

TRUE

TEMPORAL REASONING UNIVERSAL ELABORATION

True System dynamics software

MANUAL Part 60

Exercises

Last release 2014/03/07

www.true-world.com

Contents

I - EXERCISE 01.....	3
A) Statement.....	3
II - EXERCISE 02.....	3
A) Statement.....	3
III - EXERCISE 03.....	3
A) Statement.....	3
IV - EXERCISE 04.....	3
A) Statement.....	3
V - EXERCISE 05.....	4
A) Statement.....	4
VI - EXERCISE 06.....	4
A) Statement.....	4
VII - EXERCISE 07.....	4
A) Statement.....	4
VIII - EXERCISE 08.....	4
A) Statement.....	4
IX - EXERCISE 09.....	5
A) Statement.....	5
X - EXERCISE 10.....	5
A) Statement.....	5

I - EXERCISE 01

A) Statement

In nature, when a prey is killed, lions eat first, then the lionesses, the hyenas and vultures. Create an appropriate model, showing the necessary and consumed food.

See a solution in the model **Exercise01-MealsAnimals**

Create the same model with another system dynamics software such Vensim, Compare and explain the results.

II - EXERCISE 02

A) Statement

Modify the model **Exercise01-MealsAnimals**:

- The park opens in July 2014
- Tigers come only 3 months per years, startgin from december
- Tigers eat after lions and before lioness

See a solution in the model **Exercise02-OpenInJuly**

III - EXERCISE 03

A) Statement

Modify the model **Exercise02-OpenInJuly**:

- Add stocks as parameters for Animals number and what (in kg) they eat
- Adjust the flow 'Food delivery'

See a solution in the model **Exercise03-StockDependencies**

IV - EXERCISE 04

A) Statement

Modify the model **Exercise03-StockDependencies**:

- Add a flow to calculate the monthly food consumption
- Find visually in the model which animals have no food and when

See a solution in the model **Exercise04-FlowDependencies**

V - EXERCISE 05

A) Statement

Modify the model **Exercise04-FlowDependencies**:

- To simplify the model, create a global procedure to replace the code in the equations of the flows 'Lions eat', 'Lioness eat', ...

See a solution in the model **Exercise05-GlobalProcedure**

VI - EXERCISE 06

A) Statement

Modify the model **Exercise05-GlobalProcedure**:

- To simplify the model, create a vector with 5 domains for each animal.
- Affect the vector to a flow 'Eat' and the stocks: 'Animals', 'AnimalsKg', 'AnimalsNb'
- Delete unused other elements.

See a solution in the model **Exercise06-Vectorization**

VII - EXERCISE 07

A) Statement

Modify the model **Exercise06-Vectorization**:

The flow 'FoodPerMonths' returns the monthly food consumption.

Find and add other solutions to retrieve the monthly food consumption using:

- 1) ValueWrite() and ValueRead() functions: write and read a pseudo variable
- 2) Svalue() function: read a stock value

See a solution in the model **Exercise07-OtherSolutionsForMonthlyFoodConsumption**

VIII - EXERCISE 08

A) Statement

Modify the model **Exercise06-Vectorization**:

Add a flow 'FoodMissing' to detect animals that do not eat.

Display the missing food in a chart.

See a solution in the model **Exercise08-FoodMissing**

IX - EXERCISE 09

A) Statement

Modify the model **Exercise08-FoodMissing**:

Add a flow named 'FoodMissing delivery' to delivery the missing food in the stock 'FoodStock', just before the food is missing:

- use the ReStartAt() function to initialise a retro-calculation.

Add a chart to display the 'FoodMissing delivery'

After computing the model with retro-calculation:

- modify all the charts: check the box 'Values will follow the current Passing(...)', in the view 'Elements'

- open the 'ReStart' windows, by pressing the key 'R'.

- select the current passing (1 to 3) and observe the charts and the model.

- you may too select the current passing with the spin located on the right of the main window, after the field (ReSt...art.)

See a solution in the model **Exercise09-RetroCalculation**

X - EXERCISE 10

A) Statement

Modify the model **Exercise09-RetroCalculation**:

Add mirror stocks and mirror flows to retrieve and to verify the:

- Total food required per month

- Total food consumption per month

The sum of these two values must be null

- Total animals number

- Total of the AnimalsKg parameter

See a solution in the model **Exercise10-MirrorStock**