Programming with Scratch

Harvey Siy

hsiy@unomaha.edu
http://www.cs.unomaha.edu/~hsiy
Outline

• Computational thinking
• Introduction to Scratch
• Short exercise
• A more complicated example
• Discussions
Computational Thinking is…

A problem solving approach…

... solving a problem by explaining the steps needed to arrive at the solution.

To explain the steps, you have to:
• come up with the steps
• communicate them
• defend them
Explaining the steps

• come up with the steps
  – involves creativity
  – involves recognizing similar situations

• communicate the steps
  – how can I express them clearly?
  – involves creating an algorithm*
  – forms the basis of computer programming

• defend the steps
  – are these the right steps?
  – will these lead to a correct solution?
  – involves logical reasoning and critical thinking

* step-by-step procedure for solving a problem
Teaching Computational Thinking

- Coding is the most fun way to practice computational thinking.

- Lots of resources for all grade levels from Hour of Code: http://code.org/learn
Scratch

- [http://scratch.mit.edu](http://scratch.mit.edu)
- Fun and easy to learn
- Develops creativity
Development Environment
Development Environment
Development Environment
Development Environment

Scratch Development Environment

Stage

Sprite List
Sprite

- An object that performs some action
- Can define “indefinite” number of sprites in a program

Stage
- A special sprite
- 480x360
- Always in the background
Categories of blocks

- Move, rotate sprite
- Output and appearance
- Sound effects
- Controls the “pen”
- Variables and lists
- Event handling
- Control structures
- Feedback and input
- Arithmetic, string, Boolean operators
- Advanced blocks
Observations (subject to change)

- Real numbers are 64-bit floating point.
- Integers appear to be unbounded.
- Lists can only have numbers, strings, Booleans.
  - No list of lists.
  - No list of sprites.
- Recursion is not allowed
  - Except tail recursion.
Flow of execution

• **Sequence** – one instruction followed by the next

• **Decisions** – compute a condition
  – if true, follow one set of instructions
  – if false, follow an alternative set of instructions

• **Loops** – allows a set of instructions to be executed repeatedly

• **Events** – determines when to start an execution
Example 1: Computing square roots

- Use algorithm by Hero of Alexandria

To compute square root of x
1. Start with arbitrary positive value s
2. Replace s by \((s + \frac{x}{s})/2\)
3. Repeat #2 until s has stopped changing
To compute square root of x
1. Start with arbitrary positive value s
2. Replace s by (s + x/s)/2
3. Repeat #2 until s has stopped changing

Access the working version at:
http://scratch.mit.edu/projects/11656266/
Observations from Example 1

• Can make something other than games or stories (for more examples, see also http://code.google.com/p/scratch-unplugged/)

• Students can see the order of operations in expressions by the 3-D treatment of the operation templates.
Example 2: Pong

Move ball

Access the working version at:
http://scratch.mit.edu/projects/10597215/

When paddle hits ball

When paddle misses
Example 3: Sorting

- Download the Scratch programs from http://code.google.com/p/scratch-unplugged/

- Find the selection sort program.
- Gives example of:
  - Multiple sprites
  - Event handling
  - Lists
Physical interactions

- Scratch offers additional forms of interaction beyond the traditional keyboard and mouse.

- Sensor boards
  - Picoboard (currently 1.4)
  - Makey Makey
  - Lego WeDo (currently 1.4)

- Kinect (currently 1.4)

- Webcams (2.0 only)
Example 4: Webcam interaction

Detect motion by webcam

“Faster” motion detected: move sprite up

“Slower” motion detected: move sprite down

Remember to turn off webcam when done!

Access the working version at:
http://scratch.mit.edu/projects/10673482/
Makey Makey

See demo at: http://vimeo.com/60307041
Makey Makeys in action
Picoboard

- Slider
- Light sensor
- Button
- Sound sensor
- Clips
Picoboard controls (1.4)

Sensing button

- sensing button
- sensor value
- sensor condition test
Comparable languages
Alice
Snap

• Spinoff from Scratch

• Supports
  – Higher order functions
  – Recursion
  – Lists of lists
Greenfoot

A gentle transition into Java

import greenfoot.*;  // (World, Actor, GreenfootImage, and Greenfoot)

/**
 * This class defines a crab. Crabs live on the beach.
 */
public class Crab extends Animal
{
   public void act()
   {
      // Add your action code here.
   }
}
CS Unplugged

- [http://csunplugged.org](http://csunplugged.org)
- Computational thinking without computers!
Computational Thinking across the curriculum

- Multimedia
- Visual/Performing Arts
- Language Arts
- Science
- Technology
- Engineering
- Math
- Computational science
- Scientific visualization
- Information technology
- Modeling, Simulation
- Computer-aided design
- Automated theorem proving, Numerical analysis

Visual/Performing Arts

Digital humanities
Computational Thinking courses

Exploring Computer Science
http://www.exploringcs.org/

Computer Science Principles
http://www.csprinciples.org/

2/26/2014
Getting Computer Science into the K-12 Curriculum
CodeCrush
Some CS Resources


• Computer Science Teachers Association, [http://csta.acm.org](http://csta.acm.org)


  – See the list of resources near the end.
Scratch Resources

- **Website:** [http://scratch.mit.edu](http://scratch.mit.edu) (lots of examples)


- **After-school lessons:** [http://scratch.redware.com/lessonplan](http://scratch.redware.com/lessonplan)