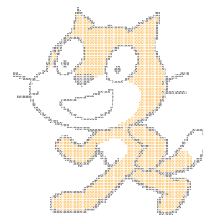
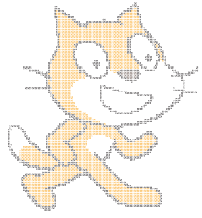


MIT Lifelong Kindergarten's Scratch

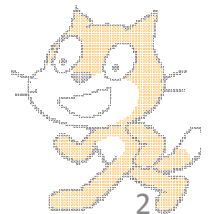
Serious Computer Science, Playful Environment
4/25/2009 Puget Sound CSTA meeting

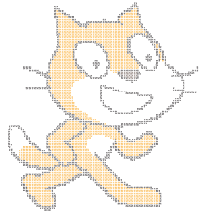




Today's goals

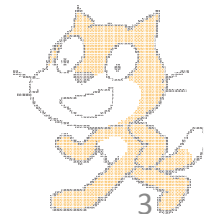
- Introduce the Scratch environment
- Demonstrate its value as a teaching tool
- NOT go through tutorials
- NOT provide lesson plans
- Explore interest for future in-depth sessions

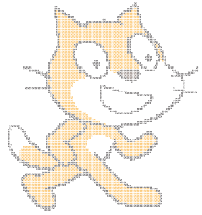




Why teach HS computer science?

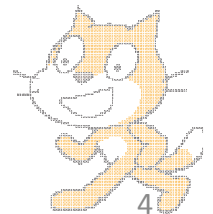
- Problem solving
 - Algorithmic thinking/“thinking like a computer”
 - Understanding of what computers can/can’t do
- Reinforcing learning from core subjects
 - Algebra
 - Geometry
 - Scientific method
- Increasing interest in computing fields

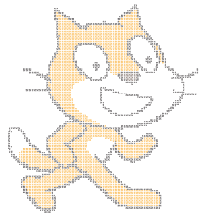




Secondary goals/means to end

- Making all students into star programmers
- Teaching all of a language's syntax
- Designing and building complex systems
 - Video games





Scratch: visual programming

- <http://scratch.mit.edu>

ROTATION STYLE
Control whether costumes rotate with the sprite.

CURRENT SPRITE INFO

TABS
Click tabs to edit scripts, costumes, or sounds.

TOOLBAR

GREEN FLAG
A way to start scripts.

PROJECT NOTES

BLOCKS PALETTE
Blocks for programming your sprites.

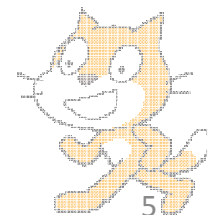
SCRIPTS AREA
Drag blocks in, snap them together into scripts.

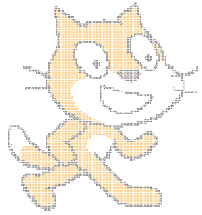
STAGE
Where your scratch creations come to life.

NEW SPRITE BUTTONS
Create a new character or object for your project.

SPRITE LIST
Thumbnails of all your sprites. Click to select and edit a sprite.

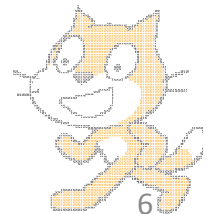
PRESENTATION MODE

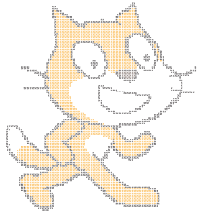




Why Scratch?

- No syntax learning-curve
- Exposure to complex topics builds intuition
 - Objects
 - Event-driven programming
 - Booleans
- Immediate feedback
- Visual environment appears “less nerdy”
- Clean interface

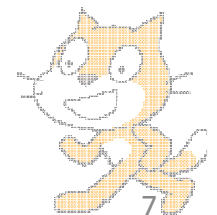
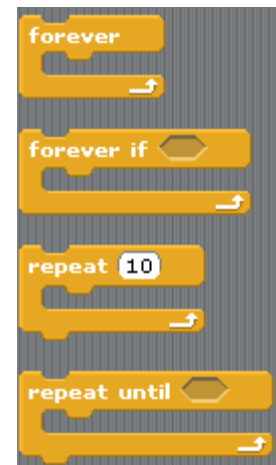
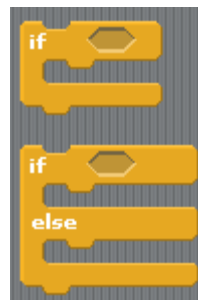


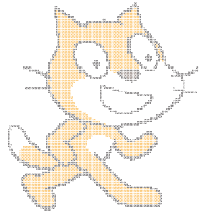


Thinking like a computer

- Understanding major programming constructs
 - Variables
 - Loops
 - Conditionals
 - Objects

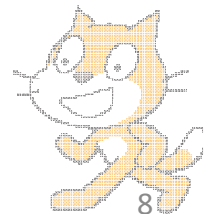
Cat X 135

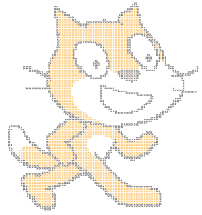




Avoiding common misconceptions

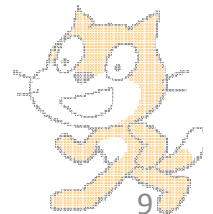
- “Loop bodies stop executing as soon as the test condition becomes false”
- “You can never have two variables with the same name”
- “It doesn’t matter what order I put my statements in”
- Have students design programs to disprove these!

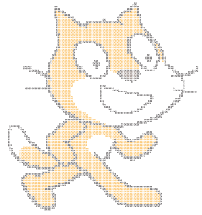




Increasing problem solving skills

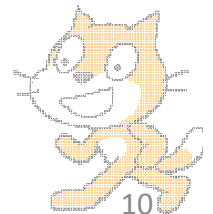
- Understanding problem statements
- Decomposing problems into solvable pieces
- Incremental design
- **Sample problem:** The user can set `gridHeight`, `gridWidth` and `gridMargin` with sliders. The cat should draw a grid of size `gridHeight` by `gridWidth`, `gridMargin` away from the edge of the screen.

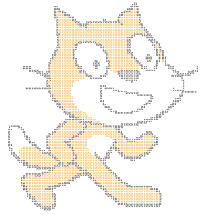




Breaking down the grid problem

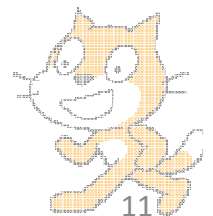
- **(Syntax)** How can we get Cat to draw a horizontal line 30 from the top of the screen?
- **(Geometry)** How can we get Cat to draw 5 evenly spaced horizontal lines?
- **(Syntax)** What kind of loop should we use?
- **(Algorithm)** How many times will it repeat?
- **(Algebra)** What if we want a margin of 30 on top and bottom?

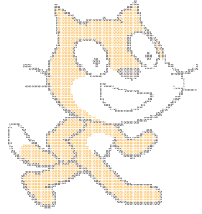




Resources

- <http://resources.scratchr.org/>
- <http://www.cs.uni-potsdam.de/~romeike/UEWettbewerb/index-english.htm>
- <http://www.redware.com/scratch/>
- <http://billkerr2.blogspot.com/2008/03/scratch-challenges-introductory.html>
- <http://www.slideshare.net/pwoessner/intro-to-scratch-presentation>
- <http://www.cs50.net/>





QUESTIONS AND DISCUSSION OF NEXT STEPS

