Mobile control interface for modular robots
Semester project, fall 2010

Gilles Cressier

Biorobotics Laboratory

Advisors:
Prof. Auke Jan Ijspeert
Rico Möckel
Stéphane Bonard
Introduction to Roombots

Roombots: modular robot composed of modules.

Figure 1: A Roombots module
Roombots: modular robot composed of modules. One Module is composed of:

Figure 1: A Roombots module
Introduction to Roombots

Roombots: modular robot composed of modules. One Module is composed of:

- 3 degrees of freedom
Introduction to Roombots

Roombots: modular robot composed of modules. One Module is composed of:

- 3 degrees of freedom
- 10 connectors

Figure 1: A Roombots module
Introduction to Roombots

Roombots: modular robot composed of modules. One Module is composed of:

- 3 degrees of freedom
- 10 connectors
- 2 active connection mechanisms (ACM)

Figure 1: A Roombots module
Motivations

Design an interface to **apprehend** and **control** Roombots:

Figure 2: Concept of the project.
Motivations

Design an interface to apprehend and control Roombots:

Figure 2: Concept of the project.
Motivations

Design an interface to **apprehend** and control Roombots:

- Provide a visualisation of the robot.

Figure 2: Concept of the project.
Motivations

Design an interface to **apprehend** and control Roombots:

- Provide a visualisation of the robot.
- Allow the user to read informations from sensors.

Figure 2: Concept of the project.
Motivations

Design an interface to apprehend and **control** Roombots:

- Provide a visualisation of the robot.
- Allow the user to read informations from sensors.
- Allow the user to modify degrees of freedom, and ACM values.

Figure 2: Concept of the project.
Design an interface to apprehend and control Roombots:

- Provide a visualisation of the robot.
- Allow the user to read informations from sensors.
- Allow the user to modify degrees of freedom, and ACM values.

Figure 2: Concept of the project.
Outline

1. Goals and challenge
   - Motivations
   - Method

2. Ideal interface to control Roombots
   - Related Works
   - Requirements
   - Specifications of a prototype
   - Conclusion

3. Implementation of a tool to control Roombots
   - Introduction
   - Presentation of the implemented Roombots Cockpit
   - Further improvements

4. Final conclusion
I split the project into two main parts.
I split the project into two main parts.

**Part I** Prototype an ideal interface to control Roombots.
I split the project into two main parts.

Part I  Prototype an ideal interface to control Roombots.

Part II  Start to implement a tool to control Roombots.
Part I: An ideal interface to control Roombots:

Plan:
Part I: An ideal interface to control Roombots:

Plan:

- See what has already been done.
Part I: An ideal interface to control Roombots:

Plan:

- See what has already been done.
- Define the requirements.
Part I: An ideal interface to control Roombots:

Plan:

- See what has already been done.
- Define the requirements.
- Create a prototype that fit the requirements.
Figure 3: **Eve: the YaMoR simulator.**
Script-based.
Use video game environment.
Figure 3: **Eve: the YaMoR simulator.**
Script-based.
Use video game environment.

Figure 4: **Molecubes software.**
Similar to Roombots.
Real-time simulation.
Figure 3: **Eve: the YaMoR simulator.**
Script-based.
Use video game environment.

Figure 4: **Molecubes software.**
Similar to Roombots.
Real-time simulation.

Figure 5: **Player/Stage/Gazebo.**
Modular approach.
Real robot control.
Guidelines for an interface for Roombots:

- Guidelines of the visualisation:
  - Readability
  - Ease to understand
  - Accessibility

- Interaction requirements:
  - Selection of objects
  - Modification of degrees
  - Modification of the value of active connection mechanisms (ACMs)
Multiple kinds of usage:
Multiple kinds of usage:

- Simulate a set of movements.
Multiple kinds of usage:

- Simulate a set of movements.
- Read information from the sensors of the real robot.
Multiple kinds of usage:

- Simulate a set of movements.
- Read information from the sensors of the real robot.
- Order simple movement to the real robot.
Multiple kinds of usage:

- Simulate a set of movements.
- Read information from the sensors of the real robot.
- Order simple movement to the real robot.
- Order general movement to the real robot.
Multiple kinds of usage:

- Simulate a set of movements.
- Read information from the sensors of the real robot.
- Order simple movement to the real robot.
- Order general movement to the real robot.

Problem

How can I handle this complexity?
Multiple kinds of usage:

- Simulate a set of movements.
- Read information from the sensors of the real robot.
- Order simple movement to the real robot.
- Order general movement to the real robot.

**Problem**

How can I handle this complexity?

**Solution I choose**

Use different modes.

**Mode:** An environment in which provided features are designed for particular kinds of tasks.
The interface would be composed of four modes:
The interface would be composed of four modes:

**Assembly mode:** To construct structures.
The interface would be composed of four modes:

**Assembly mode:** To construct structures.

**Sequence editor:** To edit sequences of movement orders.

---

**Example**

![Figure 7: A chronograph](image-url)
The interface would be composed of four modes:

**Assembly mode:** To construct structures.

**Sequence editor:** To edit sequences of movement orders.

**Low-level mode:** An expert user mode.

---

**Example**

*Figure 8: Read information from sensors.*
The interface would be composed of four modes:

**Assembly mode:** To construct structures.

**Sequence editor:** To edit sequences of movement orders.

**Low-level mode:** An expert user mode.

**High-level mode:** A non-expert user mode.

---

**Example**

Figure 9: Example of high-level suggestion.
Conclusion for this part:

- By now, tools for controlling modular robots exist, but are still focused on development.
Conclusion for this part:

- By now, tools for controlling modular robots exist, but are still focused on development.
- Implement a complete user-friendly interface for Roombots is feasible, but would require a lot of research and time.
Conclusion for this part:

- By now, tools for controlling modular robots exist, but are still focused on development.

- Implement a complete user-friendly interface for Roombots is feasible, but would require a lot of research and time.

So in the next part of the project, I implement a tool to control Roombots, focusing on visualisation of structure and basic interaction:
Conclusion for this part:

- By now, tools for controlling modular robots exist, but are still focused on development.
- Implement a complete user-friendly interface for Roombots is feasible, but would require a lot of research and time.

So in the next part of the project, I implement a tool to control Roombots, focusing on visualisation of structure and basic interaction:

**The Roombots Cockpit.**
Part II: Implementation of the Roombots Cockpit

Implementation goals:

- Allow the user to visualise a representation of the robots.
- Provide ways to select objects in the representation.
- Allow the user to change the value of a degree of freedom.
- Allow the user to change the value of ACMs.
Part II: Implementation of the Roombots Cockpit

Implementation goals:

- Allow the user to visualise a representation of the robots.
- Provide ways to select objects in the representation.
- Allow the user to change the value of a degree of freedom.
- Allow the user to change the value of ACMs.

What is not covered in the implementation:

- Construction of structures by the user.
- Physics.
- General movements.
Characteristics of the implementation:

Programming language: C++
Render engine: Ogre3D
Windows and widgets framework: Qt
Figure 10: General view of the Roombots Cockpit: Four modules on a grid.
Figure 11: Structure of six modules.
Further Improvements for the Roombots Cockpit:
Further Improvements for the Roombots Cockpit:

- Load the structure from an xml-file.
Further Improvements for the Roombots Cockpit:

- Load the structure from an xml-file.
- Determine which part of the structure has to turn.
Further Improvements for the Roombots Cockpit:

- Load the structure from an xml-file.
- Determine which part of the structure has to turn.
- Determine the consequences of connections / disconnections.
Further Improvements for the Roombots Cockpit:

- Load the structure from an xml-file.
- Determine which part of the structure has to turn.
- Determine the consequences of connections / disconnections.
- Detect collisions.
Further Improvements for the Roombots Cockpit:

- Load the structure from an xml-file.
- Determine which part of the structure has to turn.
- Determine the consequences of connections / disconnections.
- Detect collisions.
- Connect the interface to the real robot.
Further Improvements for the Roombots Cockpit:

- Load the structure from an xml-file.
- Determine which part of the structure has to turn.
- Determine the consequences of connections / disconnections.
- Detect collisions.
- Connect the interface to the real robot.
- Allow the user to construct the structure directly in the interface.
Conclusions of the project:

- First I made a case study to see what has already been done.
- Second I use this case study to elaborate a prototype of ideal interface for Roombots.
- Third I cut in the features of the prototype to determine what I would like to implement.
- Finally I implement a tool that fit the features requirements specified before.
Key points I found in designing an interface for modular robots are:
Key points I found in designing an interface for modular robots are:

- Choose an adapted representation for the visualisation.
Key points I found in designing an interface for modular robots are:

- Choose an adapted representation for the visualisation.
- Make the interactions user-friendly.
Key points I found in designing an interface for modular robots are:

- Choose an adapted representation for the visualisation.
- Make the interactions user-friendly.
- Define strategies to offer a lot of features without overloading the screen.
Personal conclusion

- Control interface for modular robots: Complex but interesting field.
Personal conclusion

- Control interface for modular robots: Complex but interesting field.
- This project allows me to:
Control interface for modular robots: Complex but interesting field.

This project allows me to:

- discover the domain of robotics.
Personal conclusion

- Control interface for modular robots: Complex but interesting field.
- This project allows me to:
  - discover the domain of robotics.
  - learn to use text, picture and video editing tools.
Personal conclusion

- Control interface for modular robots: Complex but interesting field.
- This project allows me to:
  - discover the domain of robotics.
  - learn to use text, picture and video editing tools.
  - improve my programming skills.
Personal conclusion

- Control interface for modular robots: Complex but interesting field.
- This project allows me to:
  - discover the domain of robotics.
  - learn to use text, picture and video editing tools.
  - improve my programming skills.

Thanks to prof. Ijspeert, my supervisors and the laboratory team for giving me the opportunity to discover these areas.
Goals and challenge
Ideal interface to control Roombots
Implementation of a tool to control Roombots

Final conclusion

Source of pictures

- Figure 1: S. Bonardi, Complexity reduction in optimization of modular robots locomotion using body/limbs recognition and spatial symmetries. 2010.

- Figure 2:
  - the cockpit: www rightbase nl/home.html
  - the Roombots rendered image: biorob epfl ch/research/modular

- Figure 3: birg epfl ch/page57461.html
- Figure 4: www.molecule. org
- Figure 5: www.ros.org/wiki/simulator_gazebo
- Figure 6: www.educational-freeware.com/freeware/lego-digital
- Figure 7: Home made.
- Figure 8: Source of base picture: birg epfl ch/page68135.html
- Figure 9: Source of base picture: birg epfl ch/page68135.html
- Figure 10: Screenshot of the Roombots Cockpit.
- Figure 11: Screenshot of the Roombots Cockpit.
- Figure 12: Screenshot of the Roombots Cockpit.
Goals and challenge
Ideal interface to control Roombots
Implementation of a tool to control Roombots
Final conclusion

Thank you for your attention.
## Schedule

<table>
<thead>
<tr>
<th>Monday:</th>
<th>08/09</th>
<th>13/09</th>
<th>20/09</th>
<th>27/09</th>
<th>04/10</th>
<th>11/10</th>
<th>18/10</th>
<th>25/10</th>
<th>01/11</th>
<th>08/11</th>
<th>15/11</th>
<th>22/11</th>
<th>29/11</th>
<th>06/12</th>
<th>13/12</th>
<th>20/12</th>
<th>27/12</th>
<th>03/01</th>
<th>10/01</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week: -2</td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Task:

- **Previous works**
  - Learning of principles of Roombots
  - Reviews of C++
  - Learning of Qt

- **Part I**
  - Brainstorming
  - Analyse of related works
  - Study of input management
  - Writing specifications of an ideal interface for Roombots
  - Learning of Ogre3D

- **Presentation**
  - Preparation of the midterm presentation

- **Part II**
  - Implementations, meshes and positioning
  - Input management in Ogre
  - Integration of CEGUI*
  - Integration of Qt
  - Modifications to use colors
  - Implementation of Interactions
  - Implementation of modifiers
  - Refactoring of the code
  - Modify the comments to be Doxygen compatible

### Finalisation
- Definition of the structure of the final report
- Writing of the final report

### Presentation
- Finalise the project
- Preparation of the final defence

* the integration was not successful. So I used Qt instead of CEGUI
Usage of modes

- Assembly
- Low-level
- Sequence editor
- High-level
- Record of sequence
Target platform

- Smartphone
- TabletPC
- PC
Target user

Lay user

Expert user

Features

Demos

Testing

Development
Switch between modes

Mode 1

Mode 2

Mode 3

Mode 4

Mode 1

Mode 2

Mode 3

Mode 4
Hierarchy of classes representing Roombots objects

RBOBJECT

RBGraphicObject  RBGrid  RBModule  RBSphere

RBAbstractConnector  RBCentralDOF  RBConnection  RBHalfSphere

RBAactiveConnector  RBConnector