

Lessons from Teaching HCI for a Diverse Student Population

Dr. Bowen Hui
bowen.hui@ubc.ca

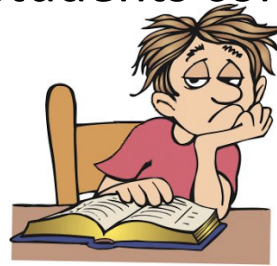
Computer Science
University of British Columbia Okanagan

Motivation

- Human computer interaction (HCI) is crucial to software acceptance and technological advances

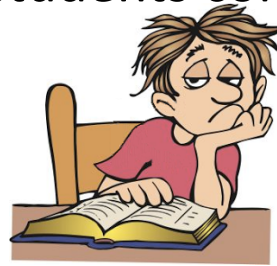
Motivation

- Human computer interaction (HCI) is crucial to software acceptance and technological advances
- Many HCI educators have reported students come with negative preconceptions about the course
 - Content was too easy
 - Grading too subjective
 - Exercises not challenging enough



Motivation

- Human computer interaction (HCI) is crucial to software acceptance and technological advances
- Many HCI educators have reported students come with negative preconceptions about the course
 - Content was too easy
 - Grading too subjective
 - Exercises not challenging enough
- Previous work to improve students' learning experience in HCI
 - Real user needs
 - Changing individualized summative assessments
 - Focusing on the design process rather than outcomes
 - Creating platform to enable exploration of design space
 - Interest based learning and project based learning
 - Rebranding HCI globally



Our Teaching Context

- Research based university
- Single undergraduate HCI course for third year CS students
 - Overview of theory and techniques
 - Probes further studies in the field
 - Touches on practical programming skills
- Mandatory for all CS majors
- Accommodates non-CS majors (~10%) and CS graduate students (~5%)



Experience 1: A Community Service Learning Approach

- Class size: < 40, prior to 2015
- **Goal: Apply design to real world problem**
- Semester long project to re-design website for non-profit organization



Experience 1: A Community Service Learning Approach

- Class size: < 40, prior to 2015
- **Goal: Apply design to real world problem**
- Semester long project to re-design website for non-profit organization
- Logistically demanding
 - Non-profit clients arranged in advance of the course
 - Students learned real user needs in situ
 - Project feedback needs to be timely and tailored to individual teams



Experience 1: A Community Service Learning Approach

- Class size: < 40, prior to 2015
- **Goal: Apply design to real world problem**
- Semester long project to re-design website for non-profit organization
- Logistically demanding
 - Non-profit clients arranged in advance of the course
 - Students learned real user needs in situ
 - Project feedback needs to be timely and tailored to individual teams
- Students worked in teams to tackle real world design problem
 - Often unbalanced in team membership
 - Students limited to building website in Wordpress
 - Students did not appreciate opportunity to work on real problem



Experience 1: A Community Service Learning Approach

- Class size: < 40, prior to 2015
- **Goal: Apply design to real world problem**
- Semester long project to re-design website for non-profit organization
- Logistically demanding
 - Non-profit clients arranged in advance of the course
 - Students learned real user needs in situ
 - Project feedback needs to be timely and tailored to individual teams
- Students worked in teams to tackle real world design problem
 - Often unbalanced in team membership
 - Students limited to building website in Wordpress
 - Students did not appreciate opportunity to work on real problem
- **Result: Approach abandoned due to increased class size**



Experience 2: A Dual-Track Approach

- Class size: 76 students in 2015
- **Goal: Meet desire to learn programming skills**
- Multiple assessments with two options: a design option and a programming option



Experience 2: A Dual-Track Approach

- Class size: 76 students in 2015
- **Goal: Meet desire to learn programming skills**
- Multiple assessments with two options: a design option and a programming option
- Students had choice
 - Students enjoyed flexibility
 - Accommodates diverse student skills
 - Some students were narrowly focused



Experience 2: A Dual-Track Approach

- Class size: 76 students in 2015
- **Goal: Meet desire to learn programming skills**
- Multiple assessments with two options: a design option and a programming option
- Students had choice
 - Students enjoyed flexibility
 - Accommodates diverse student skills
 - Some students were narrowly focused
- Increased content design
 - Assessments needed to be properly aligned
 - Content creation doubled
 - Increased administrative and grading needs



Experience 2: A Dual-Track Approach

- Class size: 76 students in 2015
- **Goal: Meet desire to learn programming skills**
- Multiple assessments with two options: a design option and a programming option
- Students had choice
 - Students enjoyed flexibility
 - Accommodates diverse student skills
 - Some students were narrowly focused
- Increased content design
 - Assessments needed to be properly aligned
 - Content creation doubled
 - Increased administrative and grading needs
- **Result: Approach not continued due to overhead**



Experience 3: A Mini-Studio Approach

- Class size: 77 students in 2017
- **Goal: Combine the pros from previous approaches**



Experience 3: A Mini-Studio Approach

- Class size: 77 students in 2017
- **Goal: Combine the pros from previous approaches**
- One-size-fits-all
 - Everyone completed the same assignments
 - Programming assignments to implement HCI techniques
 - Programming assignments were not well supported
 - Students felt programming assignments were too disconnected from concepts



Experience 3: A Mini-Studio Approach

- Class size: 77 students in 2017
- **Goal: Combine the pros from previous approaches**
- One-size-fits-all
 - Everyone completed the same assignments
 - Programming assignments to implement HCI techniques
 - Programming assignments were not well supported
 - Students felt programming assignments were too disconnected from concepts
- Mini-studio approach
 - Created small design challenges to apply concepts
 - Elicited topics from students for more realistic and relevant challenges
 - Condensed lectures to allow for studio time
 - Student attitudes toward design improved

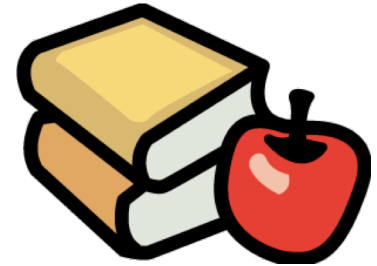


Experience 3: A Mini-Studio Approach

- Class size: 77 students in 2017
- **Goal: Combine the pros from previous approaches**
- One-size-fits-all
 - Everyone completed the same assignments
 - Programming assignments to implement HCI techniques
 - Programming assignments were not well supported
 - Students felt programming assignments were too disconnected from concepts
- Mini-studio approach
 - Created small design challenges to apply concepts
 - Elicited topics from students for more realistic and relevant challenges
 - Condensed lectures to allow for studio time
 - Student attitudes toward design improved
- **Result: Independent project needs to strengthen the ties between design, programming, evaluation**



Lessons Learned



1. Community service learning provides real design problem but not scalable.
2. CS students expect to gain programming skills that are applicable to real world application.
3. Students enjoyed having choice in the assessments, but we need to minimize overhead involved.
4. Mini-studio approach supported using condensed lectures were effective in changing students' attitude towards the “fluffiness” of design.
5. Individual, abstract assignments seem disconnected from each other.

Summary

- Teaching HCI is challenging
 - A mandatory undergraduate CS course
 - Single undergraduate HCI class
 - Accommodate non-CS majors with no programming
 - Accommodate CS graduate students
- Successes:
 - Choice, flexibility, mini-lecture format, studio-like activities, real world application
- Next step:
 - Design a project to illustrate a complete software development lifecycle

