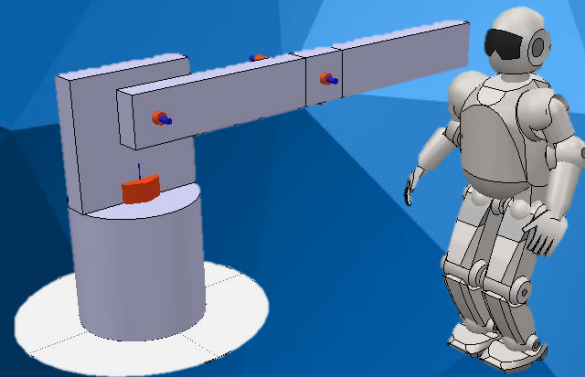




УНИВЕРСИТЕТ ИТМО

Моделирование роботов в виртуальной среде CoppeliaSim (V-REP)

Бжихатлов Ислам Асланович
bia@itmo.ru



Структура МК

- Зачем нужны симуляторы
- Обзор ПО и возможностей CoppeliaSim
- Знакомство с интерфейсом
- Демонстрация: декомпозиция готовой модели
- Практическая часть 1: Разработка модели манипулятора
- Практическая часть 2: Разработка скрипта управления манипулятором

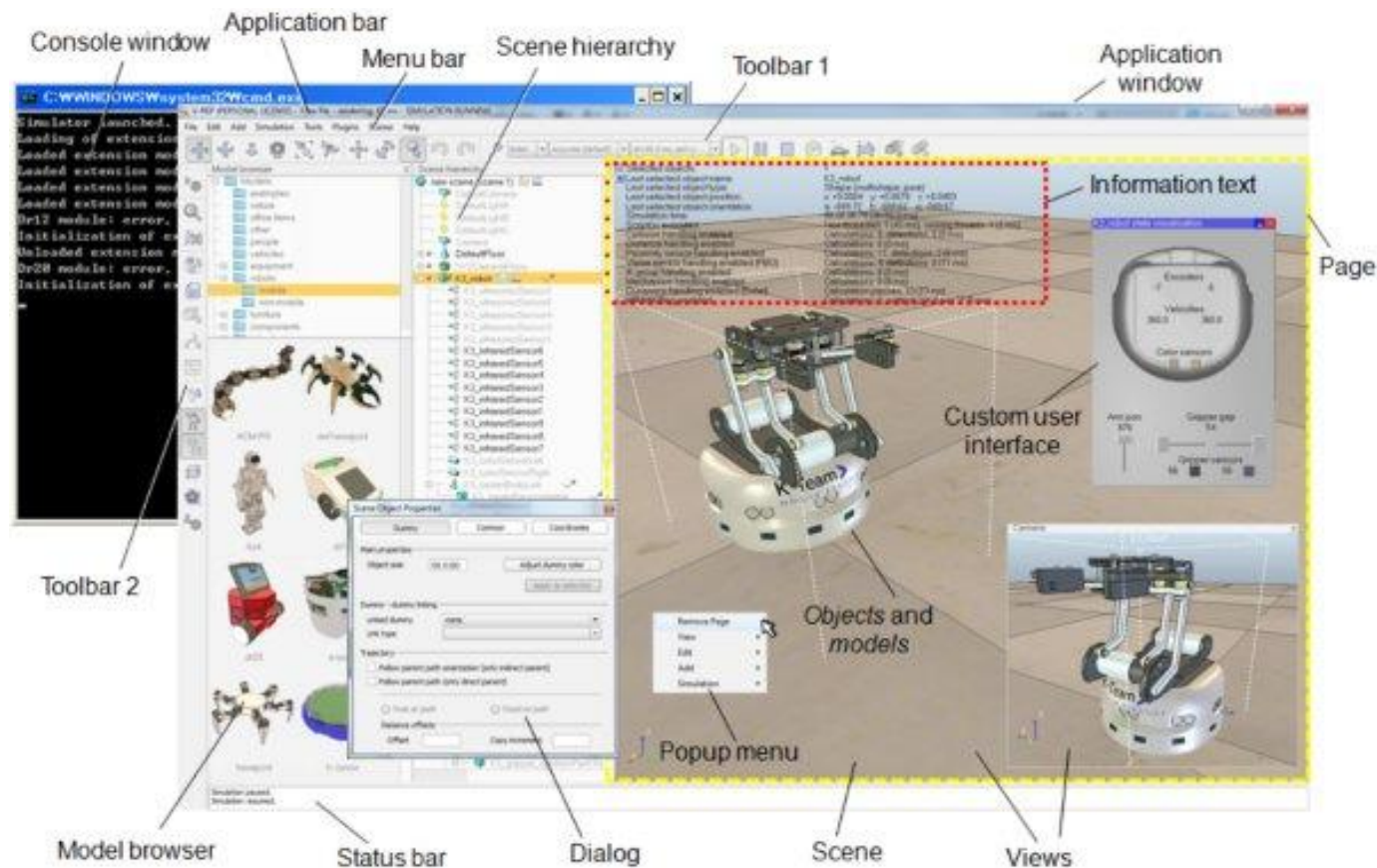
Что понадобится?

- ✓ ПК с любой операционной системой
- ✓ Установленный софт: CopelliaSim Pro Edu
(можно скачать из интернета)
- ✓ 2 часа вашего времени

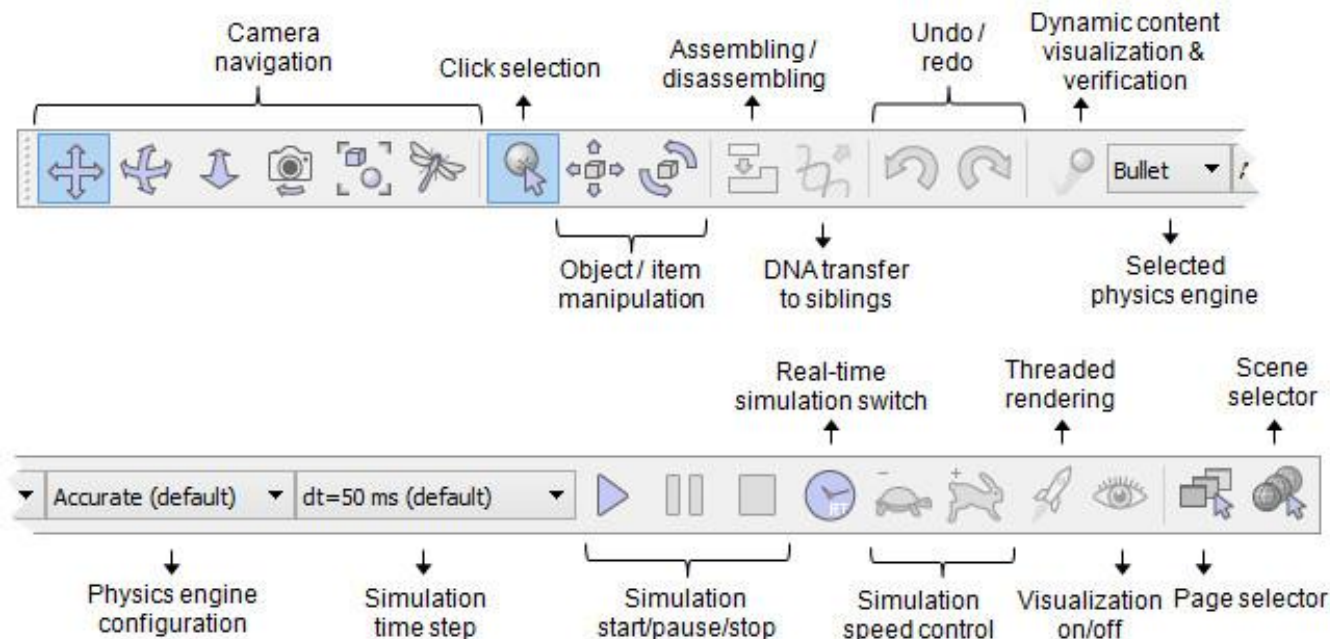
P.S. На слайдах с этой пометкой вы найдете инструкции для выполнения на компьютере



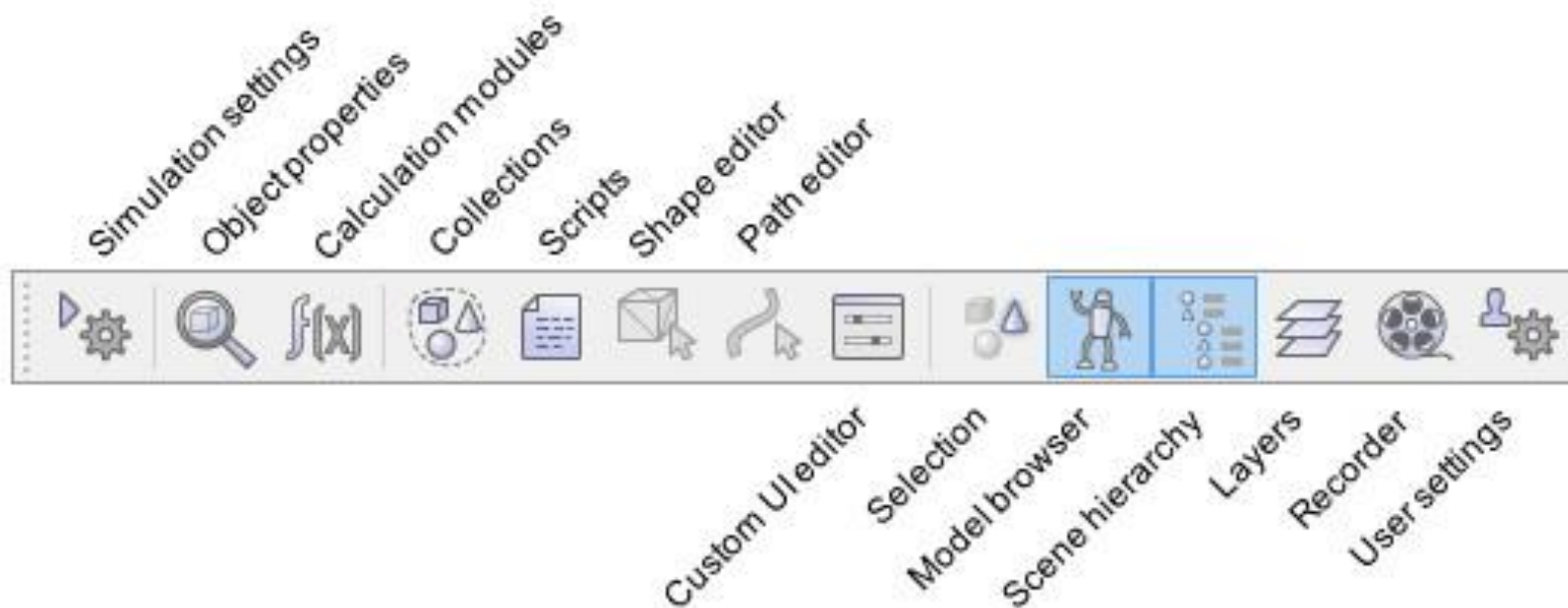
Интерфейс



Интерфейс: верхняя панель инструментов

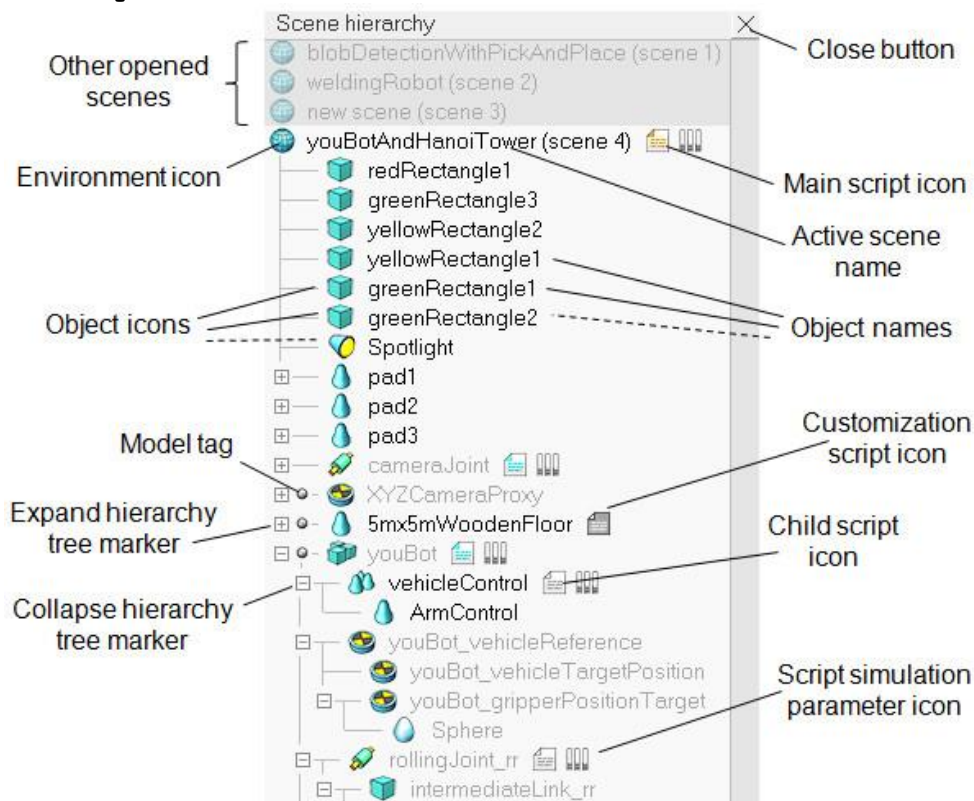


Интерфейс: боковая панель инструментов

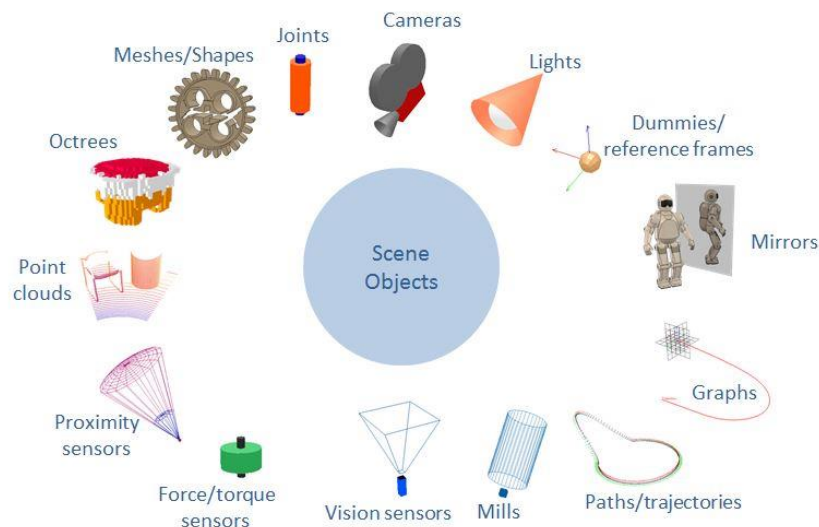


Интерфейс: иерархия сцены

«Иерархия сцены»
определяет структуру
модели, взаимосвязь всех
объектов и также
обеспечивает легкую
навигацию по объектам.



Типы объектов



Часто используются:

- ✓ Шарниры
- ✓ Формы (тела)
- ✓ Сенсоры
- ✓ Графики

Управление: скрипты

- ✓ Для управления объектами в V-REP используются скрипты (например, для управления приводом).
- ✓ Также можно задать постоянное управление приводом без использования скрипта (нельзя менять параметры привода во время симуляции).
- ✓ Выбрать объект, **Add** → **Associated child script** → **Non-Threaded**.

```

1  -- DO NOT WRITE CODE OUTSIDE OF THE if-then-end SECTIONS BELOW!! (unless
2
3  if (sim_call_type==sim_childscriptcall_initialization) then
4
5      -- Put some initialization code here
6
7
8  end
9
10
11 if (sim_call_type==sim_childscriptcall_actuation) then
12
13     -- Put your main ACTUATION code here
14
15     -- For example:
16     --
17     -- local position=simGetObjectPosition(handle,-1)
18     -- position[1]=position[1]+0.001
19     -- simSetObjectPosition(handle,-1,position)
20
21 end
22
23
24 if (sim_call_type==sim_childscriptcall_sensing) then
25
26     -- Put your main SENSING code here
27
28 end
29
30
31 if (sim_call_type==sim_childscriptcall_cleanup) then
32
33     -- Put some restoration code here
34
35 end

```

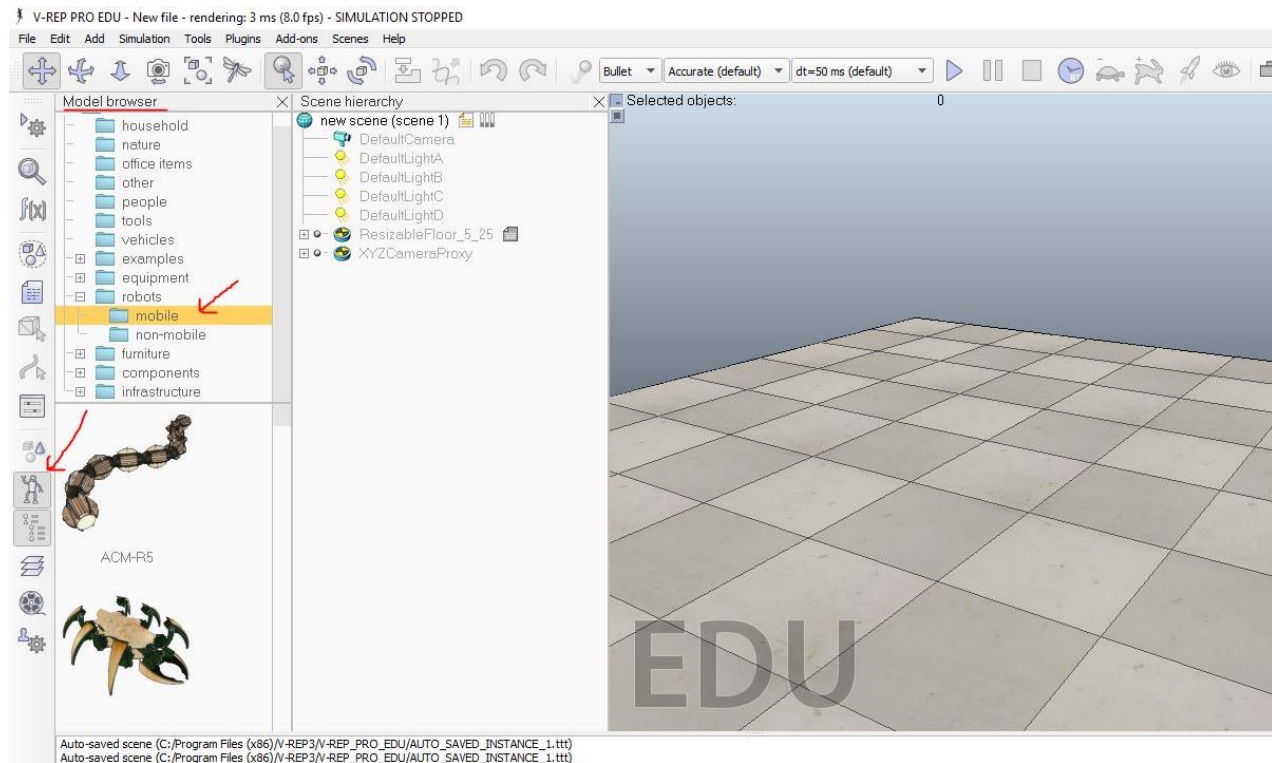
1. Блок инициализации

2. Блок управляющего кода объектов

3. Блок управляющего кода сенсоров

4. Блок завершения симуляции

Браузер моделей

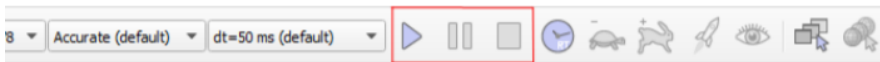


Часто используемые инструменты

- ✓ «Положение» и «ориентация»



- ✓ Запуск и остановка симуляции



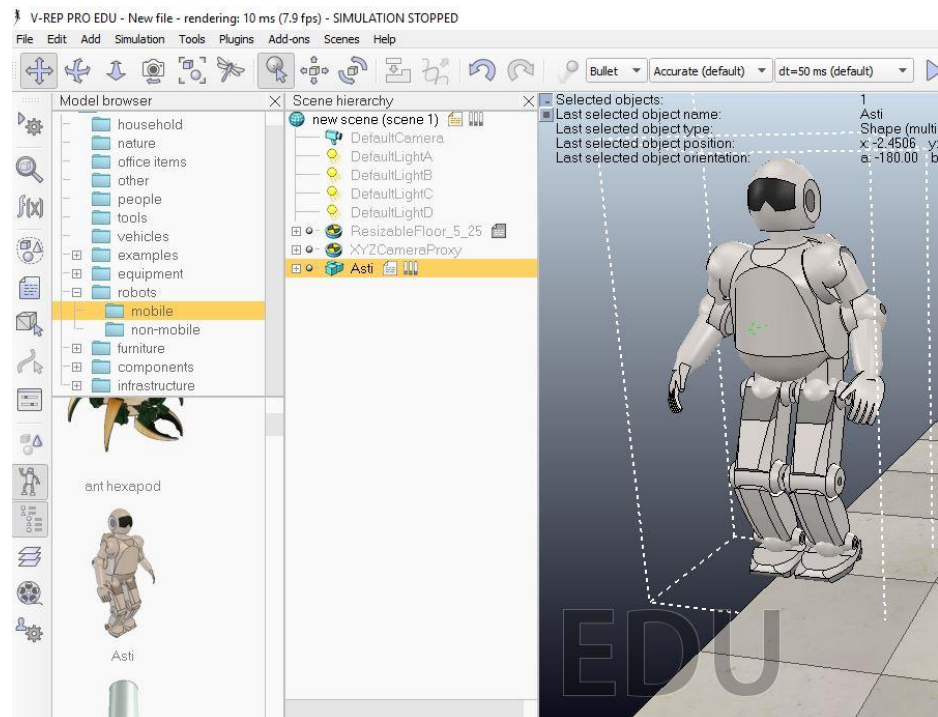
- ✓ Свойства объекта



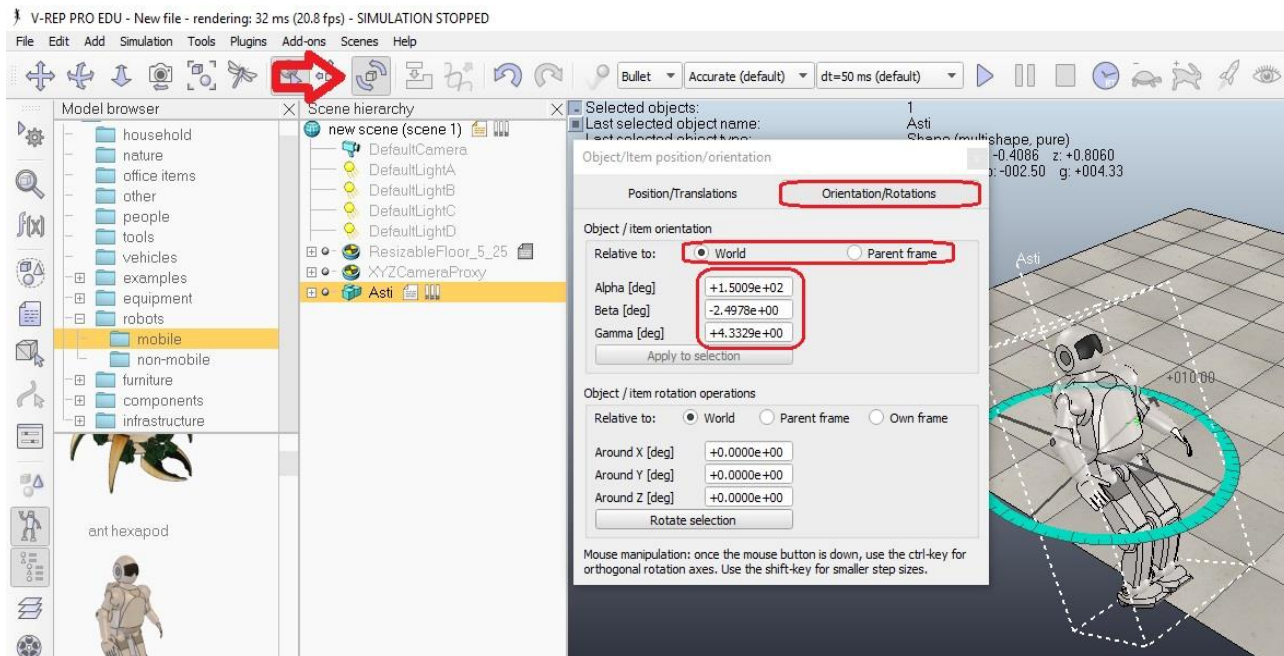
Использование готовой модели



- ✓ В окне готовых моделей находим робота Asti и перетаскиваем его в окно визуализации сценария.



Позиционирование модели в окне визуализации



Запуск симуляции



Bullet 2.78

Accurate (default)

dt=50 ms (default)

▶

⏸

⏹

🔄

🏠

🐘

🚀

👁

🖱

🔍

hape_pure)

x: +0.7716 y: +0.8224 z: +0.8059

a: -179.79 b: -000.26 g: +002.31

00:00:13.65 (dt=50.0 ms)

Non-threaded: 3 (20 ms), running threads: 0 (0 ms)

Calculations: 0, detections: 0 (0 ms)

Calculations: 0 (0 ms)

Calculations: 0, detections: 0 (0 ms)

Calculations: 0, detections: 0 (0 ms)

Calculations: 2 (0 ms)

Calculations: 0 (0 ms)

Calculation passes: 10 (17 ms)

Calculations: 0, surface cut: 0 mm²

Last selected object position:

Last selected object orientation:

Simulation time:

Script(s) executed

Collision handling enabled

Distance handling enabled

Proximity sensor handling enabled

Vision sensor handling enabled (FBO)

IK group handling enabled

Mechanism handling enabled

Dynamics handling enabled (Bullet)

Mill handling enabled

Asti user interface

Step size

Walking speed

Neck joint 1

Neck joint 2

Left shoulder joint 1

Left shoulder joint 2

Left forearm joint

Right shoulder joint 1

Right shoulder joint 2

Right forearm joint

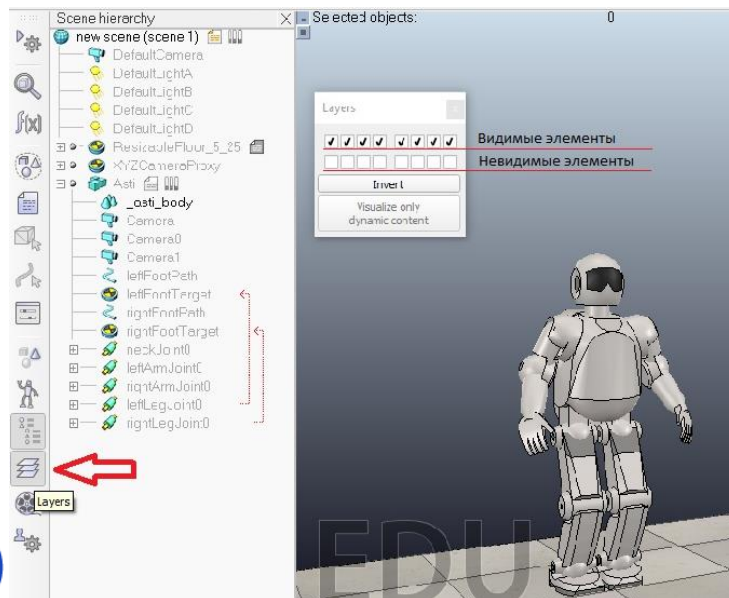
Asti

EDU

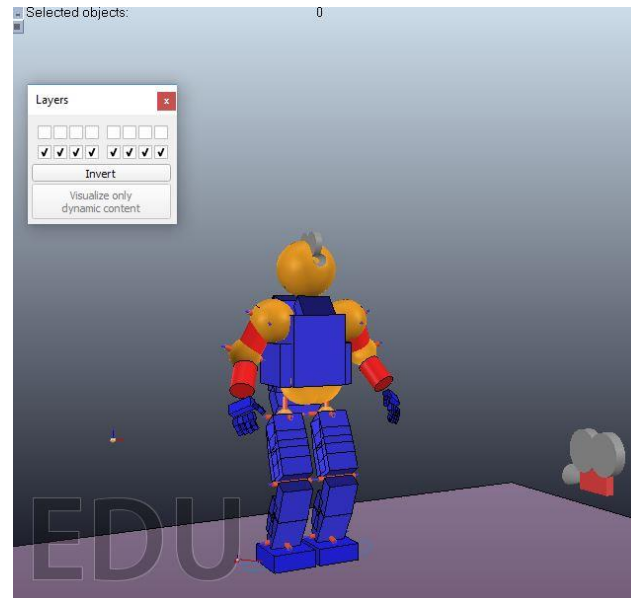
Управление слоями отображения



Визуализация



Моделирование



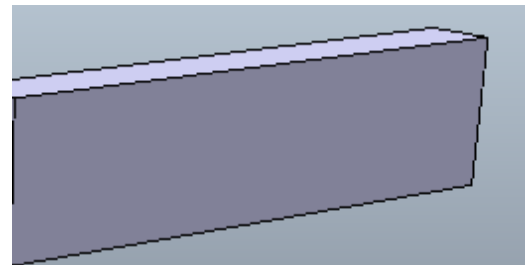
Как создать модель(твердые тела)?

✓ Используя примитивы

- Долгий процесс
- Оптимизированная геометрия
- Сложности с позиционированием

✓ Используя CAD систему

- Импортированную геометрию необходимо оптимизировать
- В CAD системе проще спроектировать



Как добавить шарниры

Всего 4 типа шарниров

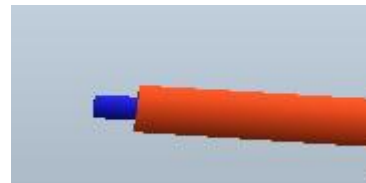
- ✓ **Revolute joints:**
- ✓ **Prismatic joints:**
- ✓ **Screws:**
- ✓ **Spherical joints:**

Шарниры можно добавить через главное меню:

Add → Joint → Revolute.

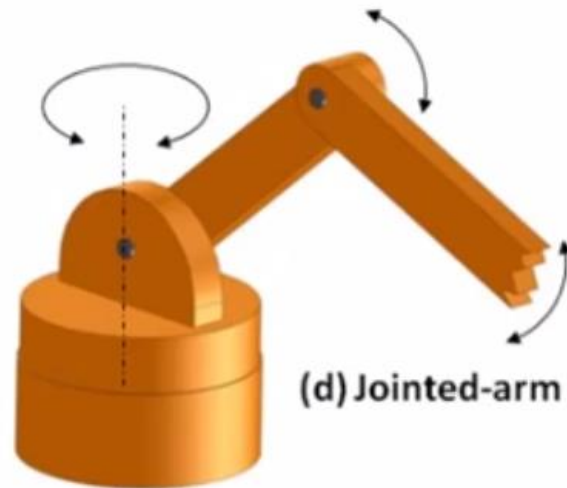
Далее необходимо их правильно спозиционировать

И задать их положение в общей структуре модели



Практическая часть 1

- ✓ Создание звеньев манипулятора
- ✓ Задать их положение и ориентацию
- ✓ Добавить шарниры
- ✓ Создать сборку (задать структуру механизма)





Создание звеньев манипулятора

✓ Add → Primitive shape → Cylinder

Primitive cylinder

X-size [m]	5.0000e-01	X-subdivisions	
Y-size [m]	5.0000e-01	Y-subdivisions	
Z-size [m]	5.0000e-01	Z-subdivisions	
Sides	32	Faces	0
Disc subdivisions	0	☒ Smooth shaded	
☐ Open ends		☐ Cone	

☒ Create dynamic and responsible shape
☒ Create pure shape Material density [kg/m³]
☐ Shape has neg. volume Inside scaling factor

✓ Add → Primitive shape → Cuboid

Primitive cuboid

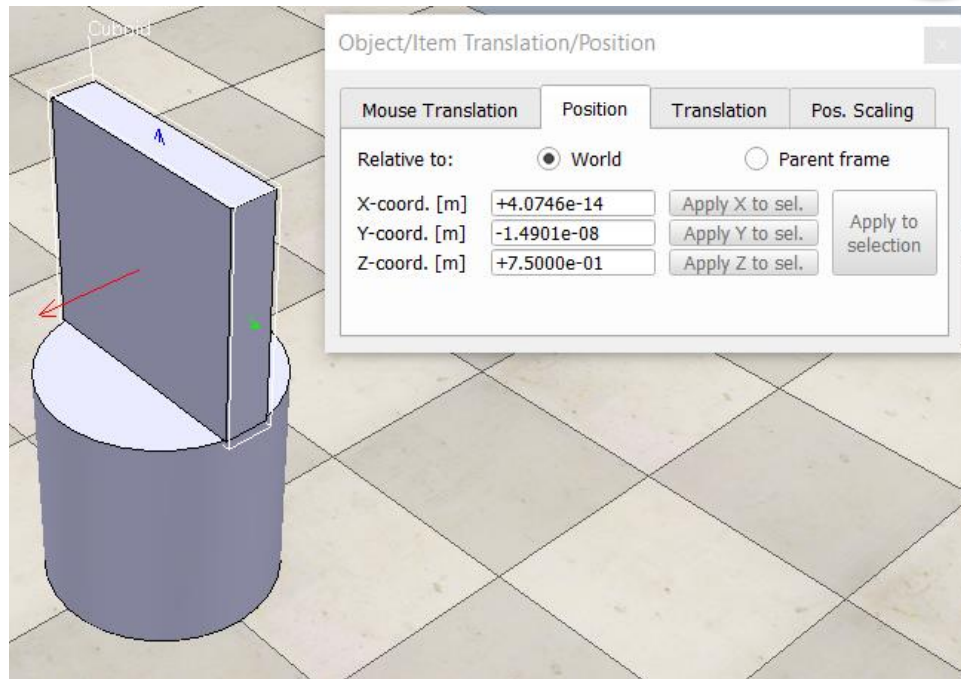
X-size [m]	1.0000e-01	X-subdivisions	0
Y-size [m]	5.0000e-01	Y-subdivisions	0
Z-size [m]	5.0000e-01	Z-subdivisions	0
Sides		Faces	
Disc subdivisions		☐ Smooth shaded	
☐ Open ends		☐ Cone	

☒ Create dynamic and responsible shape
☒ Create pure shape Material density [kg/m³]
☐ Shape has neg. volume Inside scaling factor

Задаем положение звеньев манипулятора



- ✓ Выбрать прямоугольник в иерархии сцены
- ✓ Активировать инструмент «Object/Item shift»





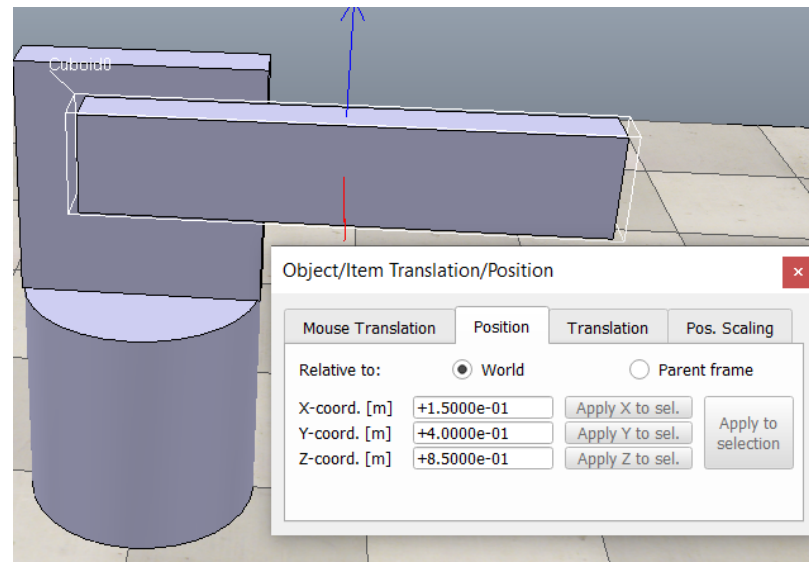
Аналогичные операции...

✓ Add → Primitive shape → Cuboid

Primitive cuboid

X-size [m]	1.0000e-01	X-subdivisions	0
Y-size [m]	1.0000e+00	Y-subdivisions	0
Z-size [m]	2.0000e-01	Z-subdivisions	0
Sides		Faces	
Disc subdivisions		☐ Smooth shaded	
☐ Open ends		☐ Cone	
☒ Create dynamic and responsible shape ☒ Create pure shape Material density [kg/m ³] 1000.0 ☐ Shape has neg. volume Inside scaling factor 0.500			
		OK	Cancel

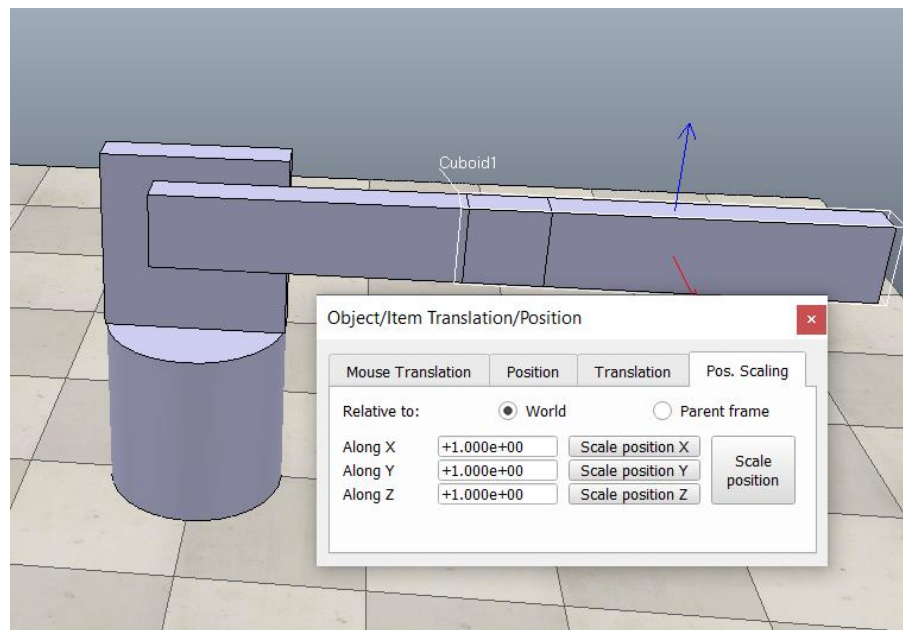
Задаем желаемое положение





Аналогичные операции...

- ✓ Копируем последнее звено и задаем положение.



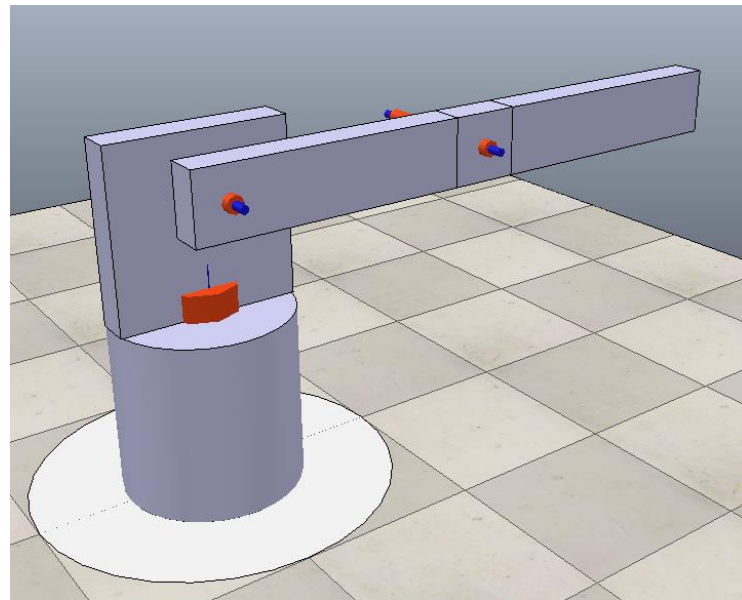


Добавляем шарниры

✓ Add → Joint → Revolute

Задаем желаемое положение для объектов типа «joint».

!!! Также переименуем все файлы, чтобы легко работать с иерархией сцены.



Задаем параметры объектов

- ✓ Выбираем объект и открываем свойства.
- ✓ Object/item properties → Show dynamic properties dialog.
- ✓ Устанавливаем динамические свойства
- ✓ Повторить для остальных звеньев.



Rigid Body Dynamic Properties
×

☒ Body is responsible

Local responsible mask

☒ ☒ ☒ ☒ ☐ ☐ ☐

Global responsible mask

☒ ☒ ☒ ☒ ☒ ☒ ☒

☒ Body is dynamic

☐ Start in sleep mode

☐ Set to dynamic if gets parent

Mass

Mass [kg]

9.817e+01

Principal moments of inertia / mass

X [m^2]

3.646e-02

Y [m^2]

3.646e-02

Z [m^2]

3.125e-02

Pos./orient. of inertia frame & COM relative to shape frame

X [m]

+0.000e+00

Alpha [deg]

+0.00e+00

Y [m]

+0.000e+00

Beta [deg]

+0.00e+00

Z [m]

+0.000e+00

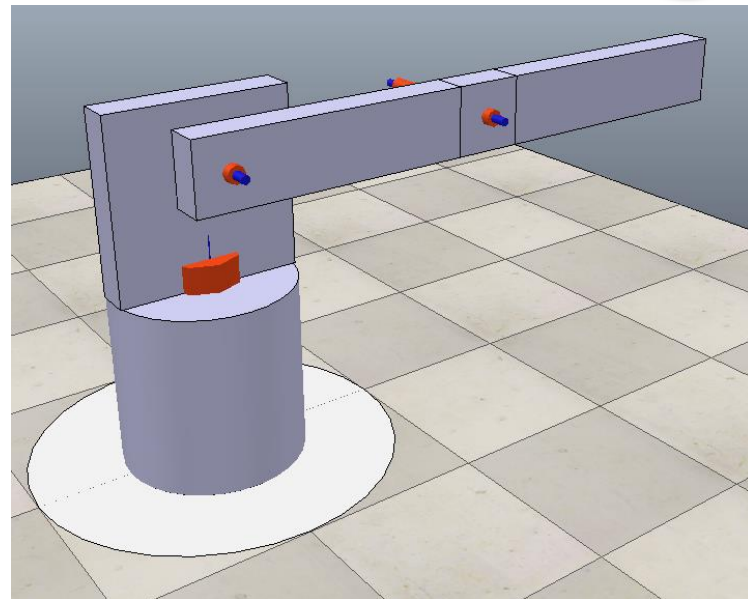
Gamma [deg]

+0.00e+00

ITMO *More than a*
UNIVERSITY

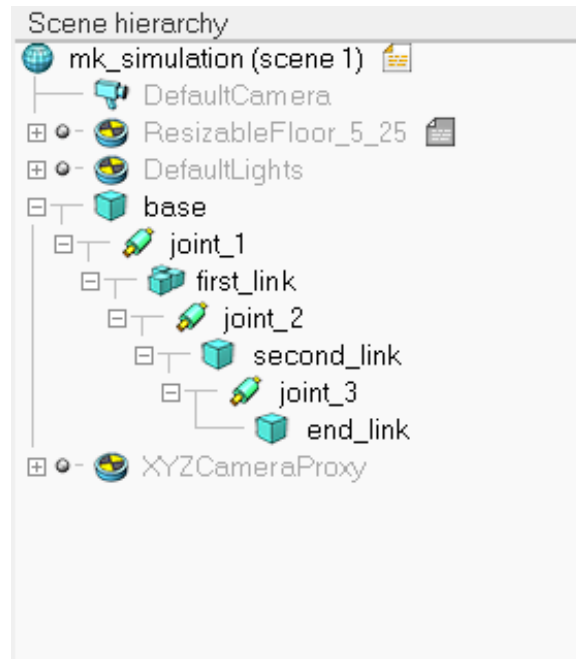
Объединение объектов

- ✓ Выбрать в иерархии сцены два объекта(через CTR+Click), затем Edit→Grouping/merging → Group selected objects.



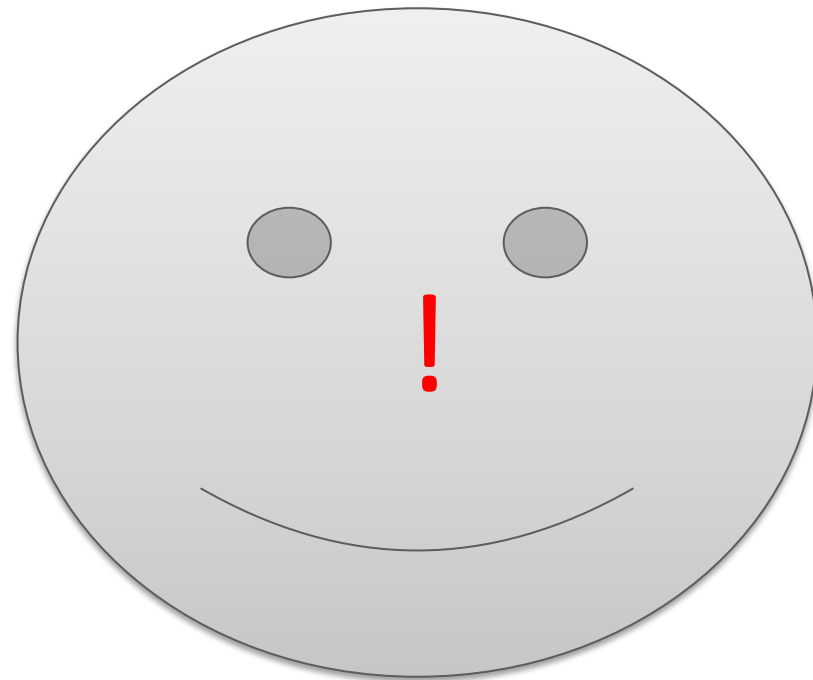
Задаем кинематические связи

- ✓ Удобнее всего это делать по принципу “перетащить и отпустить”.
- ✓ Либо выбрать 2 объекта(через CTRL), и в меню выполнить “make last selected object parent”



Запустим симуляцию

- ✓ В любой момент можно запустить симуляцию и проверить!



Задаем свойства шарниров

- ✓ Выбрать шарнир → свойства → Show dynamic properties →
- ✓ От свойств шарнира зависит, сможем ли мы управлять и каким способом!
- ✓ Увеличить момент!

Joint Dynamic Properties

Motor properties

☒ Motor enabled
 Target velocity [deg/s]
 Maximum torque [N*m]
☒ Lock motor when target velocity is zero

Apply to selection

Control properties

☐ Control loop enabled
 Target position [deg]
 Upper velocity limit [deg/s]
☐ Position control (PID)
 Proportional parameter
 Integral parameter
 Derivative parameter
☐ Spring-damper mode
 Spring constant K [N]
 Damping coefficient C [N*s]

Apply to selection

Практическая часть 2



- ✓ Добавить скрипт управления приводами манипулятора
- ✓ Настроить приводы
- ✓ Написать скрипт управления приводами

```

3 if (sim_call_type==sim_childscriptcall_initialization) then
4   -- Put some initialization code here
5   base=simGetObjectHandle('verh_dyn')
6   jointhandle=simGetObjectHandle('joint_1')
7   jointcurrentpos=simGetJointPosition(jointhandle)
8   simAddStatusbarMessage('position='..jointcurrentpos)

```

```

12 if (sim_call_type==sim_childscriptcall_actuation) then
13   result, distance=simReadProximitySensor(Sensor)
14   if (result>0) and (distance<Detection) and (distance>0) then
15     simSetJointTargetVelocity(LeftHandle, 0.008)
16     simSetJointTargetVelocity(RightHandle, 0.008)
17   else
18     simSetJointTargetVelocity(LeftHandle, 0)
19     simSetJointTargetVelocity(RightHandle, 0)
20   end
21 end

```

Создание скриптов для управления



- ✓ Выбрать «base», затем «Add→Associated child script→Nonthreaded»
- ✓ Открыть редактор можно двойным кликом по пиктограмме или через инструмент «Скрипты».



Non-threaded child script (base)

```

1 function sysCall_init()
2     -- do some initialization here
3 end
4
5 function sysCall_actuation()
6     -- put your actuation code here
7 end
8
9 function sysCall_sensing()
10    -- put your sensing code here
11 end
12
13 function sysCall_cleanup()
14    -- do some clean-up here
15 end
16
17 -- See the user manual or the availa
    
```

Простой пример скрипта



```

1 function sysCall_init()
2     -- do some initialization here
3     joint1_handler = sim.getObjectHandle('joint_1')
4 end
5
6 function sysCall_actuation()
7     -- put your actuation code here
8     sim_time = sim.getSimulationTime()
9     if sim_time > 10 then
10         sim.setJointTargetVelocity(joint1_handler, 5)
11     end
12 end
13
14 function sysCall_sensing()
15     -- put your sensing code here
16 end
17
18 function sysCall_cleanup()
19     -- do some clean-up here
20 end
21
22 -- See the user manual or the available code snippets for additional callback
23

```

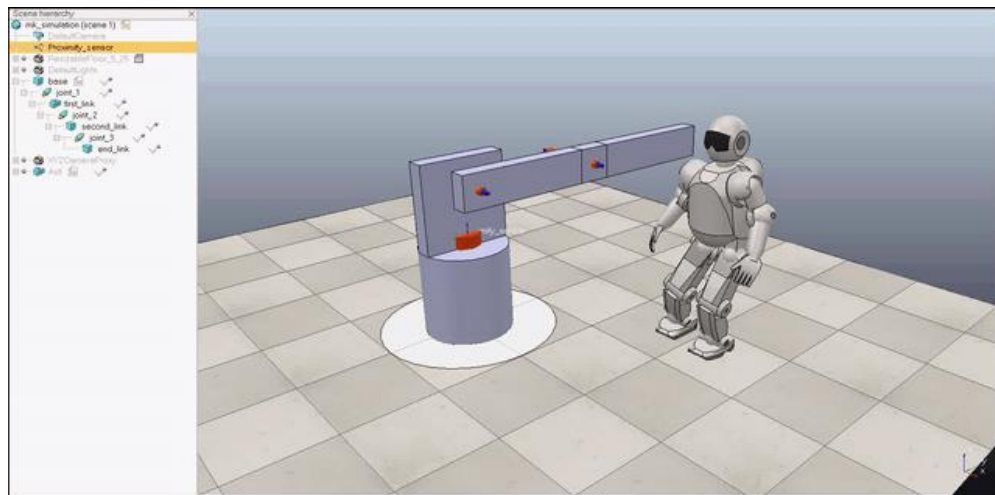

Добавление сенсора

- ✓ Add → Proximity sensor → Row type.
- ✓ Задать положение и ориентацию
- ✓ Использовать данные сенсора в скрипте.

Самостоятельно!

Итоги

- Знакомство с CoppeliaSim(V-REP), основные возможности и особенности
- Знакомство с интерфейсом
- Разработка механической составляющей роботов
- Управление механизмами с использованием сенсорной информации



Спасибо за внимание!

bia@itmo.ru

<https://vk.com/likerobots>

<https://www.youtube.com/c/likerobots>

IT'sMO *re than a*
UNIVERSITY