

Point/Counter- Point: On Using Graphical Languages to Teach Programming

J. Philip East (east@cs.uni.edu)

Michelle Friend Hutton
(mfh@pobox.com)

Start

- Introduce: session, us, session genesis
- Overview of the session
 - Kinds of "graphical" languages
 - Points we saw
 - Our conclusions
 - Resources
- Please-Discussion at most any time

Graphics in Programming

- Typing-based programming languages
 - Graphical content, Standard context, e.g., media computation, Objectdraw, games in C, GUI (Visual Basic)
 - Graphical content, Graphical context, e.g., Greenfoot, Logo & variants, Karel the Robot, Jeroo, KPL, MooseCrossing, Starlogo
- Visual programming languages (VPL), e.g., Stagecast Creator, Scratch, Alice, LegoMindstorms, Labview, Games Factory 2

(An informal categorization. Mention former: focus on

Focus on logic not syntax

■ Pro:

- Simplify entering commands
- Decrease things to memorize
- Fewer syntax errors = less frustration

■ Con

- "Real" programming languages have syntax (or you have to learn it sometime)
- Still lots to "memorize"

■ Rebuttal: Focus on one thing at a time

■ Rebuttal: Slippery slope; focus of mass education

Developing Generalized Programming Skill

- Con
 - Requires experience in multiple contexts
- Pro
 - May not be the goal
 - Can do models & simulations versus stories
 - Real programming is moving in this direction
- Rebuttal: So, consider goals and student population

It's All About Abstraction

- Pro: What's up with this "real" languages stuff, anyway
- Con: What does "Its all about abstraction" mean??
- Rebuttal: I'm right, you're wrong, so there!
- Rebuttal: I'm rubber, you're glue!
- Seriously, though ...

VPLs & Visualizing Program "State"

■ Pro

- Microworlds explicitly show progress of programs
- Aids/simplifies debugging

■ Con

- Not all programming problems are visual

VPLs Make Programming Fun!

■ Pro

- Visual contexts are engaging
- Need not be “dumbing it down”
- Can bring more people to computing

■ Con

- May be bait and switch
- But, I’m a teacher not an entertainer
- Everything cannot be fun

■ Rebuttal: VPLs make hard part (logic) less onerous

■ Rebuttal: Okay, but remember the learning goal

What Did We Miss?

Conclusions

- Identify your goal
- Identify how VPL use will advance it
- When using VPLs, provide guidance without being overly prescriptive
- For generalized programming skill, consider graphical context languages
- Using only number & text problems in CS is just wrong
- CS education must evolve; VPLs may or may not be the next species
- You teach the students you have, not the ones you want (or were)

Language Resources

- StarLogo TNG <http://education.mit.edu/starlogo-tng/>
- eToys <http://www.squeakland.org/>
- Scratch <http://scratch.mit.edu/>
- Stagecast Creator <http://www.stagecast.com/>
- Alice <http://www.alice.org/>
- Games Factory 2
<http://www.clickteam.com/eng/tgf2.php>

Literature Resources

- http://en.wikipedia.org/wiki/Visual_programming_language
- <http://www-ist.massey.ac.nz/plyons/776/VPL%20papers.html>
- <http://web.engr.oregonstate.edu/~burnett/Scaling/ScalingUp.html>
- Green, T.R.G., M. Petre, and R.K.E. Bellamy, "Comprehensibility of Visual and Textual Programs: A Test of Superlativism Against the 'Match-Mismatch' Conjecture." *Empirical Studies of Programmers: Fourth Workshop*, pp. 121-146, 1991.
- Kelleher, C and Paush, R, "Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers", *ACM Computing Surveys*, Volume 37, Issue 2 (June 2005) pp 83-137.