The MORSE Robotics simulation platform based on Blender

Gilberto Echeverría
gechever@laas.fr

Laboratoire d’Analyse et d’Architecture des Systèmes
Toulouse, France

Blender Conference
October 30, 2010
What is robotics about?

- Integration of many technologies
- From automatic to autonomous machines
- Reasoning about complex tasks
- Dealing with uncertainty
- Applications in any domain
The importance of simulation

- Less expensive
- More control over the environment
- How complex?
  - Specialized simulation of components
  - System wide simulation (whole robot)

Realism levels of a simulation
Outline

1. Requirements of a new simulator
   - Background of MORSE
   - Component library
   - General architecture

2. Current state of MORSE
   - Simulated components
   - Python Scripts in Blender
   - Future developments

3. Summary
Robotics at LAAS

- No hardware development, only control software
- Research with sensors, interaction and control
- Multiple projects
  - Robot teams performing a complex task
  - Human–robot interaction

Robot platforms
Why use Blender?

- Game Engine
- **Bullet** Physics simulation
- Realistic graphics
- Event oriented programming using **Logic Bricks**
- Python Scripts
- Modelling of robots and scenarios

Simulated scenario in Blender
Requirements for the simulation:

- Use Blender as the base platform
- Modular and reusable architecture
- “Software in the loop” philosophy
- Multi–robot simulation
- Adjustable levels of realism
Components in individual files

Very simplified modules

Middlewares add connectivity

Modifiers change data

Blender file

Python file
Middlewares

Definition

A Middleware is a type of software used to transfer data between individual components. Mainly used in distributed heterogeneous systems.

- Many middlewares are used in robotics
- They encapsulate data
- MORSE must be middleware independent
- Each component can use a different middleware
Modifications to the data

- Simulated data is “perfect”
- Data in the real world is imprecise and noisy
- Simulated data must be as close to reality as possible
- Modifications to the reference frames, scale, units, etc.

**Definition**

A **Modifier** is a program that alters the data before it is sent outside the simulator.
MORSE main loop

main.init()

FOR_EACH sensors

sensor.action()

sensor.default_action()

sensor.local_data

FOR_EACH sensor.output_modifiers

output_modifier(data)

FOR_EACH sensor.output_functions

output_function(data)

FOR_EACH actuators

actuator.action()

actuator.default_action()

actuator.input_modifiers

actuator.input_functions

FOR_EACH actuator.input_modifiers

input_modifier(data)

FOR_EACH actuator

input_function(data)

FROM CLIENTS
(possibly via middlewares)

TO CLIENTS
(possibly via middlewares)

Blender
3D world

sensor.output_modifiers
SET

sensor.output_functions
SET

actuator.input_modifiers
SET

actuator.input_functions

(from component_config.py)

Gilberto Echeverría
The MORSE simulator
Requirements of a new simulator

Current state of MORSE

Background of MORSE

Component library

General architecture

FOR_EACH sensors

sensor.default_action()

sensor.local_data

FOR_EACH sensors

output_modifier(data)

FOR_EACH sensors

output_function(data)

FOR_EACH actuators

actuator.default_action()

actuator.action()

FOR_EACH actuators

input_modifier(data)

FOR_EACH actuators

input_function(data)

Blender 3D world

FROM CLIENTS (possibly via middlewares)

TO CLIENTS (possibly via middlewares)

Gilberto Echeverría

The MORSE simulator
Requirements of a new simulator
Current state of MORSE
Background of MORSE
Component library
General architecture

FOR_EACH actuators

actuator.action()

FOR_EACH actuator.input_functions

FOR_EACH actuator.input_modifiers

input_modifer(data)

input_function(data)

sensor.local_data

FOR_EACH sensors

output_function(data)

output_modifier(data)

FOREACH sensor.output_modifiers

FOREACH sensor.output_functions

FOR_EACH actuators

actuator.default_action()
Control software of each robot can run on different CPU
- Middlewares are used to communicate the computers
- A single instance of MORSE controls the simulation
Outline

1. Requirements of a new simulator
   - Background of MORSE
   - Component library
   - General architecture

2. Current state of MORSE
   - Simulated components
   - Python Scripts in Blender
   - Future developments

3. Summary
Robots

- All terrain ground robot (DALA)
- Mobile platform with robotic arm (JIDO)
- Robotic helicopter (RESSAC)

Gilberto Echeverría

The MORSE simulator
Robots

- All terrain ground robot (DALA)
- Mobile platform with robotic arm (JIDO)
- Robotic helicopter (RESSAC)
Robots

- All terrain ground robot (DALA)
- Mobile platform with robotic arm (JIDO)
- Robotic helicopter (RESSAC)
Programming of components

- The behaviour of sensors and actuators is scripted in Python
- Using the predefined functions in Blender
- Cameras use the VideoTexture module
- SICK sensor uses vertex editing
- Other sensors use rayCastTo function
Building a robot in MORSE

Linking of components in different files
External Python files

- Python scripts inside the .blend files are difficult to handle using version control systems
- Use of external Python modules
- Blender files reference the Python modules and classes
- Modules are dynamically loaded
- Initialization scripts allow better control of components in the scene
Logic brick programming for modules
Logic brick programming for modules
Simulation examples

Single robot simulation
Simulation examples

Team of terrestrial robots following a path
Things to improve in MORSE

- Current limitations of Blender
  - Entirely dependant on Blender’s predefined functions to interact with the simulated world
  - Game Engine does not have access to all of Blender’s functionality
  - Linking multiple files is not easy
  - Management of time inside the GE

- Future plans
  - Switch to Blender 2.5 and Python 3
  - Armatures in GE (including IK)
  - Configurable GUI
  - Synchronisation of multiple MORSE simulators
Outline

1. Requirements of a new simulator
   - Background of MORSE
   - Component library
   - General architecture

2. Current state of MORSE
   - Simulated components
   - Python Scripts in Blender
   - Future developments

3. Summary
Summary

- Blender’s Game Engine provides an **interactive** platform
- Great **graphic detail** and **physics simulation**
- Python offers **dynamic** module programming
- The MORSE project can be used in various conditions thanks to a **modular structure** and **middleware compatibility**
Contact, questions, more information:

- E-mail: gechever@laas.fr
- Developers list: morse-dev@laas.fr
- Website: http://www.laas.fr/morse

THANKS FOR YOUR ATTENTION!!!