Robot ON!
A Serious Game for Improving Programming Comprehension

Michael Miljanovic, Jeremy Bradbury
Software Quality Research Lab
University of Ontario Institute of Technology
Oshawa, Ontario, Canada
{michael.miljanovic, jeremy.bradbury}@uoit.ca

http://www.sqrlab.ca
Motivation

• Many students struggle in introductory computer programming courses
• Students especially struggle to understand code they did not write themselves
• Most serious games about programming involve players writing their own code
  – Students lack programming comprehension skills
Robot ON!

• Players take the role of a programmer trying to activate a series of ‘Mech’ systems
• Controlling a robotic avatar, the player must complete **puzzle tasks** to finish each level
  – Each puzzle is based on understanding existing code
Robot ON!

• Two dimensional
  – The player can run, climb, or fall through lines of code
  – Mimics a real code environment by allowing navigation similar to a word processor cursor

• Tools provided to the player allow them to interact with different puzzle tasks
  – Each line of code may have any number of puzzle tasks
// This program calculates pi.

// This program is a prototype!

string my_word = "hello there!";

int my_number = 2;

if (my_number <= 1) {
    my_number = 0;
} else {
    my_number = 10;
}

bool varname;

/*mybool = true;*/

/*mybool is true;*/
Robot ON! Puzzle Tasks

• Puzzle tasks focus on understanding key programming concepts:
  – Variable values
  – Data type identification
  – Program statements
  – Control flow
/**/ This program calculates pi.

/**/ This program is a prototype!

    string my_word = "hello there!";

    int my_number = 2;
    if (my_number <= 1){
        my_number = 0;
    }
    else{
        my_number = 10;
    }

        bool varname;
        /*mybool = true;*/
        /*mybool is true;*/
Robot ON! Puzzle Tasks

- Puzzle tasks focus on understanding key programming concepts:
  - Variable values
  - **Data type identification**
  - Program statements
  - Control flow
```cpp
// This program calculates pi.

// This program is a prototype!

string my_word = "hello there!";

int my_number = 2;

if (my_number <= 1){
    my_number = 0;
}

else{
    my_number = 18;
}

bool varname;

//mybool = true;++

//mybool is true;+

// Tasks:
// ACTIVATE the beacons in the right order.
// CHECK the values of the variables.
// NAME the variables with appropriate names.
// COMMENT the lines that describe the code,
// UN-COMMENT the code that is correct.

Available Tools:

Activator: 1

Time remaining: 946 seconds
```
Robot ON! Puzzle Tasks

• Puzzle tasks focus on understanding key programming concepts:
  – Variable values
  – Data type identification
  – Program statements
  – Control flow
/**< This program calculates pi. 

/**< This program is a prototype!

string my_word = "hello there!";

int my_number = 2;

if (my_number <= 1){
    my_number = 0;
}

else{
    my_number = 18;
}

bool varname;

/**<mybool = true;++/

/**<mybool is true;++/
Robot ON! Puzzle Tasks

• Puzzle tasks focus on understanding key programming concepts:
  – Variable values
  – Data type identification
  – Program statements
  – Control flow
This program calculates pi.

This program is a prototype!

string my_word = "hello there!";

int my_number = 2;

if (my_number <= 1){
    my_number = 0;
}

else{
    my_number = 18;
}

bool varname;

/*mybool = true;*/

/*mybool is true;*/
Customizing Robot ON!

• Robot ON! was designed to be customizable and extendible by course instructors
  – Levels can be created for different programming languages
  – Puzzle tasks can be created to target specific students and learning materials
  – Creating puzzle tasks provides students a chance to learn through failure!
<code>
<badcomment size="1" righttext = "This program counts from 1 to 10!">
   This program counts from 10 to 1.
</badcomment>
<oncomment size="1">
   "This program counts from 1 to 10!"
</oncomment>
void main()
{
   for(int i = 1;i &lt;=10; i++)
   {
      cout &lt;&lt;i &lt;&lt; " " &lt;&lt; endl;
      if (i % 5 == 0)
      {
         cout &lt;&lt; "div by 5!" &lt;&lt; endl;
         &lt;beacon actnums="0,1"&gt;&lt;/beacon&gt;
      }
   }
}</code>
Evaluating Robot ON!

• Is the Robot ON! game *playable* by undergraduate students?

• Does Robot ON! give players sufficient *skills* to work with a new programming language (i.e., achieve learning outcomes)?

• Do students *enjoy* playing the Robot ON! game?
Future Work

• **Planned Evaluation**
  – Two part study of usability and learning outcomes

• **Incorporation into introductory courses**
  – Robot ON! is intended to supplement existing courses

• **Open-source**
  – Robot ON! is available to other institutions for community improvement on Github

• **Wider application of CS game-based learning**
  – Robot ON! game could be adapted to include other software concepts (e.g. parallelization)
Summary

• Robot ON! is a **learning supplement** for introductory programming courses to aid students in developing programming comprehension skills

• Robot ON! can be a starting point for the creation of an **improved** game for teaching debugging or other concepts

• Robot ON! can be **customized and extended** by instructors in accordance with their learning materials
Robot ON!

A Serious Game for Improving Programming Comprehension

Michael Miljanovic, Jeremy Bradbury
Software Quality Research Lab
University of Ontario Institute of Technology
Oshawa, Ontario, Canada
{michael.miljanovic, jeremy.bradbury}@uoit.ca

http://www.sqrlab.ca

ROBOT ON! IS AVAILABLE AT:
https://github.com/sqrlab/RobotON