General	
Show At Runtime	false
Type	double
Default Value	2*day()
Editor	
Editor Control	TEXT_BOX

Parameter: timeService

Name	Value
General	
Show At Runtime	false
Type	double
Default Value	2*day()
Editor	
Editor Control	TEXT_BOX

Parameter: memDepth

Name	Value
General	
Type	int
Editor	
Editor Control	TEXT_BOX

Parameter: maxStock

Name	Value
General	
Type	int
Editor	
Editor Control	TEXT_BOX

Parameter: playerIndex

Name	Value
------	-------

General	
Type	int
Editor	
Editor Control	TEXT_BOX

Parameter: facOrdersPerDay

Name	Value
General	
Show At Runtime	false
Type	double
Default Value	5
Editor	
Editor Control	TEXT_BOX

Parameter Hide Conditions:

Parameter	Condition	Value
playerIndex	!=	5

Dynamic Event: Delivery

Name	Value
General	
Action	//deliver the shipment destination.receive(shipment); productOnWay -=shipment.amount;

Parameters:

Name	Type
shipment	Shipment
destination	Port

Dynamic Event: Servicing_order

Name	Value
General	
Action	Order order=waitingOrders.getFirst(); orders.addLast(order); waitingOrders.removeFirst(); ordered = order.amount;

Parameters:

Name	Type
msg	Order

Dynamic Event: Manufacturing

Name	Value
General	

Action	<pre>//add manufactured items to the inventory curReceive = amount; inventory += amount; expected -= amount; factoryTotalCreationLeftTime = max(-0.01,</pre>
--------	---

Parameters:

Name	Type
amount	int

Function: backlogCalculation

Name	Value
General	
Return Type	int
Code	
Body	<pre>//calculate the total backlogged order int bd = 0; for(Order d : orders) bd += d.amount; curBacklog=bd; return bd;</pre>

Function: toString

Name	Value
General	
Access Type	public
Return Type	String
Code	
Body	<pre>String s = ""; switch(playerIndex){ case Main.RETAILER: s = "Retailer"; break; case Main.WHOLESALE: s = "Wholesaler"; break; case Main.DISTRIBUTOR: s = "Distributor"; break; case Main.FACTORY: s = "Factory"; break; default: s = "Wrong player"; }</pre>

Variable: inventory

Name	Value
General	
Type	int
Initial Value	initInventory

Variable: expected

Name	Value
General	
Type	int
Initial Value	0

Variable: backlogCostTotal

Name	Value
General	

Name	Value
Type	double

Variable: storageCostTotal

Name	Value
General	
Type	double

Variable: productOnWay

Name	Value
General	
Type	int

Variable: curBacklog

Name	Value
General	
Type	int

Variable: curReceive

Name	Value
General	
Type	int

Variable: curSend

Name	Value
General	
Type	int

Variable: curOrder

Name	Value
General	
Type	int

Variable: prevBacklog

Name	Value
General	
Type	int

Variable: factoryTotalCreationLeftTime

Name	Value
General	
Type	double
Initial Value	-0.01

Variable: ordered

Name	Value
General	
Type	int

Collection: orders

Name	Value
General	
Collection Class	java.util.LinkedList
Element Class	Order

Collection: waitingOrders

Name	Value
General	
Collection Class	java.util.LinkedList
Element Class	Order

Collection: ordersMemStat

Name	Value
General	
Collection Class	java.util.ArrayList
Element Class	Integer

Port: portLeft

Name	Value
General	
Show name	false
Incoming Message Type	Order
Outgoing Message Type	Shipment
On Receive	<pre> if(playerIndex == Main.RETAILER){ orders.addLast(msg); //traceln(); //traceln("Получен заказ от рынка на партию в размере " + msg.amount); ordered = msg.amount; } else{ //add incoming order to the queue waitingOrders.addLast(msg); //Servicing ends before day start create_Servicing_order(timeService-0.01, msg); </pre>

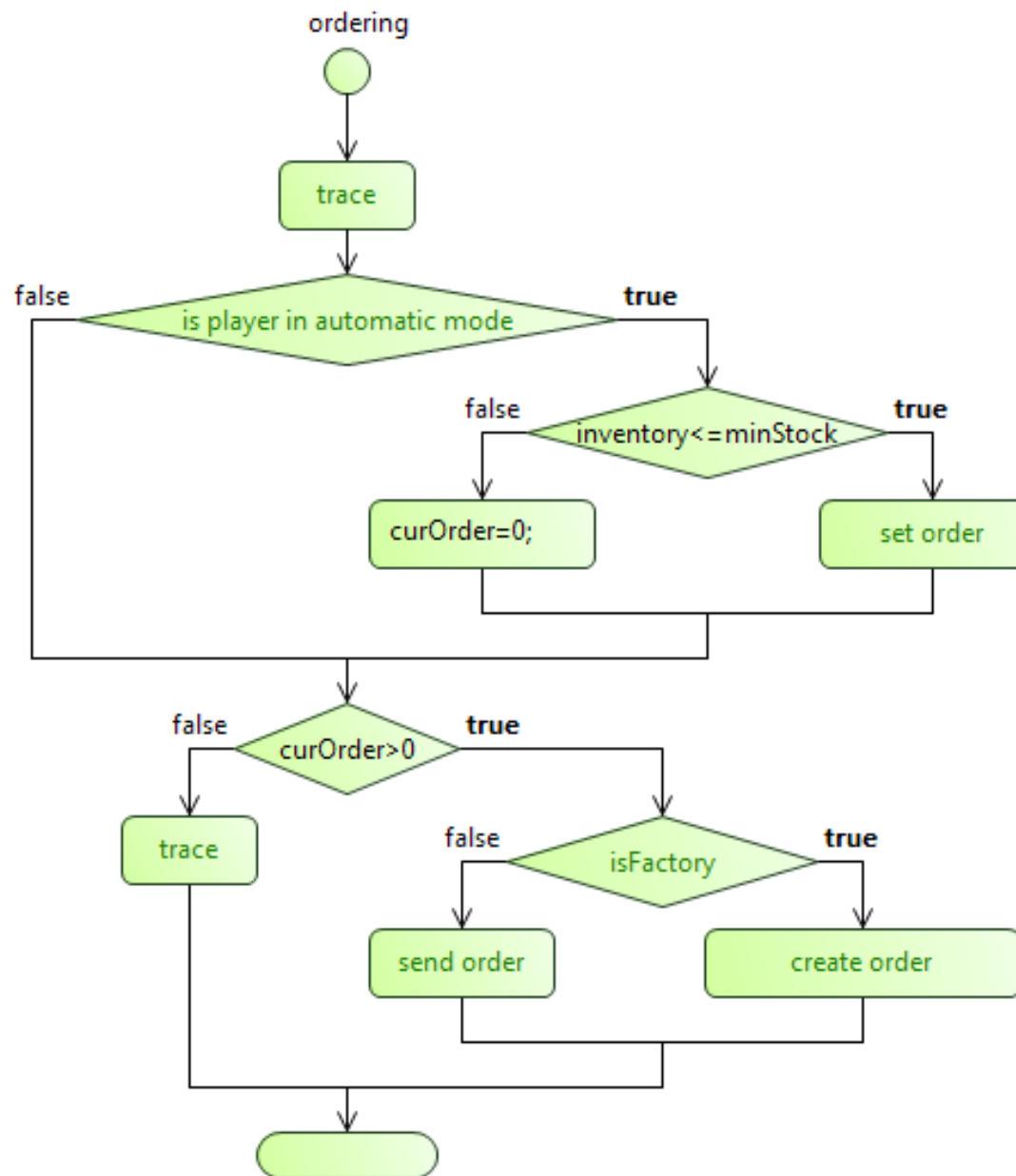
Port: portRight

Name	Value
General	
Show name	false
Incoming Message Type	Shipment

Name	Value
Outgoing Message Type	Order
On Receive	<pre>curReceive = msg.amount; //add arrived items to the inventory inventory += msg.amount; //and remove from expected shipments expected -= msg.amount; if(playerIndex == Main.RETAILER){ //println("Получили товара " + msg.amount + " ", expected шаг" + expected);</pre>

Action Chart: ordering

Name	Value
General	
Return Type	void



Return: returnStatement1

Decision: decision11

Name	Value
General	
Condition	curOrder>0

Code: code18

Name	Value
General	
Comment	trace
Action	//traceIn(", no order, total expected " + expected);

Code: code13

Name	Value
General	
Comment	trace
Action	//trace(this + ": inventory " + inventory + ", curBacklog " + curBacklog);

Decision: decision6

Name	Value
General	
Comment	is player in automatic mode
Condition	players.size() != 4 && !players.contains(this.toString())

Decision: decision9

Name	Value
General	
Condition	inventory<=minStock

Code: code16

Name	Value
General	
Action	curOrder=0;

Code: code15

Name	Value
General	

Comment	set order
Action	curOrder = (maxStock - inventory) + (backlogCalculation() - curReceive) - expected; //alternate variant of ordering //curOrder=maxStock-inventory-expected;

Name	Value
	//curOrder += max(0, backlogCalculation()-prevBacklog); ordersStatistics.add(curOrder);

Decision: decision26

Name	Value
General	
Comment	isFactory
Condition	playerIndex == Main.FACTORY

Code: code20

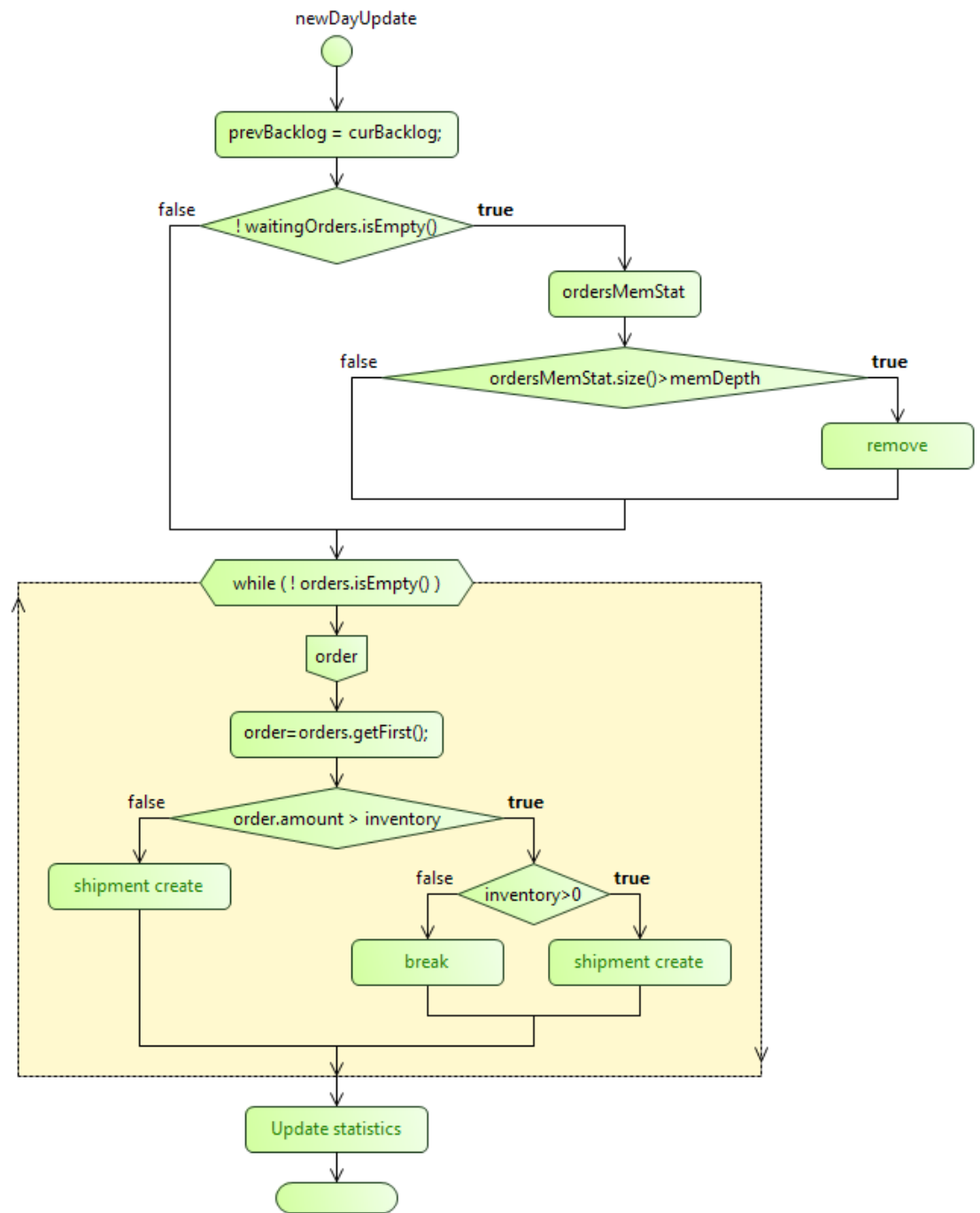
Name	Value
General	
Comment	send order
Action	expected += curOrder; portRight.send(new Order(curOrder, Player.this.portRight)); //traceIn(" ordered: " + curOrder + " total expected " + expected);

Code: code42

Name	Value
General	
Comment	create order
Action	expected +=curOrder; double creationSumTime = day() * curOrder / facOrdersPerDay; double ordersLeft = curOrder; while(ordersLeft > 0){ int newOrder = (int)min(ordersLeft, facOrdersPerDay); double newOrderTime = day() * newOrder / facOrdersPerDay; create_Manufacturing(newOrderTime + factoryTotalCreationLeftTime, newOrder); factoryTotalCreationLeftTime += newOrderTime; ordersLeft -= newOrder; }

Action Chart: newDayUpdate

Name	Value
General	
Return Type	void



Code: code7

Name	Value
General	
Action	prevBacklog = curBacklog;

Decision: decision4

Name	Value
General	
Condition	! waitingOrders.isEmpty()

Code: code12

Name	Value
General	
Action	ordersMemStat.add(0,waitingOrders.getLast().amount);

Decision: decision7

Name	Value
General	
Condition	ordersMemStat.size()>memDepth

Code: code14

Name	Value
General	
Comment	remove
Action	ordersMemStat.remove(memDepth);

While Loop: whileLoop1

Name	Value
General	
Condition	! orders.isEmpty()

Local Variable: order

Name	Value
General	
Type	Order

Code: code17

Name	Value
General	
Action	order=orders.getFirst();

Decision: decision8

Name	Value
General	
Condition	order.amount > inventory

Decision: decision10

Name	Value
------	-------

General	
Condition	inventory>0

Code: code19

Name	Value
General	
Comment	shipment create
Action	<pre>Shipment shipment = new Shipment(inventory); create_Delivery(timeSend-0.01, shipment, order.destination); source.inject(1); //increment productOnWay productOnWay +=inventory; //decrement amount in current order order.amount -= inventory; curSend = inventory; inventory = 0;</pre>

Code: code21

Name	Value
General	
Comment	break
Action	break;

Code: code22

Name	Value
General	
Comment	shipment create
Action	<pre>Shipment shipment = new Shipment(order.amount); create_Delivery(timeSend-0.01, shipment, order.destination); source.inject(1); //increment productOnWay productOnWay +=order.amount; //decrement inventory level inventory -= order.amount; curSend = order.amount; //remove order from the queue orders.removeFirst();</pre>

Code: code28

Name	Value
General	
Comment	Update statistics

Action	<pre>backlogCostTotal += backlogCalculation()*backlogCost; storageCostTotal += inventory*inventoryCost; storingCostStat.update(); backloggingCostStat.update(); inventoryStatistics.update(); backlogStatistics.update();</pre>
--------	--

Return: returnStatement2

Delay: shipmentDelay

Name	Value
General	
Type	Delay<T extends Entity>
Generic Parameters Substitute	Entity
Java Package Name	com.xj.anylogic.libraries.enterprise
Embedded Object Collection Type	ARRAY_LIST_BASED

Embedded Object Parameters:

Name	Value
Delay time is	false
Delay time	timeSend-0.01
Speed	10
Capacity	1
Maximum capacity	true
On enter	entity.setShape(shipmentImage);
Animation guide shape	shipmentDeliveryLine
Animation type	Animator.PATH
Animation direction	true
Enable statistics	false

Delay: orderDelay

Name	Value
General	
Type	Delay<T extends Entity>
Generic Parameters Substitute	Entity
Java Package Name	com.xj.anylogic.libraries.enterprise
Embedded Object Collection Type	ARRAY_LIST_BASED

Embedded Object Parameters:

Name	Value
Delay time is	false
Delay time	timeService-0.01
Speed	10
Capacity	1
Maximum capacity	true
On enter	entity.setShape(orderImage);
Animation guide shape	orderDeliveryLine
Animation type	Animator.PATH
Animation direction	true
Enable statistics	false

Enter: orderEnter

Name	Value
------	-------

General	
Type	Enter<T extends Entity>
Generic Parameters Substitute	Entity
Java Package Name	com.xj.anylogic.libraries.enterprise

Name	Value
Embedded Object Collection Type	ARRAY_LIST_BASED

Exit: exit

Name	Value
General	
Type	Exit<T extends Entity>
Generic Parameters Substitute	Entity
Java Package Name	com.xj.anylogic.libraries.enterprise
Embedded Object Collection Type	ARRAY_LIST_BASED

Exit: exit1

Name	Value
General	
Type	Exit<T extends Entity>
Generic Parameters Substitute	Entity
Java Package Name	com.xj.anylogic.libraries.enterprise
Embedded Object Collection Type	ARRAY_LIST_BASED

Source: source

Name	Value
General	
Type	Source<T extends Entity>
Generic Parameters Substitute	Entity
Java Package Name	com.xj.anylogic.libraries.enterprise
Embedded Object Collection Type	ARRAY_LIST_BASED

Embedded Object Parameters:

Name	Value
Arrivals defined by	Source.MANUAL
Arrival rate	1
Interarrival time	exponential(1)
Modify rate	false
Rate expression	baseRate
Entities per arrival	1
Limited number of arrivals	false
Maximum number of arrivals	Integer.MAX_VALUE
New entity	new Entity()
Entity animation shape	shipmentImage
Unique shape for each entity	false
Enable rotation	false
Enable vertical rotation	true

Time Plot: plot

Name	Value
General	

Name	Value
Time Window	10
Vertical Scale	AUTO
Analysis Auto Update	true
Recurrence	0.1
Dataset Samples To Keep	1000
Advanced	
x	0
y	10
Width	240
Height	140
Appearance	
Show Legend	true
Legend Place	SOUTH
Label Format	MODEL_TIME_UNITS

Plot Items:

Title	Type	Dataset / Value	Point Style	Color	Line	Width	Interpolation
inventory	value		NONE	blue	true	1	LINEAR
orders	dataset	ordersStatistics	NONE	green	true	1	LINEAR

Statistics: storingCostStat

Name	Value
General	
Discrete	false
Statistics Value	inventory * inventoryCost
Analysis Auto Update	false

Statistics: backlogingCostStat

Name	Value
General	
Discrete	false
Statistics Value	backlogCalculation() * backlogCost
Analysis Auto Update	false

Data Set: ordersStatistics

Name	Value
General	
Axis Data Freeze X Axis	true
Axis Data Vertical Y Axis	curOrder
Dataset Samples To Keep	10000
Analysis Auto Update	false

Data Set: inventoryStatistics

Name	Value
------	-------

Name	Value
General	
Axis Data Freeze X Axis	true
Axis Data Vertical Y Axis	backlogCalculation() > 0 ? (-1) * backlogCalculation() : inventory
Dataset Samples To Keep	10000
Analysis Auto Update	false

Data Set: backlogStatistics

Name	Value
General	
Axis Data Freeze X Axis	true
Axis Data Vertical Y Axis	curBacklog
Dataset Samples To Keep	10000
Analysis Auto Update	false

Java Class: Shipment

Name	Value
General	
Java Class Type	JAVA_CLASS
Text	<pre>public class Shipment implements java.io.Serializable { Shipment(int a) { amount = a; } int amount; public String toString() { return "Shipment[" + Utilities.format(amount) + "]; } /** * This number is here for model snapshot storing purpose
 * It needs to be changed when this class gets changed */</pre>

Java Class: Order

Name	Value
General	
Java Class Type	JAVA_CLASS
Text	<pre>public class Order implements java.io.Serializable { Order(int a, Port<?, ?> dest) { amount = a; destination = dest; } int amount; Port<?, ?> destination;</pre>

Name	Value
------	-------

```

        return "Order[" + Utilities.format(amount) + "]"
from " + destination.getActiveObject().getFullName();
    }

    /**
     * This number is here for model snapshot storing
purpose<br>
     * It needs to be changed when this class gets changed
     */
    private static final long serialVersionUID =
60004720857002485502L;

```

Simulation Experiment: Simulation

Name	Value
General	
Active Object Class	Main
Random Number Generation Type	fixedSeed
Seed Value	1
Advanced	
Maximum Available Memory	64
Differentiation Equations Method	EULER
Mixed Equations Method	RK45_NEWTON
Algebraic Equations Method	MODIFIED_NEWTON
Absolute Accuracy	1.0E-5
Time Accuracy	1.0E-5
Relative Accuracy	1.0E-5
Fixed Time Step	0.0010
Presentation Top Group Persistent	true
Model Time	
Stop Option	Stop at specified time
Initial Time	0.0
Final Time	100.0
Presentation	
CPU Time Balance	ratio_1_2
Execution Mode	realTimeScaled
Real Time Scale	1.0
Window	
Title	BeerGame : Simulation
Enable Panning	false
Enable Zoom	false
Real Time Of Simulation	false

Timings ① timeOrder ② timeOrder ③ timeOrder ④ timeOrder ⑤ timeOrder ⑥ timeOrder ⑦ timeOrder ⑧ timeOrder	Costs ① holdingCost ② storageCost ③ invCost ④ invCost ⑤ invCost ⑥ invCost ⑦ invCost ⑧ invCost	Min-max strategy ① invCost ② invCost ③ invCost ④ invCost ⑤ invCost ⑥ invCost ⑦ invCost ⑧ invCost
--	--	---



Beer Distribution Game Tutorial

Introduction to the Beer Game

The purpose of the Beer Game is to experience systemic effects and to introduce the need for supply chain and network management. Specifically it shows:

- How single parts in a system influence each other
- How individual thinking affects from systemic thinking (network thinking)
- Potentials for systems optimization and the use of information systems

The Beer Game setup

The Beer Game supply chain consists of four stages:

- The retailer has to fulfill the end consumer's orders
- The wholesaler has to fulfill the retailer's orders
- The distributor has to fulfill the wholesaler's orders
- The factory has to produce the beer to fulfill the distributor's orders

The goal

Your goal is to minimize your cost!

There are two different kinds of cost:

- Inventory cost: Items in stock cost x per week in holding cost.
- Backorder cost: If an incoming order cannot be fulfilled, the items are outstanding and have to be put on "backorder" in the following day(s). Each item on backorder costs y and y are set before the game starts.

General playing procedure

The game runs in days and it starts in day 1. In each day, each supply chain group has to proceed with the following steps:

- Receive new deliveries
- Receive orders in the inbox
- Calculate the total amount that will be shipped ("Your delivery"), note down the amount, and ship to the next player
- Calculate a new order amount, and send order

Settings

You can choose game mode and play for retailer, wholesaler, distributor, factory or any combination of them. Every step you can order beer at your supplier. Every order will be automatically processed and product will be sent if enough amount is available at the inventory. Also you should consider the delays on servicing your order and on delivery.

Variable: invRet

Name	Value
General	
Show At Runtime	false
Type	int
Initial Value	50

Variable: invWho

Name	Value
General	
Show At Runtime	false
Type	int
Initial Value	50

Variable: invDis

Name	Value
General	
Show At Runtime	false
Type	int
Initial Value	50

Variable: invFac

Name	Value
General	
Show At Runtime	false
Type	int
Initial Value	50

Variable: sRet

Name	Value
General	
Show At Runtime	false
Type	int
Initial Value	30

Variable: sWho

Name	Value
General	
Show At Runtime	false
Type	int
Initial Value	30

Variable: sDis

Name	Value
General	
Show At Runtime	false
Type	int
Initial Value	30

Variable: sFac

Name	Value
General	
Show At Runtime	false
Type	int
Initial Value	30

Variable: storageCost

Name	Value
General	
Type	double
Initial Value	0.5

Variable: backlogCost

Name	Value
General	
Type	double
Initial Value	1

Variable: timeWhoSend

Name	Value
General	
Show At Runtime	false
Type	int
Initial Value	2

Variable: timeDisSend

Name	Value
General	
Show At Runtime	false
Type	int
Initial Value	2

Variable: timeFacSend

Name	Value
General	
Show At Runtime	false
Type	int
Initial Value	2

Variable: timeFacOrder

Name	Value
General	
Show At Runtime	false
Type	int
Initial Value	2

Variable: timeWhoOrder

Name	Value
General	
Show At Runtime	false
Type	int
Initial Value	2

Variable: timeDisOrder

Name	Value
General	
Show At Runtime	false
Type	int
Initial Value	2

Variable: memDepth

Name	Value
General	
Type	int
Initial Value	50

Variable: SRet

Name	Value
General	
Show At Runtime	false
Type	int
Initial Value	80

Variable: SWho

Name	Value
------	-------

General	
Show At Runtime	false
Type	int
Initial Value	90

Variable: SDis

Name	Value
------	-------

Name	Value
General	
Show At Runtime	false
Type	int
Initial Value	100

Variable: SFac

Name	Value
General	
Show At Runtime	false
Type	int
Initial Value	120

Variable: facOrdersPerDay

Name	Value
General	
Show At Runtime	false
Type	int
Initial Value	20

Collection: players

Name	Value
General	

Collection Class	java.util.ArrayList
Element Class	String

Slider: slider4

Name	Value
General	
Orientation	HORIZONTAL
Link To	true
Link	sRet
Minimum Value	0
Maximum Value	99
Dynamic: Enable	getState() == IDLE
Action	<pre>if (SRet<value){ SRet=(int)value + 1; slider8.setValue(SRet); }</pre>
Advanced	
x	-130
y	-70
Width	180
Height	30

Slider: slider8

Name	Value
General	
Orientation	HORIZONTAL
Link To	true
Link	SRet
Minimum Value	0
Maximum Value	200
Dynamic: Enable	getState() == IDLE
Action	<pre>if(value < sRet) { SRet = sRet + 1; slider8.setValue(SRet); }</pre>
Advanced	
x	-130
y	10
Width	180
Height	30

Group: group1

Name	Value
Advanced	
x	385
y	550
Dynamic	
Dynamic: Visible	(players.size() != 4 && !players.contains("Wholesaler"))

Slider: slider5

Name	Value
General	
Orientation	HORIZONTAL
Link To	true
Link	sWho
Minimum Value	0
Maximum Value	99
Dynamic: Enable	getState() == IDLE
Action	if (SWho<value){ SWho=(int)value + 1; slider9.setValue(SWho); }
Advanced	
x	-100
y	-70
Width	180
Height	30

Slider: slider9

Name	Value
General	
Orientation	HORIZONTAL
Link To	true
Link	SWho
Minimum Value	0
Maximum Value	200
Dynamic: Enable	getState() == IDLE
Action	if(value < sWho) { SWho = sWho + 1; slider9.setValue(sFac); }
Advanced	

Name	Value
x	-100
y	10
Width	180
Height	30

Slider: slider10

Name	Value
General	
Orientation	HORIZONTAL
Link To	true
Link	SDis

Name	Value
Minimum Value	0
Maximum Value	200
Dynamic: Enable	getState() == IDLE
Action	<pre>if(value < sDis) { SDis = sDis + 1; slider10.setValue(sFac); }</pre>
Advanced	
x	-60
y	10
Width	180

Height	30
--------	----

Slider: slider7

Name	Value
General	
Orientation	HORIZONTAL
Link To	true
Link	sFac
Minimum Value	0
Maximum Value	99
Dynamic: Enable	getState() == IDLE
Action	<pre>if (SFac<value){ SFac=(int)value + 1; slider10.setValue(SFac); }</pre>
Advanced	
x	-80
y	-70
Width	180
Height	30

Slider: slider11

Name	Value
General	
Orientation	HORIZONTAL
Link To	true
Link	SFac
Minimum Value	0
Maximum Value	200
Dynamic: Enable	getState() == IDLE

Action	<pre> if(value < sFac) { SFac = sFac + 1; slider11.setValue(SFac); } </pre>
Advanced	
x	-80
y	10
Width	180
Height	30

Button: button

Name	Value
General	
Label Text	Run the model
Action	<pre> if (getState() == IDLE){ run(); if (players.size() == 0){ getEngine().setRealTimeScale(1); } else { getEngine().setRealTimeScale(4); } } </pre>

Name	Value
	<pre> } getEngine().getPresentation().setPresentable(getEngine().getRoot()); </pre>
Advanced	
Text Color	black
Font Name	Dialog
Font Size	22
Bold Font Style	true
Italic Font Style	true
x	776
y	640
Width	205
Height	40

Dynamic	
Dynamic: Label	getState() == IDLE ? "Run the model" : "Back to name"

Slider: slider

Name	Value
General	
Orientation	HORIZONTAL
Link To	true
Link	invRet
Minimum Value	0
Maximum Value	100
Dynamic: Enable	getState() == IDLE
Advanced	
x	30
y	390
Width	180
Height	30

Slider: slider1

Name	Value
General	
Orientation	HORIZONTAL
Link To	true
Link	invWho
Minimum Value	0
Maximum Value	100
Dynamic: Enable	getState() == IDLE
Advanced	
x	285
y	390
Width	180
Height	30

Slider: slider2

Name	Value
General	
Orientation	HORIZONTAL
Link To	true
Link	invDis
Minimum Value	0
Maximum Value	100
Dynamic: Enable	getState() == IDLE
Advanced	
x	540
y	390
Width	180
Height	30

Slider: slider3

Name	Value
General	
Orientation	HORIZONTAL
Link To	true
Link	invFac
Minimum Value	0
Maximum Value	100
Dynamic: Enable	getState() == IDLE
Advanced	
x	795
y	390
Width	180
Height	30