Emote Scenario 2 User Guide

Installation

To install the system simply copy and paste the folder containing the scenario modules on your computer. To be sure each module will be able to run, verify that the system matches the requirements listed in the Appendix.

Each module is a Thalamus module. This means that system may run with the modules distributed on several computers, as long as they are connected to the same network. This is particularly useful when the scenario has to run on low end computers, where a single computer cannot stand the work load of several modules together.

Execution

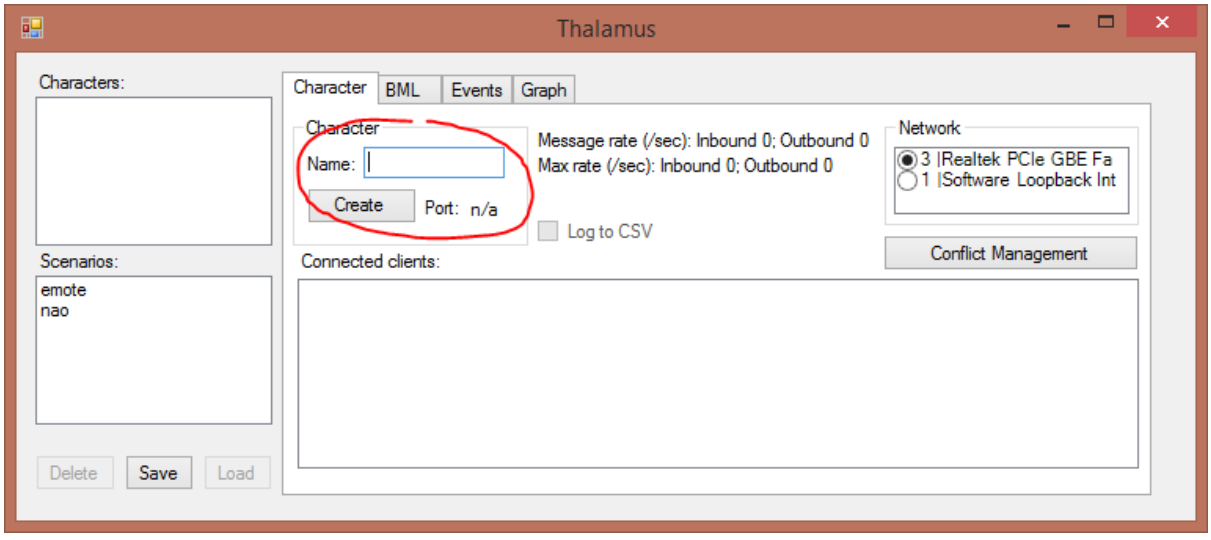
Thalamus

As described before, the scenario is made up by several thalamus modules. Thalamus is a middleware that allows message exchanging among several entities called Thalamus modules (or clients).

To run the scenario, thalamus must be running in the system.

After copying the scenario files on the computer, run the *Thalamus/ThalamusStandalone.exe* executable to start the Thalamus server. Due to Windows restrictions, **ThalamusStandalone and its modules must be run as administrator.**

Once the Thalamus window opens, create a new Thalamus Character writing its name in the "*Name*" textbox and clicking on the "*Create*" button as showed in the image below.



Thalamus Modules

To run the scenario a set of thalamus modules need to be run:

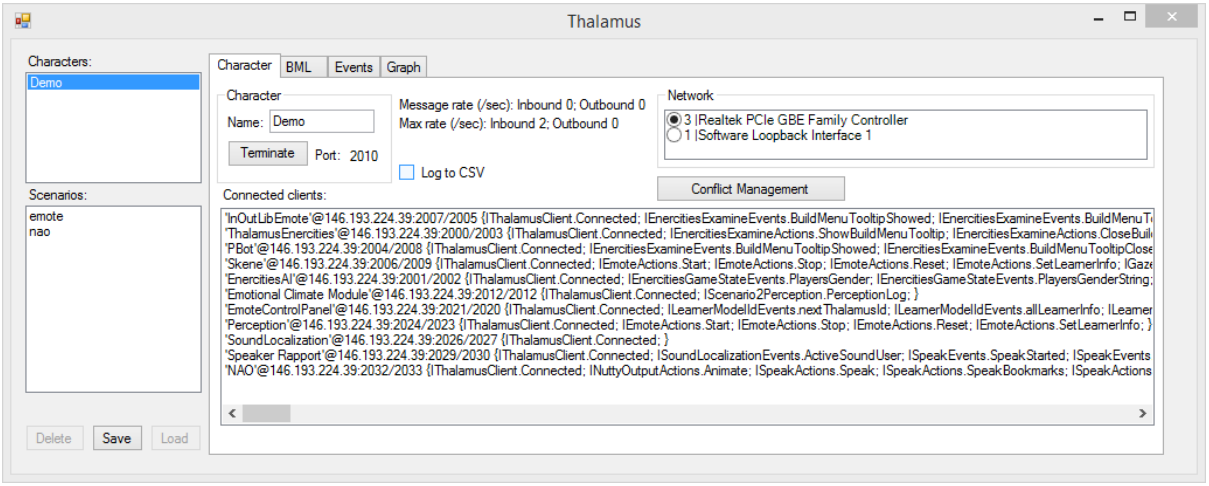
- ControlPanel
- ECModule
- EnercitiesAI
- LogicWeb

- NAOThalamusGUI
- Perception
- Skene
- SpeakerDetector
- SpeakerRapport
- ThalamusEnercities
- PBot
- LearnerModelThalamus

Each module can run on a different machine, as long as the machine is connected to the same network where the machine running Thalamus is and **there are no firewalls or antivirus blocking the connection between Thalamus and its modules**.

When running a Thalamus module, it will search for the first available Thalamus character on the network and connect to it.

Once every module is connected, the Thalamus Standalone interface should look like in the following picture:



Thalamus and Thalamus Modules command line arguments

Both Thalamus and the Thalamus Modules accept command line arguments to perform automatically certain tasks. Each module may respond to different arguments, specified in the *Readme.txt* file located in the module root directory.
Normally, every thalamus module accept a **first string argument** indicating the name of the Thalamus Character the module will connect to. This is especially useful in a network where more Thalamus Character are present.

Perception

The perception module is a resources greedy module that is better to be run on a dedicated machine. It sends to the Thalamus Character data coming from sensors as: **Kinect 2, Okao** and **qSensors**.

This module divides into several pieces, each managing different sensors. The main Perception module can be started from the file *"Perception/perception.exe"* and controls Kinect and qSensors.

To receive input from a camera using Okao, a separate module in *"OKAOModule/OKAOmodule_Release.exe"* must be started.

For detailed instructions on how to use the Perception module please refer to its *readme.txt* file.

NAOThalamusGUI

This module is used to connect the Thalamus character to the NAO robot. It uses the *bonjour* protocol to detect any NAO robot connected on the network and it automatically starts the python script it contains.

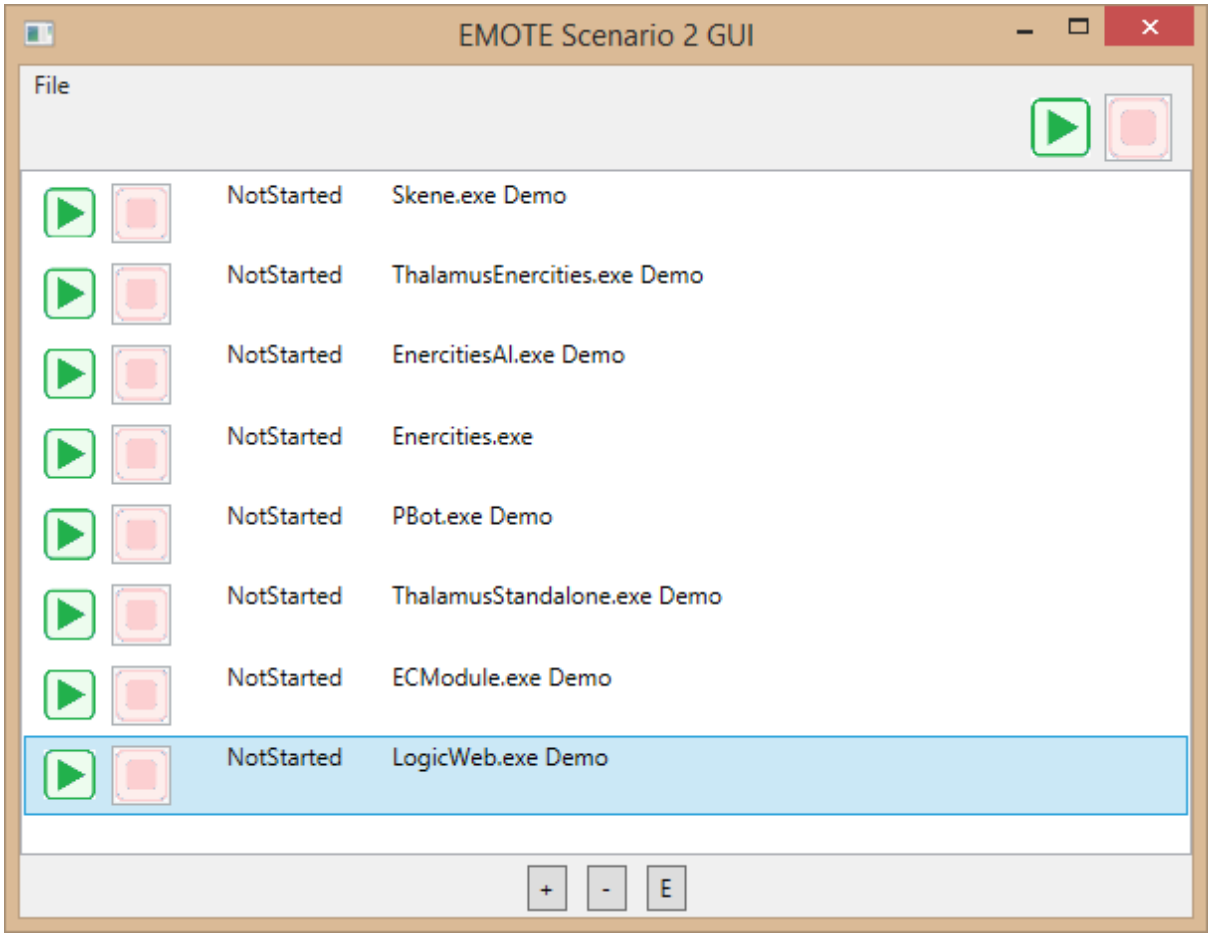
If there are more than one NAO robots connected to the network or the robot cannot automatically be found, it may be necessary to **manually specify the NAO robot's IP address**. To do so start the NAOThalamusGUI module from a shortcut and set as **second command line argument** the IP of the robot the module should connect to.

Once the module connects to the robot it will check if in the robot are installed the python scripts and the behaviour necessary to control the robot. If it doesn't find them or they are outdated, an *"Install"* button will appear next to the feature that needs to be updated or installed (python or behaviour). It is usually good practice to keep the robot updated with the latest version of the python scripts and behaviours.

Note:

This modules suppose that the NAO robot username and password to access to its ssh command shell are both set to *"nao"*.

1 Click Start



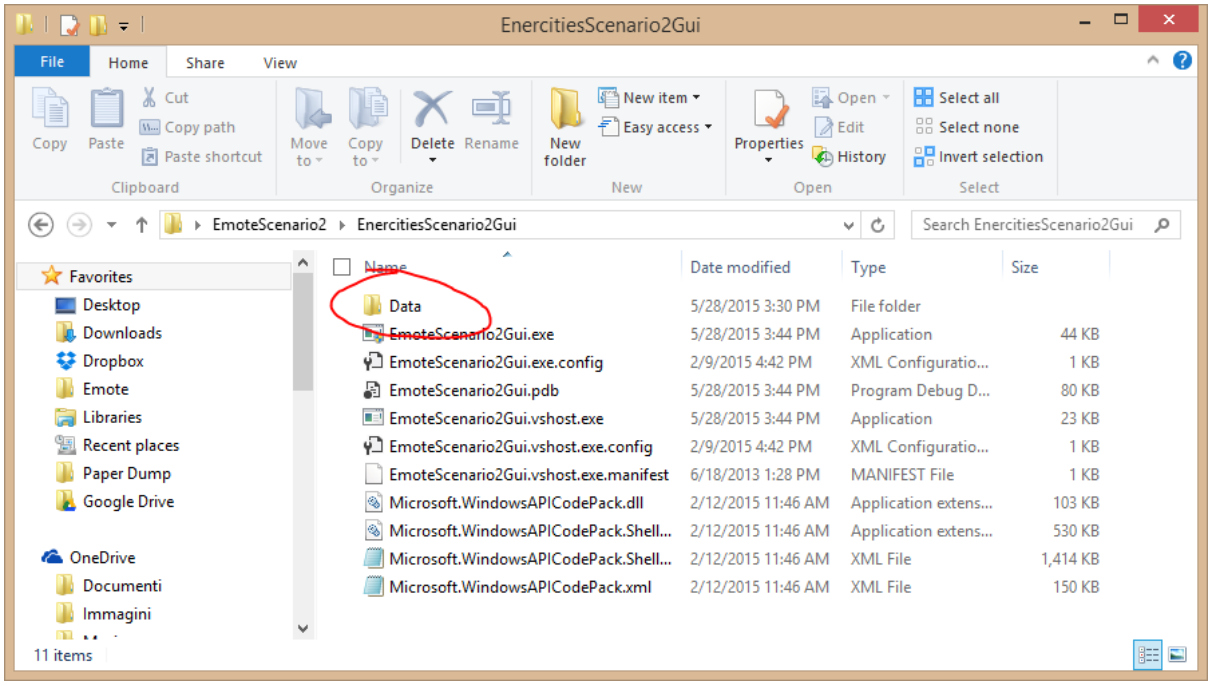
To make it easy to run the scenario, a small utility located in *"EnercitiesScenario2Gui/EmoteScenario2Gui.exe"* can be used. Its purpose is to automatize the start of all the modules necessary to run the scenario.

Basic usage:

- Using the buttons on the bottom of the window it's possible to add, remove or edit a module in the list.
- Using the Play/Stop buttons on the upper right corner of the window, it's possible to Run or Stop all the modules in the list.
- Using the Play/Stop buttons on the left of each module in the list, it's possible to Run or Stop that specific module.
- When running a module, the EmoteScenario2Gui stores information regarding the module's window location and size. Next time the module will be run, it will be placed and sized as it was in the latest saved state.

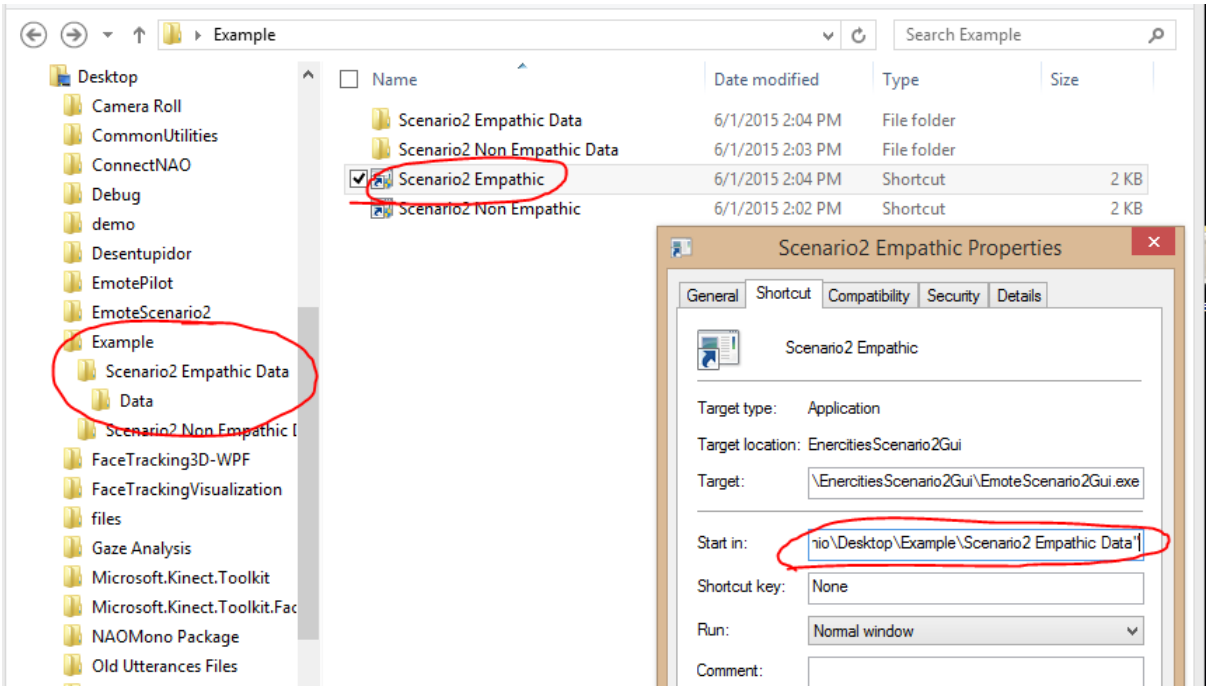
Data file:

The list of modules is saved in a folder called *"Data"* located in the EmoteScenaio2Gui working directory.



The application allows to have only one saved list, but since the path of the saved list is relative to the working directory, it is possible to have several link to the main application, each with a different working directory.

Example:



In this example there are two shortcuts to the *EmoteScenario2Gui.exe* utility, but each one has a different working directory:

- *Scenario2 Empathic:* Desktop\Example\Scenario 2 Empathic Data\
- *Scenario2 Non Empathic:* Desktop\Example\Scenario 2 Non Empathic Data\

With this trick it is possible to have two different lists saved into two separate folders. In this way it is possible to have presets of modules to be run for specifics needs.

Note:

Even though in this guide it's about how to use *EmoteScenario2Gui.exe* to run Thalamus and Thalamus modules, it can work for any runnable application on Windows.

So if, for example, every time the scenario run another application like a screen capture needs to be run, it is possible to add it to the modules list.

Issues

There are a few known issues that is good to be aware of:

- Sometimes, especially when copying a data file from one computer to another, it is possible that the position and size of the window of a module in the list get messed up and prevent the window to be visible. To fix this select the module on the main window; click on the edit button (the *"E"* button on the bottom). Confirm clicking on the *"Edit"* button of the dialog window that will appear.
- Sometimes, when a module crashes, the EmoteScenario2Gui.exe utility crashes as well leaving all the already started modules running. All the modules will keep running normally but even restarting the EmoteScenario2Gui.exe utility it won't be possible to control them with it.

Thalamus Logs

Every message sent to and coming from a Thalamus module or a Thalamus server is logged in several locations.

- For each Thalamus Module there is a *"Log"* folder located in the working directory containing the log of every message received and sent from that specific module.
- In the working directory of the Thalamus Server (ThalamusStandalone.exe) there is a *"Log"* folder containing a *ThalamusMaster* log and several other logs named after each running module's name.
  The *ThalamusMaster* log contains ALL the messages received and sent by and from the Thalamus server.
  All the other logs named after their respective module contains exactly the same information contained in each module's Log folder.

Since the **ThalamusMaster log contains all the message** coming and going from and to all the modules, this is the log that makes more sense to be analysed in a second moment. All the other files contains redundant information limited to the specific module that can still be useful for other tasks.

Study Workflow (Binaries Folder)

When running a study with Scenario 2 it is recommended to split the modules into two machines:

- **Multitaction**:
  - ECModule *
  - EnercitiesAI *
  - LogicWeb *
  - Skene *
  - SpeakerDetecto
  - SpeakerRapport
  - ThalamusEnercities
  - PBot
  - Enercities Game *
  - ThalamusStandalone (Thalamus server)
- **Second machine**:
  - Perception
  - OKAOModule
  - LearnerModelJava *(see appendix)*
  - LearnerModelThalamus
  - ControlPanel
  - NAOThalamusGUI

** Modules that MUST be restarted for every session*

With such setup all the modules that need to be restarted every session are installed in the Multitaction, while all the modules that can keep running are located on the second machine.

In this way, it is possible to create a list of modules in the EmoteScenario2Gui utility for the multitaction and *play* and *stop* the entire list at every session.

Since each thalamus client and server are logging continuously until they are closed, restarting the module list at every session brings the added benefit to thave **Thalamus Logs divided by session**.

Appendix

List of modules and applications needed to run the scenario:

- ControlPanel\ControlPanel.exe
- ECModule\ECModule.exe
- Enercities\Enercities.exe *
- EnercitiesAI\EnercitiesAI.exe
- LogicWeb\LogicWeb.exe
- NAOThalamusGUI\NAOThalamusGUI.exe
- PBot\PBot.exe
- Skene\Skene.exe
- SpeakerDetector\SpeakerDetector.exe
- SpeakerRapport\SpeakerRapportGui.exe
- ThalamusEnercities\ThalamusEnercities.exe
- Thalamus\Thalamus.exe
- Perception\Perception.exe
- LearnerModelThalamus\LearnerModelThalamus.exe
- OKAOModule\OKAOModule_Release.exe *
- Eclipse with Tomcat and LearnerModule java code **

** These are standalone applications and do not connect to a Thalamus character.*
*** To install and run the learner model and the map application please refer to the guide at: https://github.com/AidanJones/EmoteMaps/*

Libraries Requirements

- An installed version of Microsoft Office 2010.
  If there is no Office 2010 installed on the machine it is possible to alternately execute the intallers contained into the *"TO_INSTALL_IN_CASE_OF_ERRORS"* folder.