Chapter 3: Programming

CS105: Great Insights in Computer Science
Announcements

• Until further notice, you will NOT have to simplify Boolean expressions to receive full credit

• You must write up your own homework

• I put up a reference on binary addition, subtraction and two’s complements
Return of the Clicker
Google Words
The Vector

- We can build (or at least imagine!) lots of circuits.
- We can even think about state machines that use circuits to do various things over time.
- We’re headed towards using these ideas to create a programmable computer.
- Hillis has us take a detour to talk about programming first.
Many different languages have been devised for programming.

Each provides a way of writing a kind of script along with rules for the computer to interpret the script as instructions.

They can all do pretty much the same things, but make it easier to say some things than others.

How many languages can you name?
Scratch

- Developed by the “Lifelong Kindergarten” group at MIT.
- Allows users to make media-rich programs by clicking together blocks.
- Share your creations à la YouTube.
- The miniNim program is an example.
Programming Window
Things

- Performance area
  - stage (background)
  - sprites (objects)
    - scripts (behavior)
    - costumes (appearance)
    - sounds (available sounds)
More Things

- Run/Stop
- Script inventory
  - Motion
  - Looks
  - Sound
  - Pen
  - Control
- Sensing
- Numbers
- Variables
Shapes

- Trigger
- Statement
- Ending statement
- Boolean value
- Numeric value
Motion Inventory

- move 10 steps
- turn ← 15 degrees
- turn → 15 degrees
- point in direction 90°
- point towards
- go to x: 0 y: 0
- go to
- glide 1 secs to x: 0 y: 0

- change x by 10
- set x to 0
- change y by 10
- set y to 0
- if on edge, bounce
- x position
- y position
- direction
color, fisheye, whirl, pixelate, mosaic, brightness, ghost
Sound Inventory

- play sound "meow"
- play sound "meow" until done
- stop all sounds

- play drum 48 for 0.2 beats
- rest for 0.2 beats
- play note 60 for 0.5 beats
- set instrument to 1

- change volume by -10
- set volume to 100%
- change tempo by 20
- set tempo to 60 bpm

large collection of synthesized sounds
Pen Inventory

clear
pen down
pen up
set pen color to 1
change pen color by 10
set pen color to 0
change pen shade by 10
set pen shade to 50
change pen size by 1
set pen size to 1
stamp
Control Inventory

- when clicked
- when space key pressed
- when Sprite1 clicked
- wait 1 secs
- forever
- repeat 10

- broadcast and wait
- when I receive
- forever if
- if

- if
- else
- wait until
- repeat until
- stop script
- stop all
Sensing Inventory

- mouse x
- mouse y
- mouse down?

- key space pressed?
- touching ▼ ?
- touching color ▼ ?
- color ▼ is touching ▼ ?
- distance to ▼

- reset timer
- timer
- x position ▼ of Sprite1 ▼
- loudness
- loud?
- slider ▼ sensor value
- sensor ▼ button pressed ▼ ?
Numbers Inventory

abs, sqrt, sin, cos, tan, asin, acos, atan, ln, log, e^, 10^
Variables Inventory

Make a variable
Delete a variable

set value to 0
change value by 1
show variable value
hide variable value

Make a list
Delete a list

add thing to mem
delete 1 of mem
insert thing at 1 of mem
replace item 1 of mem with thing

item 1 of mem
length of mem
Logo in Scratch

• Logo is a language invented to help people (kids?) learn to program.

• Scratch is the language we’ve been using for examples and demos in this class. It’s a descendent of Logo, in many ways.

• Hillis uses Logo for his examples in this chapter.

• I’ll translate them into Scratch so it fits better with our other examples.
Drawing Commands

- `pen down`
- `move 100 steps`
- `turn 90 degrees`
- `move 100 steps`
New Command

when I receive square
pen down
move 100 steps
turn left 90 degrees
move 100 steps
turn left 90 degrees
move 100 steps
turn left 90 degrees
move 100 steps
Do It!

broadcast square and wait
Use It!

when I receive window and broadcast square and wait
broadcast square and wait
broadcast square and wait
broadcast square and wait
broadcast window and wait
Command With Variable

- when I receive `squareSize`
- pen down
- move `size` steps
- turn ↘ 90 degrees
- move `size` steps
- turn ↘ 90 degrees
- move `size` steps
- turn ↘ 90 degrees
- move `size` steps

- clear
- set `size` to 100
- broadcast `window` and wait
- set `size` to 50
- broadcast `window` and wait