Reflections on the Development of Interactive Learning Tools

SCOTT H. HAWLEY
DEPARTMENT OF CHEMISTRY & PHYSICS

BELMONT SOTL WORKSHOP
MAY 11, 2016

They say you don't really know something…

…until you try to ~~teach it~~ write a teaching 'app' about it!

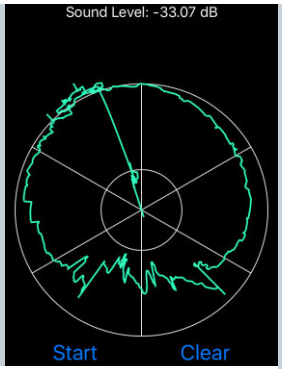
Instructional Tech apps are created to help students, and yet…

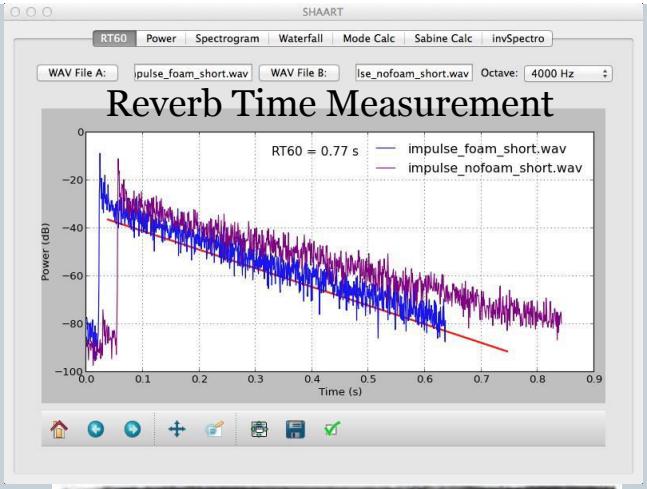
The 'app' writing process *also* benefits the instructor:

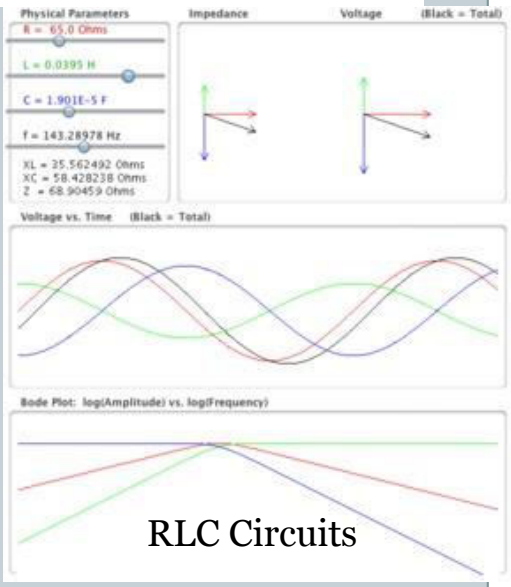
- builds depth of comprehension
- clarifies understanding / illuminates misunderstanding
- offers opportunities for discovery

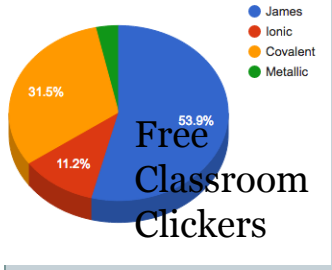
Sample of ‘Apps’ Developed While at Belmont

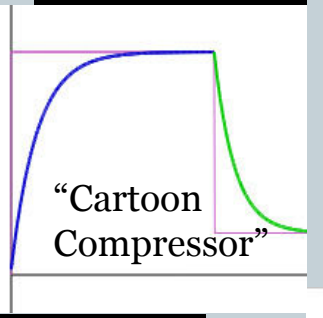
Languages: Python, JavaScript, PHP, Swift (none of which I knew before starting)
Platforms: Web, Mac & iOS

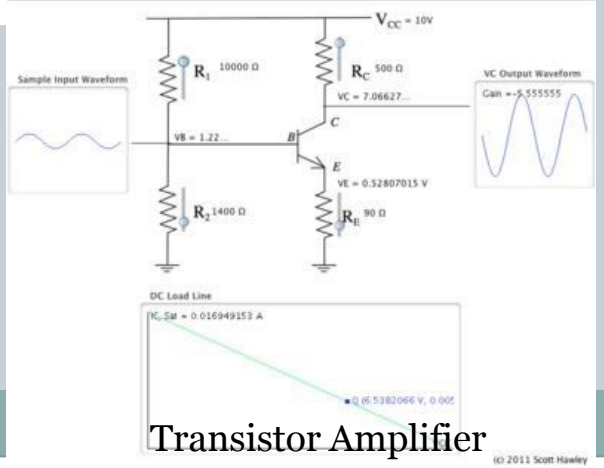


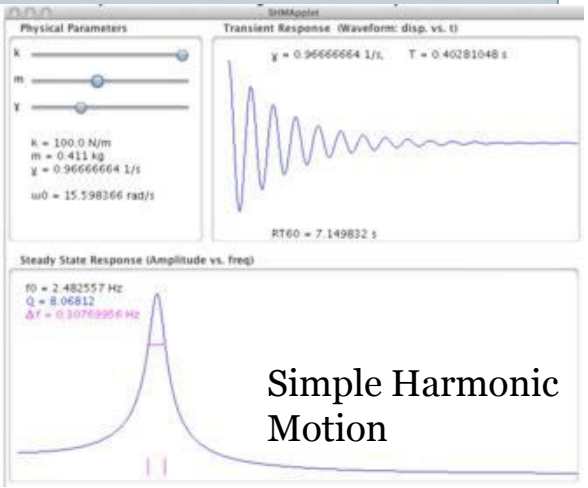












Why Write Instructional Tool Apps?

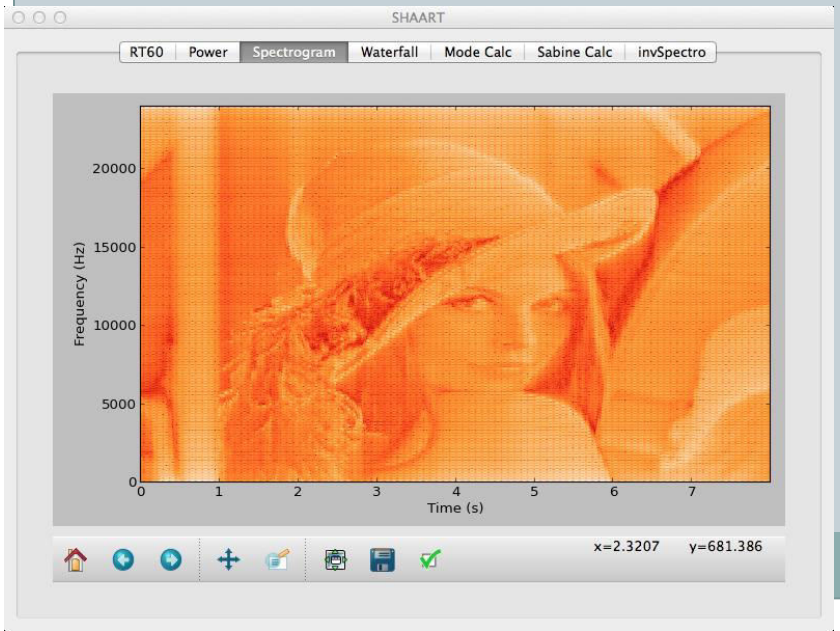
- Necessity
  - e.g., Acoustic toolkit, for measuring reverb times & other lab tasks
- Convenience – 'there ought to be a way…'
  - e.g., Clicker system, Polar Pattern Plotter
- Control / Customization
  - e.g., Physics Problem Parser & online HW, for randomizing questions
- Enrich & accelerate the learning experience
  - Provide visualization, interactivity & 'tactile' learning experience
  - Emphasize concepts over (math) details
    - e.g., Transistor Amp demo, RLC Circuit demo
- Comprehension
  - e.g., Compressor demo
- Service
  - e.g., Knobility, for Audio II documentation (which I don't teach)
- Help Stay Current on Latest Tech
  - Convolution Reverbs, WebAudio, Spatial Audio, Neural Networks

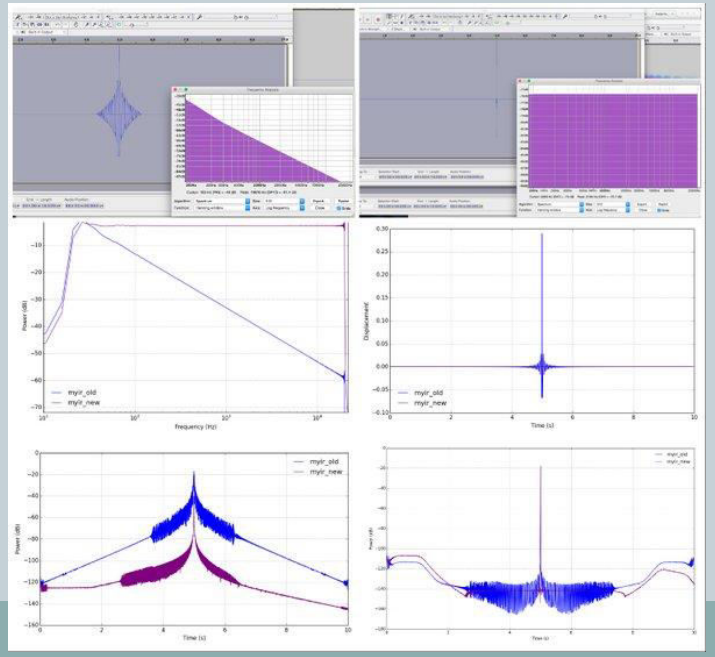
How Hard Is It?

- Can be intimidating
  - Delayed JavaScript/HTML5 transition despite student encouragement
- Tons of free tutorials available on the internet
  - Grab one and get started!
  - Wrote Knobility (iOS app) in basically a week and half, never having tried programming in Swift, started with one online tutorial and gradually built it.
  - Learned Python (for SHAART) via learnpython.org
  - Learned PHP (for RhymeLink & Homeworks) similarly
- Tons of questions answered, e.g. on StackExchange

“…builds depth of comprehension”

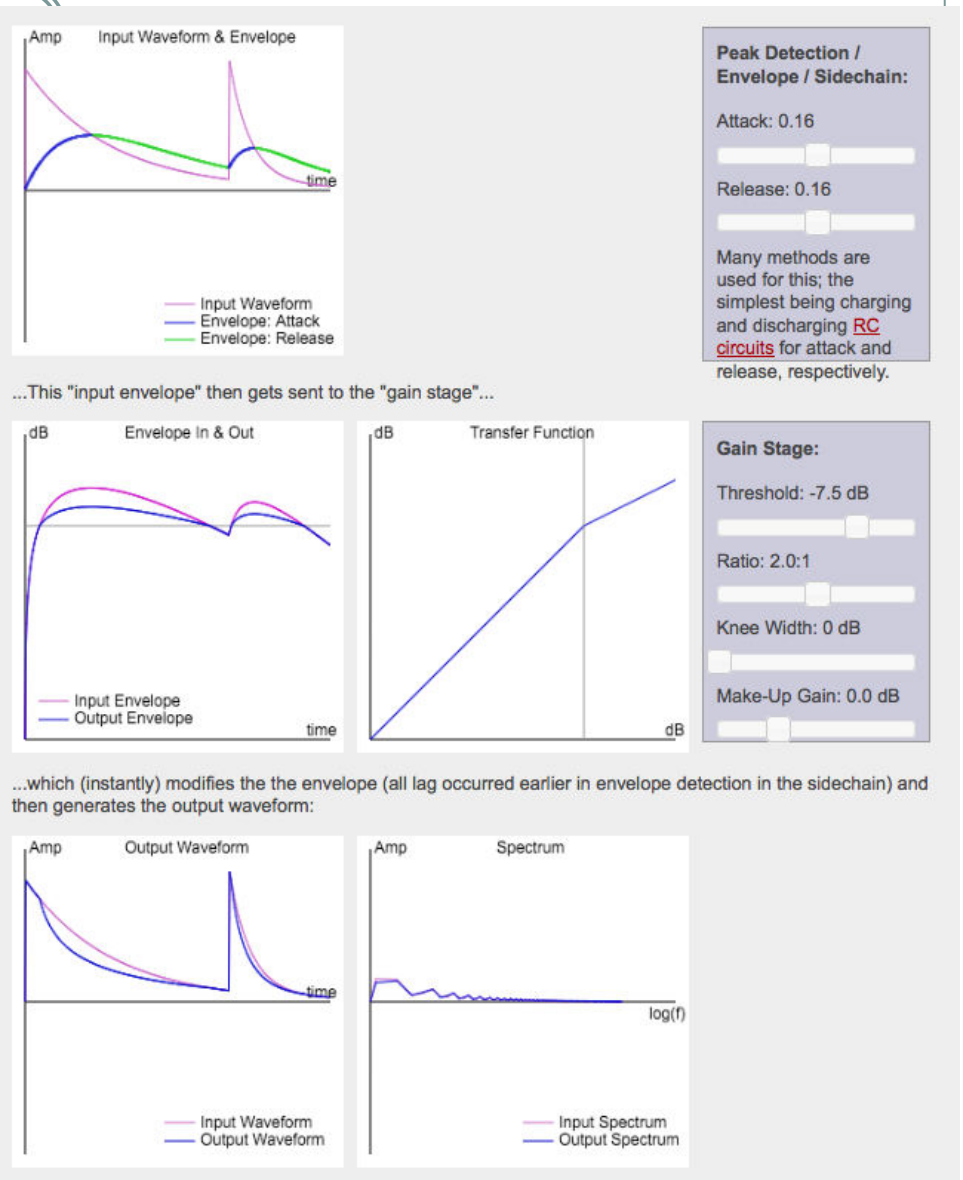
- In writing Acoustic Toolkit “SHAART”:
  - Lots of details in code, e.g. power spectrum calc’s
  - Additions like picture-to-waveform required writing an Inverse Short Time Fourier Transform (ISTFT) *from scratch*
  - Impulse Response measurements required understanding & implementing “de-pinking” (??)





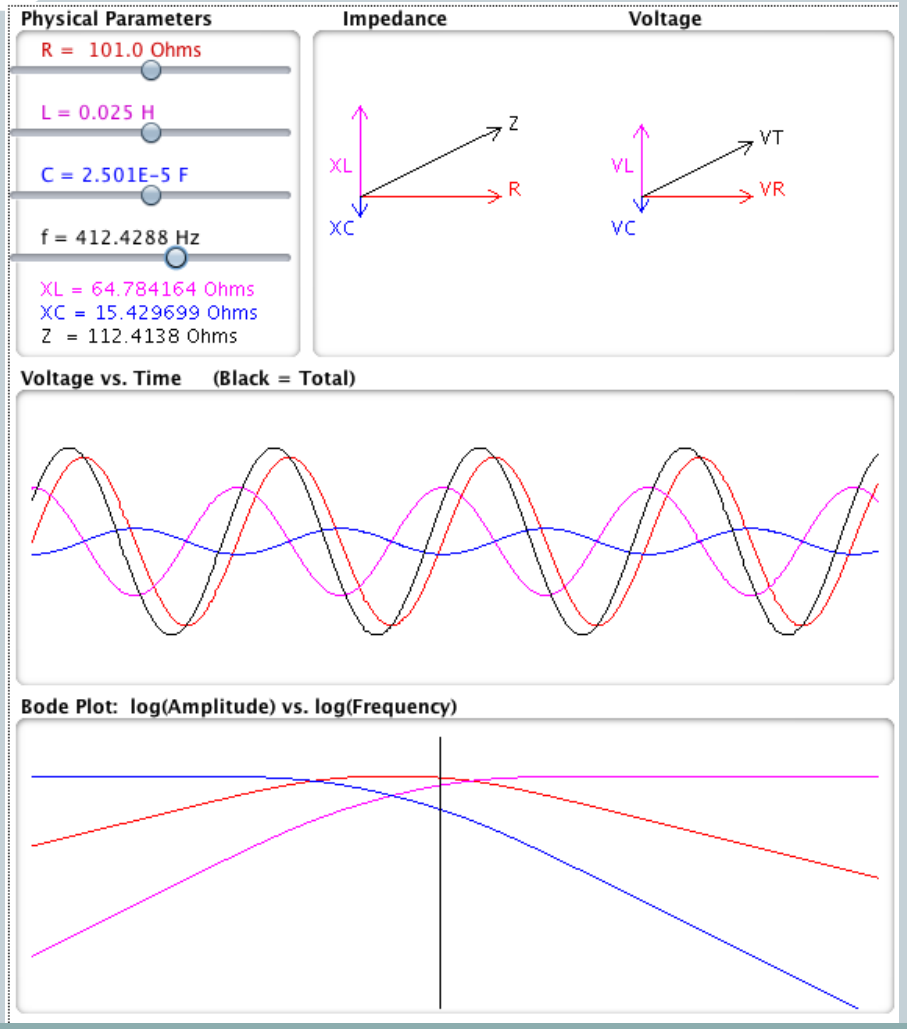
“…clarifies (mis)understanding”

- In writing Cartoon Compressor…
  - Discovered while writing that I was simply, *completely wrong* about the details how an audio compressor worked
  - After ‘getting it right’: “Ohhhh, now it all seems so simple!”



“…offers opportunities for discovery”

- In Series RLC Circuit demo…
- …noticed that changing frequency simply causes total voltage vector to ***rotate!***
- Never knew that before!



Caveats / Issues

- Use: Students often don't take advantage of app ☹
- Obsolescence: Language or implementation may fall out of favor / become unsupported: Flash, Java,…
  - One could argue that ALL implementations will inevitably become obsolete
  - Example: My Java demos, now rewriting in HTML5
- Platform:
  - Binary apps vs. student OS version
  - Web browser lack of support
- Forgetting
  - Had to relearn Swift over the weekend to write new app!

Scholarship & Moving On…

- Peer-reviewed app publication possible, e.g. via MERLOT
- Software development counts as Scholarship II for TP&L
  - App Store publication *very* stringent, could be Schol. I (Scholarship of Application)

- Brand new app as of today:
- Will use for lab tomorrow!

My apps page, including Clicker instructions:
http://hedges.belmont.edu/~shawley/physapplets