Experiences with the RoboNewbie Project

Hans-Dieter Burkhard, Monika Domanska

Humboldt University Berlin

Understanding Emerges by Doing

The best way to understand Robotics
is experimenting with robots.

NaoTeamHumboldt with
Robots Nao of French
Company Aldebaran
at Campusparty 2012

Requirements for Education

- Complex Robots (like Humanoids):

  Perception, Motion, Control.
- Holistic View:

  Complete tasks using all robot parts.
- Motivating Application:

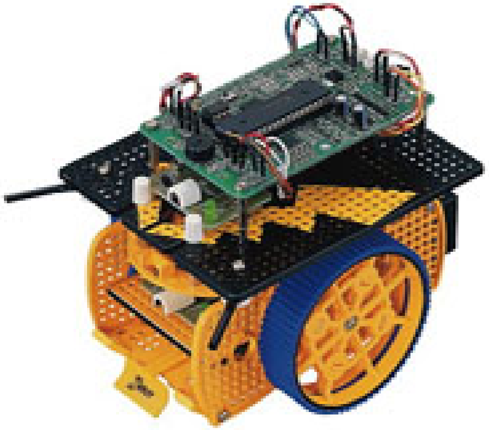
  Known tasks like household, game playing, …
- Scalable Tasks:

  Start with simple tasks (no special knowledge on Robotics).

  Extensions with challenging tasks.
- Minimal costs:

  Resources, Maintenance, Guidance.

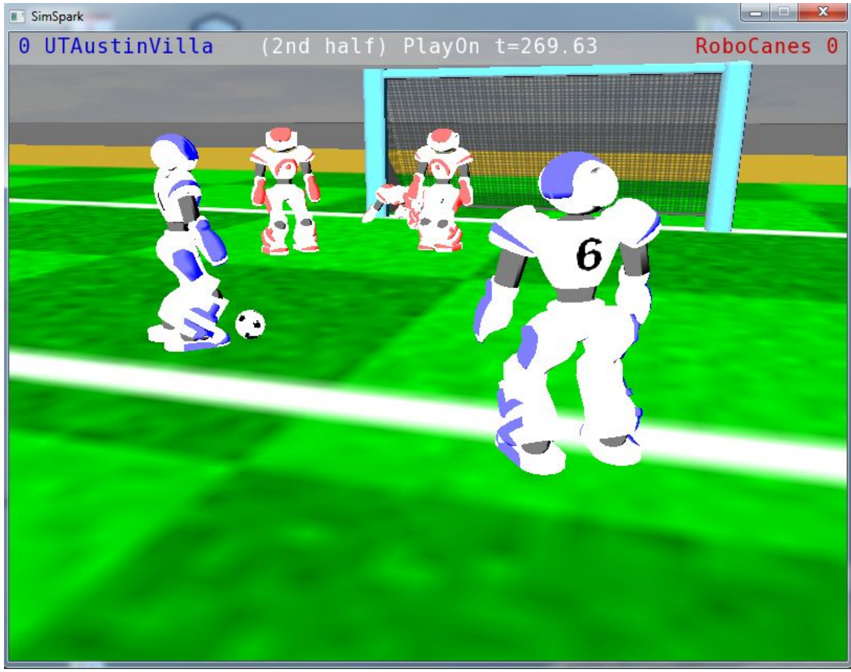
Many Toolkits for Basic Experiments



But more Complex Robots …

… require high efforts for materials, construction, maintenance

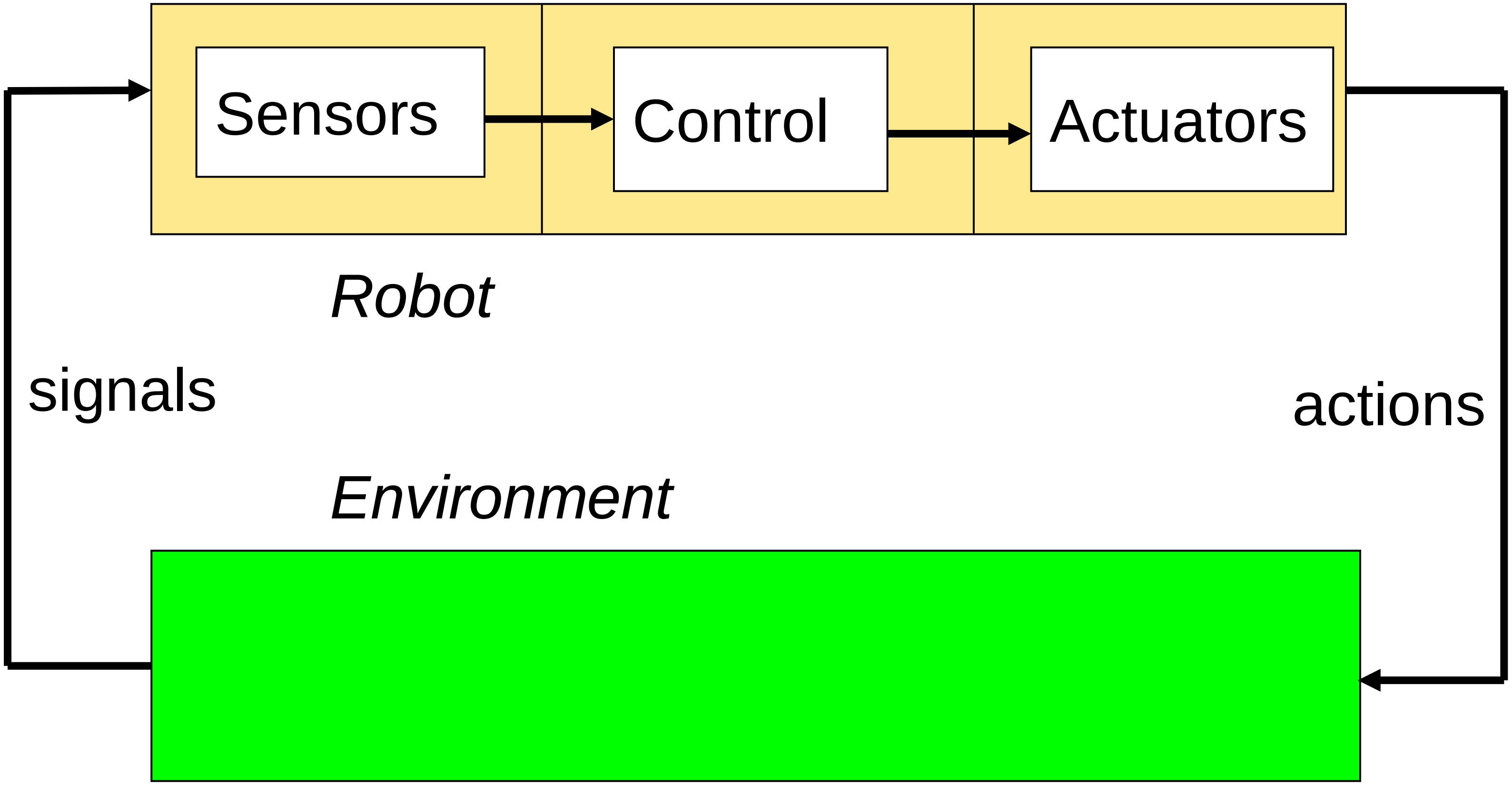
Simulation as Alternative



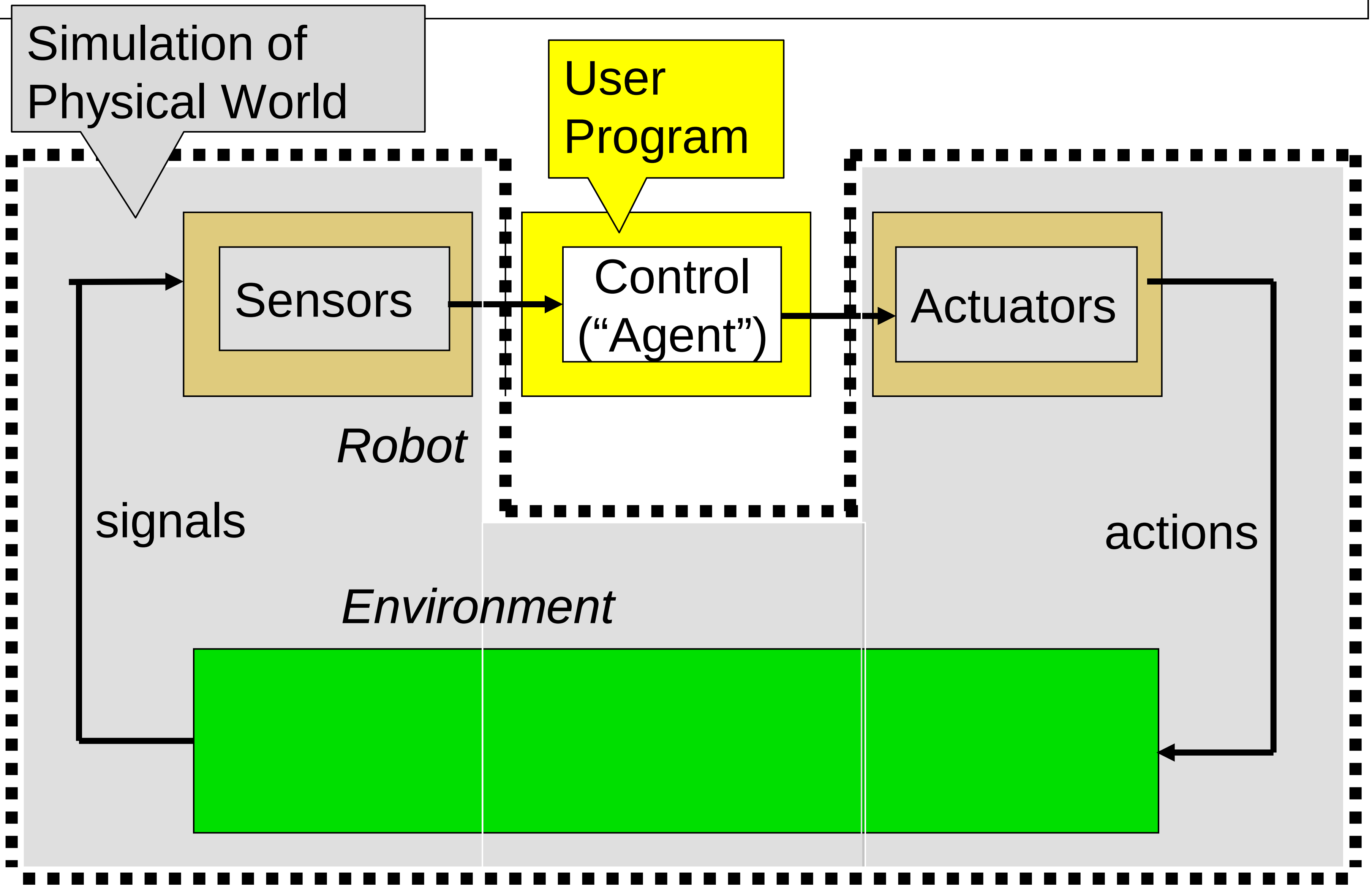
Simulated Robots should

- Behave like real robots
- Have sensors, motors, controls like real robots

Real Robot in Real World



Simulation

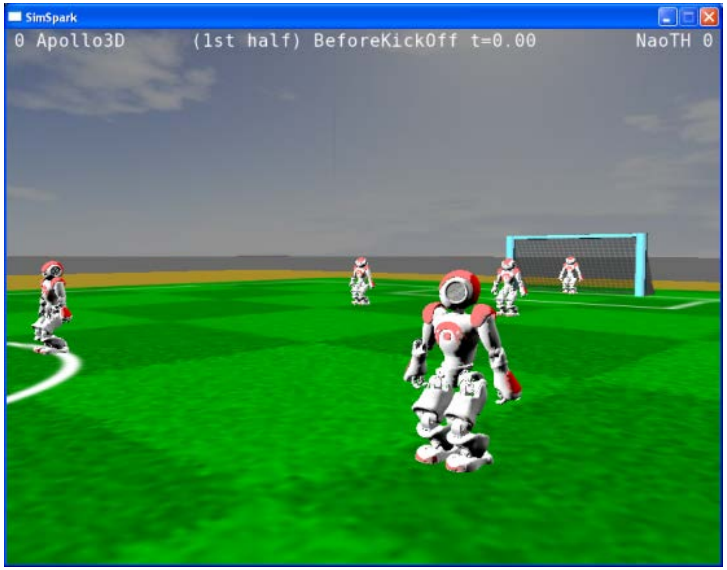

Simulation of Physical World
User Program
Sensors
Control ("Agent")
Actuators
Robot
signals
actions
Environment

RoboCup SimSpark

Physical simulation by Open Dynamic Engine (ODE)
for body dynamics of robot Nao and the soccer environment.

- field and ball
- bodies of players
- soccer rules (“referee”)

.



0 Apollo3D (1st half) BeforeKickOff t=0.00 NaoTH 0

Users can program own robot controls as “agents” which communicate with SimSpark by messages containing sensory information resp. action commands.

Simulation by RoboCup SimSpark

Communication
via protocols (TCP)

<u>Effector messages</u>

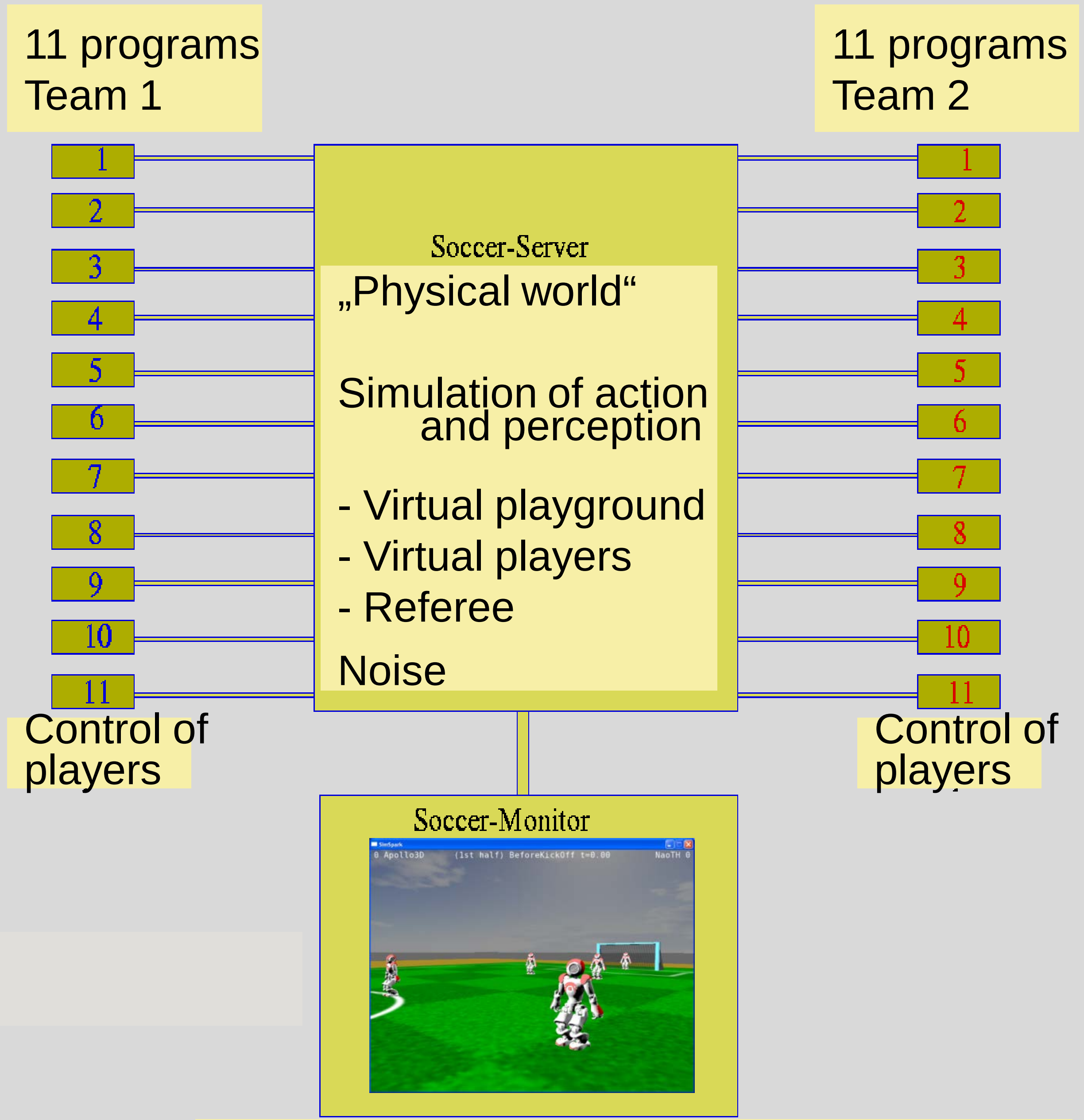
Motor commands

similar to real robot

<u>Perceptor messages</u>

Vision, acoustic, inertial,

….



Server and Monitor developed
by volunteers of RoboCup community

The RoboNewbie Project: Support for Programming Soccer Robots

Diploma Thesis by Monika Domanska at Humboldt University

Framework based on Java and Netbeans.

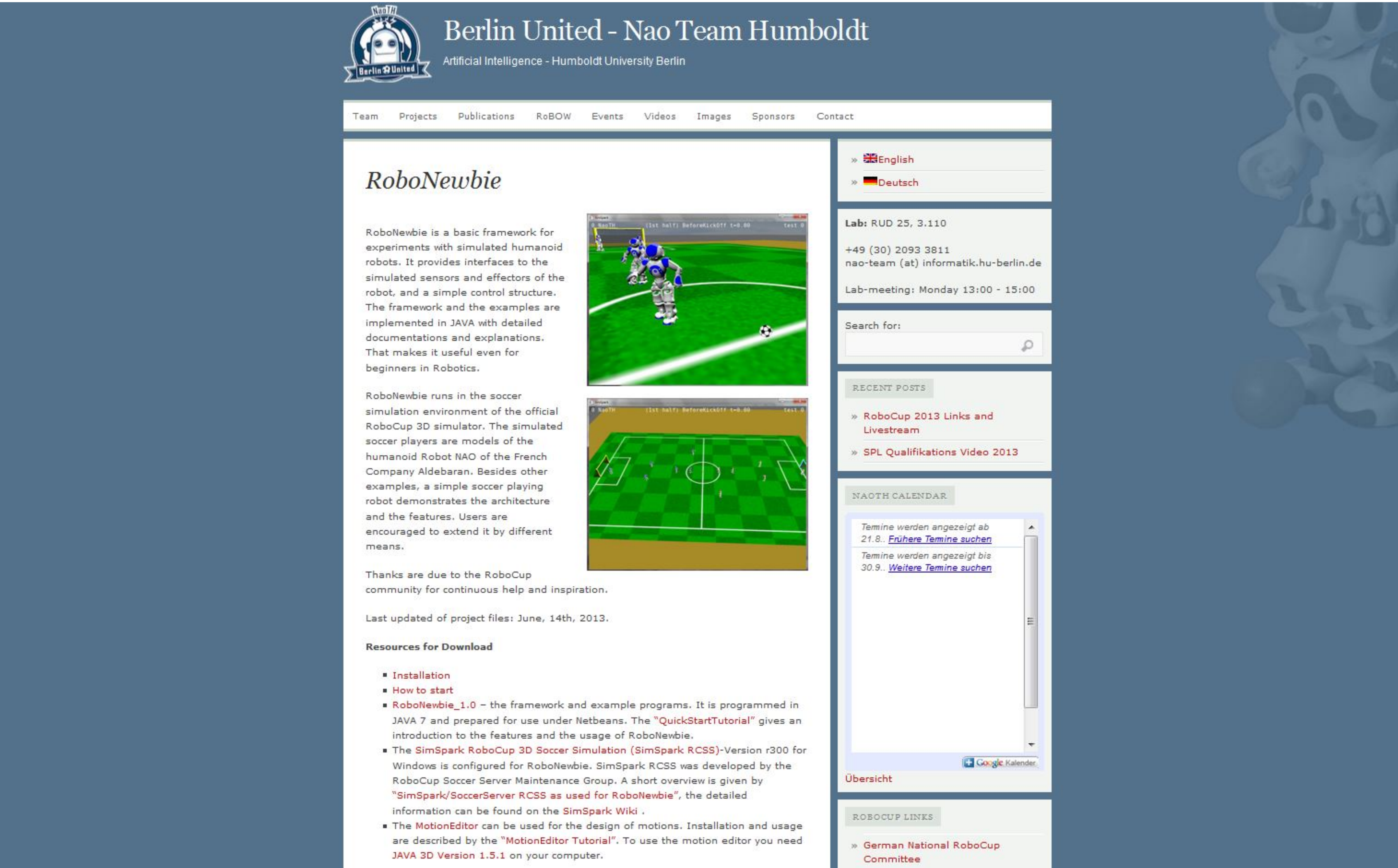
Hides non-robotics aspects (e.g. communication with Simspark)

Provides basic examples/exercises for experiments.

Users can concentrate on robot aspects:

- Direct access to sensor/perceptor data.
- Easy understandable structure (sense-think-act).
- Basic motion skills.
- Open for challenging extensions including Machine Learning.

All programs and materials for download from our website
http://www.naoteamhumboldt.de/projects/robonewbie/



Berlin United - Nao Team Humboldt

Artificial Intelligence - Humboldt University Berlin

RoboNewbie

RoboNewbie is a basic framework for experiments with simulated humanoid robots. It provides interfaces to the simulated sensors and effectors of the robot, and a simple control structure. The framework and the examples are implemented in JAVA with detailed documentations and explanations. That makes it useful even for beginners in Robotics.

RoboNewbie runs in the soccer simulation environment of the official RoboCup 3D simulator. The simulated soccer players are models of the humanoid Robot NAO of the French Company Aldebaran. Besides other examples, a simple soccer playing robot demonstrates the architecture and the features. Users are encouraged to extend it by different means.

Thanks are due to the RoboCup community for continuous help and inspiration.

Last updated of project files: June, 14th, 2013.

Resources for Download

- Installation
- How to start
- RoboNewbie_1.0 – the framework and example programs. It is programmed in JAVA 7 and prepared for use under Netbeans. The "QuickStartTutorial" gives an introduction to the features and the usage of RoboNewbie.
- The SimSpark RoboCup 3D Soccer Simulation (SimSpark RCSS)-Version r300 for Windows is configured for RoboNewbie. SimSpark RCSS was developed by the RoboCup Soccer Server Maintenance Group. A short overview is given by "SimSpark/SoccerServer RCSS as used for RoboNewbie", the detailed information can be found on the SimSpark Wiki .
- The MotionEditor can be used for the design of motions. Installation and usage are described by the "MotionEditor Tutorial". To use the motion editor you need JAVA 3D Version 1.5.1 on your computer.

Lab: RUD 25, 3.110

+49 (30) 2093 3811
nao-team (at) informatik.hu-berlin.de

Lab-meeting: Monday 13:00 - 15:00

RECENT POSTS

- RoboCup 2013 Links and Livestream
- SPL Qualifikations Video 2013

NAOTH CALENDAR

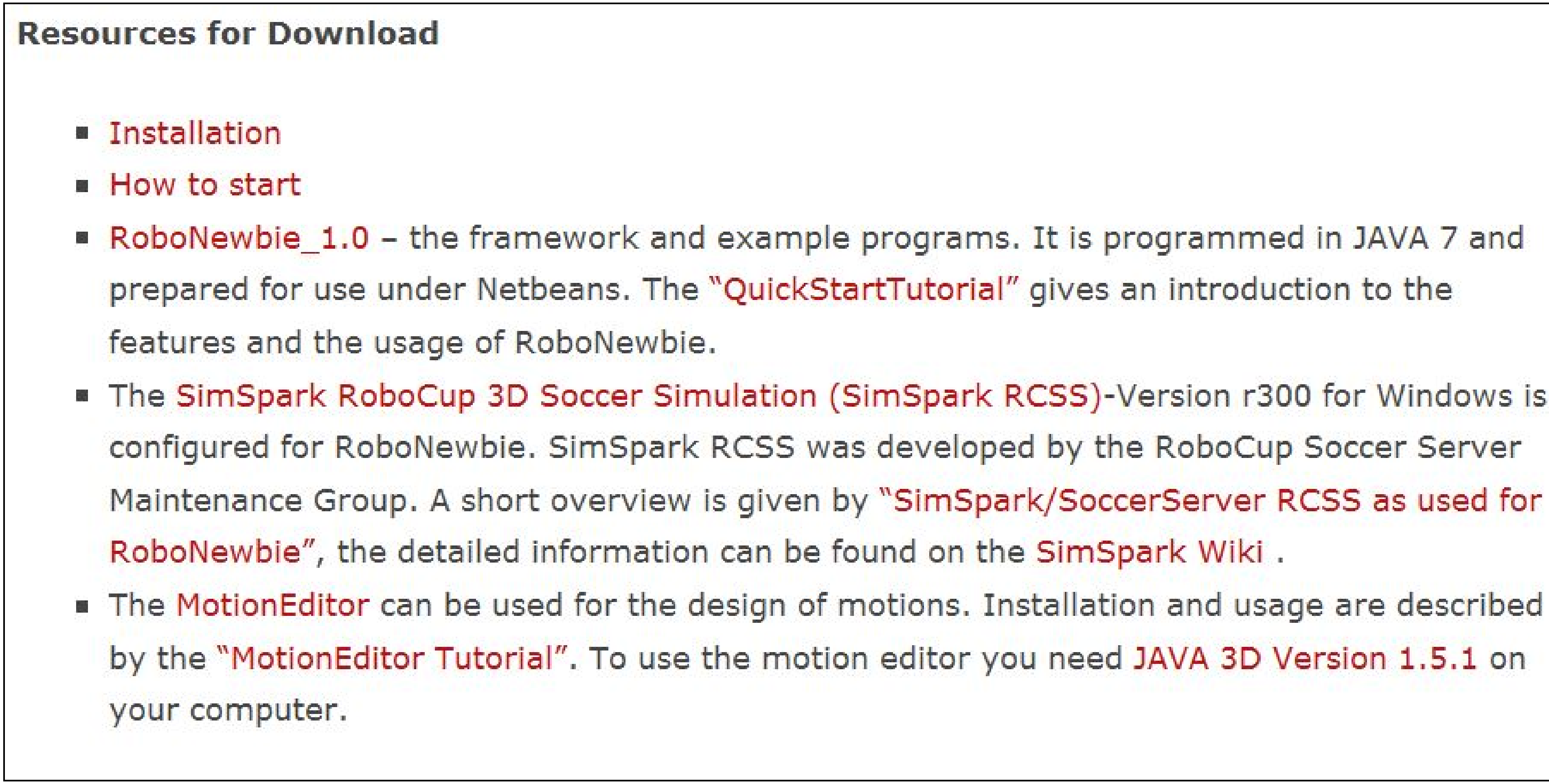
Termine werden angezeigt ab 21.8.. Frühere Termine suchen

Termine werden angezeigt bis 30.9.. Weitere Termine suchen

ROBOCUP LINKS

- German National RoboCup Committee

All programs and materials for download from our website
http://www.naoteamhumboldt.de/projects/robonewbie/



Resources for Download

- Installation
- How to start
- RoboNewbie_1.0 – the framework and example programs. It is programmed in JAVA 7 and prepared for use under Netbeans. The "QuickStartTutorial" gives an introduction to the features and the usage of RoboNewbie.
- The SimSpark RoboCup 3D Soccer Simulation (SimSpark RCSS)-Version r300 for Windows is configured for RoboNewbie. SimSpark RCSS was developed by the RoboCup Soccer Server Maintenance Group. A short overview is given by "SimSpark/SoccerServer RCSS as used for RoboNewbie", the detailed information can be found on the SimSpark Wiki .
- The MotionEditor can be used for the design of motions. Installation and usage are described by the "MotionEditor Tutorial". To use the motion editor you need JAVA 3D Version 1.5.1 on your computer.

Resources

General Resources

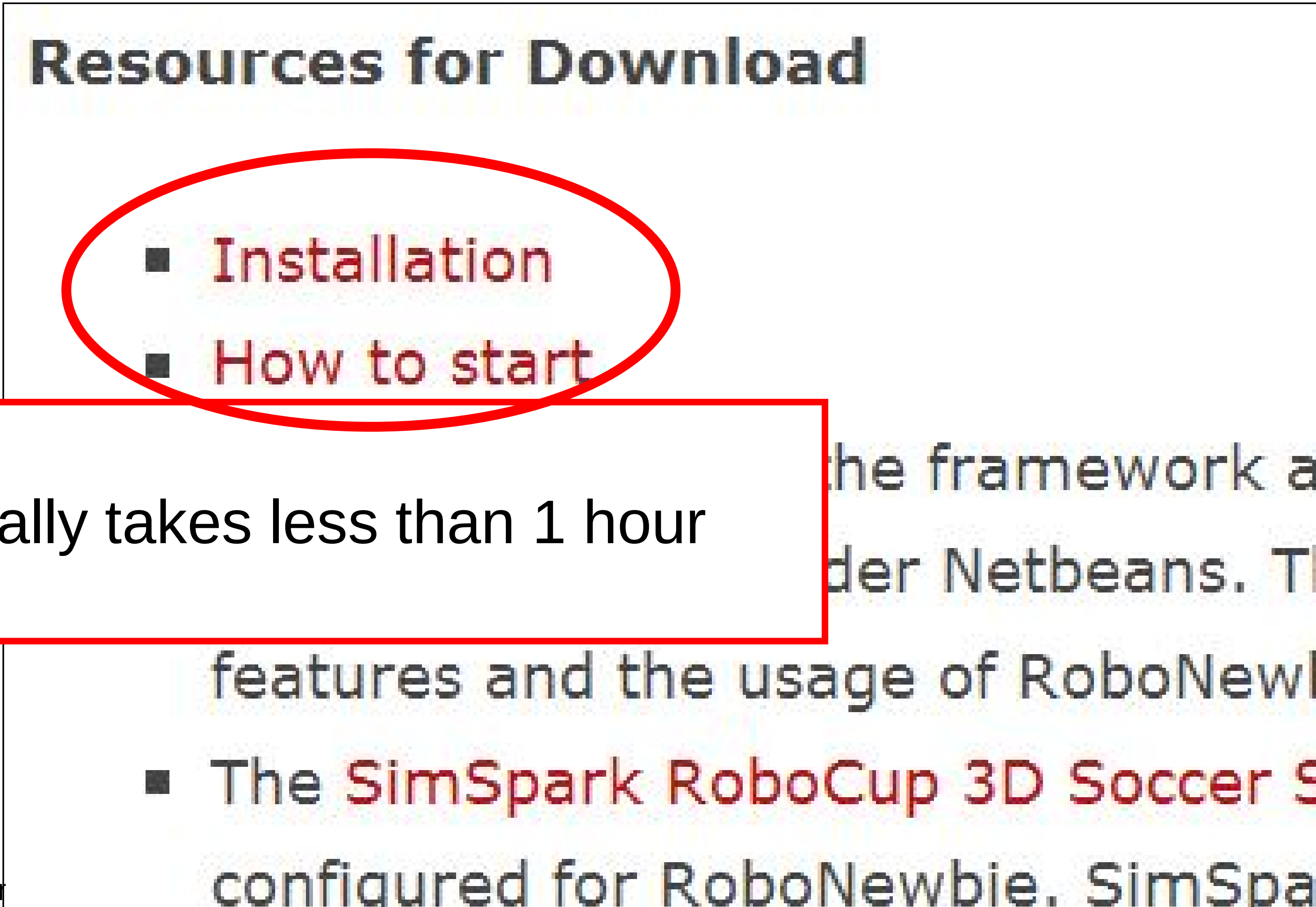
- Windows XP or newer
- Java Development Kit 7
- NetBeans (v. 7.1 or later, JavaSE or JavaEE)
- JAVA 3D

Project Resources for **Download**

- SimSpark Soccer Server
  (executable file for simulation)
- RoboNewbie frame work
  (communication, preprocessing, example programs)
- Motion Editor
- Information materials

Download and Installation

are guided by documents



Resources for Download

- Installation
- How to start

…he framework a…

…der Netbeans. Th…

features and the usage of RoboNewb…

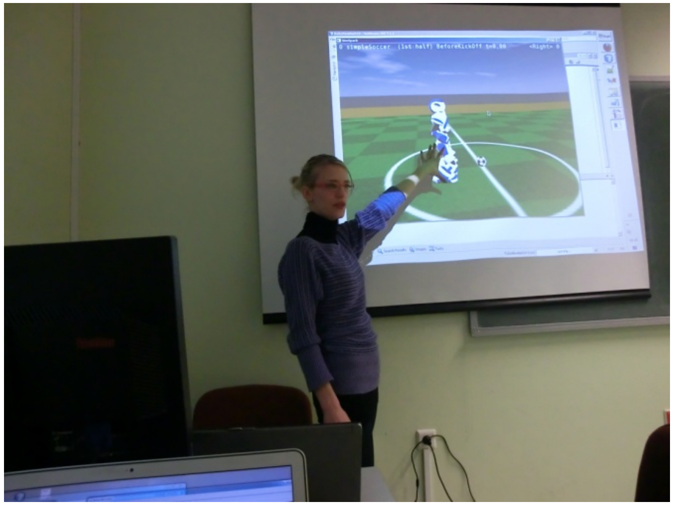
- The SimSpark RoboCup 3D Soccer S…

configured for RoboNewbie. SimSpa…

It usually takes less than 1 hour

First Steps

After download, first experiments are possible.



Can be guided

by instructor (1-2 hours).

Support

by detailed program documentation

and by several documents:

- *“How to start”* describes the usage of programs.
- *“SimSpark/SoccerServer RCSS as used for RoboNewbie”*

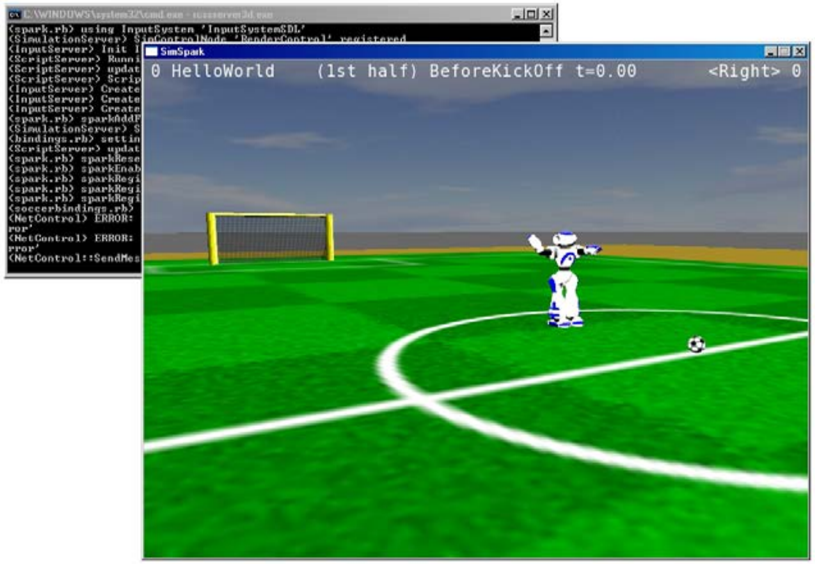
  describes the simulator and its (soccer) rules.
- *“Motion Editor”* describes the motion editor.

The Quick Start Tutorial

Leads through first experiments.

Example programs can be modified by the users.

First example: Let the robot raise its arms.



```
<spark.rb> using InputSystem 'InputSystemSDL'
<SimulationServer> SimControlNode 'RenderControl' registered
<InputServer> Init I
<ScriptServer> Runni
<ScriptServer> updat
<ScriptServer> Scrip
<InputServer> Create
<InputServer> Create
<InputServer> Create
<spark.rb> sparkAddF
<SimulationServer> S
<bindings.rb> settin
<ScriptServer> updat
<spark.rb> sparkRese
<spark.rb> sparkEnab
<spark.rb> sparkRegi
<spark.rb> sparkRegi
<spark.rb> sparkRegi
<soccerbindings.rb>
<NetControl> ERROR:
ror'
<NetControl> ERROR:
rror'
<NetControl::SendMes
```

```
0 HelloWorld    (1st half) BeforeKickOff t=0.00      <Right> 0
```

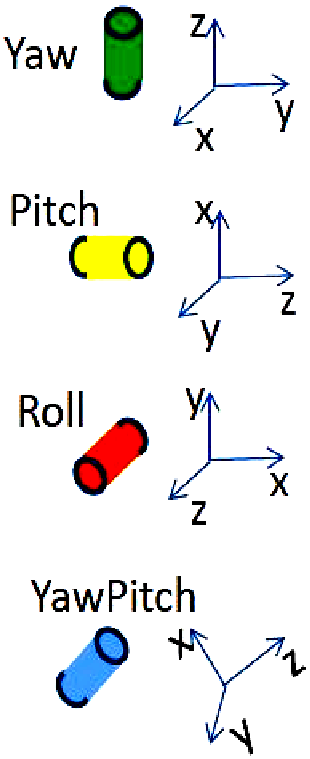
Exercise 1: Change program Agent_BasicStructure

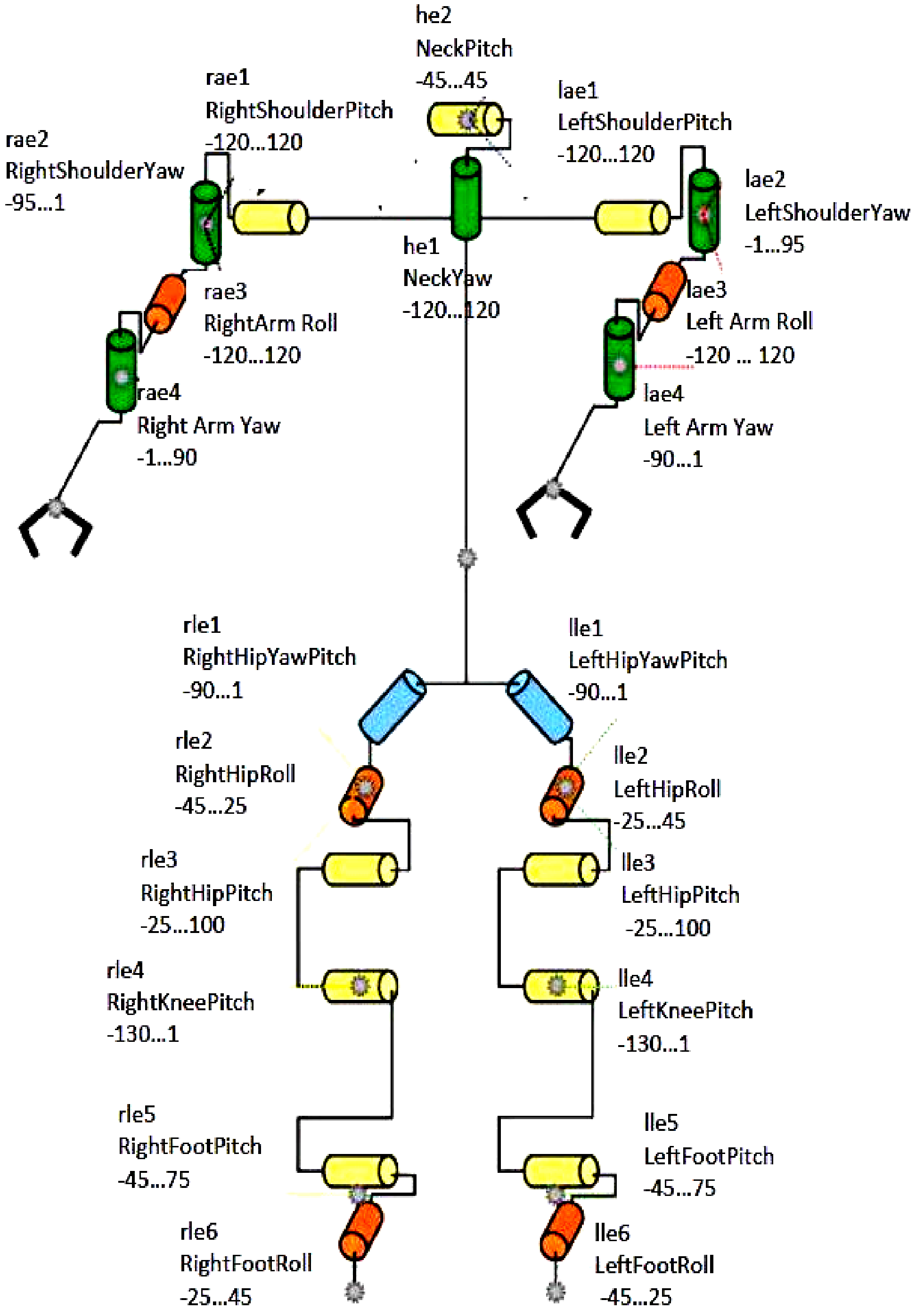
1) Open the RoboNewbie-Project in NetBeans.
   Make a new package for your own agents.
   Copy *Agent_BasicStructure* into your own package.

2) Start SimSpark with *"[SimSpark root directory]/rcssserver3d.exe"*
   Navigate at the monitor window with left mouse button, arrow keys, …

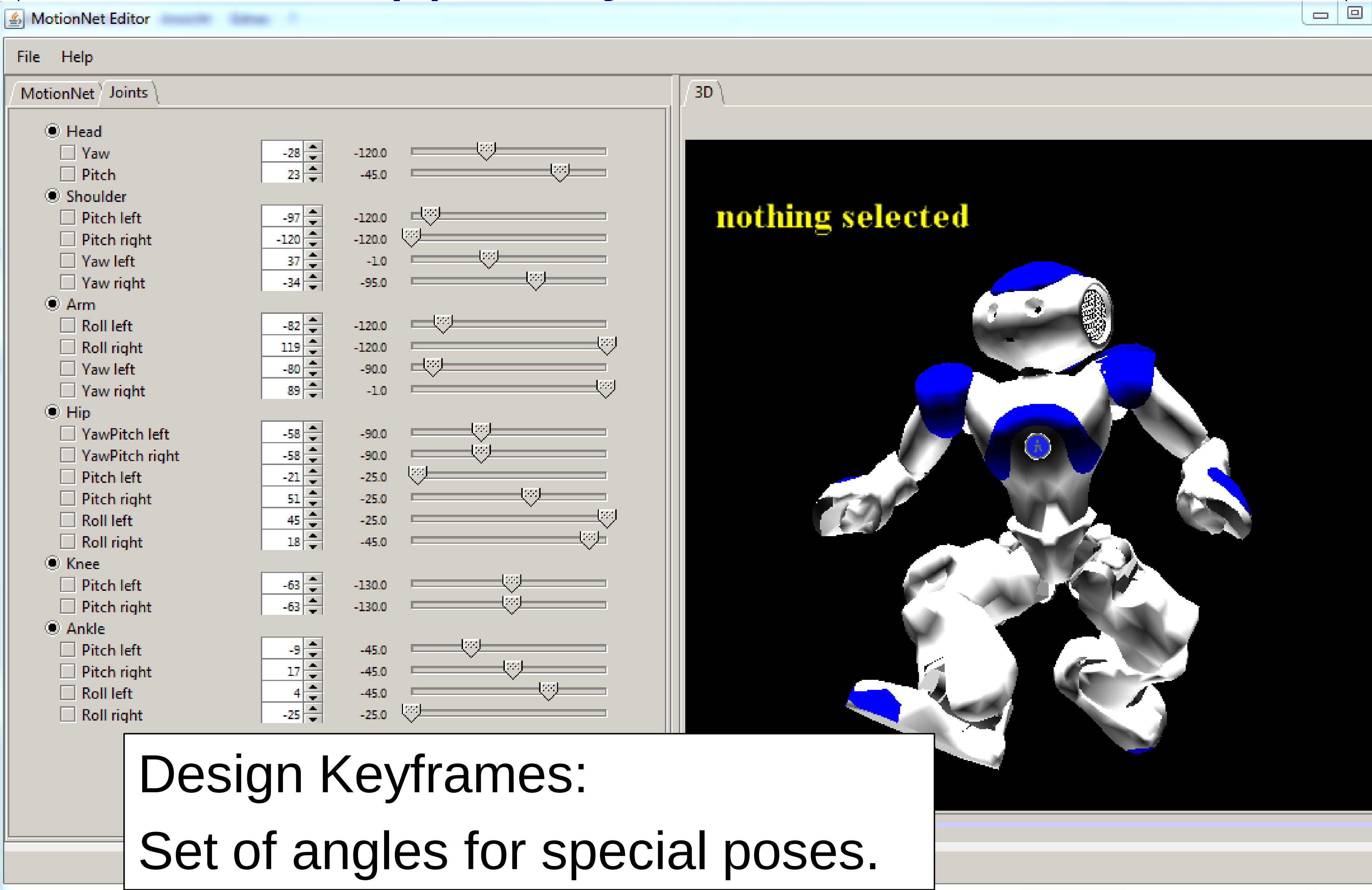
3) Run your class *Agent_BasicStructure* to test if everything works
   (in NetBeans: right click on the filename → Run File).

4) Try changes:
   - ➔ Change initial position
   - ➔ Change effector commands in method run():
     Try other velocities (they range from -2π to 2π).
     Try other robot joints …


Yaw
z
y
x
Pitch
x
z
y
Roll
y
x
z
YawPitch
x
z
y


he2
NeckPitch
-45...45
rae1
RightShoulderPitch
-120...120
lae1
LeftShoulderPitch
-120...120
rae2
RightShoulderYaw
-95...1
lae2
LeftShoulderYaw
-1...95
he1
NeckYaw
-120...120
rae3
RightArm Roll
-120...120
lae3
Left Arm Roll
-120 ... 120
rae4
Right Arm Yaw
-1...90
lae4
Left Arm Yaw
-90...1
rle1
RightHipYawPitch
-90...1
lle1
LeftHipYawPitch
-90...1
rle2
RightHipRoll
-45...25
lle2
LeftHipRoll
-25...45
rle3
RightHipPitch
-25...100
lle3
LeftHipPitch
-25...100
rle4
RightKneePitch
-130...1
lle4
LeftKneePitch
-130...1
rle5
RightFootPitch
-45...75
lle5
LeftFootPitch
-45...75
rle6
RightFootRoll
-25...45
lle6
LeftFootRoll
-45...25

Support by Motion Editor



● Head		
□ Yaw	-28	-120.0
□ Pitch	23	-45.0
● Shoulder		
□ Pitch left	-97	-120.0
□ Pitch right	-120	-120.0
□ Yaw left	37	-1.0
□ Yaw right	-34	-95.0
● Arm		
□ Roll left	-82	-120.0
□ Roll right	119	-120.0
□ Yaw left	-80	-90.0
□ Yaw right	89	-1.0
● Hip		
□ YawPitch left	-58	-90.0
□ YawPitch right	-58	-90.0
□ Pitch left	-21	-25.0
□ Pitch right	51	-25.0
□ Roll left	45	-25.0
□ Roll right	18	-45.0
● Knee		
□ Pitch left	-63	-130.0
□ Pitch right	-63	-130.0
● Ankle		
□ Pitch left	-9	-45.0
□ Pitch right	17	-45.0
□ Roll left	4	-45.0
□ Roll right	-25	-25.0

nothing selected

Design Keyframes:

Set of angles for special poses.

Support by Logger

Runtime debugging is difficult because of synchronization (simulator cycle lasts 20 msec).

- Printout messages may take too much time.
- Breakpoints stop the agent, but not the simulation.

Debug messages are collected by the class Logger.
Collected messages are printed out after finish.

Further Examples

Usage of Perceptors:

Exercise 2: Lift an arm when it sees another robot.

Motion:

Exercise 3: Obstacle avoidance (using given skills).

Exercise 4: Implement a skill for kicking the ball.

Exercise 5: Improve the program *agentSimpleSoccer*

Exercise 5 is used for a final competion.

Course Competition

Become the Soccer Champion of the Fast Scoring Competition !

The task is to score as soon as possible.

The example *agentSimpleSoccer* pushes the ball towards the goal. During 10 minutes it almost reaches the goal with the ball. You can use this program as an inspiration for your task.

You can modify and extend it with new motions, better perception and more intelligent behavior. You can even program a team of up to 4 robots which cooperatively perform the task.

Example *Robo1* by Luka Unuk (Rijeka) (on SimSpark Version Rijeka-Course)

Courses

(*): Funded by DAAD programs

5.-7.6. 2012, University of Novi Sad 4 hours, 12 participants. (*)
(first test of the concepts)

12.-18.8.2012 Summer school Ohrid 30 hours, 23 participants. (*)

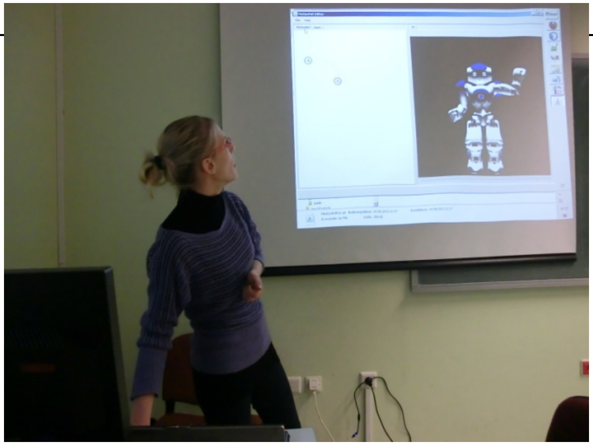
Oct-Dec. 2012 Humboldt University Berlin, 32 hours, 10 participants.
(part of regular Course "Cognitive Robotics“ for Diploma students)

25.2.-1.3.2013 Vistula University Warsaw, 30 hours, 30 participants.

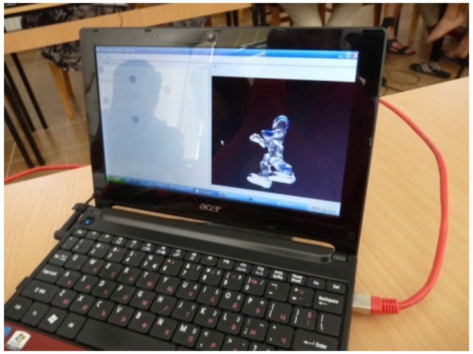
12.-21.3.2013 University of Novi Sad, 32 hours, 9 participants. (*)

21.-29.5. 2013 University of Rijeka, 24 hours, 18 participants. (*)

Courses



Team Work



Participants

- Need programming skills with Java and Netbeans to understand and modify the agent programs.

- Some physical and mathematical background is needed to understand the theoretical and practical issues of Robotics.

- Preferably, participants should work in teams of 3-5 participants, mixed by different skills of its members.

Technical Resources

Preferable:

Participants have individual computers (laptops) with installed programs.

Alternative:

Computers with installed programs in a lab.

Needs a responsible administrator.

Test of programs in advance.

Organisational Issues

- Early information for installation.

- Lectures and exercises ae mixed. Exercises start on the first day. More exercises in the beginning to become familiar with the problems. Further exercises parallel to the courses.

- Later exercises concentrate on the competition. Mostly by homework, but with common discussions about ideas.

- Each participant prepares a written report on her/his efforts for the competition (ideas, implementations, results).

Organisation of Competition as final event

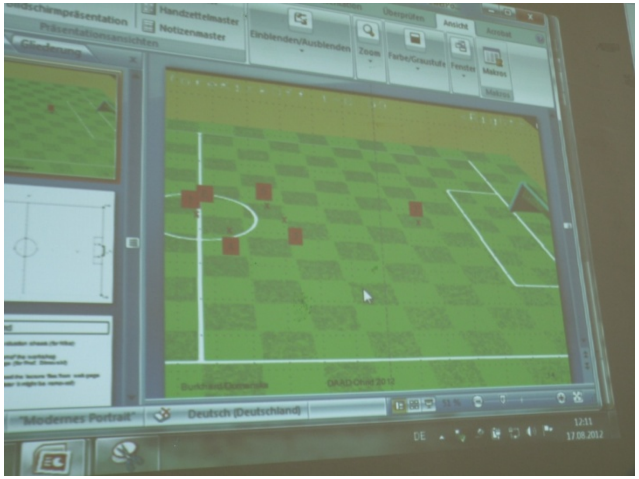
(Task: Improve *agentSimpleSoccer* for faster scoring.)

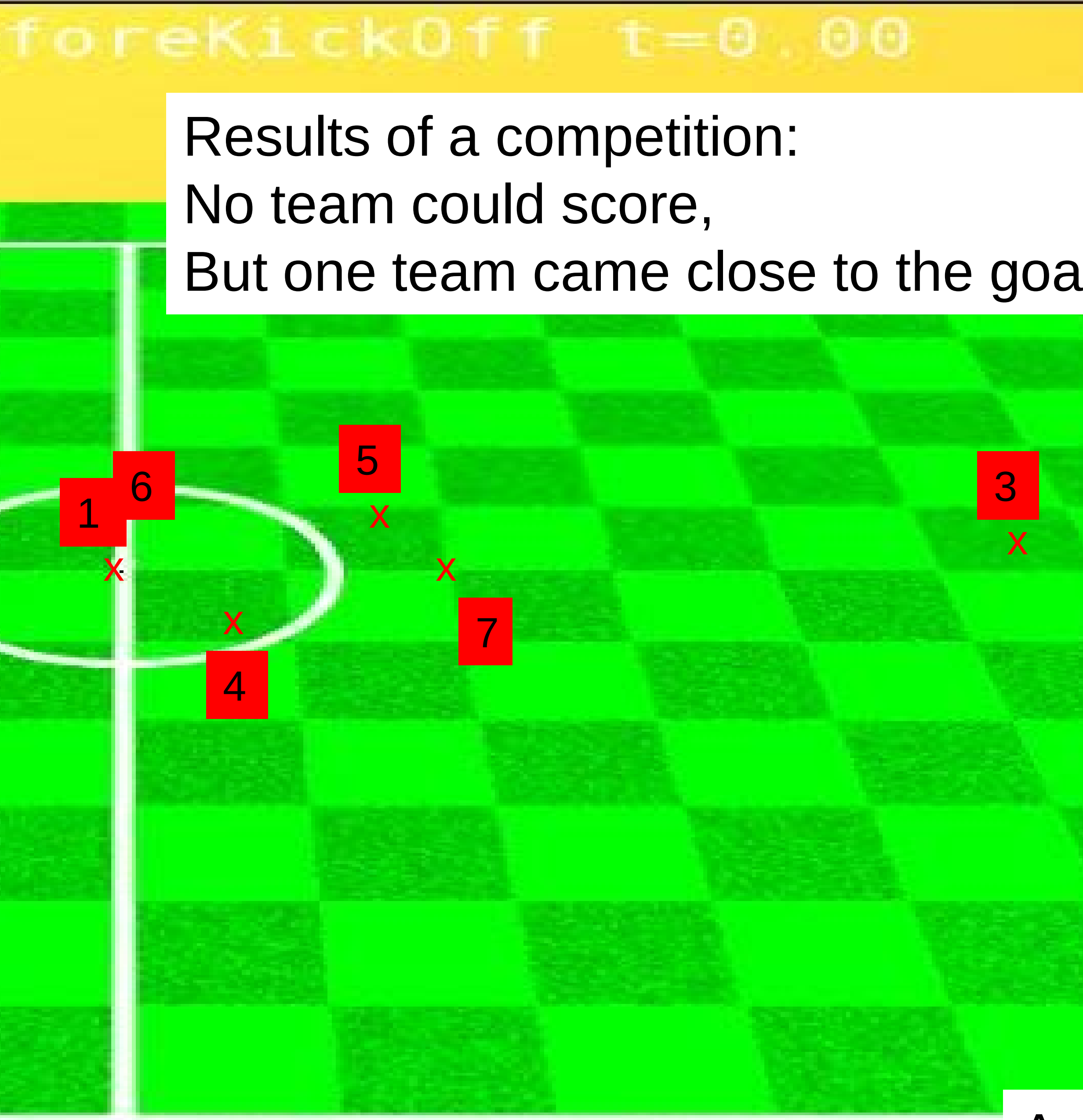
Competition with tight schedule, strict and transparent rules:

- Each team gives a presentation about its efforts.
- Each team has only one trial of only 3 minutes.
- Ranking by fastest scoring times

  resp. minimal distances to the goal after the 3 minutes.

Competition

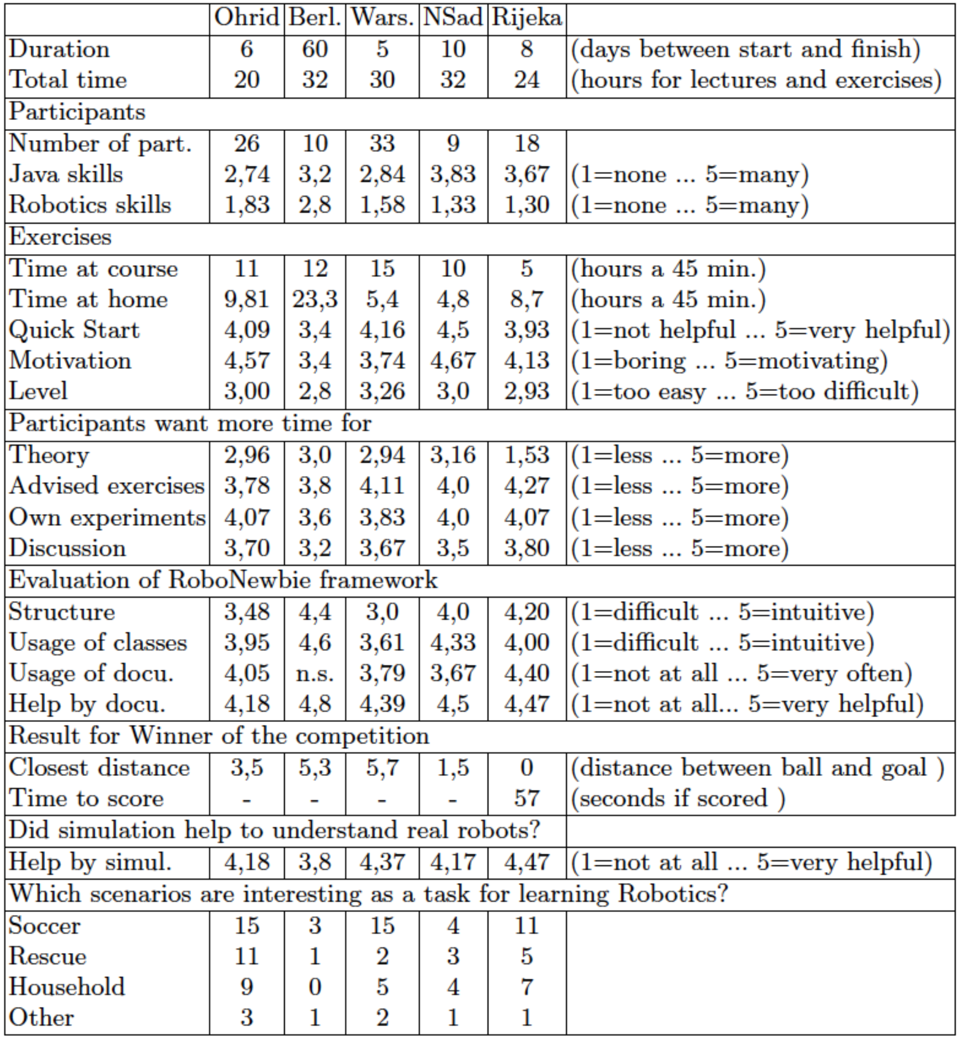




Results of a competition:
No team could score,
But one team came close to the goal.

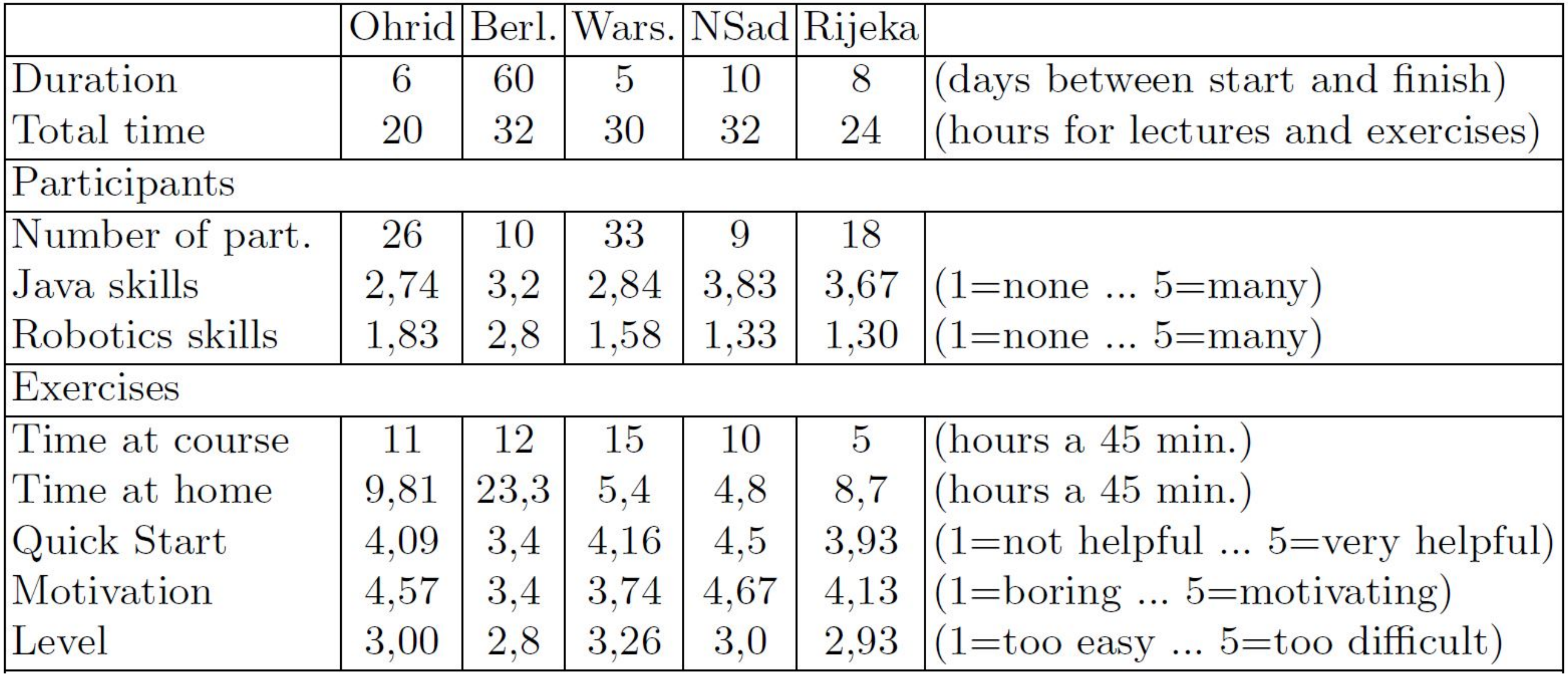
Award Ceremony

Evaluation by Students



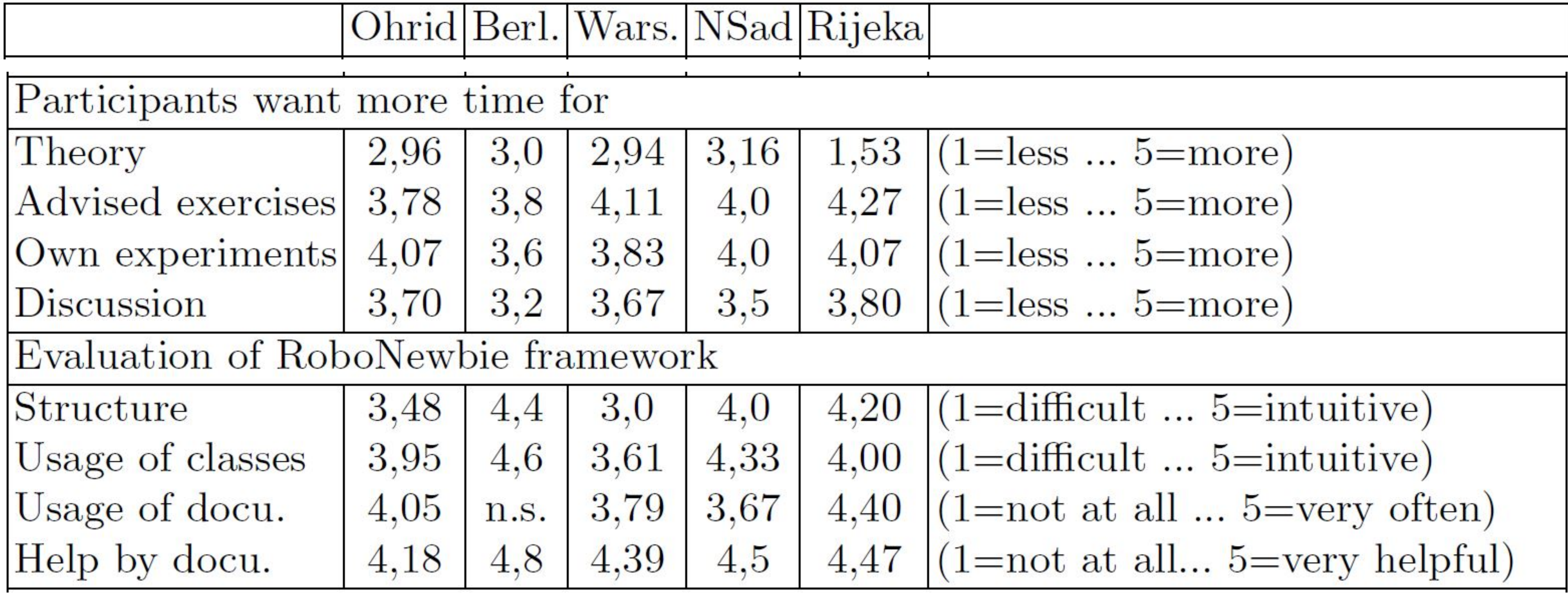
	Ohrid	Berl.	Wars.	NSad	Rijeka	
Duration	6	60	5	10	8	(days between start and finish)
Total time	20	32	30	32	24	(hours for lectures and exercises)
Participants						
Number of part.	26	10	33	9	18	
Java skills	2,74	3,2	2,84	3,83	3,67	(1=none ... 5=many)
Robotics skills	1,83	2,8	1,58	1,33	1,30	(1=none ... 5=many)
Exercises						
Time at course	11	12	15	10	5	(hours a 45 min.)
Time at home	9,81	23,3	5,4	4,8	8,7	(hours a 45 min.)
Quick Start	4,09	3,4	4,16	4,5	3,93	(1=not helpful ... 5=very helpful)
Motivation	4,57	3,4	3,74	4,67	4,13	(1=boring ... 5=motivating)
Level	3,00	2,8	3,26	3,0	2,93	(1=too easy ... 5=too difficult)
Participants want more time for						
Theory	2,96	3,0	2,94	3,16	1,53	(1=less ... 5=more)
Advised exercises	3,78	3,8	4,11	4,0	4,27	(1=less ... 5=more)
Own experiments	4,07	3,6	3,83	4,0	4,07	(1=less ... 5=more)
Discussion	3,70	3,2	3,67	3,5	3,80	(1=less ... 5=more)
Evaluation of RoboNewbie framework						
Structure	3,48	4,4	3,0	4,0	4,20	(1=difficult ... 5=intuitive)
Usage of classes	3,95	4,6	3,61	4,33	4,00	(1=difficult ... 5=intuitive)
Usage of docu.	4,05	n.s.	3,79	3,67	4,40	(1=not at all ... 5=very often)
Help by docu.	4,18	4,8	4,39	4,5	4,47	(1=not at all... 5=very helpful)
Result for Winner of the competition						
Closest distance	3,5	5,3	5,7	1,5	0	(distance between ball and goal)
Time to score	-	-	-	-	57	(seconds if scored)
Did simulation help to understand real robots?						
Help by simul.	4,18	3,8	4,37	4,17	4,47	(1=not at all ... 5=very helpful)
Which scenarios are interesting as a task for learning Robotics?						
Soccer	15	3	15	4	11	
Rescue	11	1	2	3	5	
Household	9	0	5	4	7	
Other	3	1	2	1	1	

Evaluation by Students



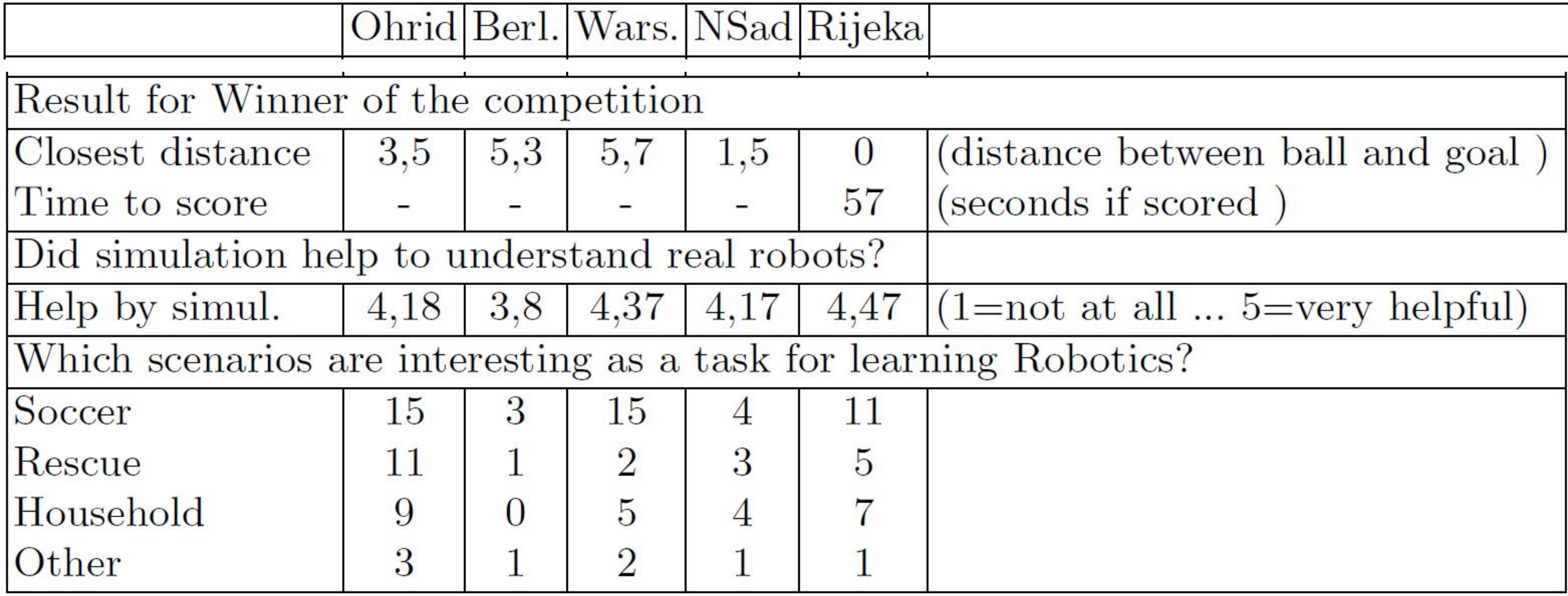
	Ohrid	Berl.	Wars.	NSad	Rijeka	
Duration	6	60	5	10	8	(days between start and finish)
Total time	20	32	30	32	24	(hours for lectures and exercises)
Participants						
Number of part.	26	10	33	9	18	
Java skills	2,74	3,2	2,84	3,83	3,67	(1=none ... 5=many)
Robotics skills	1,83	2,8	1,58	1,33	1,30	(1=none ... 5=many)
Exercises						
Time at course	11	12	15	10	5	(hours a 45 min.)
Time at home	9,81	23,3	5,4	4,8	8,7	(hours a 45 min.)
Quick Start	4,09	3,4	4,16	4,5	3,93	(1=not helpful ... 5=very helpful)
Motivation	4,57	3,4	3,74	4,67	4,13	(1=boring ... 5=motivating)
Level	3,00	2,8	3,26	3,0	2,93	(1=too easy ... 5=too difficult)

Evaluation by Students



	Ohrid	Berl.	Wars.	NSad	Rijeka	
Participants want more time for						
Theory	2,96	3,0	2,94	3,16	1,53	(1=less ... 5=more)
Advised exercises	3,78	3,8	4,11	4,0	4,27	(1=less ... 5=more)
Own experiments	4,07	3,6	3,83	4,0	4,07	(1=less ... 5=more)
Discussion	3,70	3,2	3,67	3,5	3,80	(1=less ... 5=more)
Evaluation of RoboNewbie framework						
Structure	3,48	4,4	3,0	4,0	4,20	(1=difficult ... 5=intuitive)
Usage of classes	3,95	4,6	3,61	4,33	4,00	(1=difficult ... 5=intuitive)
Usage of docu.	4,05	n.s.	3,79	3,67	4,40	(1=not at all ... 5=very often)
Help by docu.	4,18	4,8	4,39	4,5	4,47	(1=not at all... 5=very helpful)

Evaluation by Students



	Ohrid	Berl.	Wars.	NSad	Rijeka	
Result for Winner of the competition						
Closest distance	3,5	5,3	5,7	1,5	0	(distance between ball and goal)
Time to score	-	-	-	-	57	(seconds if scored)
Did simulation help to understand real robots?						
Help by simul.	4,18	3,8	4,37	4,17	4,47	(1=not at all ... 5=very helpful)
Which scenarios are interesting as a task for learning Robotics?						
Soccer	15	3	15	4	11	
Rescue	11	1	2	3	5	
Household	9	0	5	4	7	
Other	3	1	2	1	1	

Conclusions: Scenario and Exercises

Amount of exercises was sufficient for short courses.

Additional exercises for longer courses and e-learning:

- Different behaviors
  (line following, balancing, dancing, ...)
- Advanced Motions
  (model based, biologically inspired motions, Machine Learning)
- Advanced perception
  (world models, mapping, other robots,...)
- Coordination strategies
  (protocols, swarm behavior, …)
- Competitive soccer players
  (e.g. for RoboCup competitions).

Problems with Predefined Skills

Users should understand the pre-implemented algorithms.

Users should be motivated for own experiments.

But some advanced methods
may be behind the scope of beginners
e.g.

- advanced motions
- preprocessing for perception

Advanced Motions

… are needed e.g. for coordinated behavior.

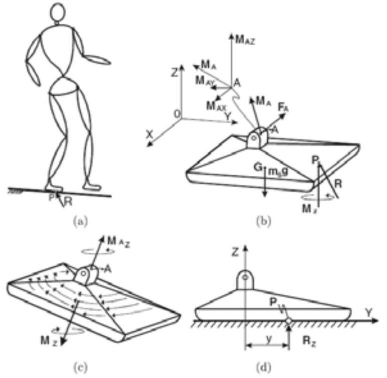


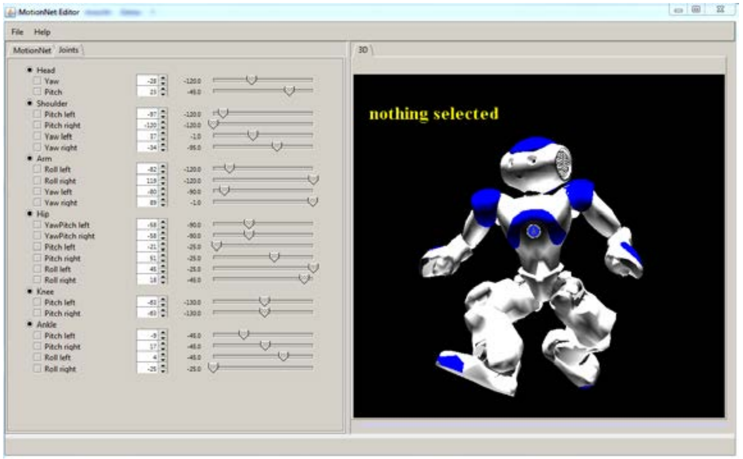
Fig. 1. Biped mechanism and forces acting on its sole.

Takes too much time to develop during a course.

Could be provided by pre-programmed methods.

Drawback:

Decreased motivation when participants should make own experiences by defining (simple) motions.



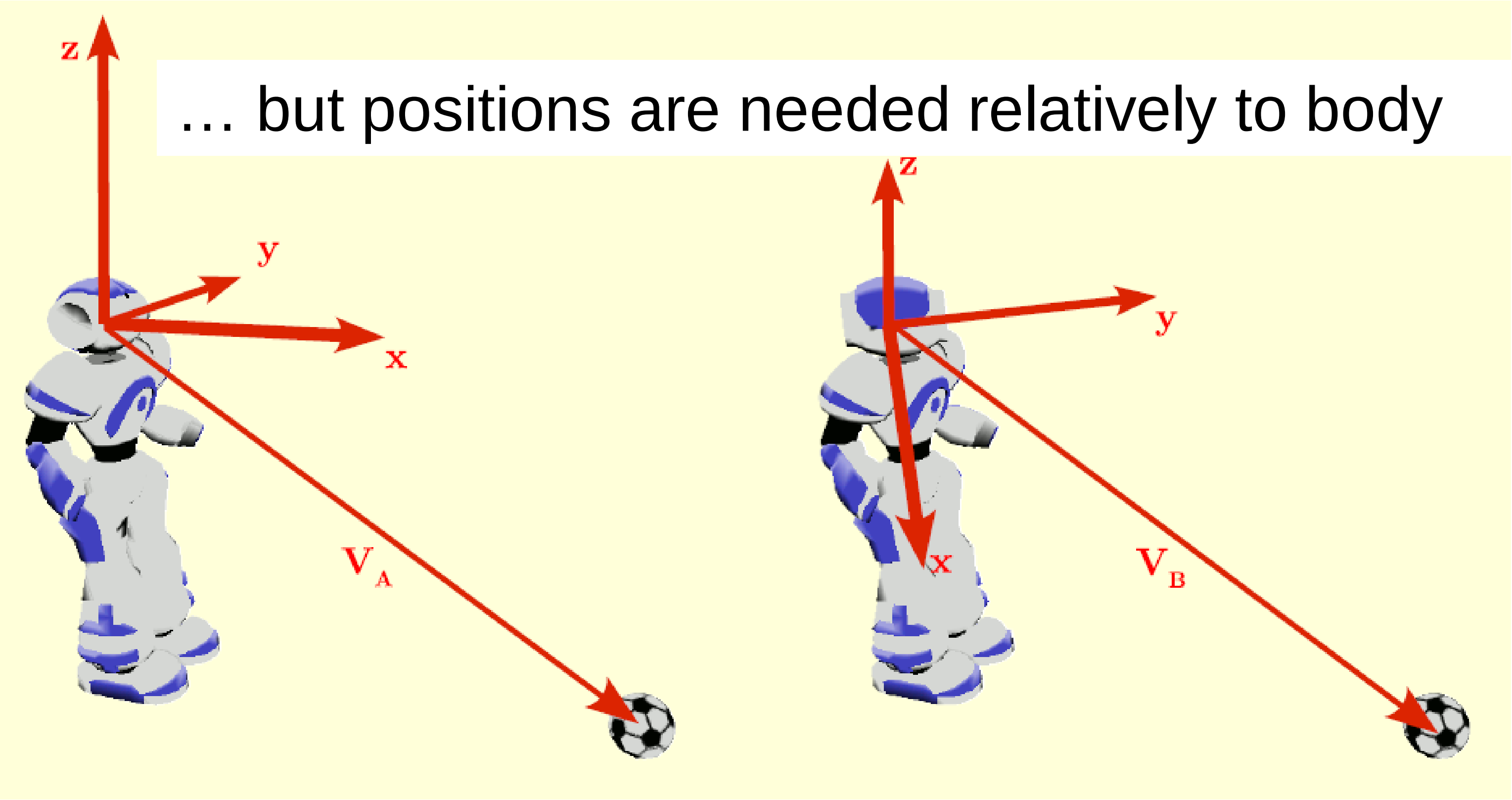
nothing selected

- Head
  - □ Yaw -28 -120.0
  - □ Pitch 21 -49.0
- Shoulder
  - □ Pitch left -87 -120.0
  - □ Pitch right -120 -120.0
  - □ Yaw left 37 -1.0
  - □ Yaw right -34 -95.0
- Arm
  - □ Roll left -82 -120.0
  - □ Roll right 119 -120.0
  - □ Yaw left -80 -90.0
  - □ Yaw right 89 -1.0
- Hip
  - □ YawPitch left -58 -90.0
  - □ YawPitch right -58 -90.0
  - □ Pitch left -21 -25.0
  - □ Pitch right 51 -25.0
  - □ Roll left 45 -25.0
  - □ Roll right 18 -45.0
- Knee
  - □ Pitch left -63 -130.0
  - □ Pitch right -63 -130.0
- Ankle
  - □ Pitch left -9 -45.0
  - □ Pitch right 17 -45.0
  - □ Roll left 4 -45.0
  - □ Roll right -25 -25.0

Preprocessing for Perception

Objects are perceived

with coordinates relatively to the camera in the head …



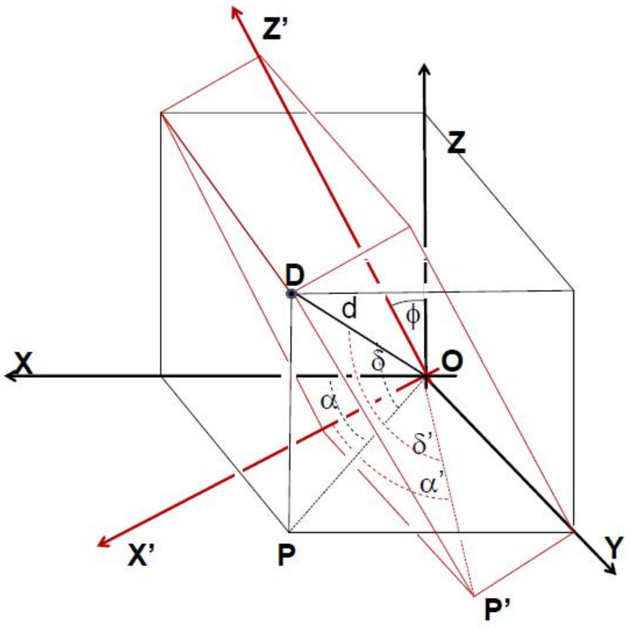
… but positions are needed relatively to body

Preprocessing for Perception

Simple transformation in RoboNewbie
by adding offsets for head angles.

Drawback: Deviations in position for near balls.

But correct transformations
would need
complex geometrical calculations.



Next Steps

- Further courses with unversity students.
- First courses at secondary schools.

- Extensions by new exercises.
- Preparation for Machine Learning with evolutionary methods.
- Preparation of e-Learning materials.

Thank you!

By the way:

Next RoboCup Competition

will be in Brazil

TeenSize Finales at RoboCup 2013 in Eindhoven
NimbRo (Germany) vs. CIT Brains (Japan): 5:0

Coming soon...

Mark your calendar:

July, 19-25th, 2014
João Pessoa, Brazil