User Interface Design

7 Dec 2009
CMPT140
Dr. Sean Ho
Trinity Western University
Quiz 6: 10min, 20pts

● Compare and contrast: stacks and queues. [5]
  ○ Describe the operations on each.

● Convert the following into Reverse Polish Notation and show all steps to evaluate: [4]
  ○ 10 - 7 * 2 > - 5 and 9 == 2 ** 3

● What is a widget? [2]

● Define and contrast: [4]
  ○ Unit testing vs. integration testing

● What are the principles of the spiral model of development? Contrast with waterfall/WADES [5]
Quiz 6 answers: #1-2

- Compare and contrast: stacks and queues. [5]
  - Describe the operations on each.
  - Stacks: LIFO, push(), pop()
  - Queues: FIFO, enqueue(), dequeue()

- Convert the following into Reverse Polish Notation and show all steps to evaluate: [4]
  
  - $10 - 7 \times 2 > -5$ and $9 == 2 ** 3$
  - $10 7 2 * -5 - > 9 2 3 ** ==$ and
  - $10 \rightarrow 10 7 \rightarrow 10 7 2 \rightarrow 10 14 \rightarrow -4 \rightarrow -4 5 \rightarrow$
  - $-4 -5 \rightarrow$ True $\rightarrow$ T 9 $\rightarrow$ T 9 2 $\rightarrow$ T 9 2 3 $\rightarrow$ T 9 8
  - $\rightarrow$ T F $\rightarrow$ False
Quiz 6 answers: #3-5

- What is a **widget**? [2]
  - Interactive graphical **component** of UI
  - Responds to user **input**

- Unit testing vs. integration testing: [4]
  - Unit: test individual **component** in isolation
  - Integration: how components work **together**

- What are the principles of the **spiral** model of development? Contrast with **waterfall/WADES** [5]
  - Spiral is **multiple** iterations of waterfall: requirements may **change**, need to **adapt**
Human-computer interface

- WIMP/GUIs are just one kind of an interface between humans and computers
  - Input and Output
- Automated bank machine touch-screen
- Touch-tone keypad
- Voice recognition, speech synthesis
- Car steering wheel and pedals
- Head tracking / eye tracking
- EEG
- Other possibilities?
Some UI design principles

- Know your users: programmers? Man-on-the-street? Grandma
- Be consistent: names, colours, layout, parts of speech
- Use metaphors carefully: desktop, canvas
- Use multiple levels of complexity
  - Let the user tradeoff safety for control
- Always show the current state of the program:
  - Waiting for password? Processing?
  - Progress bars (that don't lie!)
More UI design principles

- From Constantine+Lockwood
- Structure: hierarchy, layout: windows, tabs, etc.
- Simplicity: make common tasks easy
  - Epicentre: design around primary purpose
- Visibility: need-to-know basis
- Feedback: current state, errors, etc.
- Tolerance: be flexible to user mistakes, save user data / user's hard work
- Reuse: consistent naming, behaviour
A few UI case studies

- BBC website
- Vincent Laforet website
- Blender3D application
- Google Calendar