A Racket-Based Robot to Teach First-Year Computer Science

K. Androutsopoulos, N. Gorogiannis, M. Loomes, M. Margolis, G. Primiero, F. Raimondi, P. Varsani, N. Weldin, A. Zivanovic

Department of Computer Science
School of Science and Technology
Middlesex University, London
http://www.cs.mdx.ac.uk

Middlesex University

European Lisp Symposium 2014
1. The really (in)famous precedent

2. The context

3. Racket & Mirto

4. Applications

5. Assessment & Evaluation
1 The really (in)famous precedent

2 The context

3 Racket & Mirto

4 Applications

5 Assessment & Evaluation
1. The really (in)famous precedent

2. The context

3. Racket & Mirto

4. Applications

5. Assessment & Evaluation
Computer Science at Middlesex University

- New Computer Science programme for the academic year 2013/2014

- Teach students how to become autonomous learners

- *Racket*: solid mathematical background and language-independent programming skills

- *Real hardware*: Arduino, Raspberry Pi, and the Robotic Platform *Mirto*

- Completely revised delivery and assessment methods:
  - no modules or courses
  - activities run seamlessly across the projects
  - Assessment through *Student Observable Behaviours (SOBs).*
Week structure

- **General Lecture**: introduction to topic and related project;

- **Design Workshop**: design skills in software or hardware, systems engineering (UML), HCI, security;

- **Programming Workshop**: exercises, master-classes, coaching sessions, restricted to Racket;

- **Physical Computing Workshop**: from simple logic gates to microcontrollers (Arduino) and other specialist devices controlled through Racket;

- **Synoptic Workshop**: 4 hours to investigate foundations, design, build, test and discuss projects.
Three Projects

1. traffic light system
2. dungeon game
3. MIddlesex RoboTic PlatfOrm – MIRTO
The Platform

Base platform:

- two HUB-ee wheels with motors and encoders (to measure actual rotation)
- front and rear castors
- two bump sensors
- an array of six infra-red sensors
- a rechargeable battery pack
- an Arduino microcontroller board

Top layer:

- a Raspberry Pi connected to the Arduino
- Linux with Racket (current version 5.93)
- USB-WiFi adapter for SSH and network
- Additional: cameras, microphones and text to speech with speakers
MIRTOlib

- Library developed by the teaching team
- Takes care of low-level serial communications
  \[(\text{send-sysex-int-msg} \ #\text{x7D} \ 5 \ \text{power})\]
- Students deal only with high-level Racket programs
  \[(\text{define} \ (\text{setMotors} \ \text{speed1} \ \text{speed2}))\]
  \[(\text{setMotor} \ 0 \ \text{speed1})\]
  \[(\text{setMotor} \ 1 \ \text{speed2}))\]
- Students can read IR values with
  \[(\text{getIR} \ 2)\]
The really (in)famous precedent

The context

Racket & Mirto

Applications

Assessment & Evaluation
(define proportional (- error 2000))

;;; Integral component: we reset to 0 when error is 0
(cond ( (= 0 proportional) (set! intError 0))
    (else (set! intError (+ intError proportional)))
)

;;; we assume dt constant, so this is just the difference
;;; If derivative < 0, we moved to the left of the line
(define derivative (- proportional (- prevError 2000)))
(set! prevError error)

;;; The correction is the sum of a proportional component,
;;; integral component and a derivative component.
(define correction (+ (* Kp proportional)
                        (* Ki intError)
                        (* Kd derivative)))

(cond
  ((> correction 0) ;; we are to the right
   (setMotors PWR (- PWR correction)))

  (else ;; we are to the left
   (setMotors (+ PWR correction) PWR))
)

(Middlesex University)
Others

- Speech-recognition: PocketSphinx connected to Racket
- Graphical Interface using X on Pi
- Web-server running on Pi
- Twitter controlled Robot
1. The really (in)famous precedent

2. The context

3. Racket & Mirta

4. Applications

5. Assessment & Evaluation
SOBs

1. **Threshold level**: essential to pass the year.

2. **Typical level**: expected for a good honours degree.

3. **Excellent level**: identifies outstanding achievements.
### SOBs Tool

#### Figure: Entering and searching SOBs

- **SOB ID**: 1
- **Level**: Threshold
- **Topic**: Racket
- **SOB**: Enter simple expressions, including nested brackets and symbols bound to values into the interaction window, execute them and explain what is happening. **Keywords**: expression | binding | block 1
- **Start Date**: 07.10.2013
- **Expected Completion Date**: 18.10.2013

- **SOB ID**: 2
- **Level**: Threshold
- **Topic**: Racket
- **SOB**: Use simple list commands including list, first, rest, cons, reverse, length and append to solve problems posed in a very explicit way. **Keywords**: lists | block 1
- **Start Date**: 14.10.2013
- **Expected Completion Date**: 25.10.2013

- **SOB ID**: 3
- **Level**: Threshold
- **Topic**: Racket
- **SOB**: Use define, lambda and cond, with other language features as appropriate, to create and use a simple function. **Keywords**: define | lambda | cond | block 1
- **Start Date**: 14.10.2013
- **Expected Completion Date**: 25.10.2013
## SOBs Tool

### Figure: Student list with SOBs

<table>
<thead>
<tr>
<th>S.No</th>
<th>Student Number</th>
<th>First Name</th>
<th>Last Name</th>
<th>Email</th>
<th>Threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>M00...</td>
<td></td>
<td></td>
<td>@live.mdx.ac.uk</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>M00...</td>
<td></td>
<td></td>
<td>@live.mdx.ac.uk</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>M00...</td>
<td></td>
<td></td>
<td>@live.mdx.ac.uk</td>
<td>5 !</td>
</tr>
<tr>
<td>4</td>
<td>M00...</td>
<td></td>
<td></td>
<td>@live.mdx.ac.uk</td>
<td>0 ✔️</td>
</tr>
<tr>
<td>5</td>
<td>M00...</td>
<td></td>
<td></td>
<td>@live.mdx.ac.uk</td>
<td>0 ✔️</td>
</tr>
<tr>
<td>6</td>
<td>M00...</td>
<td></td>
<td></td>
<td>@live.mdx.ac.uk</td>
<td>0 ✔️</td>
</tr>
<tr>
<td>7</td>
<td>M00...</td>
<td></td>
<td></td>
<td>@live.mdx.ac.uk</td>
<td>0 ✔️</td>
</tr>
</tbody>
</table>
SOBs Tool

**Figure** : Observing a SOB for a student

(Middlesex University)
SOBs Tool

Figure: Student view: position with respect to class
Evaluation & Conclusion

- 85% success rate
- Average 90% attendance
- All students have progressed beyond threshold SOBs
- https://github.com/fraimondi/myrtle/
  (software and design files)
Conclusion

Thanks and feel free to come and see MIRTO!