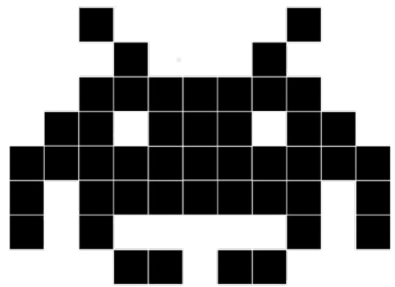
A Teaching System To Learn Programming:
The Programmer's Learning Machine

Martin Quinson, Gérald Oster
Lorraine University, France

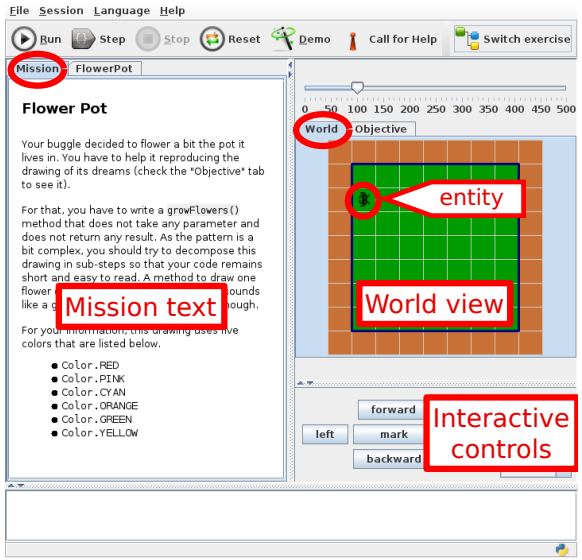
July 8., 2015
itiCSE 2015, Vilnius

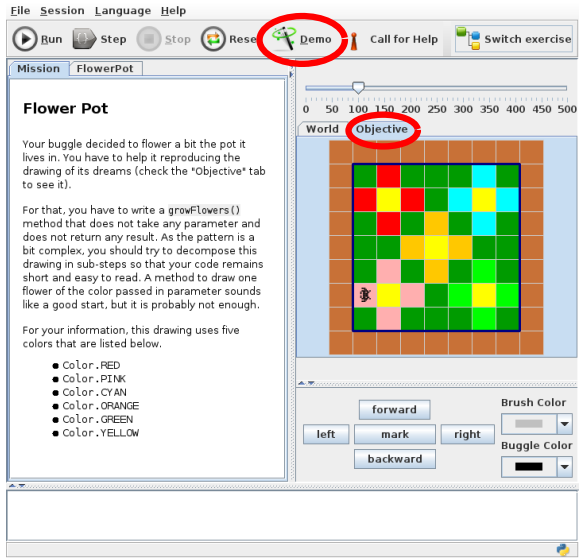


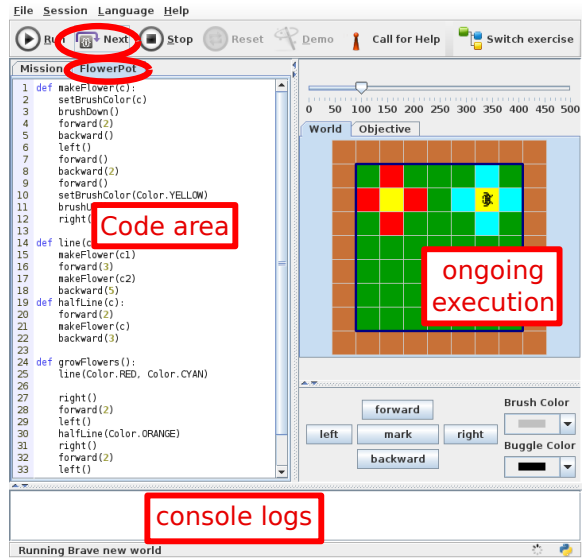
The Programmer's Learning Machine

- Interactive Exerciser to Learn Programming
- Loosely tutored practical sessions at University, College

Short Feedback Loop

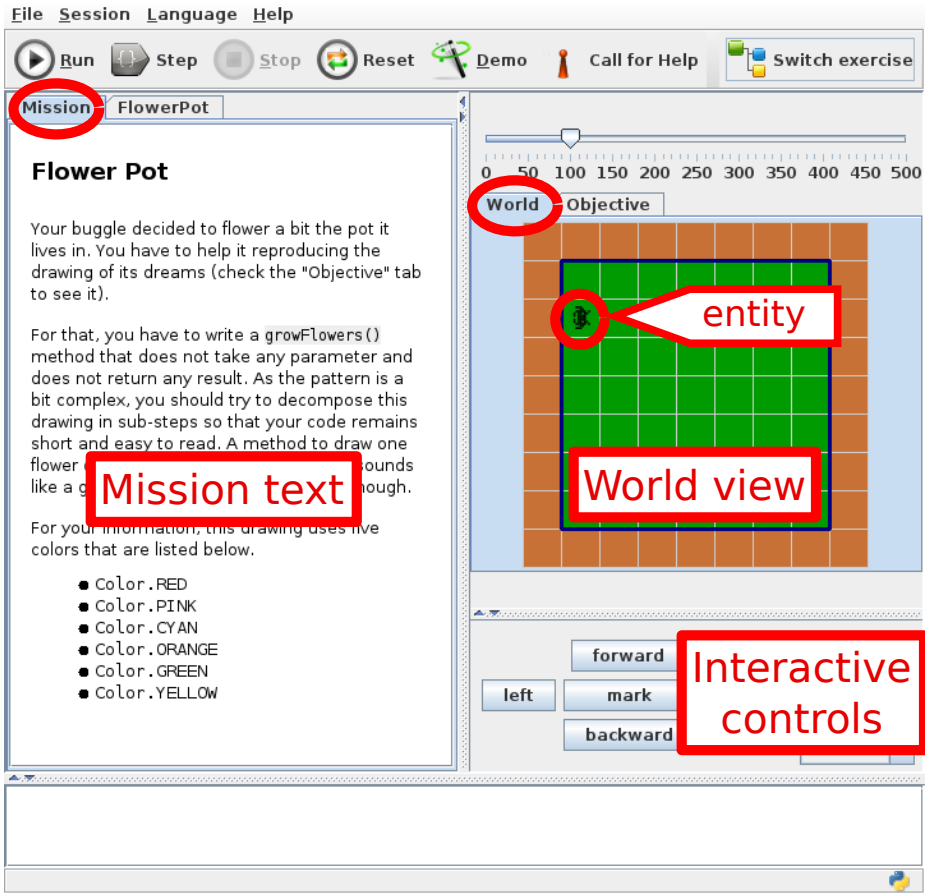






The Programmer's Learning Machine

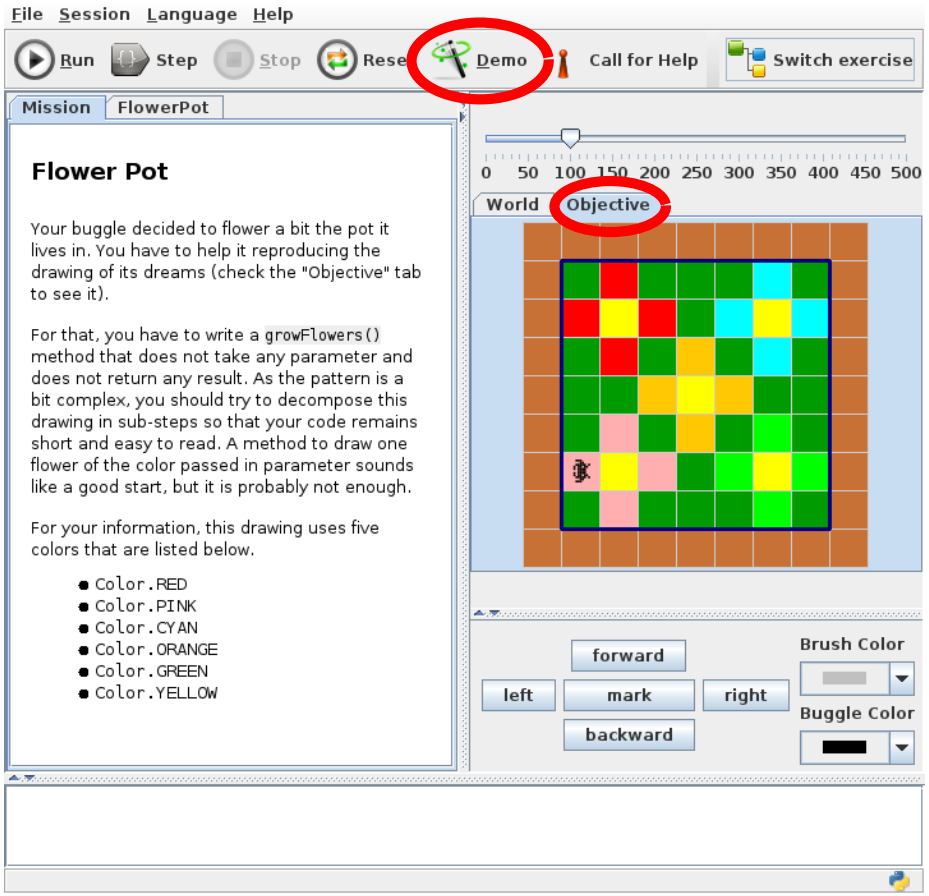
- Interactive Exerciser to Learn Programming
- Loosely tutored practical sessions at University, College



Read the mission. Inspect the initial state

The Programmer's Learning Machine

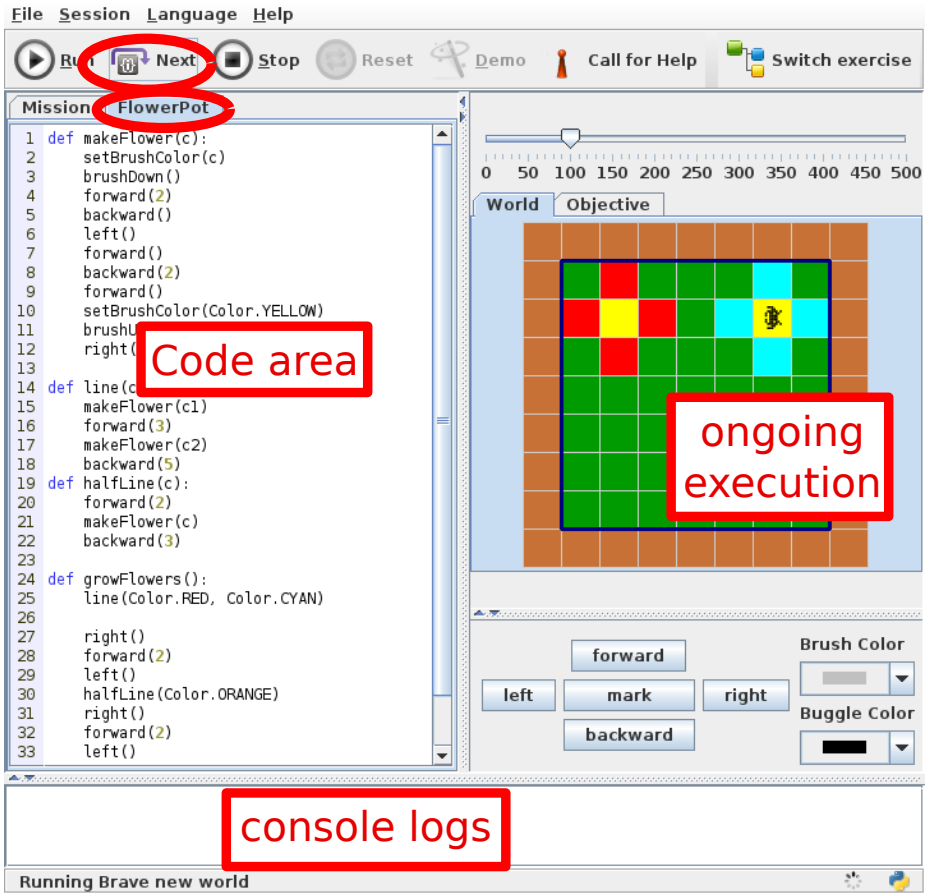
- Interactive Exerciser to Learn Programming
- Loosely tutored practical sessions at University, College



Compare to the objective state. Check the demo?

The Programmer's Learning Machine

- Interactive Exerciser to Learn Programming
- Loosely tutored practical sessions at University, College



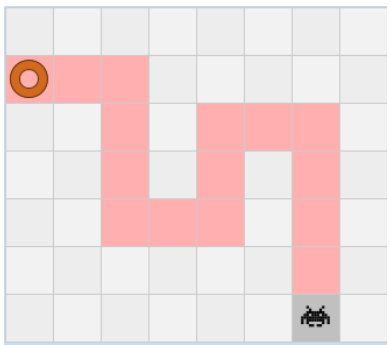
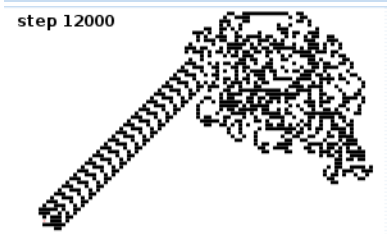
Write your code; Run it: it moves!

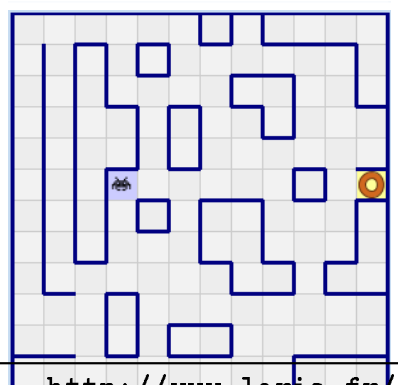
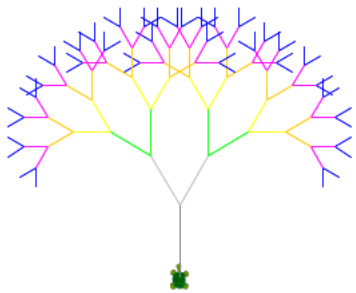
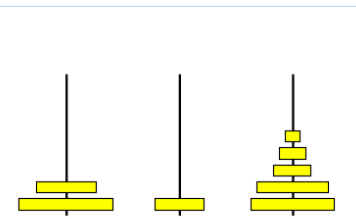
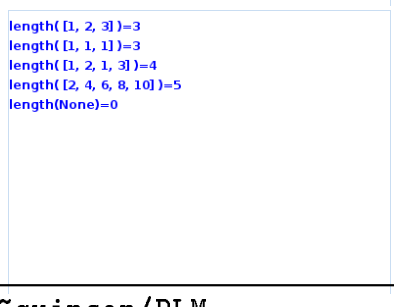
Yet Another IDE for Learners??

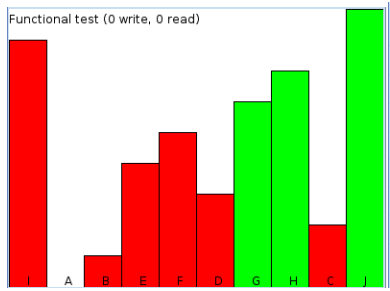
- Exerciser $\neq$ IDE: Many exercises, Coherent progressions
- Multi-language: Java/Scala/Python (+ C/javascript/Blockly/Ruby)
- Multi-lingual: English/French/Brasilian (+ Italian/Chinese)
- Multi-universe: Several kind of Micro-Worlds

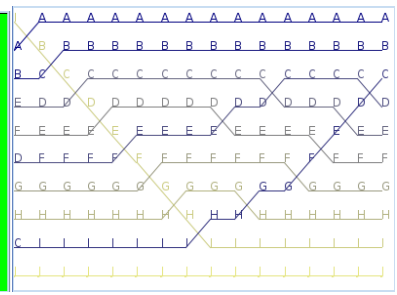
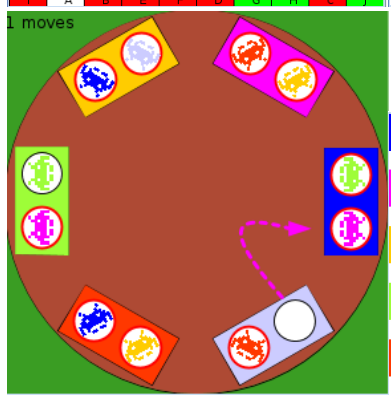
Yet Another IDE for Learners??

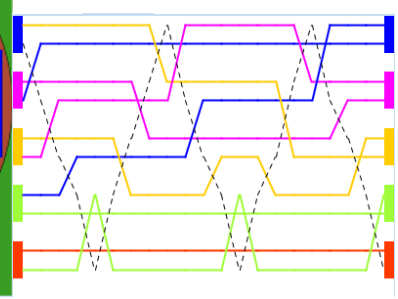
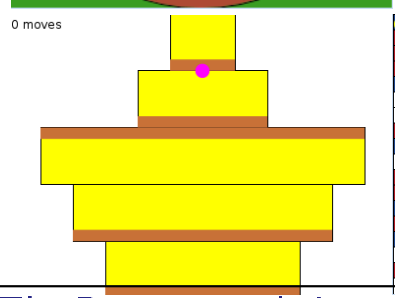
- Exerciser ≠ IDE: Many exercises, Coherent progressions
- Multi-language: Java/Scala/Python (+ C/javascript/Blockly/Ruby)
- Multi-lingual: English/French/Brasilian (+ Italian/Chinese)
- Multi-universe: Several kind of Micro-Worlds

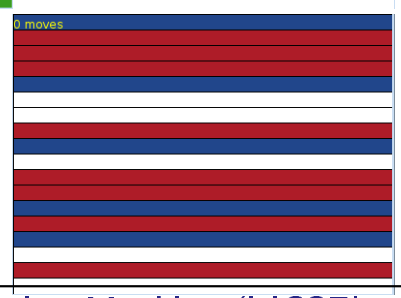



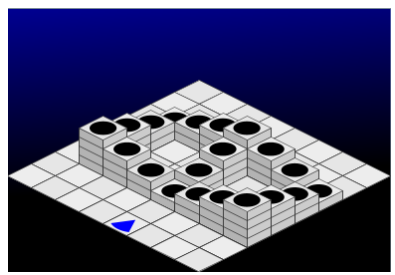
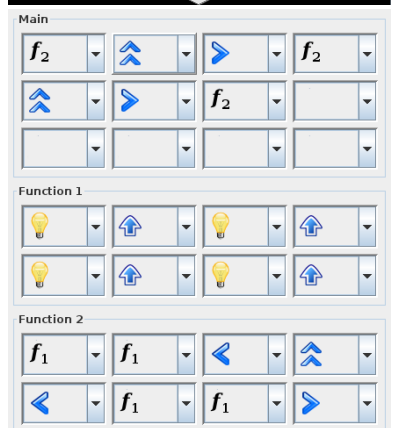





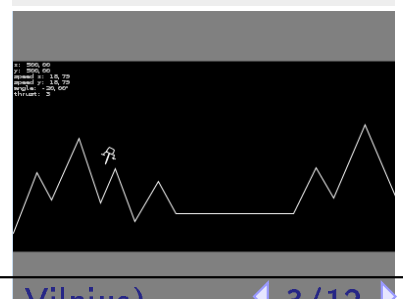






Teaching Material: CS0

Tactical Programming

- Solve all initial syntax difficulties
- Don't get distracted when the algorithms become hairly

Imperative Kernel (regardless of the language)

- Instructions & Comments; Conditionals
- *while* loops; *switch cases*
- Variables
- *for* loops and *do/while*
- Methods, Functions, Parameters
- Arrays
- Many application and Summative exercises

200 scenarized exercises (10 to 50+ hours)

Teaching Recursion in CS1

Recursion made Simple

- Split problem; Have a friend solve sub-problem; Gather solutions

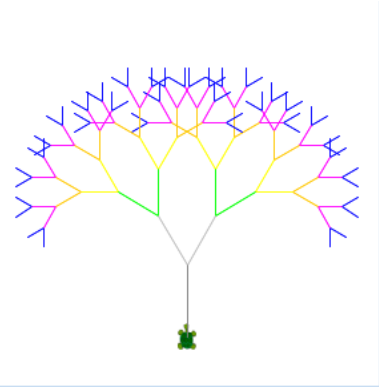
Teaching Recursion in CS1

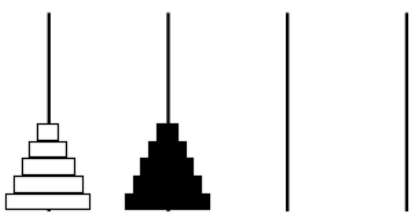
Recursion made Simple

- Split problem; Have a friend solve sub-problem; Gather solutions

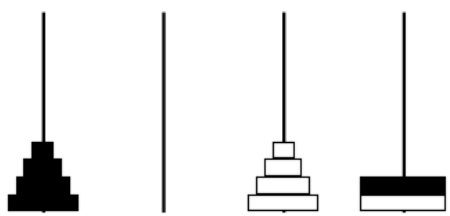
Supporting Micro-worlds

- Recursive Lists: length(), isMember(), etc
- Turtle Recursion: Trees, Fractals
  - `subtree()` provided, toward decomposition
- Hanoï and variations toward decomposition
  - Linear, Cyclic, Bicolor, Tricolor, etc

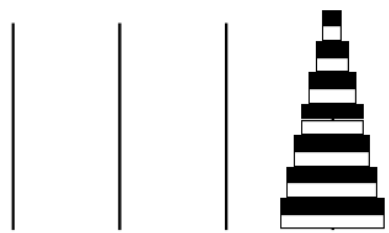




Initial Settings



Intermediate Step

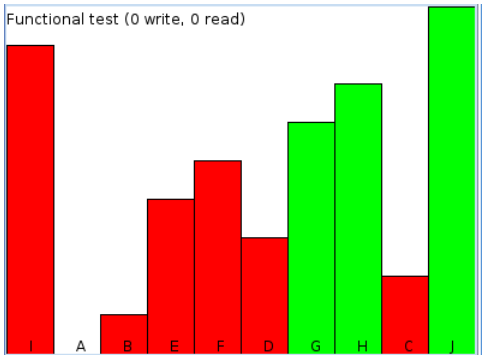


Final Goal

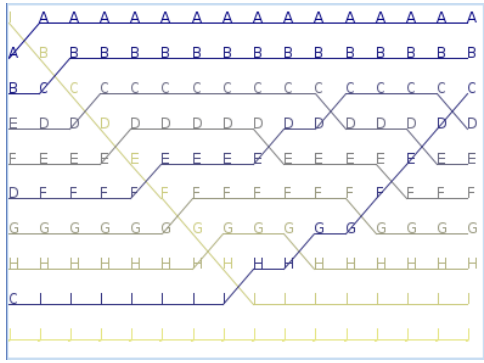
Teaching Sorting in CS1

Discover: go through a dozen algorithms (enforces #operations)

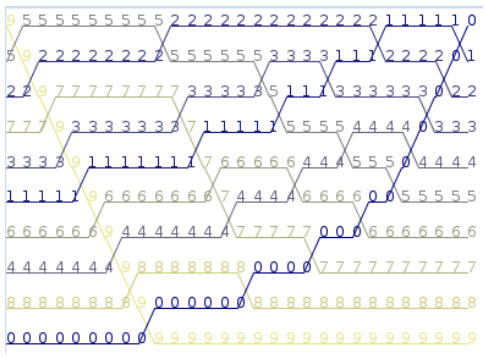
- State view, and History view to help understanding



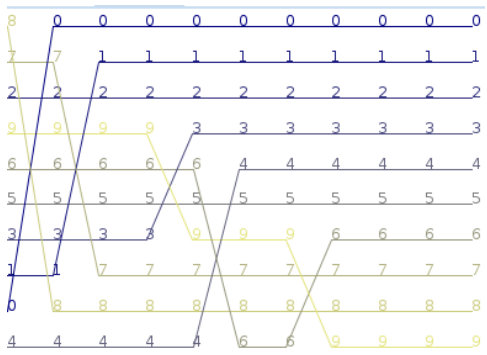
State view



Selection sort

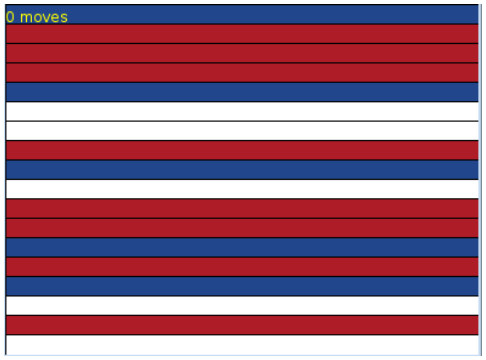


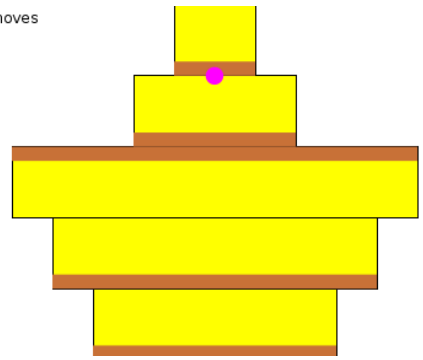
Bubble sort

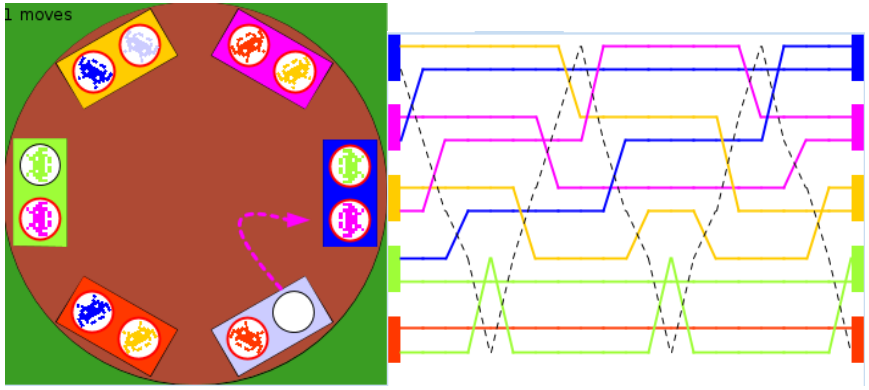


Selection sort

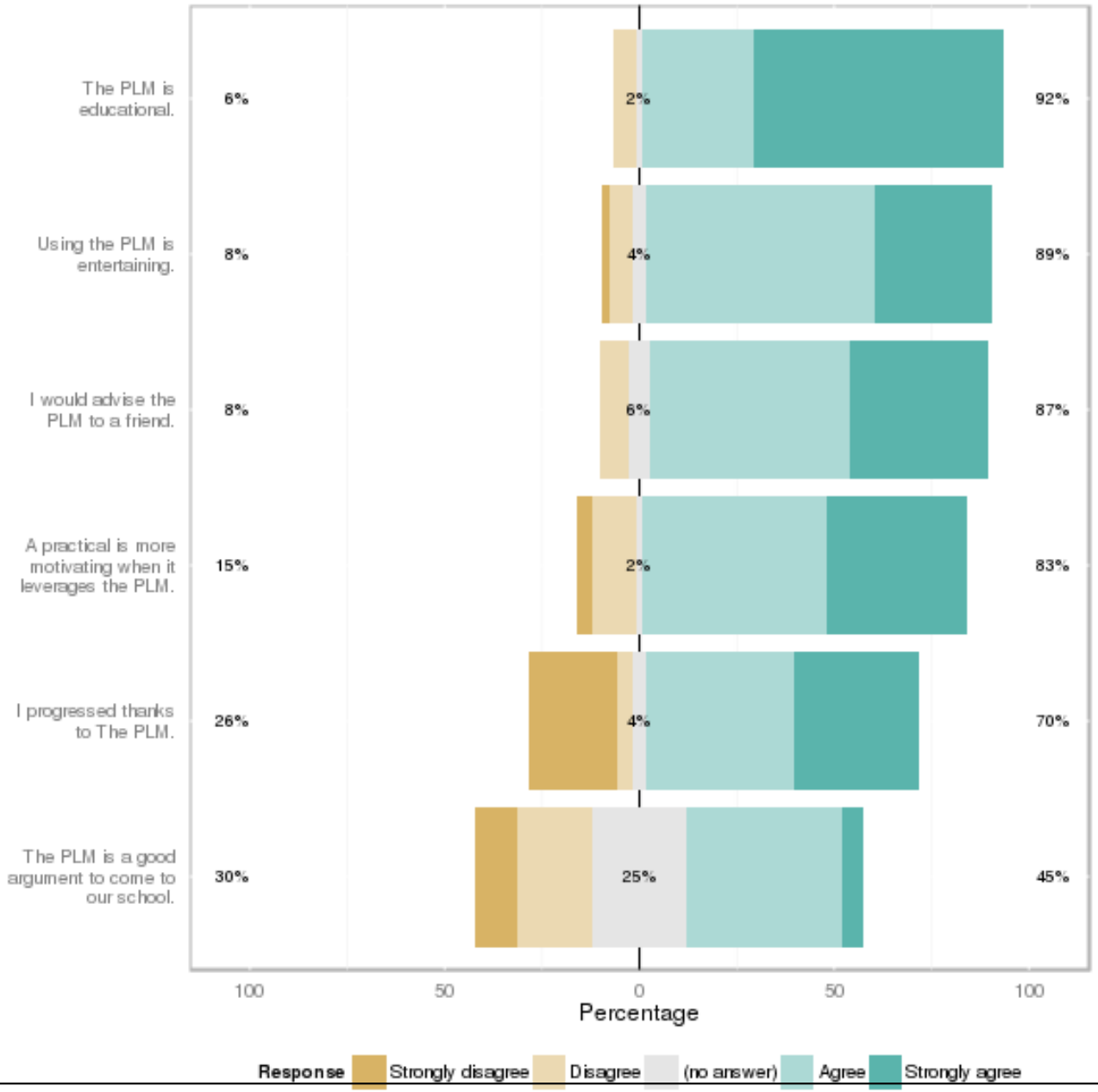
Apply: Dutch Flag, Pancakes or Pebble Motion Problem







Students' Feedback

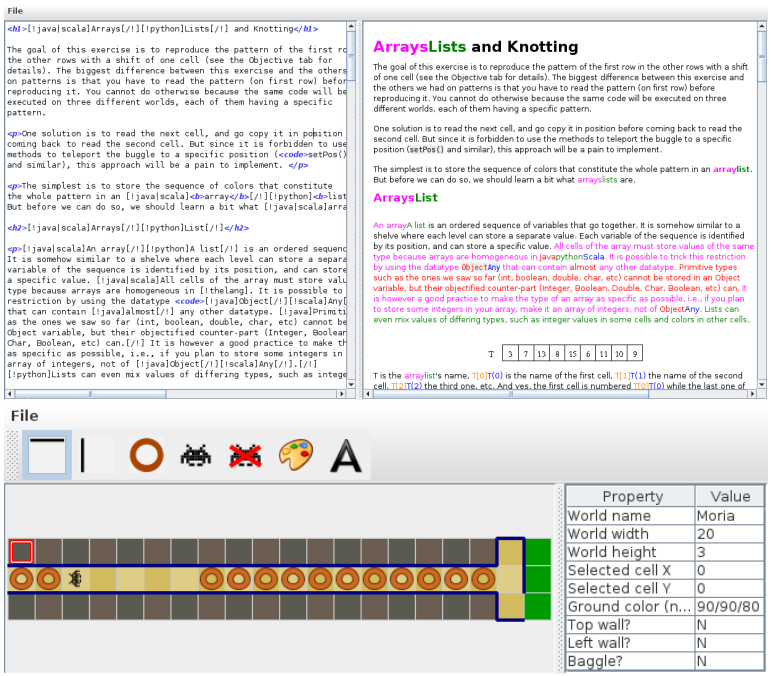


Creating new Material

- Lesson: Just a graph of exercises (list + dependencies)

Creating new Material

- Lesson: Just a graph of exercises (list + dependencies)
- Exercise: Mission + Initial Worlds + Correcting Entities

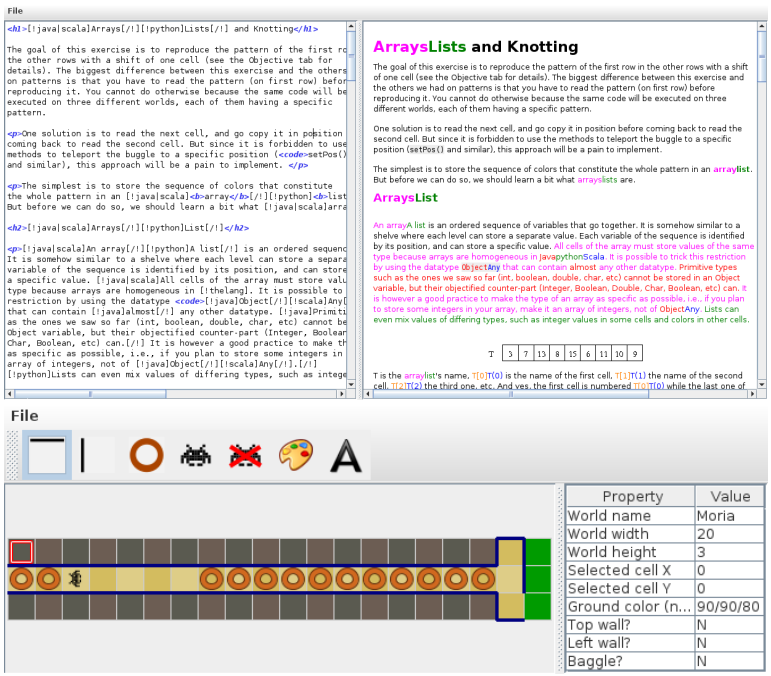



```
public class SlugTrackingEntity
             extends SimpleBuggle {

   // Some additional (hidden) code
   /* BEGIN TEMPLATE */
   boolean isFacingTrail() {
      // Write your code here
      /* BEGIN SOLUTION */
      if (isFacingWall())
         return false;
      forward();
      boolean res =
         getGroundColor().equals(Color.green);
      backward();
      return res;
      /* END SOLUTION */
   }
   /* END TEMPLATE */
}
```

Creating new Material

- Lesson: Just a graph of exercises (list + dependencies)
- Exercise: Mission + Initial Worlds + Correcting Entities

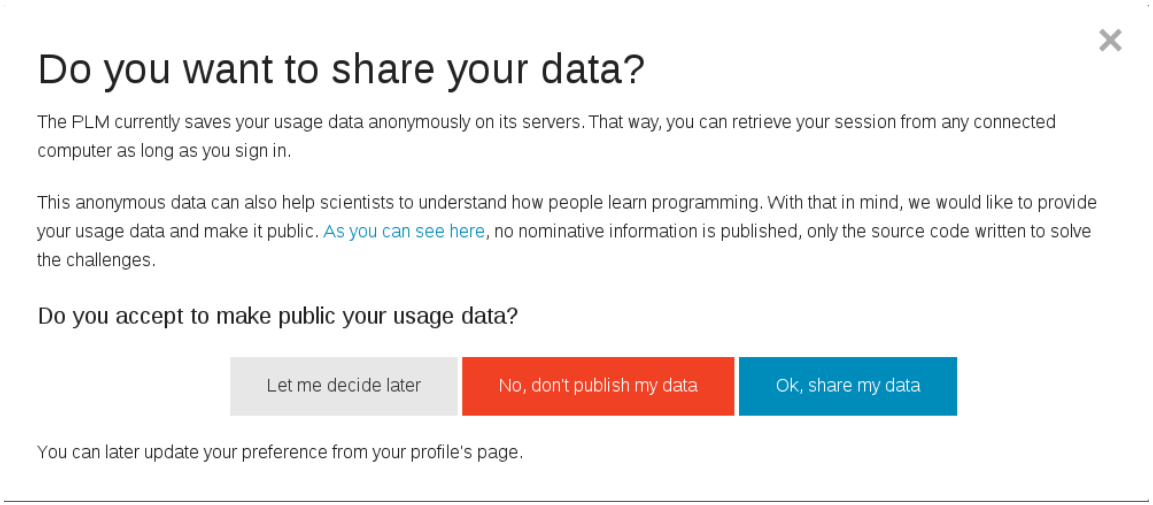



```
public class SlugTrackingEntity
             extends SimpleBuggle {

   // Some additional (hidden) code
   /* BEGIN TEMPLATE */
   boolean isFacingTrail() {
      // Write your code here
      /* BEGIN SOLUTION */
      if (isFacingWall())
         return false;
      forward();
      boolean res =
         getGroundColor().equals(Color.green);
      backward();
      return res;
      /* END SOLUTION */
   }
   /* END TEMPLATE */
}
```

- Micro-world: World + View + Entity + ControlPanel
  - 400 to 1400 lines

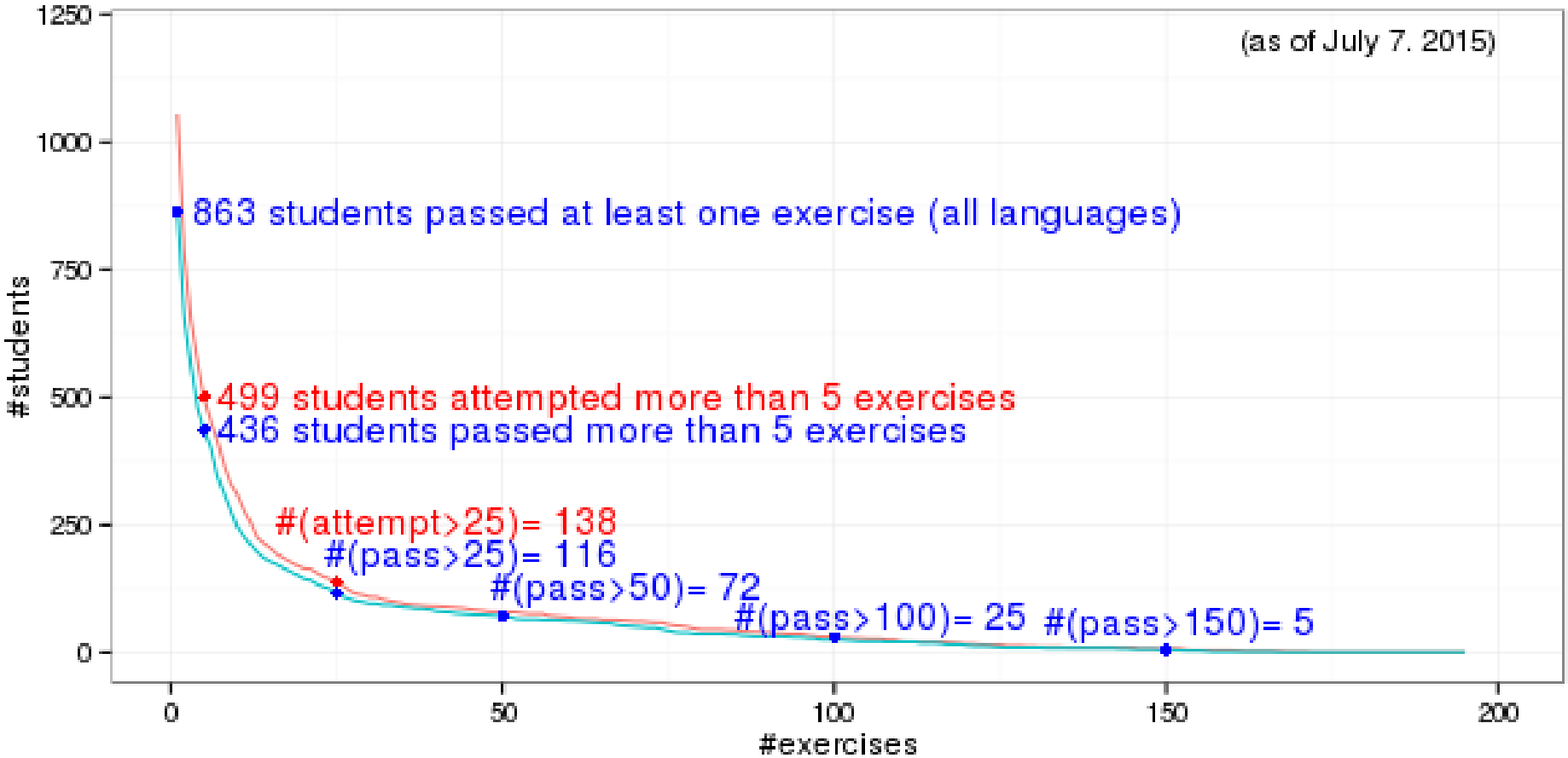
Helping Researchers on CSER



- Each attempt is saved anonymously, so that it can be rerun
- 133,636 code submissions (as of yesterday):
  - 17,480 success; 42,693 compilation errors; 73,463 failures
  - Scala: 87,514 (66%); Python: 28,066 (21%); Java: 16,669 (13%)

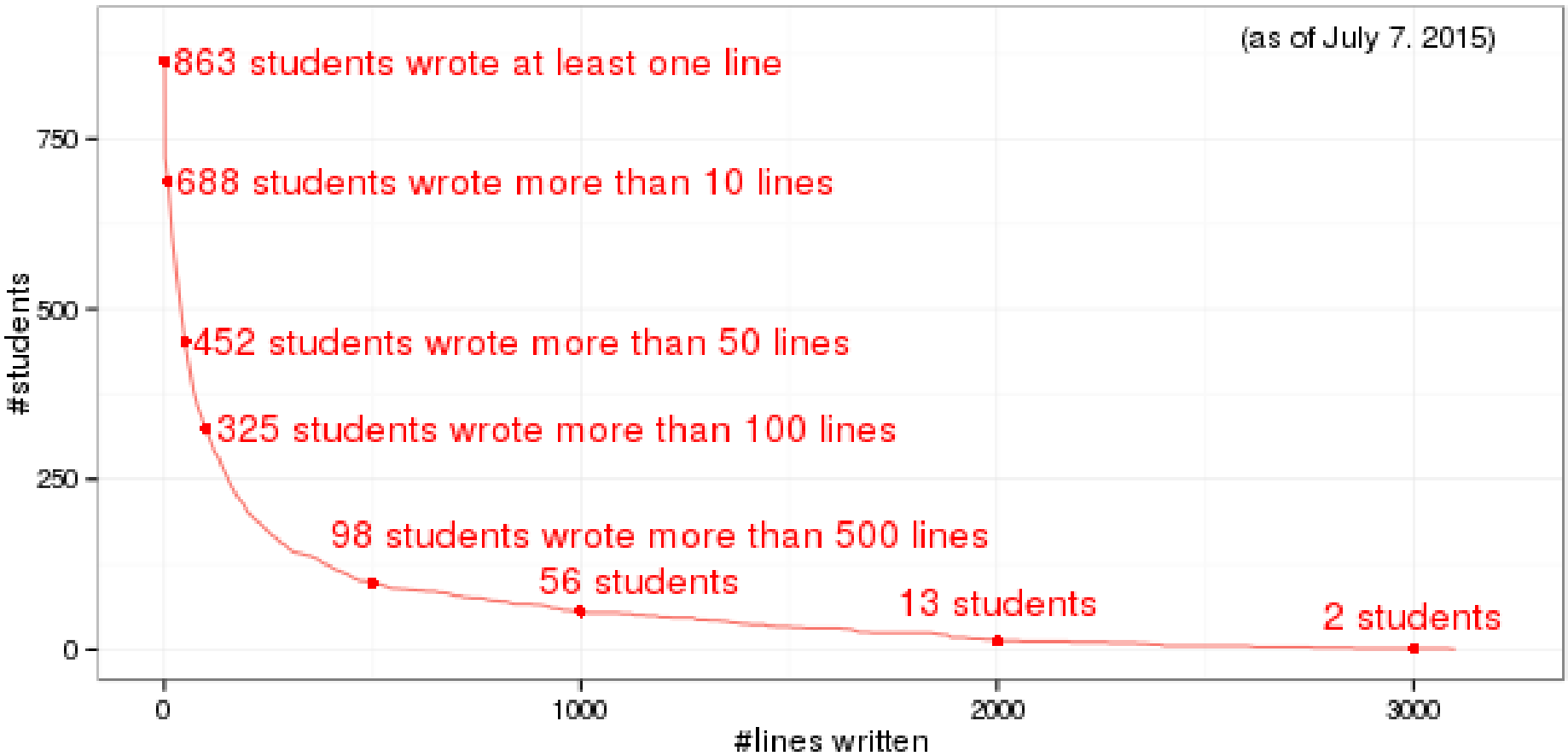
Exhibiting Longer Learning Path

Cumulative count of students that passed more than X exercises



Exhibiting Longer Learning Path

Cumulative count of students that wrote more than X lines of code



- `https://github.com/BuggleInc/PLM-data`
- We provide a crawler — contact me.

Current Work on the PLM

From a Java application to a Web-based service

- Beta version: `http://plm.telecomnancy.univ-lorraine.fr:9000/`
- Tests, Distributed Judges and Dockers ongoing
- Currently at watershed! In production in September

Teacher Console

- Dashboard of Class Progress
- Alerts: Detect stuck students (hard!)

Personalized Remediation for the MOOCs

- Clusterize the logic errors (not compile errors)
- Provide an adapted text to common ones, shown automatically

Conclusions

The PLM helps several audiences

- Learners: work at their own peace with a serious «game»
- Teachers: automated+monitoring $\rightsquigarrow$ more time with students
- Authors: reuse of non-functional code, instrumented feedback
- Researchers: corpus of data

Future Works:

- New worlds, new exercises, new languages, polishing
- Integrated Exercise and Material Editor
- Integrated Q&A System, community-based programming resource

Join us!

Martin.Quinson@loria.fr