Resources available at: https://goo.gl/4rDN8y

TOOLS FOR WHO?

Matthew Turner

Department of Psychology, GSU

mturner46@gsu.edu

MAIN POINTS

- Two points:
  1. Students should build things
  2. We need to let our advanced students work with more powerful tools
- Is this a problem only in the sciences?

ASIDE: CODING CONUNDRUM!

- Over the course of the last two days I have met a lot of people who seem put off my "coding"
- Coding is seen as a kind of mystic art
- I find this odd—most of the people here are humanists, writing should be natural, shouldn't it?
- Historically coding **was** somewhat tedious, but modern languages are tightly fitted to the topics for which they are used
- Needless to say: this presentation presupposes that coding is justified as an activity

ACTIVE DESIGN

- Building things is important:
  - It exposes details present in problems that students don't recognize are there when just reading a description
  - It reveals the unexpressed (and unconscious) assumptions that students bring to a problem
  - When a student's built solution fails, it usually reveals the trouble spot directly in the failure
  - Embodying knowledge in artifacts (physical or software) forces students to co-create their understanding (producers versus consumers!)

ADVANCED STUDENTS

- One of my main concerns is how we train our advanced students
  - We have essentially abandoned them when it comes to software
  - Tools are specialized and domain specific (by topic/curriculum)
  - We artificially limit the students to simple tools which only work for simple problems which hurts their education

  - Most of the software we use to teach has training wheels on and we **never** take them off

ADVANCED STUDENTS

- We have to give the students powerful tools
  - There IS a **cost** to this – more powerful tools are harder to use and take training
  - But the cost is **worth it** in terms of understanding and applicability
- There is also a cultural shift needed
  - Students will resist this!
  - Faculty will resist this, too!

CONTRAST

- I want to contrast the two extremes in software in education:
  - Demos (toys)
    - Used to show a (single) concept to a class or laboratory
    - Hide details
  - Languages or Extensible Systems
    - Can be used very broadly across topics
    - Exposes details

My contention is very simple: we need fewer of the former and more of the latter!

DEMOS

- Demos are used in math, statistics, and much of science education
  - Teacher-centric - they show one thing and usually hide the details
  - They are focused exclusively on making things **easy** and **engaging**
  - Engagement is mostly defined by **look-and-feel** in demos: **colorful**, **pretty**, **entertaining**, etc.
- While this is one form of engagement, there are others:
  - Treating the students as participants
  - Allowing the students choice in how to do things
  - Giving the students the ability to ask new questions and explore topics of interest to them

DEMOS

- Demos are used in math, statistics, and much of science education
  - details
  - **gaging**
  - **colorful**,

Languages engage students on these things:

- While this is one form of engagement, there are others:
  - Treating the students as participants
  - Allowing the students choice in how to do things
  - Giving the students the ability to ask new questions and explore topics of interest to them

MAIN POINTS

- Two points:
  1. Students should build things
     - **We do not do enough of this**
  2. We need to let our advanced students work with more powerful tools
     - **Even at more advanced levels, we depend on software that is easy to use, rather than capable of making meaningful results**

EXAMPLE: MONTY HALL PROBLEM

- Player is on a game show:
  - Host shows 3 doors to the player
  - Behind one door is a sports car
  - Behind the other 2 are (stinky) goats
- Play proceeds like this:
  - Player picks a door
  - Host shows a goat (from an un-selected door!)
  - Player gets a chance to switch doors

Problem:

1. Should the player switch doors?
2. If the player switches doors does this change the chance of winning?

LINKS TO MONTY HALL GAMES

- http://www.stayorswitch.com/
- http://www.math.ucsd.edu/~crypto/Monty/monty.html
- http://www.shodor.org/interactivate/activities/SimpleMontyHall/
- http://www.grand-illusions.com/simulator/montysim.htm
- http://montyhallpuzzle.appspot.com/