SLego: A Squeak Implementation of Lego Mindstorms

Alexandre Bergel
Institut fuer Mathematik und Informatik
University Bern

bergel@iam.unibe.ch

Plan

1. Goal: Mapping between Morph and Lego Mindstorms
2. What is the Lego Mindstorms kit?
3. Slego
    1. Environment
    2. Compiler
4. Demo
5. Future

Mapping between Lego Mindstorms and Morphics

- Define a behavior for concrete robots using morphics
- A morph is a graphical component in Squeak
- Once a morph behavior defined, the goal is to make a Lego Mindstorms system acting as it
- 3 steps are needed :
  - Extract the behavior of a morph
  - Compile it for the Lego Mindstorms
  - Download the bytecode resulting to this last

Introduction of Lego Mindstorms 1/2

- Is a robot-kit-in-a-box product from Lego
- Originally made at the MIT
- It consists of :
  - A computer module (called **RCX**)
  - An inventory of many TECHNICS part (elementary bricks)
  - Sensors and motors
- An environment made by Lego which defines a visual langage (having `if`, `loop`, ...)

Introduction of Lego Mindstorms 2/2

- The RCX is composed of:
  - 3 inputs for sensors (such as push buttons or light detectors)
  - 3 outputs for motors or lights
  - A microprocessor
  - 32 kb of RAM

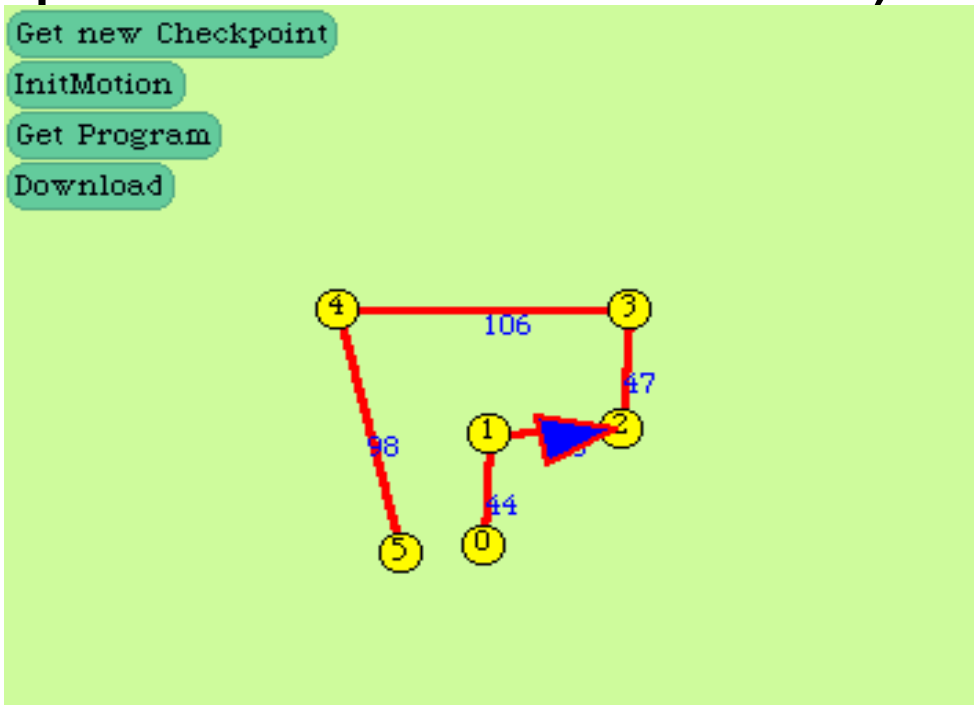
- The communication between a computer and a RCX is done via radio or infra-red

Slego

- Slego = playarea + compiler + serial port communication
- Two kinds of communication
- *Direct use* needs to have the RCX always connected to the computer. Each command sent to the RCX is immediately executed
- *Program use* downloads bytecodes from a computer and delays their execution

Slego

- Playarea is used for defining the behavior of a morph. This behavior is translated into a list of instructions
- Current implementation only supports cars (two independent coaxial wheels)



Compiler

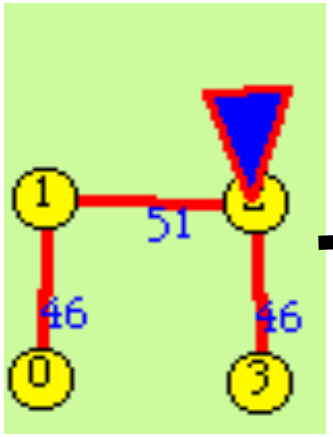
- *Takes* as input a program composed of high level instructions such as: *turnLeft, turnRight, walk, ...*
- *This program is compiled in an intermediate language composed of primitive instructions such as: turnOnLeftMotor, turnOffLeftMotor, sound, ...*
- *This last language acts as an assembler*

Takes as input a program composed of high level instructions such as: turnLeft, turnRight, walk, ...

This program is compiled in an intermediate language composed of primitive instructions such as: turnOnLeftMotor, turnOffLeftMotor, sound, ...

This last language acts as an assembler

Compiler: an example



(walk 46)
(turnRight 90)
(walk 51)
(turnRight 90)
(walk 46)

TurnOnLeftMotor
TurnOnRightMotor
wait 5
TurnOffLeftMotor
TurnOffRightMotor

TurnOnLeftMotor
wait 2
TurnOffLeftMotor

.
.
.

0010
1110
1010
1011
...

RCX

Demo

- Web: http://minnow.cc.gatech.edu/squeak/2412
- Mail: `bergel@iam.unibe.ch`

Future

- Manage new sensors in Slego
- Give accuracy
- Extends Slego to other robot systems (antlj, ...)