

# Art 382: Introduction to Interactive Media

[mcdo@umbc.edu](mailto:mcdo@umbc.edu)

Lecture 12, March 5

# Schedule

- Take Roll
- Looking at backstories
- Using ActionScript
- Making a button

# Backstories

- one for each character
- long-term : why are they who they are?
- short-term : what happened earlier that day?

let's talk about programming

# what is a computer?

- Computers can't do much, but they can do that not-much very, very quickly.
- Computers are CPU's and RAM
- RAM is a huge list of numbers– bytes
- The CPU can
  - Copy numbers from one place to another
  - Do arithmetic
- From these two operations, everything else

# what do programs do?

- move numbers around in the computer
- some numbers control hardware
  - pixels on the screen
  - sounds coming out of the speakers
  - whether the hard drive is spinning
  - keyboard LEDs
- mostly the numbers are just invisibly inside the computer, in lists

# numbers represent data

- pixel color data
- letters of the alphabet : text files
- programmers invent what the numbers mean
  - they assign meanings to numbers
  - "a" is letter number 64
  - Omar is person number 453342

# meaning

- how is meaning assigned to text?
  - tradition
  - convention
- what do lines really mean?
- you'll be doing the same thing as a programmer

# how programs do meaning

- programs move numbers around
- the numbers have had meanings assigned to them
- the meaning of a program is
  - how meanings are assigned to its numbers
  - what the program does to the numbers
- that's really all-- kinda disappointing

# simple operations build up

- here's a list of numbers: temperature observations for a place for a day
  - 10 bytes
- make a list of 20 years' worth of observations for that place
  - 73k
- get this data for 13,000 places
  - 905 MB
- That's all the weather in the world in your whole life

# more simple operations

- get weekly highs and lows and averages
- to "compare" a week's worth of weather is to add up the differences in high and low and averages
  - one new number, "weekly difference"
- add up the weekly differences of each place
  - one new number, "dissimilarity"
- find minimum-dissimilarity places for each place

# simple ops, complex result

- "minimum-dissimilarity" is defined by me
- it is a number
  - "find the smallest number in a set" is a good task to give a computer
- it reflects reality
  - places with large dissimilarities have different weather
- applications: real-estate speculation, social networking for gardeners, energy use predictions

# summary: what programs do

- simple operations on large collections of numbers
- the numbers could mean anything
- complexity through repetition

# programming in windows

- "moving numbers around" doesn't tell you how to get the mouse to work
- there is an enormous complexity jump from text-processing to windows
- but most of the complexity is hidden!
- HTML is a good intro to this

# <img /> tags

- make an image come up
- you don't know:
  - the jpg format (or any other)
  - where to move that data to in RAM
  - how to tell the browser to draw a color
  - anything about internet file retrieval
- you do know
  - put in an "<img ...", makes an image appear
  - some attributes
  - that's enough!

# same thing in windows

- you'll have some numbers that you use to control your program
  - the "text-processing" model
  - this part, you invent, and you have to understand it
- you'll have some "commands" that control Flash, its window, the mouse
  - the "<img/>" model
  - you just make this part work
  - you don't want to understand it much

# Flash simplifies

- Flash is platform independent
  - .swf's run the same way on PC, Mac, LeapPads, whatever has a flash player
- Flash wraps the platform with flashiness
  - you don't deal with Mac mouse vs. PC mouse
  - you deal with Flash Mouse
  - Adobe deals with Apple, M\$, & the others
- Some disadvantages
  - slower
  - some platform features not supported.

# your program has two parts

- a part you understand completely
  - simple operations on numbers
  - you assign meaning to the numbers
- an interface with Flash
  - you tell flash to do things for you

# enough philosophy

- two ways to get code into flash
  - a script layer
  - external code files
- script layers are easier; we'll start there

# your first script

- open flash
- make a sliding box animation, 40 frames
- make a second layer, called "scripts"
- put a keyframe on the last frame
- bring up the "actions" window
  - window -> actions (F9)
- type in "gotoAndPlay(20);"
- note little "a" on your frame
- ctrl-enter

# the ActionScript editor

- built in to flash
- a programmer's text editor
  - with lots of extra features-- later!
- scripts are not shown on-stage
  - what a mess that would be
- scripts are attached to keyframes
  - go to wrong frame, no script!

# script layers

- put all your code on the script layer
  - industry convention
  - much, much easier to debug
  - not the way they used to do it
    - code in symbols
    - bad!
- hitting "enter" does not run scripts!

# testing scripts

- hitting "enter" does not test scripts!
- ctrl-enter does
- sometimes, scripts don't work yet, but you want to see the animation

# gotoAndPlay(20);

- a command to flash
- sets the frame number
  - to 20
  - and then plays from there
  - much stuff happening!!
    - there's a frame counter
    - you set it
    - that change gets noticed by flash
    - who knows what else!
    - don't care! just play the movie!

# gotoAndPlay(20);

- a "function call"
  - note the parentheses right after the word
  - functionName(stuff)
- with a "parameter"
  - 20
  - 20 is sent to the "gotoAndPlay" function
  - like an attribute in HTML
- jargon: we just called gotoAndPlay with an argument of 20

# when do scripts run?

- we just made an animation loop
- except, the first time, the first 10 frames play
- and, eternally afterwards, we skip them
  
- a script is run when flash gets to the frame to which it is attached

# playing with jumps

- put a keyframe at 15
- add the script "gotoAndPlay(25);"
- jumpy!
  
- can you make it play backwards?
  - one keyframe per frame?