

# Presentation MORSE

## Qu'est-ce que M.O.R.S.E ?

### Definition

The Modular OpenRobots Simulation Engine

### Histoire

Créé par le **LASS-CNRS** (Laboratoire d'Analyse et d'Architecture des Systèmes). Puis repris par d'autres universités (Georgia Tech, Dassault Aviation, INRIA, ISAE SUPAERO ...)

### En quelques mots

- simulateur d'environnement 3D
- focus sur le rendering (fait par Blender)
- Physique géré par Bullet Library
- entierement écrit en Python (sauf les calculs de rendering 3D et de la physique)

## Comment ca marche ?

### Utilisation ultra-simple

```
# Creation d'un 'workspace'
morse create mysim
# Lancement d'une simulation
morse run mysim
```

### Utilisation Simple - Analyse du 'workspace'

Architecture:

- **data**: dossier contenant les fichiers blender (robots, environnement)
- **scripts**: dossier contenant les scripts pour vos interagir avec vos simulations (controle, lecture des capteurs..etc)
- **src**: dossier contenant les définitions des robots, de leur capteurs, et actionneurs (c'est aussi la que vous mettez vos equations d'états) *\_\_default.py*: script python pour configurer la simulation

### **Script Python (default.py)**

- robot
- capteurs
  - kinect
  - camera
  - ...
- publication de données

### **Fichiers blender**

tout l'environnement peut être défini sous blender. Il y a certains paramètres à ajouter pour faire qu'ils interagissent avec l'environnement correctement

### **Scripts**

Python (import pymorse) pour contrôler et accéder aux données

### **Lien avec des middlewares pour faire des systèmes plus complexes**

MORSE s'interface très bien avec des middlewares:

- ROS
- MOOS
- YARP
- PocoLibs

Mais aussi avec des sockets

### **Utilisation Avancée**

Equation d'état

#### **Créer son robot**

#### **Ajouter des actuateurs**

#### **Ajouter des capteurs (Optionel)**