ITMO UNIVERSITY

V-REP quick start

Teacher: Islam Bzhikhatlov.

St-Petersburg, 2018

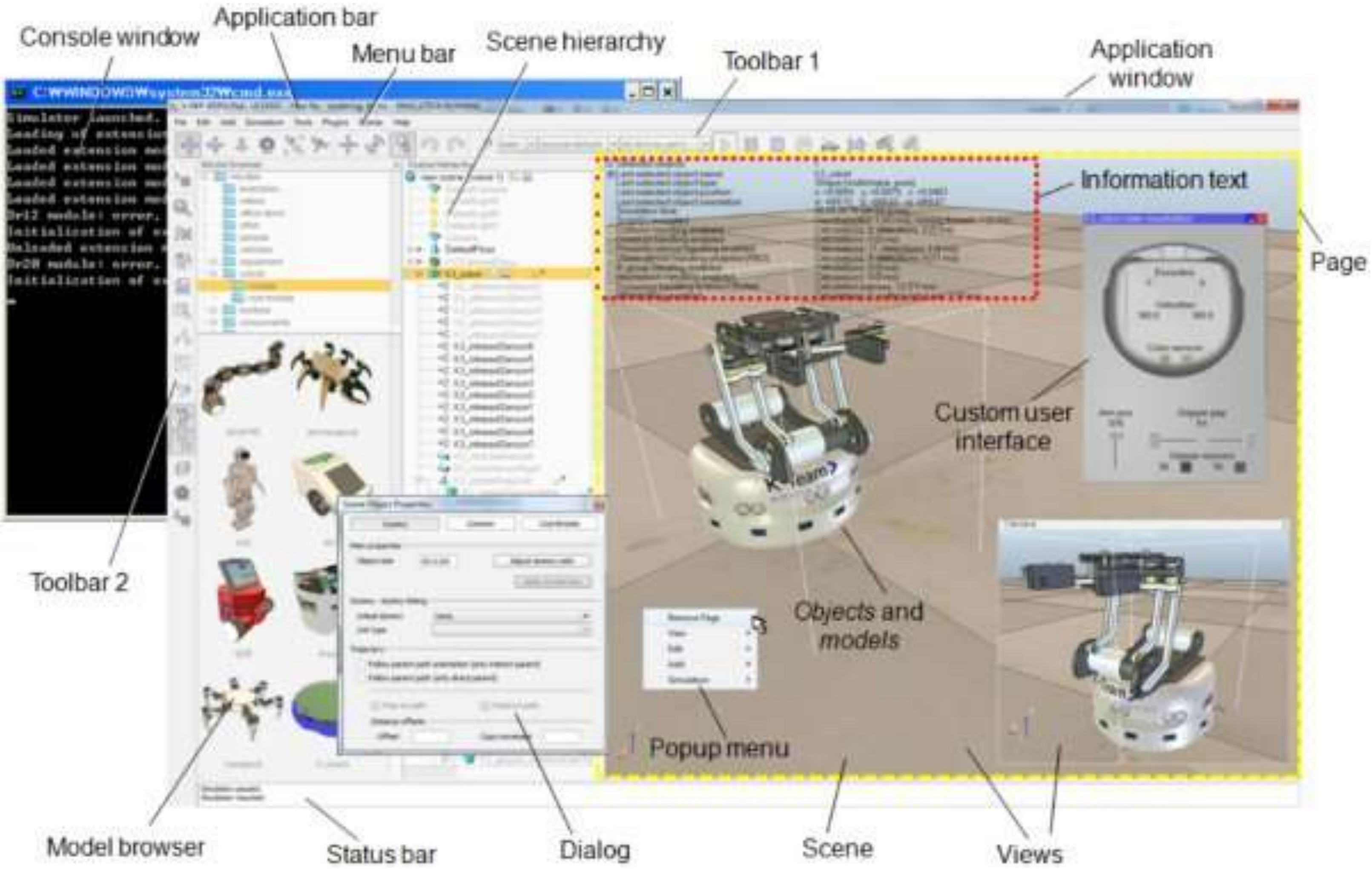
ITMO UNIVERSITY

Teacher: Islam Bzhikhatlov

Virtual Robotics Experimentation Platform

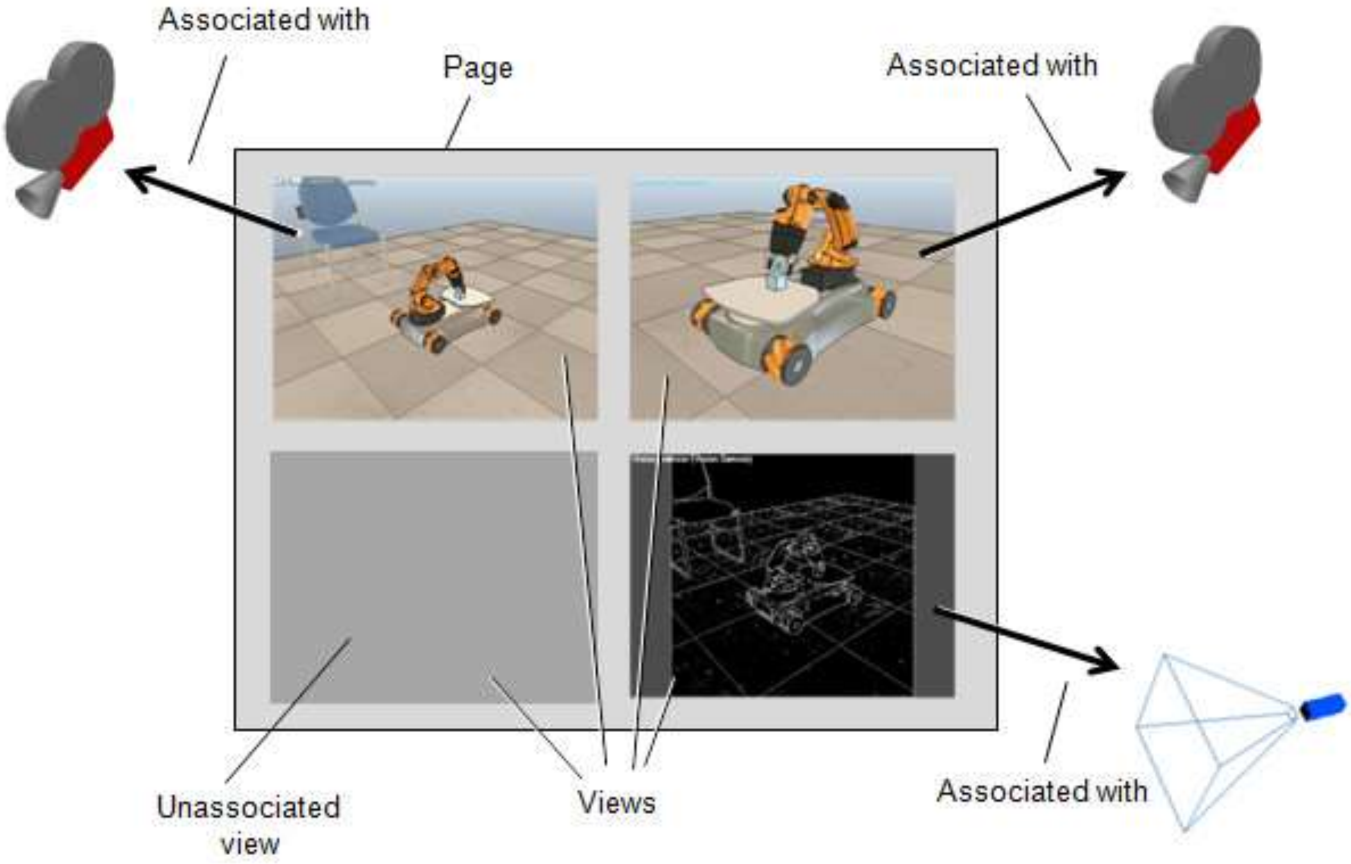
V-REP is the robot simulator with integrated development environment, is based on a distributed control architecture: each object/model can be individually controlled via an embedded script, a plugin, a ROS node, a remote API client, or a custom solution. This makes V-REP very versatile and ideal for multi-robot applications. Controllers can be written in C/C++, Python, Java, Lua, Matlab or Octave.

User Interface

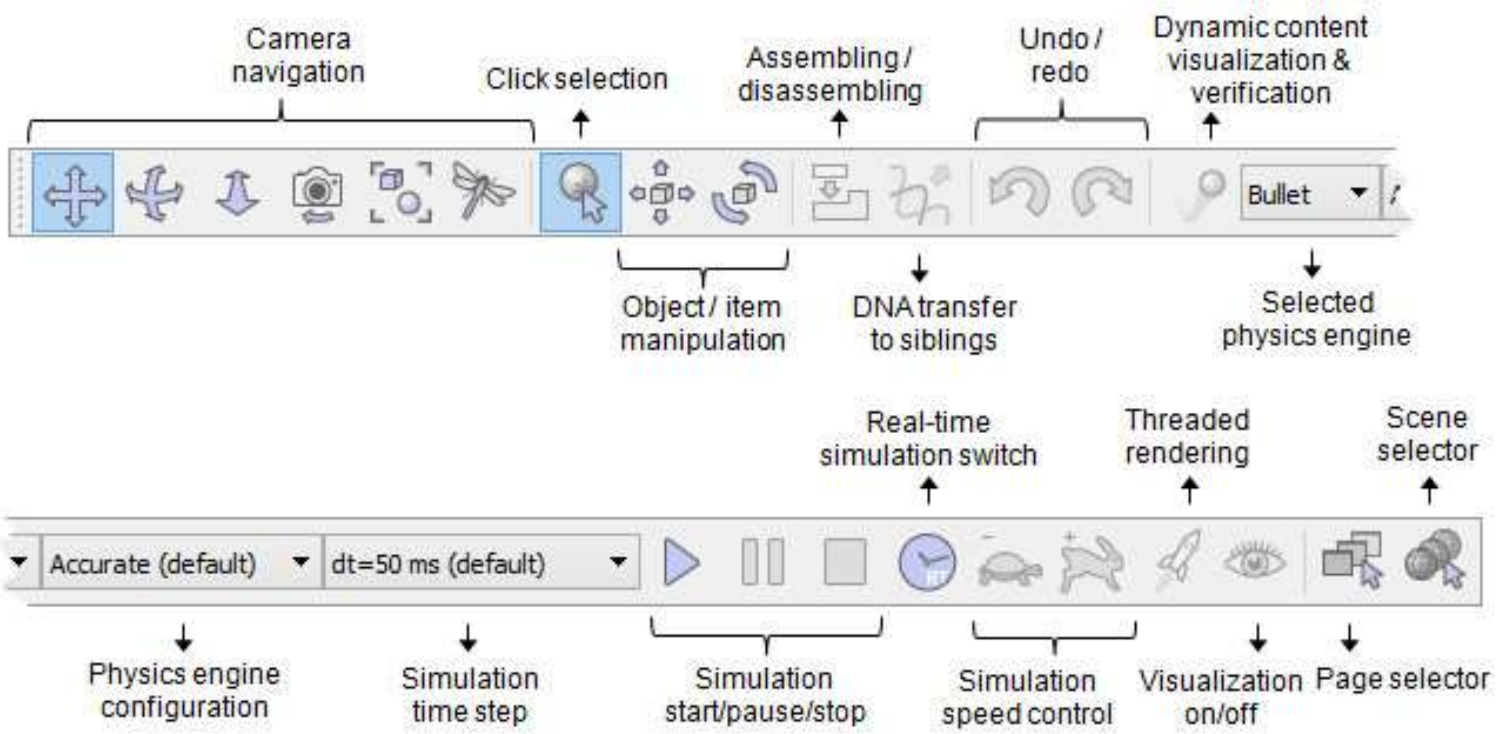


```
Simulator launched.
Loading of extensio
Loaded extension mod
Loaded extension mod
Loaded extension mod
Loaded extension mod
Dr12 module: error.
Initialization of ex
Unloaded extension m
Dr20 module: error.
Initialization of ex
_
```

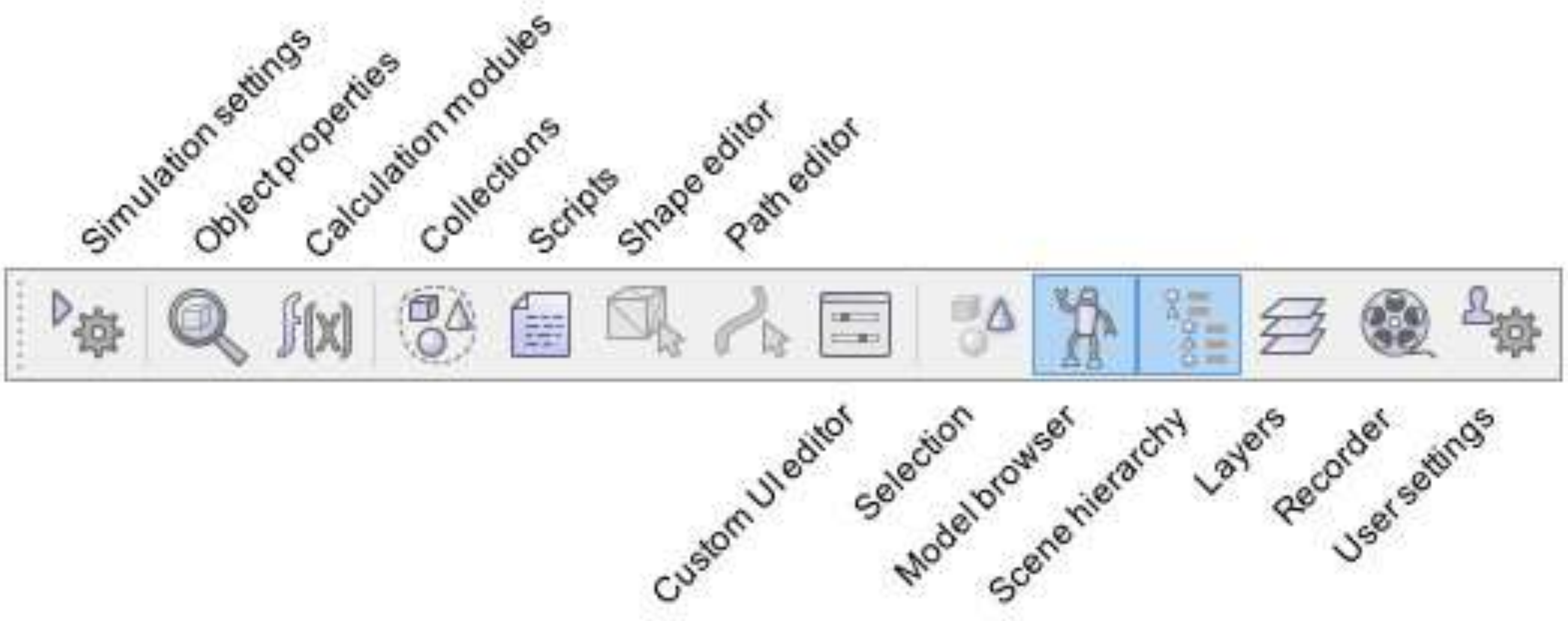
Pages and views



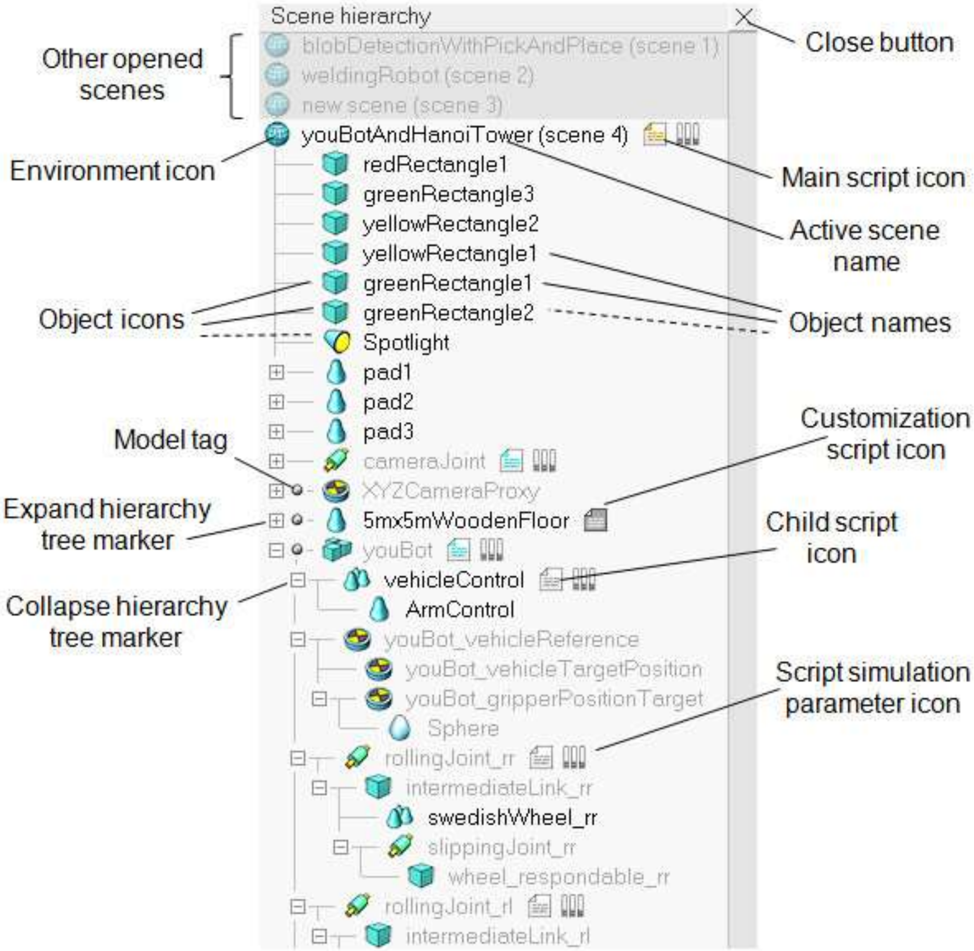
Main Toolbar



Secondary Toolbar



The components of Simulation

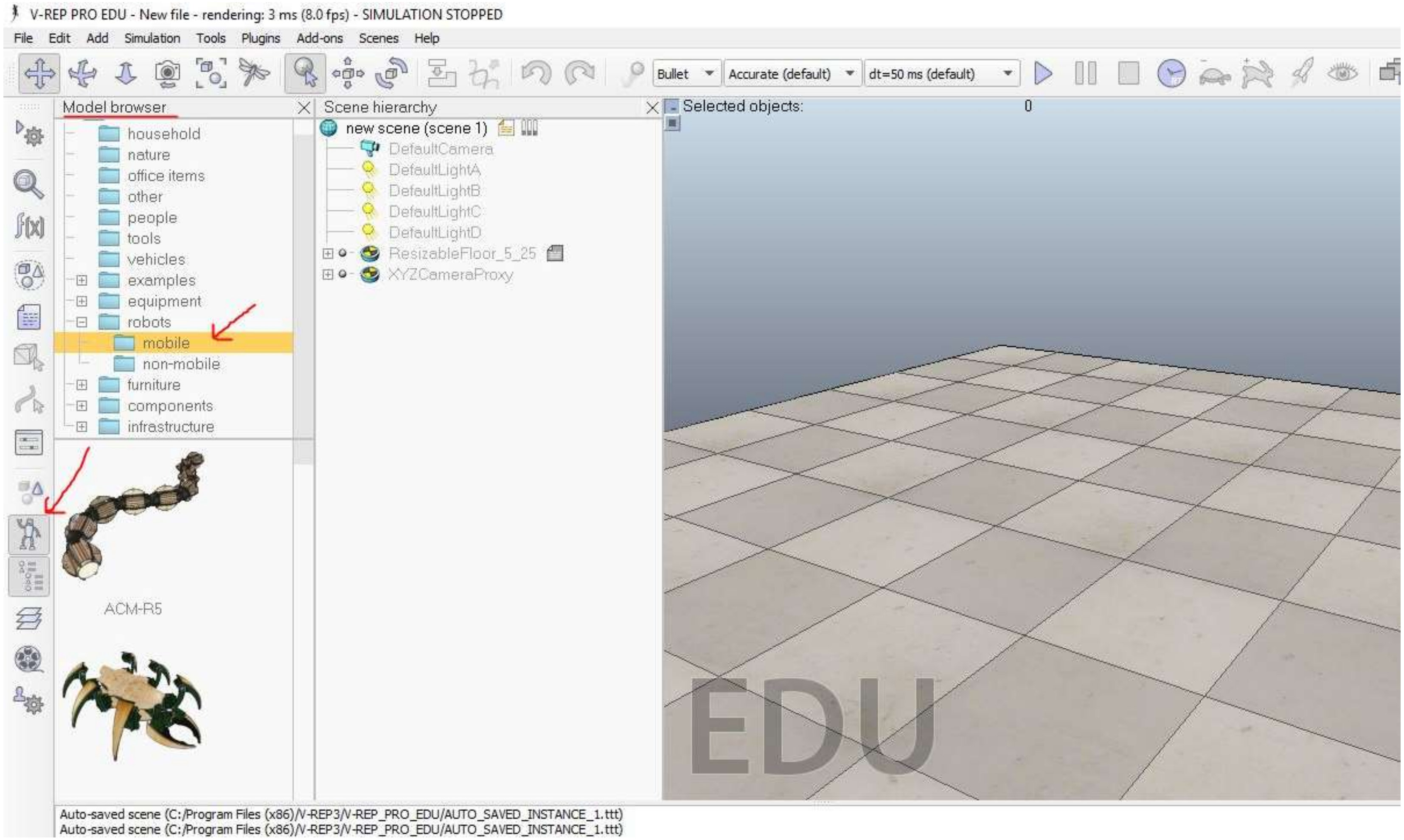


Time for some fun

In V-REP we have some prepared robots, which can be considered and analyzed by students.

Use model browser!

Model Browser



V-REP PRO EDU - New file - rendering: 3 ms (8.0 fps) - SIMULATION STOPPED

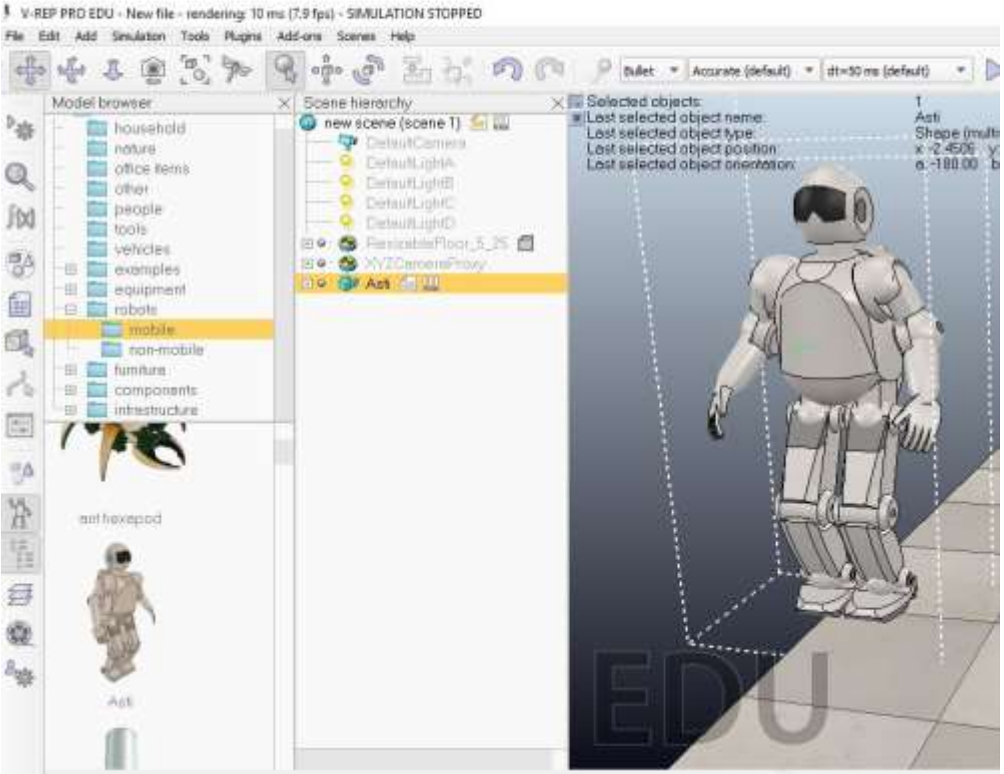
Bullet
Accurate (default)
dt=50 ms (default)

Selected objects: 0

```
Auto-saved scene (C:/Program Files (x86)/V-REP3/V-REP_PRO_EDU/AUTO_SAVED_INSTANCE_1.ttt)
Auto-saved scene (C:/Program Files (x86)/V-REP3/V-REP_PRO_EDU/AUTO_SAVED_INSTANCE_1.ttt)
```

Let’s try to use prepared models

Drag and drop to our environment the mobile robot named ASTI and start the simulation.

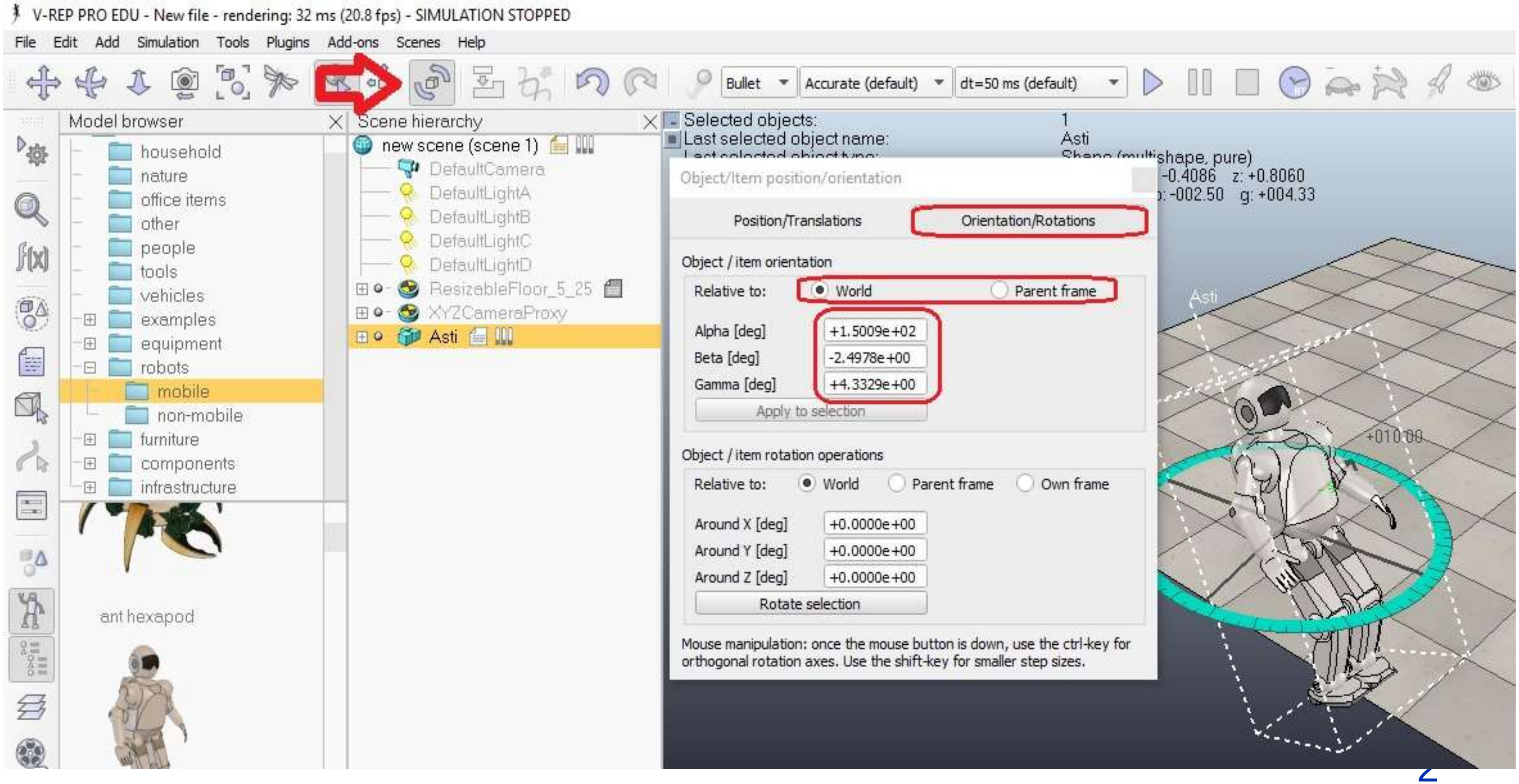


V-REP PRO EDU - New file - rendering: 10 ms (7.9 fps) - SIMULATION STOPPED

Bullet | Accurate (default) | dt=50 ms (default)

Selected objects: 1
Last selected object name: Asti
Last selected object type: Shape (multi
Last selected object position: x: -2.4506 y:
Last selected object orientation: a: -180.00 b

Basic tools: Position and Orienation



Selected objects: 1
Last selected object name: Asti
Last selected object type: Shape (multishape, pure)
-0.4086 z: +0.8060
b: -002.50 g: +004.33

Object/Item position/orientation

Position/Translations | Orientation/Rotations

Object / item orientation

Relative to: ● World ○ Parent frame

Alpha [deg] +1.5009e+02
Beta [deg] -2.4978e+00
Gamma [deg] +4.3329e+00

Apply to selection

Object / item rotation operations

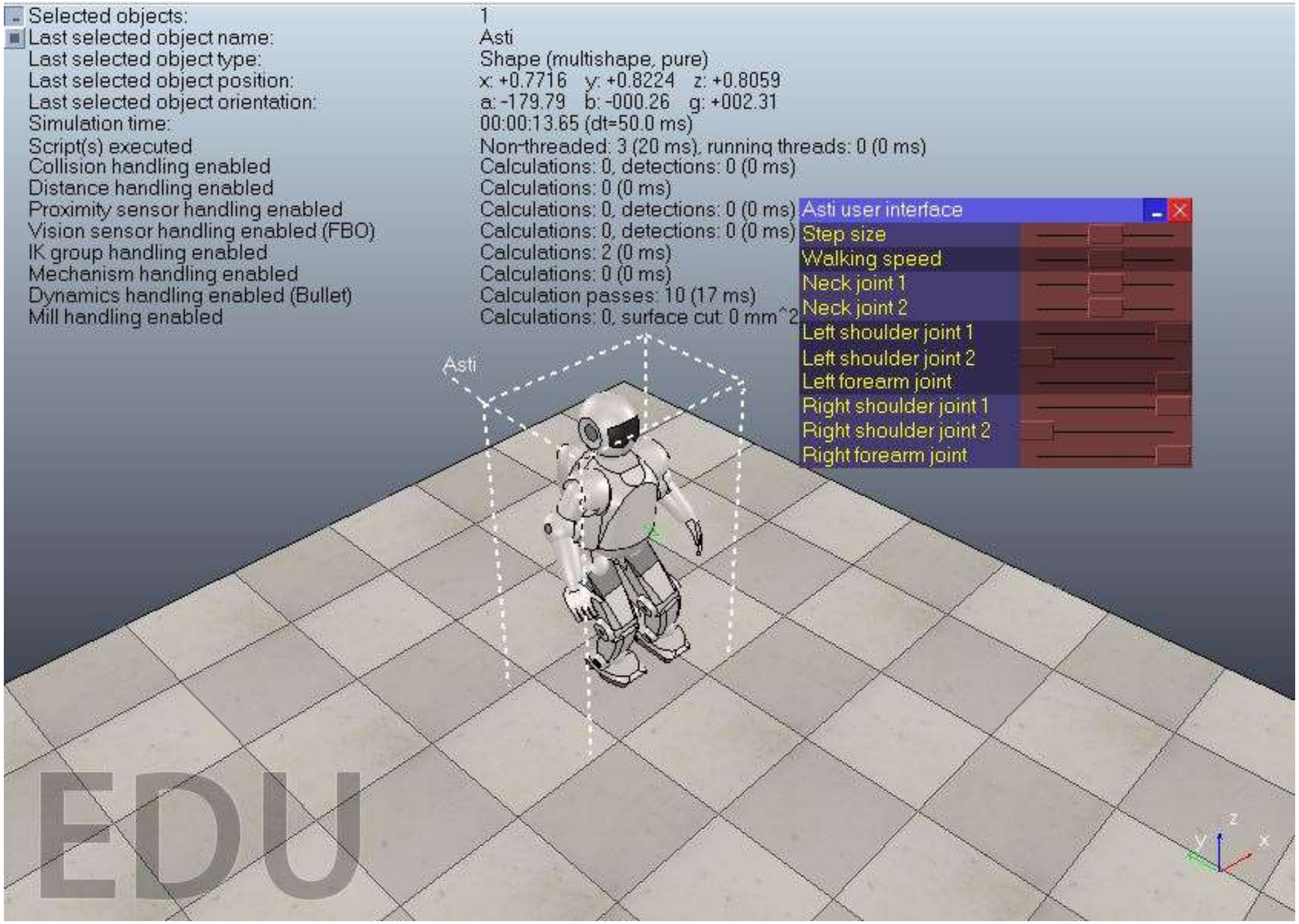
Relative to: ● World ○ Parent frame ○ Own frame

Around X [deg] +0.0000e+00
Around Y [deg] +0.0000e+00
Around Z [deg] +0.0000e+00

Rotate selection

Mouse manipulation: once the mouse button is down, use the ctrl-key for orthogonal rotation axes. Use the shift-key for smaller step sizes.

Custom user interface

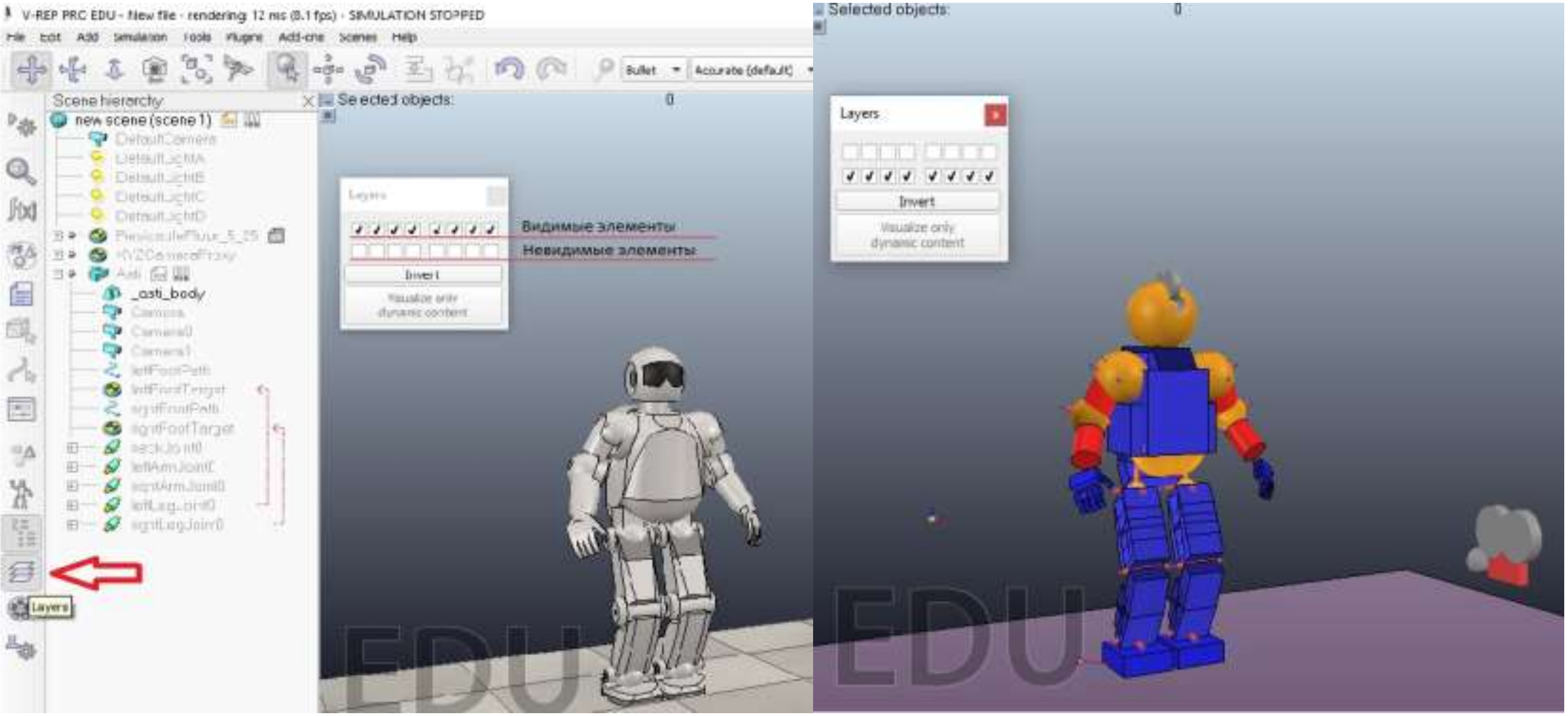


```
Selected objects:                           1
Last selected object name:                  Asti
Last selected object type:                  Shape (multishape, pure)
Last selected object position:              x: +0.7716   y: +0.8224   z: +0.8059
Last selected object orientation:           a: -179.79   b: -000.26   g: +002.31
Simulation time:                            00:00:13.65 (dt=50.0 ms)
Script(s) executed                          Non-threaded: 3 (20 ms), running threads: 0 (0 ms)
Collision handling enabled                  Calculations: 0, detections: 0 (0 ms)
Distance handling enabled                   Calculations: 0 (0 ms)
Proximity sensor handling enabled           Calculations: 0, detections: 0 (0 ms)
Vision sensor handling enabled (FBO)        Calculations: 0, detections: 0 (0 ms)
IK group handling enabled                   Calculations: 2 (0 ms)
Mechanism handling enabled                  Calculations: 0 (0 ms)
Dynamics handling enabled (Bullet)          Calculation passes: 10 (17 ms)
Mill handling enabled                       Calculations: 0, surface cut: 0 mm^2
```

Layers

We have 10 layers, our components can be located in any layer.

Pay Attention! The layers allow only to change the visibility!



Composing of simulation structures

In V-REP connection of solid parts is organized by model hierarchy.

- We need to determine base element
- Another rigged connected part or connected parts should be associated to base element (or base-child components).

Main divisions in V-REP

- ✓ Visualization
- ✓ Physics
- ✓ Scripts
- ✓ Properties

Visualization in V-REP

We have special types of ”Shapes” for visualization. The modeling of dynamic properties based on “shapes” also.

Shapes for visualization is not mandatory for simulation.

Physics

In V-REP standard package we have physic engines:

- Bullet
- ODE
- Vortex
- Newton

This physic engines allows to calculate all the properties of our system in real time with high accuracy. The precision is not 100%.

More information about V-REP

- ✓ Sensors
- ✓ Graphs (for data output in real time).
- ✓ Custom user interface.
- ✓ License for education purposes.

Scripts

There are two types of scripts in V_REP:

- ✓ Non-threaded child scripts
- ✓ Threaded

We can use as much scripts as we need. The scripts should be associated to some v-rep component.

Non-threaded child scripts

This type of script is recommended when it is appropriate for simulation.

“Non-threaded” means that every time they are called, they should perform some task and then return control.

If control is not returned, then the whole simulation halts(stop).

Non-threaded child scripts are executed in a cascaded way: child scripts are executed starting with root objects (or parentless objects), and ending with leaf objects (or childless objects).

Non-threaded child scripts example

This type of script contains 4 parts.

```
if (sim_call_type==sim_childscriptcall_initialization) then
  --  initialization script
end
if (sim_call_type==sim_childscriptcall_actuation) then
   -- executing main script
end
if (sim_call_type==sim_childscriptcall_sensing) then
   --
end
if (sim_call_type==sim_childscriptcall_cleanup) then
   --
end
```

Non-threaded child scripts example

```
function sysCall_init()
 --
end
function sysCall_actuation()
 --
end
function sysCall_sensing()
    --
end
function sysCall_cleanup()
    --
end
```

Threaded

This kind of script «execute just once».

When a threaded child script's execution is still underway, it will not be launched a second time.

Threaded child scripts have several weaknesses compared to non-threaded child scripts if not programmed appropriately: they are more resource-intensive, they can waste some processing time, and they can be a little bit less responsive to a simulation stop command.

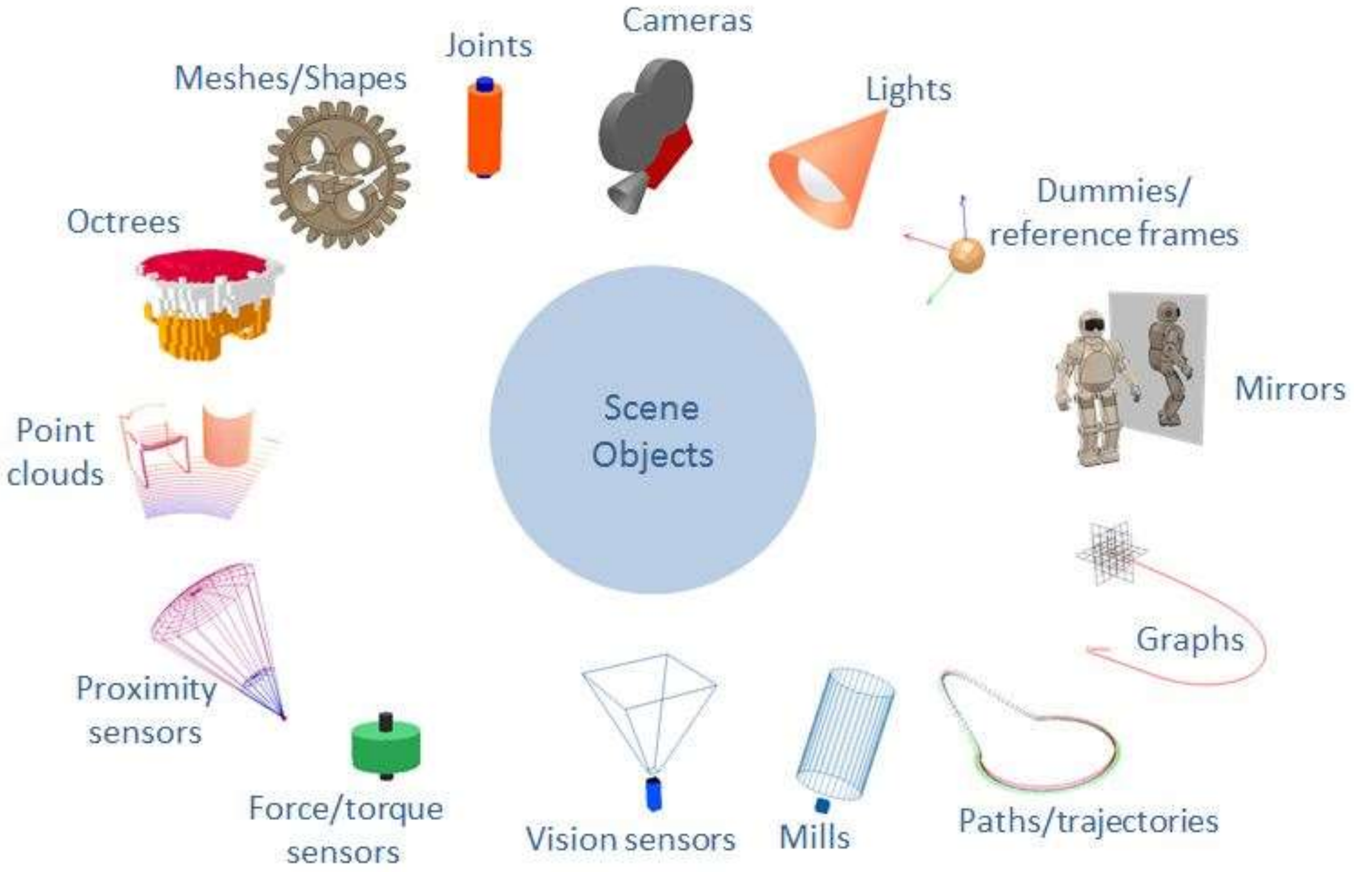
Threaded script example

```
threadFunction=function()
 while
    -- main code
simGetSimulationState()~=sim_simulation_advancing_abouttostop do
    -- this loop wastes precious computation time since we should only read
    -- new values when the simulation time has changed (i.e. in next simulation
    --- step).
end
 -- Put some initialization code here:
sensorHandleFront=simGetObjectHandle("DoorSensorFront")
-- Here we execute the regular thread code:
 -- Put some clean-up code here:
```

Scenes and models

Scenes and models are V-REP's main simulation elements. A model is a sub-element of a scene, clearly marked as model. A scene may contain any number of models.

Scene objects



ITMO UNIVERSITY

Thank you for attention!

bia@corp.ifmo.ru