Logo++
*An educational language*

Project Manager: Kam Lai
Language Guru: Yi-Yin Chang
System Architect: Jiao Li
System Integrator: Xinyuan Wang
System Tester: Bohong Zhao
Motivation

• Students should learn to code.

http://www.youtube.com/watch?v=nKlu9yen5nc
Problems of the current languages

• C – too much
• Scratch – not easy
• Traditional Logo – not motivating

Student-friendly programming language needs to be:

– Feedback-driven
– Easy to write/read
– Progressive learning
– Motivating

Logo++
Quick example

set x 144
repeat 5 { FD 100 RT x }
Syntactic constructs

• Easy to start with:
  – Simple syntax
  – Default data type
  – No semicolon

• Positive transfer for learning other programming languages:
  – Conditional, iteration and recursion statement
  – Function definition
# Syntactic Definitions

<table>
<thead>
<tr>
<th>Format</th>
<th>Alias</th>
<th>Functionality</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forward number</td>
<td>FD</td>
<td>Moves forward by given distance</td>
<td>Forward 100</td>
</tr>
<tr>
<td>Back number</td>
<td>BK</td>
<td>Moves backward by given distance</td>
<td>Back 80</td>
</tr>
<tr>
<td>Left number</td>
<td>LT</td>
<td>Turns left by given angle</td>
<td>Left 50</td>
</tr>
<tr>
<td>Right number</td>
<td>RT</td>
<td>Turns right by given angle</td>
<td>Right 90</td>
</tr>
<tr>
<td>SetSpeed number</td>
<td>SS</td>
<td>Sets speed to given value</td>
<td>SetSpeed 10</td>
</tr>
<tr>
<td>Teleport (X,Y)</td>
<td>TP</td>
<td>Teleports to the target location</td>
<td>Teleport (100, 300)</td>
</tr>
<tr>
<td>GPS</td>
<td></td>
<td>Prints the current location</td>
<td>GPS</td>
</tr>
<tr>
<td>Print “word”</td>
<td></td>
<td>Prints content of given argument</td>
<td>Print “Hello”</td>
</tr>
<tr>
<td>Color “color”</td>
<td></td>
<td>Sets the color of the current pen</td>
<td>Color “red”</td>
</tr>
<tr>
<td>Origin</td>
<td>Home</td>
<td>Returns current turtle to the center (home)</td>
<td>Home</td>
</tr>
<tr>
<td>ClearScreen</td>
<td>CS</td>
<td>Clears the canvas and returns all turtles back home</td>
<td>ClearScreen</td>
</tr>
<tr>
<td>Wrap</td>
<td></td>
<td>Turtle(s) could warp to the other side when reaching border</td>
<td>Wrap</td>
</tr>
<tr>
<td>Fence</td>
<td></td>
<td>Turtle(s) will stop once reaching border</td>
<td>Fence</td>
</tr>
<tr>
<td>PenUp</td>
<td>PU</td>
<td>Turtle(s) won’t draw when moving</td>
<td>PenUp</td>
</tr>
<tr>
<td>PenDown</td>
<td>PD</td>
<td>Turtle(s) will draw when moving</td>
<td>PenDown</td>
</tr>
<tr>
<td>ShowTurtle</td>
<td>ST</td>
<td>Displays turtle on GUI</td>
<td>ShowTurtle</td>
</tr>
<tr>
<td>HideTurtle</td>
<td>HT</td>
<td>Hides turtle on GUI</td>
<td>HideTurtle</td>
</tr>
<tr>
<td>Fill</td>
<td></td>
<td>Fills current enclosure area with pen color</td>
<td>Fill</td>
</tr>
<tr>
<td>Reset</td>
<td></td>
<td>Clears the symbol table and ClearScreen</td>
<td>Reset</td>
</tr>
</tbody>
</table>
Workflow of the interpreter

1. A user enters a command `FD 10*5` into the GUI.
2. The string `FD 10*5` is received by LOGOPP class, and it is sent to LOGOInterpreter.
3. Parse
Workflow of the Interpreter (cont.)

Internal structure:

- LOGOCommandNode
  - root
  - id = “FD”
  - children

- LOGOOperatorNode
  - children[0]
    - id = “*”
    - children

- LOGOLeaf
  - children[0]
    - id = “10”
    - children

- LOGOLeaf
  - children[1]
    - id = “5”
    - children
Workflow of the Interpreter (cont.)

4. The root.run() is called, which recursively evaluates the nodes in the tree in postorder:

Leaf = Double 10
Leaf = Double 5
Operator = 10*5 = 50
Command = FD 50

The FD command will call a static method LOGOBasic.forward() to modify the canvas.
Tools and Environment We Used

GitHub
Dropbox
Google docs
Java
eclipse
Clover
ANTLR
Run-time Environment

• Need to install latest Java Runtime Environment (JRE)
• Run the Java Archive (Logo++.jar) directly or compile from source code
Fully Integrated Interface

Output (graph)

Output (text)

Input

Status Bar
Testing

1. Regression test
2. Trace the error-causing code from the output and report to the responsible person
# Team Management

## Timeline

<table>
<thead>
<tr>
<th>Task Name</th>
<th>Duration</th>
<th>Predecessors</th>
<th>Resource Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language of the paper</td>
<td>3 days</td>
<td></td>
<td>Jia, Irene, Ken, Xinyuan, Na</td>
</tr>
<tr>
<td>Language tutorial</td>
<td>2 days</td>
<td></td>
<td>Jia, Irene, Ken, Xinyuan, Na</td>
</tr>
<tr>
<td>Language reference manual</td>
<td>1 day</td>
<td></td>
<td>Jia, Irene, Ken, Xinyuan, Na</td>
</tr>
<tr>
<td>Working compiler and dump</td>
<td>2 days</td>
<td></td>
<td>Jia, Irene, Ken, Xinyuan, Na</td>
</tr>
<tr>
<td>Final project report</td>
<td>1 week</td>
<td></td>
<td>Jia, Irene, Ken, Xinyuan, Na</td>
</tr>
<tr>
<td>Language</td>
<td>1 week</td>
<td></td>
<td>Jia, Irene, Ken, Xinyuan, Na</td>
</tr>
<tr>
<td>Language syntax and semantics</td>
<td>1 week</td>
<td></td>
<td>Jia, Irene, Ken, Xinyuan, Na</td>
</tr>
<tr>
<td>Grammar implementation</td>
<td>1 week</td>
<td></td>
<td>Jia, Irene, Ken, Xinyuan, Na</td>
</tr>
<tr>
<td>Basic Expressions</td>
<td>1 week</td>
<td></td>
<td>Jia, Irene, Ken, Xinyuan, Na</td>
</tr>
<tr>
<td>Basic Statements</td>
<td>1 week</td>
<td></td>
<td>Jia, Irene, Ken, Xinyuan, Na</td>
</tr>
<tr>
<td>Command</td>
<td>2 weeks</td>
<td></td>
<td>Jia, Irene, Ken, Xinyuan, Na</td>
</tr>
<tr>
<td>Conditional, iteration and functions</td>
<td>2 weeks</td>
<td></td>
<td>Jia, Irene, Ken, Xinyuan, Na</td>
</tr>
<tr>
<td>Module</td>
<td>3 weeks</td>
<td></td>
<td>Jia, Irene, Ken, Xinyuan, Na</td>
</tr>
<tr>
<td>Checklist</td>
<td>1 week</td>
<td></td>
<td>Jia, Irene, Ken, Xinyuan, Na</td>
</tr>
</tbody>
</table>

## Communicate is the key!
Lessons learned

- A team, not a collection of parts
- Collaborate and cooperate
- Speak up and participate
- Learn from your fellow team members
Conclusion

• An updated version of Logo tailored for elementary and secondary education
• Simple and clear syntax, easy to get started and transferable
• Progressive and expandable to complicated programs

• Logo++: should be the first programming language schools teach students