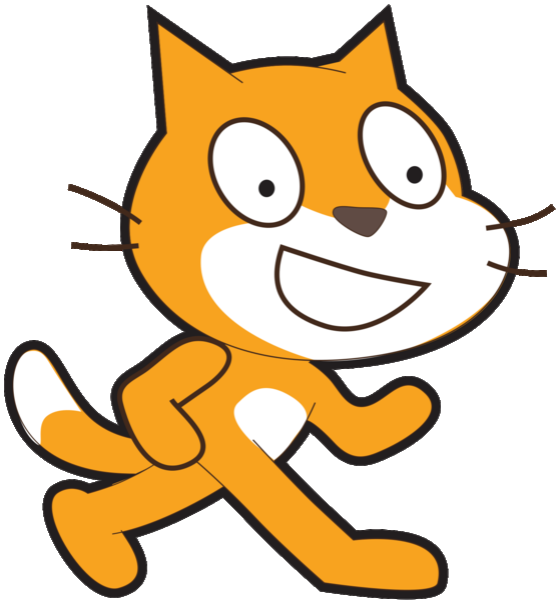
Programming with Scratch

Harvey Siy
hsiy@unomaha.edu
http://www.cs.unomaha.edu/~hsiy



Getting Computer Science into the K-12 Curriculum
CodeCrush

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 <u>algorithm</u>*
  - forms the basis of computer programming

** step-by-step procedure for solving a problem*

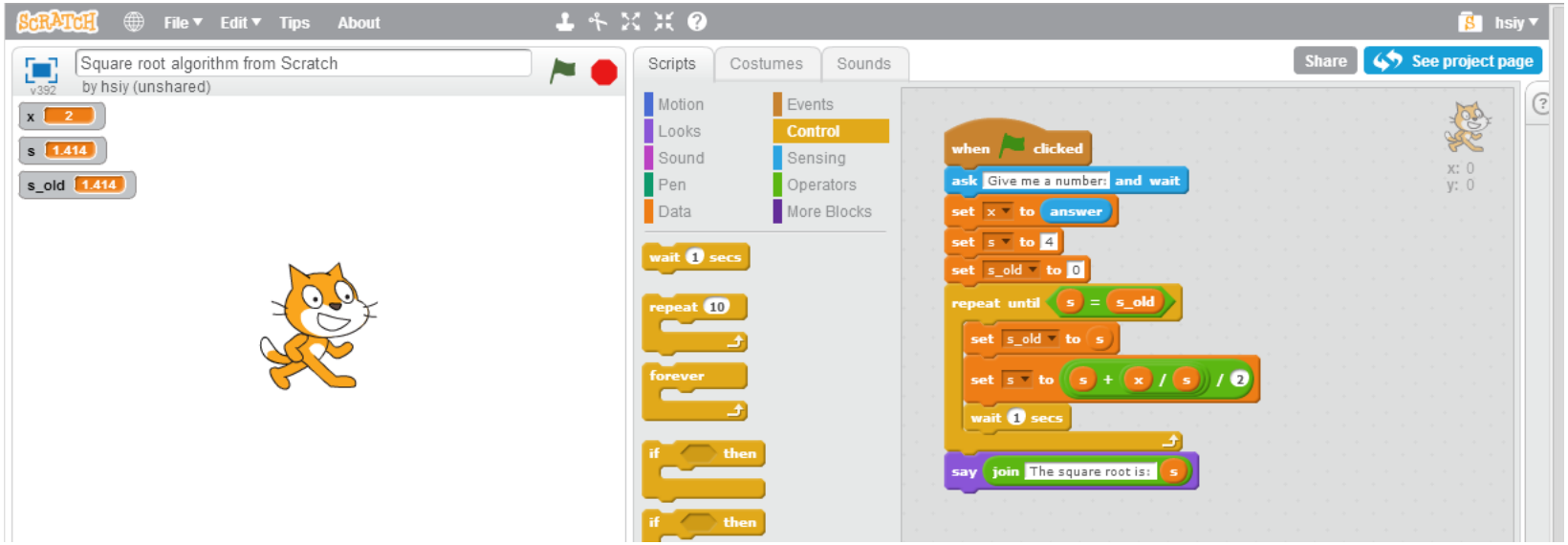
- defend the steps
  - are these the *right* steps?
  - will these lead to a *correct* solution?
  - involves *logical reasoning* and *critical thinking*

Teaching Computational Thinking

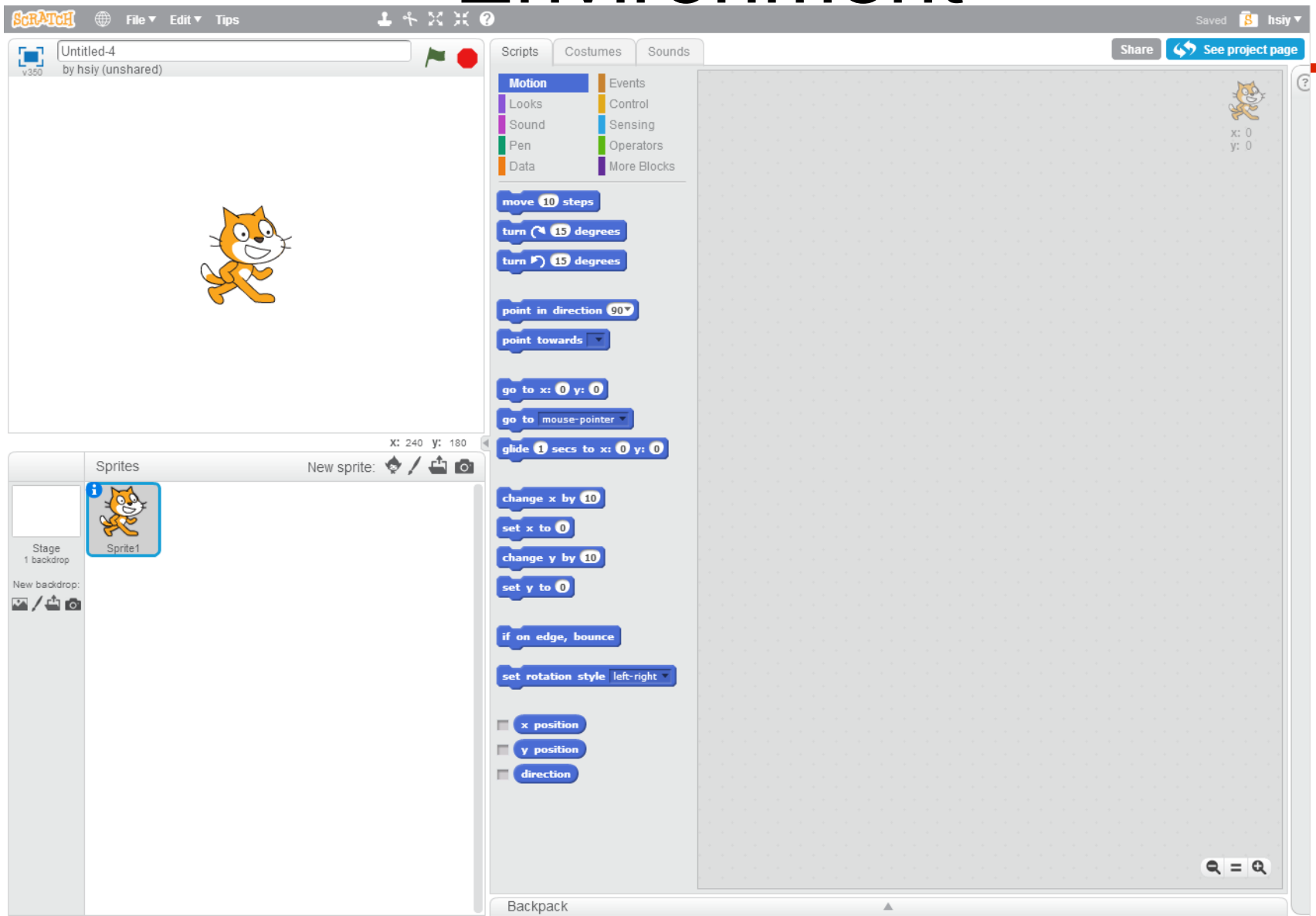
- Coding is the most fun way to practice computational thinking.
- Lots of resources for all grade levels from Hour of Code: http://code.org/learn

Scratch

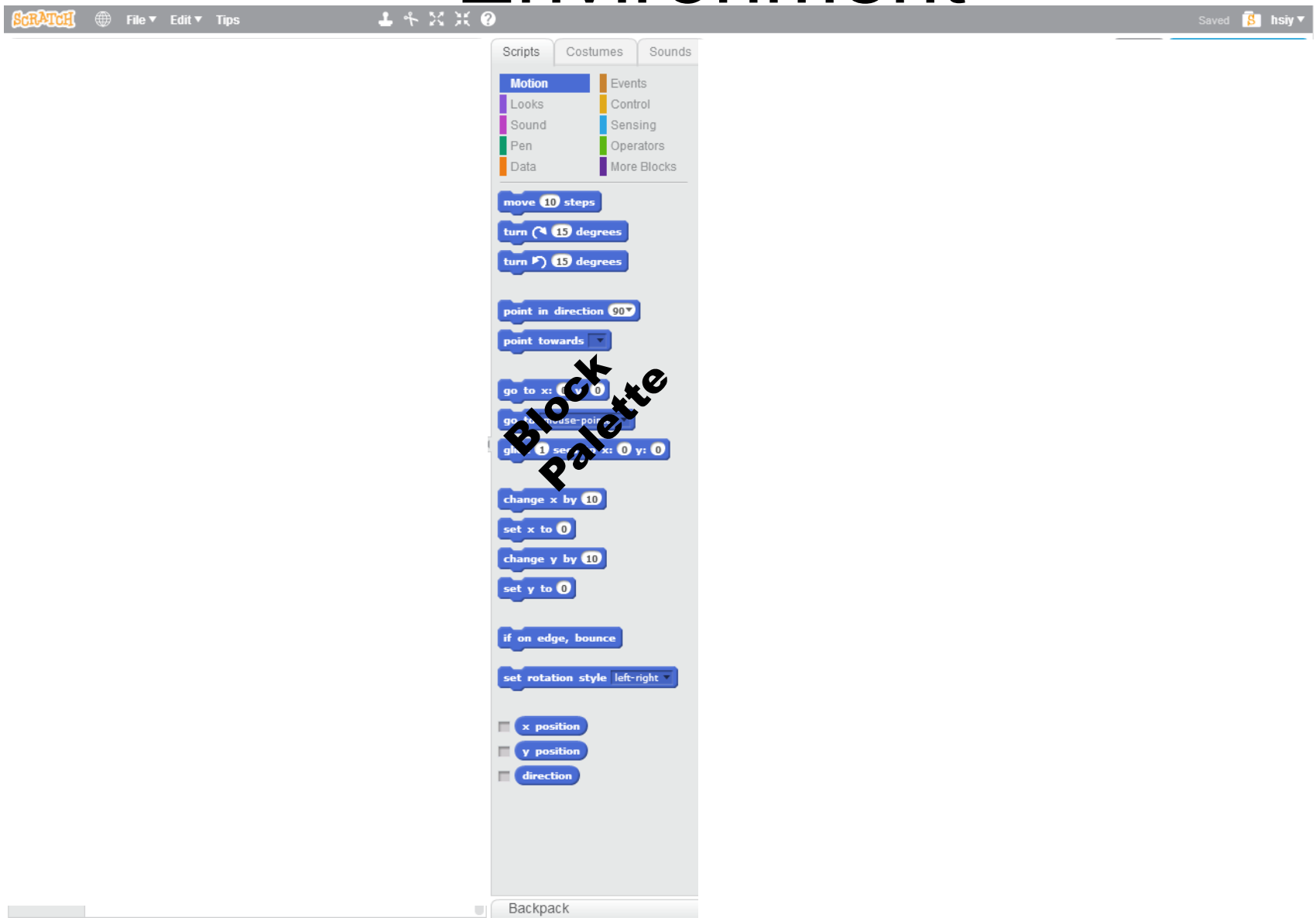
- http://scratch.mit.edu
- Fun and easy to learn
- Develops creativity



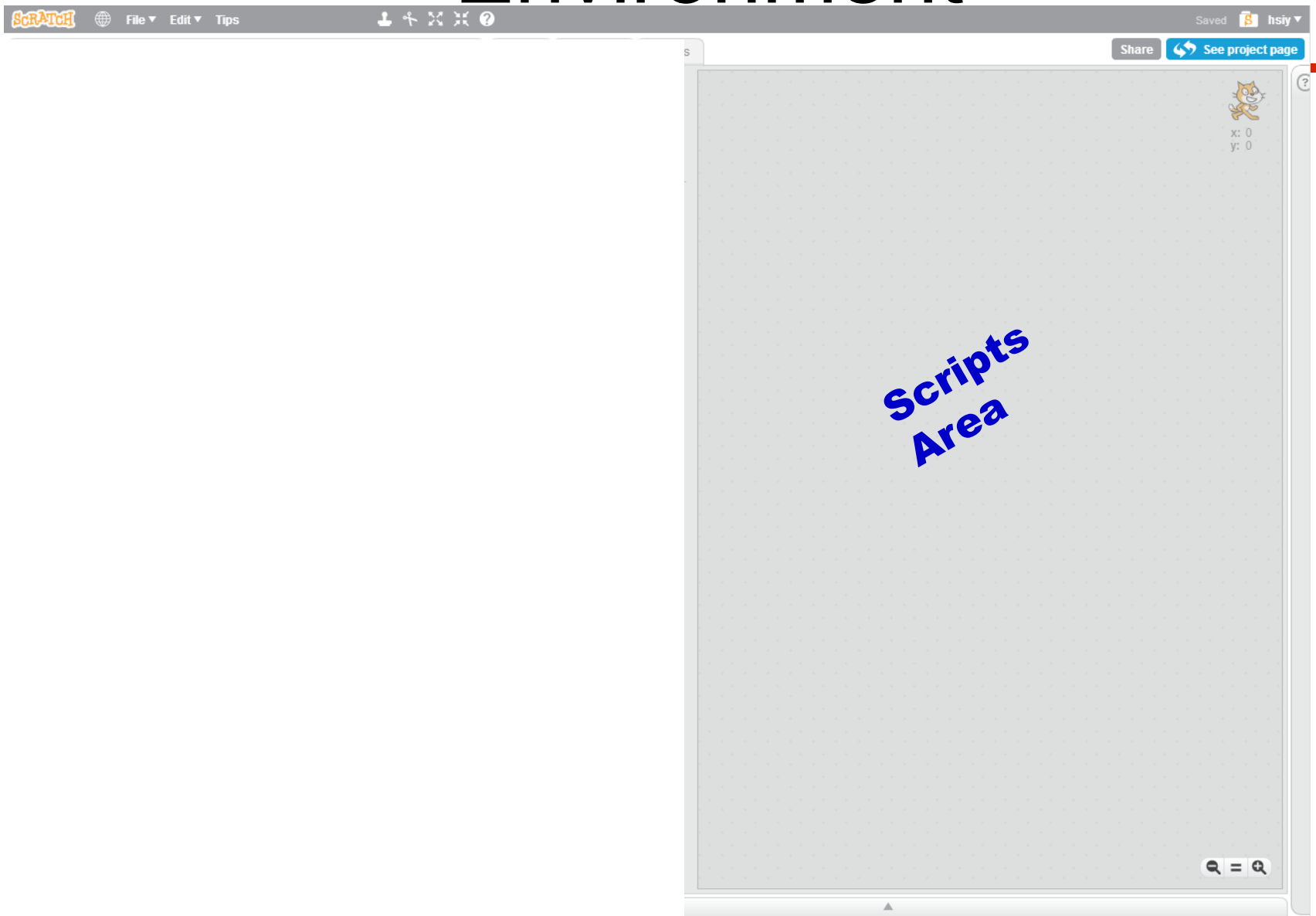
Development Environment



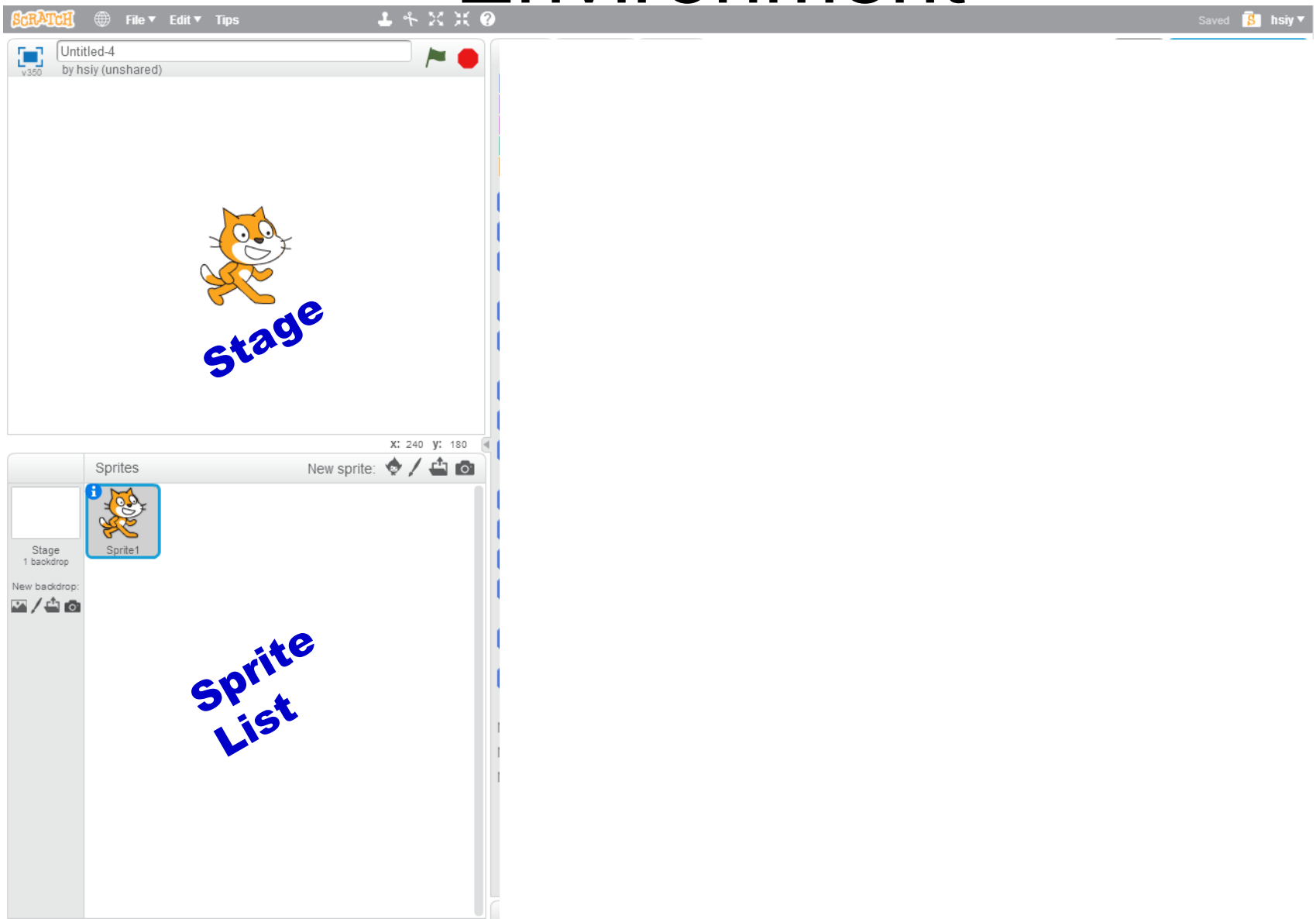
Development Environment



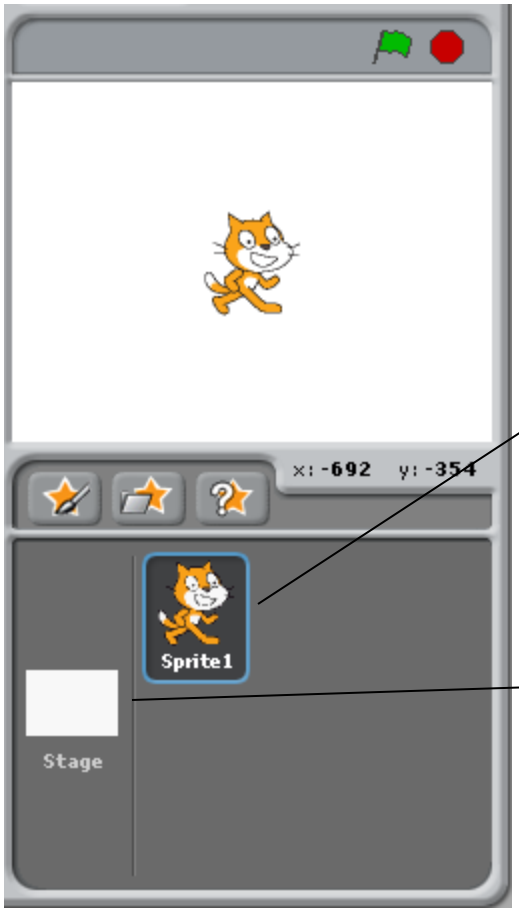
Development Environment



Development Environment



Sprite

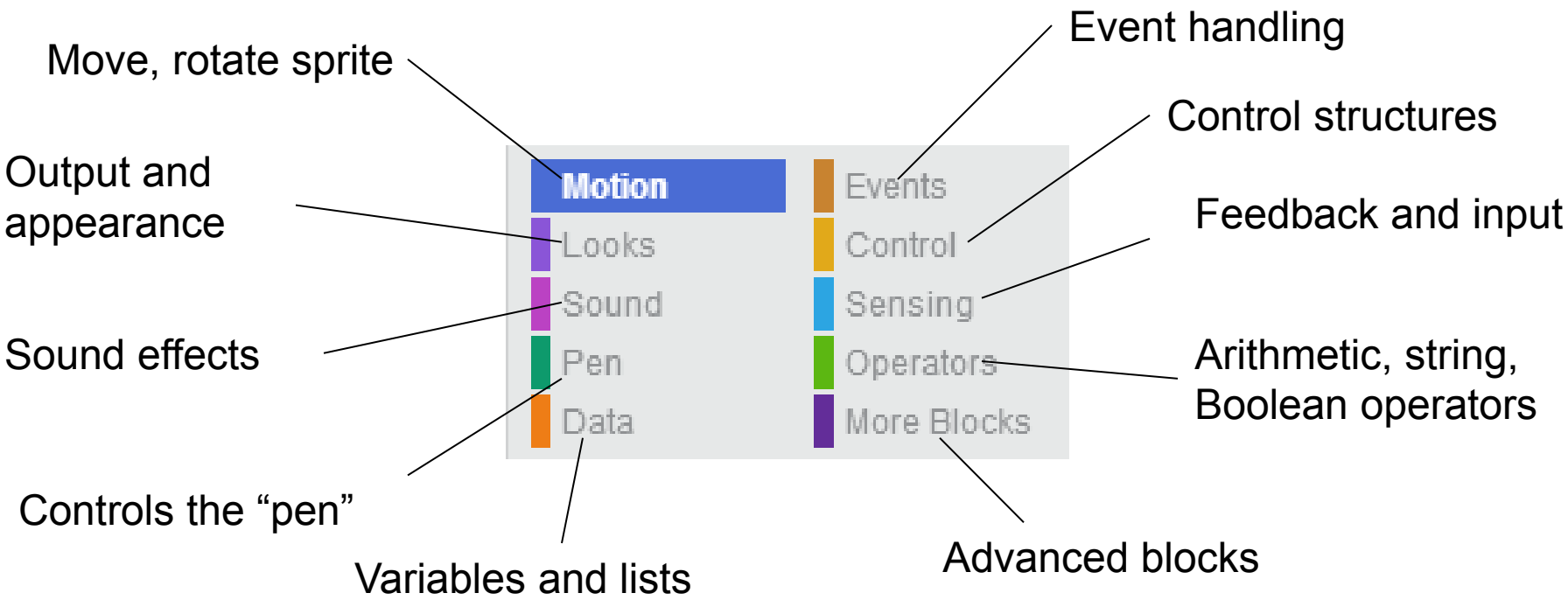


The stage and sprites can have scripts.

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



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

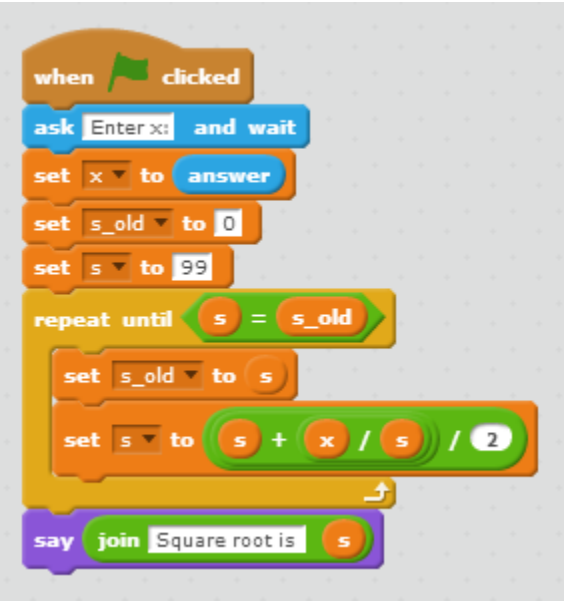
To compute square root of x

1. Start with arbitrary positive value s
2. Replace s by (s + x/s)/2
3. Repeat #2 until s has stopped changing

Solution

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



Need to make variables

Access the working version at: http://scratch.mit.edu/projects/11656266/

Observations from Example 1

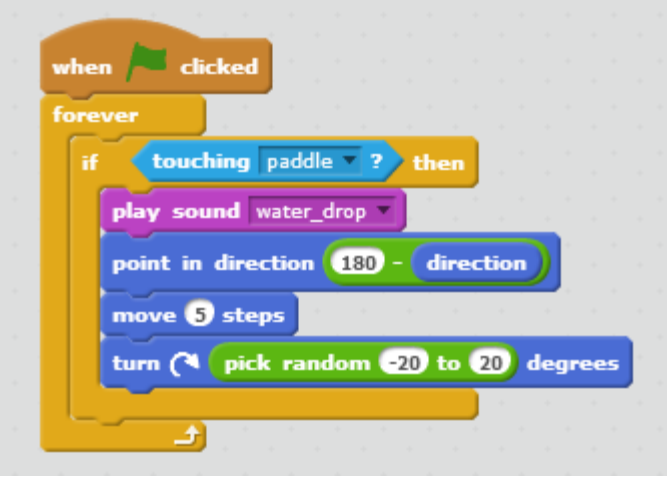
- Can make something other than games or stories (for more examples, see also http://code.google.com/p/scratch-unplugged/)

- Students can see the order of operations in expressions by the 3-D treatment of the operation templates.

Example 2: Pong

Move ball

When paddle hits ball



When paddle misses

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

Example 3: Sorting

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

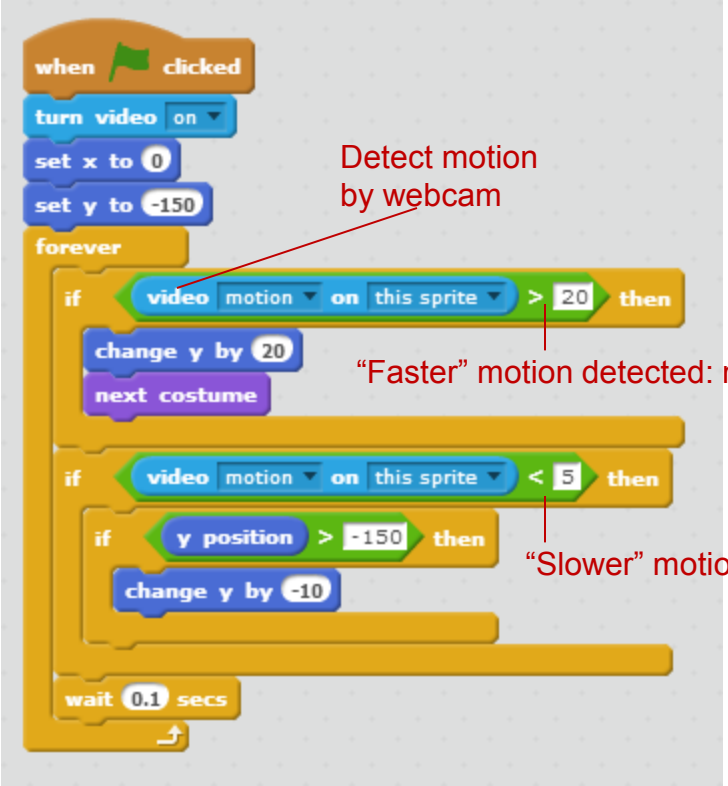
- Find the selection sort program.
- Gives example of:
  - Multiple sprites
  - Event handling
  - Lists

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

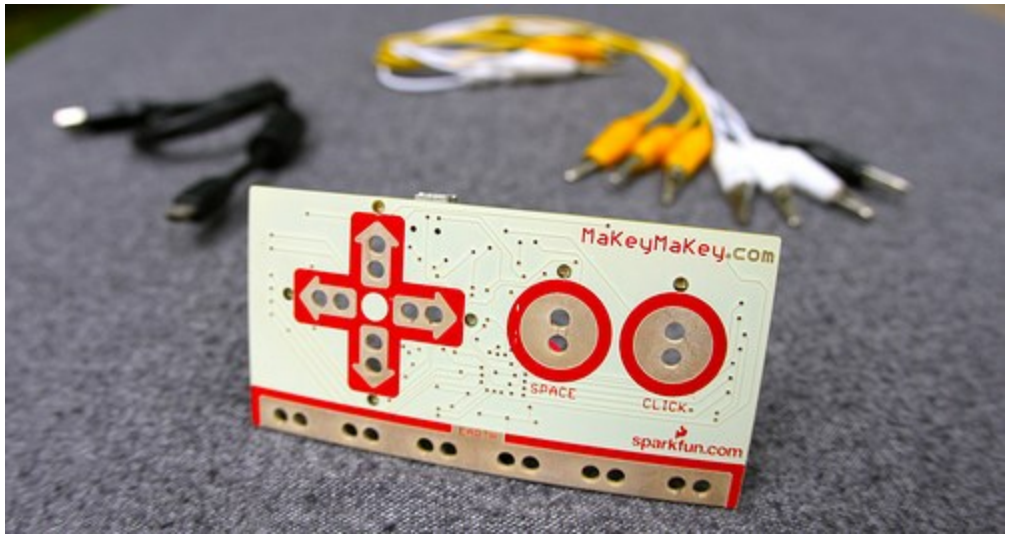
"Faster" motion detected: move sprite up

"Slower" motion detected: move sprite down

Remember to turn off webcam when done!

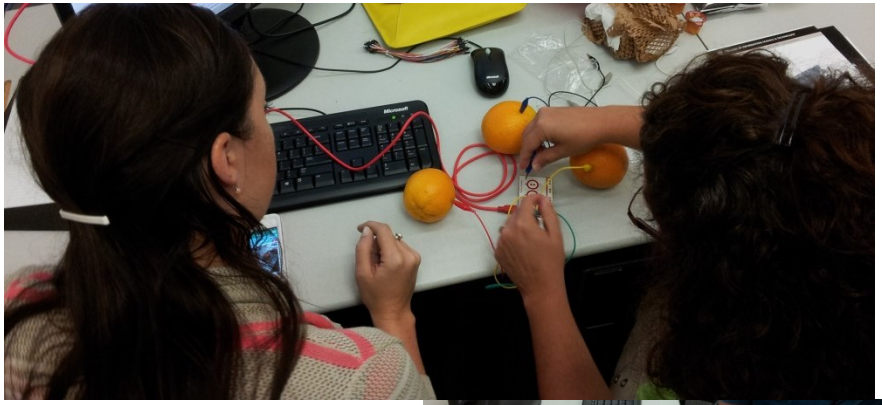
Access the working version at:
http://scratch.mit.edu/projects/10673482/

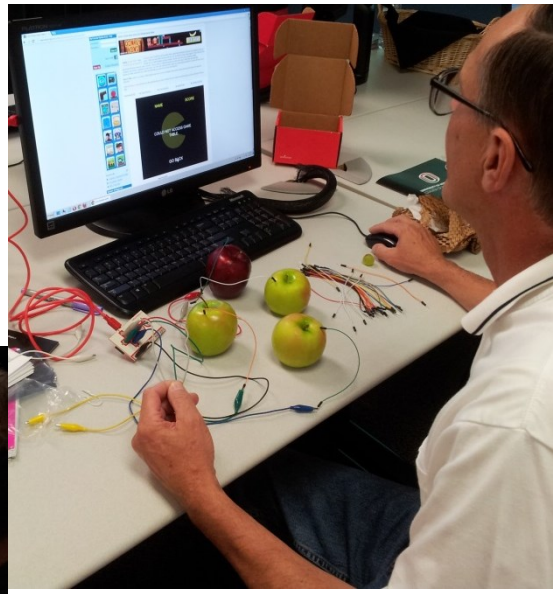
Makey Makey



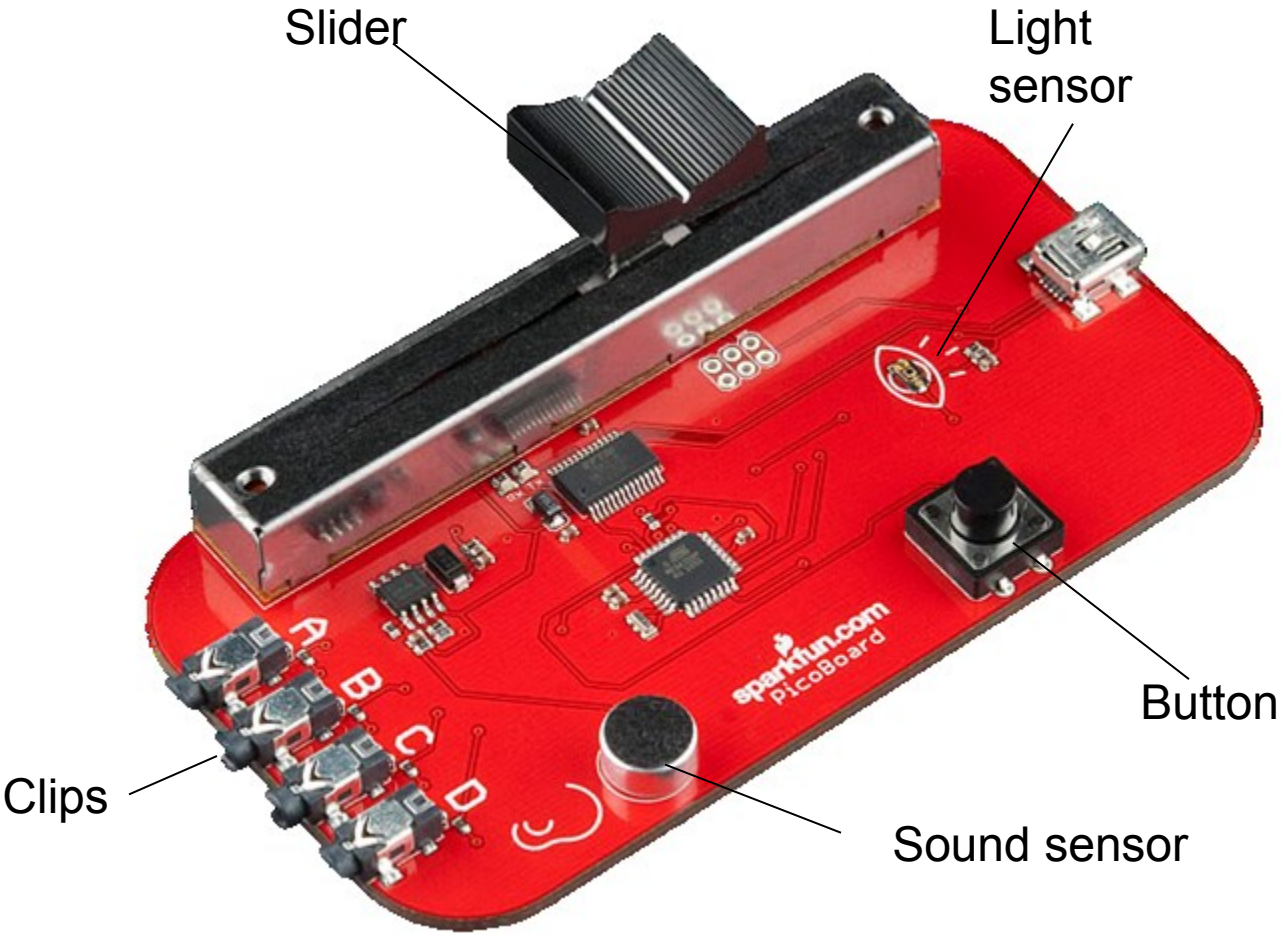
See demo at: http://vimeo.com/60307041

Makey Makeys in action

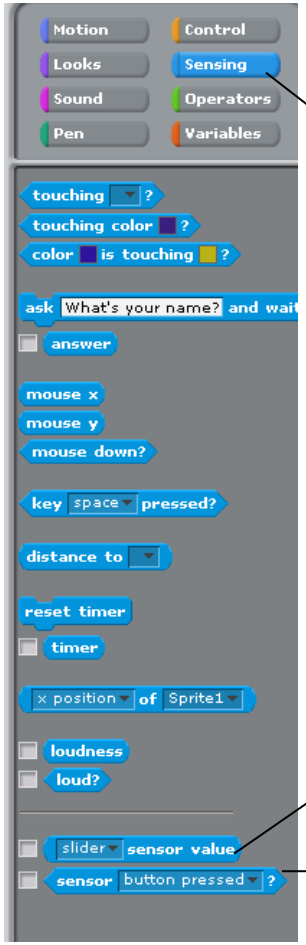




Picoboard


Slider
Light sensor
Button
Clips
Sound sensor
sparkfun.com
picoBoard

Picoboard controls (1.4)



Sensing button

sensor value

sensor condition test

Comparable languages

Alice

Alice (2.2 03/01/2012) D:\tools\Alice2.2\AliceAndWhiteRabbit.a2w

File Edit Tools Help

Play Undo Redo

world
- camera
- light
- ground
- whiteRabbit
  - rightArm
  - head
  - leftLeg
  - leftArm
  - rightLeg
  - tail
- aliceLiddell

ADD OBJECTS

Events create new event

When the world starts, do world.my first method

world.my first method

world.my first method *No parameters* create new parameter

No variables create new variable

Do in order
- aliceLiddell.neck.head point at camera more...
- aliceLiddell say Oh, hello there! *duration* = 2 seconds *fontSize* = 30 more...
- whiteRabbit point at camera more...
- whiteRabbit say Uhm, yes. Hello there! *duration* = 2 seconds *fontSize* = 30 more...
- aliceLiddell say My name is Alice Liddell. *duration* = 2 seconds *fontSize* = 30 more...
- whiteRabbit say And I am the White Rabbit. *duration* = 2 seconds *fontSize* = 30 more...

Do in order | Do together | If/Else | Loop | While | For all in order | For all together | Wait | print | //

whiteRabbit's details

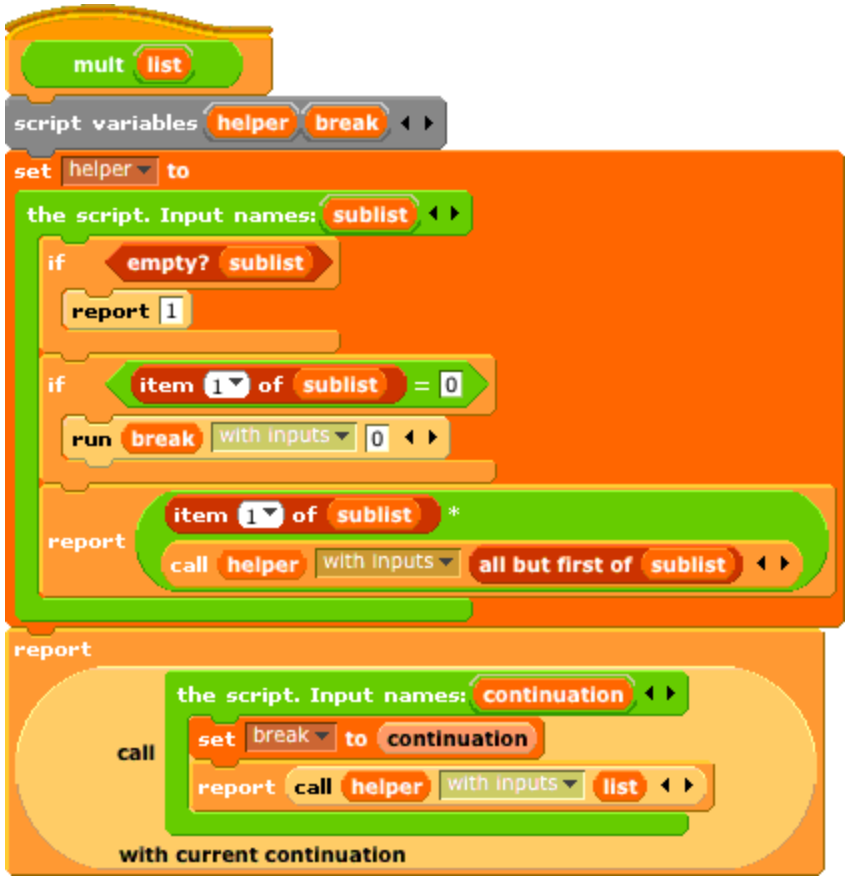
properties | methods | functions

create new method

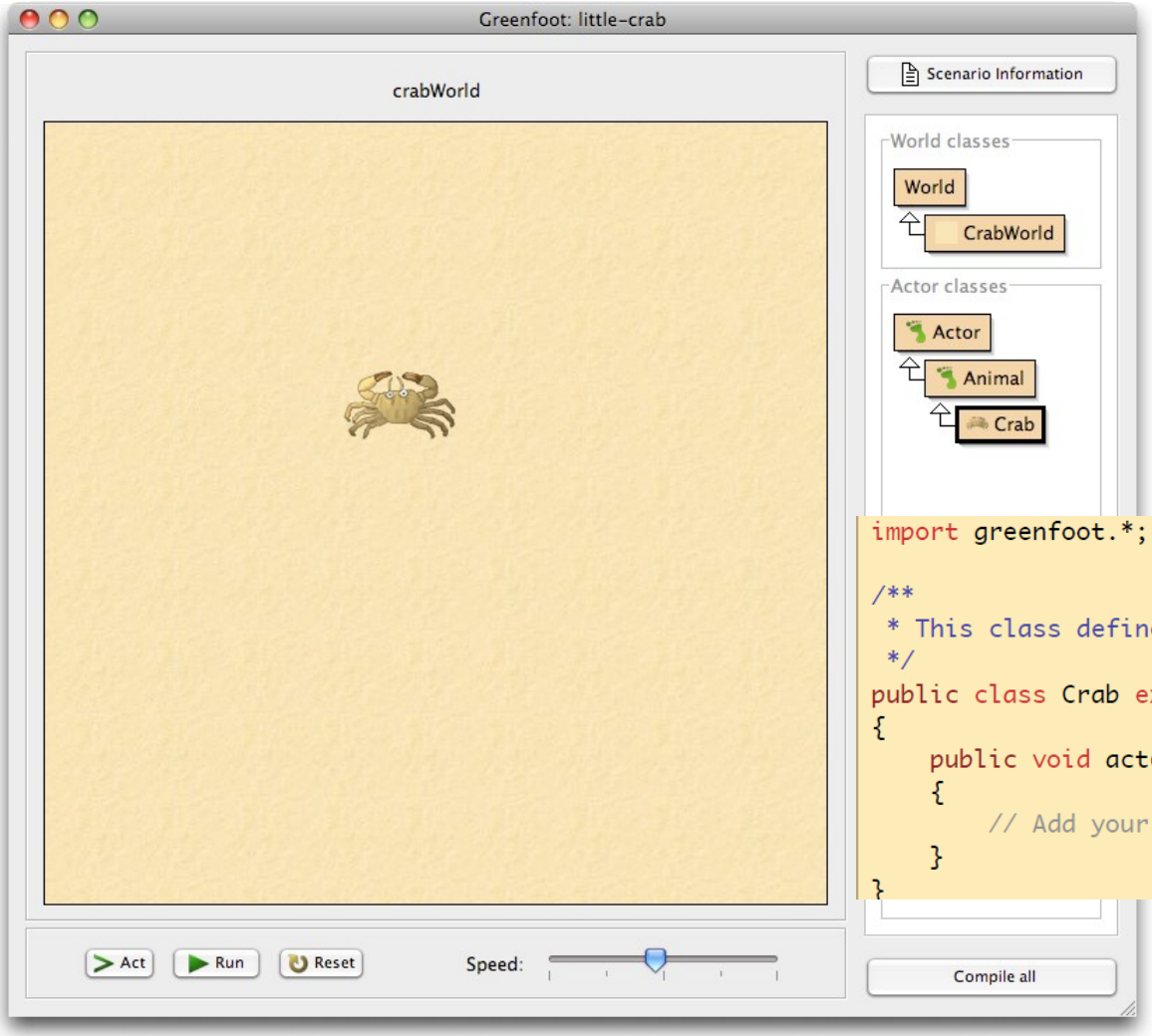
whiteRabbit move
whiteRabbit turn
whiteRabbit roll
whiteRabbit resize
whiteRabbit say
whiteRabbit think

Snap

- Spinoff from Scratch
- Supports
  - Higher order functions
  - Recursion
  - Lists of lists



Greenfoot

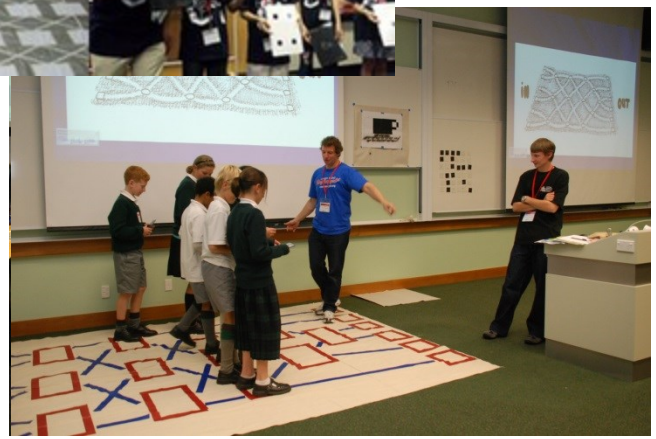


A gentle transition into Java

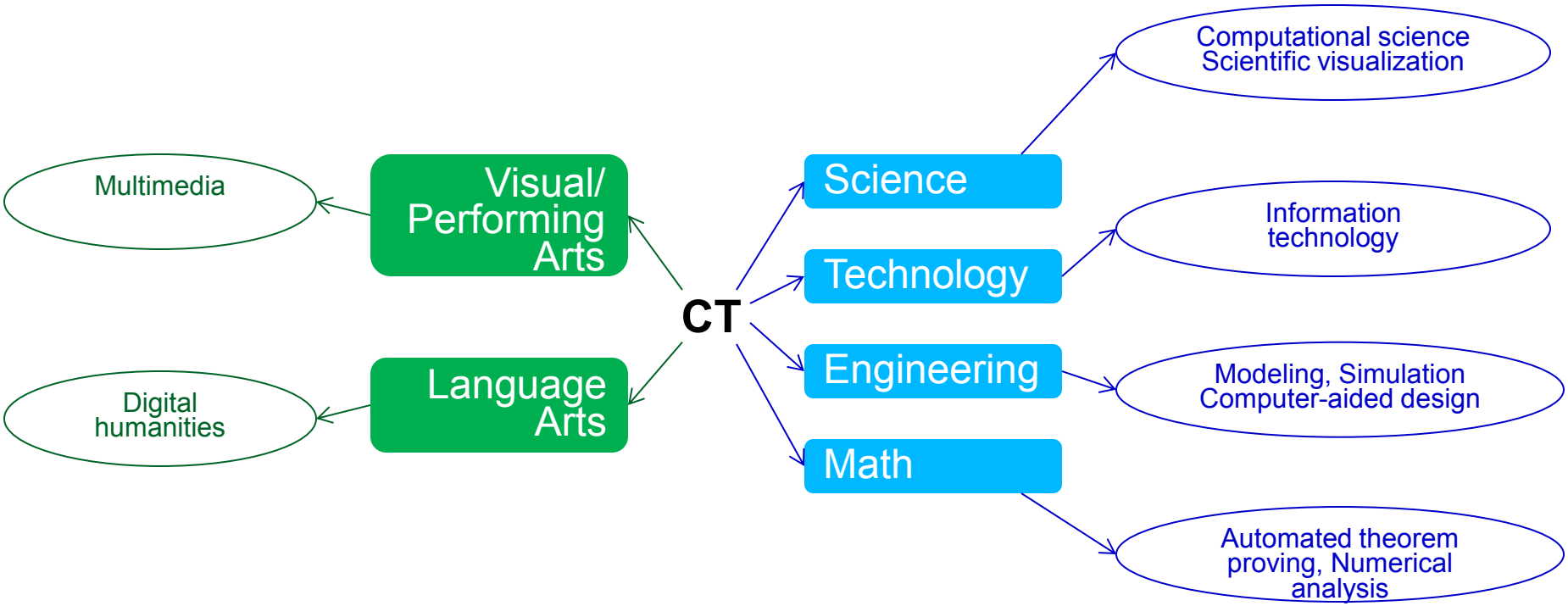
```
import greenfoot.*;  // (World, Actor, GreenfootImage, and Greenfoot)

/**
 * 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
- Computational thinking without computers!



Computational Thinking across the curriculum



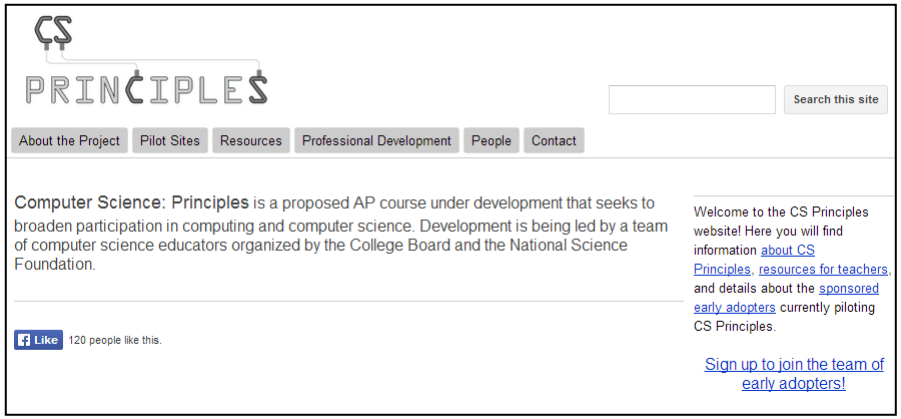
Computational Thinking courses

Exploring Computer Science

http://www.exploringcs.org/

Computer Science Principles

http://www.csprinciples.org/



Some CS Resources

- J. Wing. Computational Thinking, *Communications of the ACM*, March 2006.
- Computer Science Teachers Association, http://csta.acm.org
- Google. Exploring Computational Thinking, http://www.google.com/edu/computational-thinking.
- CS Unplugged, http://csunplugged.org.
- Yadav. Computational Thinking in K-12 Education, http://cs4edu.cs.purdue.edu/_media/ct-in-k12_edps235.pdf.
  - See the list of resources near the end.

Scratch Resources

- Website: http://scratch.mit.edu (lots of examples)
- Curriculum Guide: http://scratched.media.mit.edu/resources/scratch-curriculum-guide-draft
- After-school lessons: http://scratch.redware.com/lessonplan