Scratch:
Making Programming Easy and Fun

John Maloney
Lifelong Kindergarten Group
MIT Media Laboratory

My Software Passions

- Smalltalk
- Other fun, dynamic programming languages
- Implementing such languages
- User Interface frameworks
- Frameworks for sound and music
- Empowering everyone to be programmers

Overview

- What is Scratch?
- Who uses it?
- Why was it created?
- What makes programming not easy and fun?
- How does Scratch address those problems?
- What are some systems with similar goals?
- Where can you learn more?

What is Scratch?

DEMO

Scratch Statistics

- Website: 620k accounts, 1.3 million projects
- Ages 9-19 most prolific creators (peak at 13)
- 2 million downloads from website
- XO laptops (1.85 million deployed)
- Schools: 2200 educators on ScratchEd website
- 50 languages

Inspirations

Roots:

- Logo (~1967)
- Smalltalk (1972)

Direct Influences:

- Morphic UI Framework (1994)
- Squeak Smalltalk (1995)
- Etoys (1996)
- Logo Blocks (1995)

The Catalyst

Computer Clubhouse (started 1993):

- Informal setting, self-directed activities
- Youth highly engaged with media, but not programming
- No suitable programming tools
- Scratch NSF Proposal (2003)

(Declining CS enrollment not yet a concern in 2003)

Not Easy

- Difficult to get started
- Syntax and data types
- Cryptic error messages
- Execution is invisible
- Data is invisible
- Overwhelmingly huge API's

Not Fun

- Easy programs are boring; fun ones difficult
- Errors crash application
- Edit-compile-run cycle
- Must restart after every change
- Programming is often solitary

Demo Time!

DEMO

Scratch is Easier!

Professional Language	Scratch
Difficult to get started	Palette, tinkerability, sample projects, website
Syntax and data types	Blocks programming
Cryptic error messages	Do something; no backtalk!
Execution is invisible	Stack & block highlighting
Data is invisible	Variable and list monitors
Overwhelmingly huge API's	~140 blocks

And More Fun!

Professional Language	Scratch
Fun programs are difficult	Sprite model simplifies use of images, animation, and sound
Errors crash application	“Failsoft” commands
Edit-compile-run cycle	Liveness and tinkerability
Restart after every change	Fix problems in context
Programming is often solitary	Scratch website supports feedback and collaboration

Observations

- Beginners and experts need different tools
- Perhaps some ideas from Scratch could make programming more fun for experts, too...
- Engagement and motivation are key
- A good first impression is essential
- Programming is still challenging (but fun!)

Related Systems

- Alice, Storytelling Alice (www.alice.org)
- Android App Inventor (appinventor.googlelabs.com)
- BYOB (byob.berkeley.edu)
- DesignBlocks (www.designblocks.net)
- Etoys (www.squeakland.org)
- Greenfoot (www.greenfoot.org)
- Kodu (research.microsoft.com/en-us/projects/kodu)
- PicoCricket (www.picocricket.com)
- And many others...

Learning More

- **Scratch: Programming for All**, CACM Nov. 2009
- **The Scratch Programming Language and Environment**, TOCE Oct. 2010, to appear
- **Directness and Liveness in the Morphic User Interface Construction Environment**, UIST 1995
- More papers at: info.scratch.mit.edu/Research

scratch.mit.edu