Design Thinking for Developers

Zack Jordan
2017-06-27

- What do ~~users~~ **humans** need?
  (examples)
  - mental models
    - ???
  - consistency
    - JAVASCRIPT
      - [ ] + [ ]
      - [ ] + { }
      - { } + [ ]
      - { } + { }
  - limited memory load
    - CSS
    - display: flex
      flex direction
      flex wrap
      flex flow
      flex grow
      flex shrink
      flex basis
      flex
      align-down
      align-content
      flex start
      flex end
      center
      space between
      space around
      stretch
      align-self
      justify content
      order
  - feedback
    -
  - error handling
    - hard to make
    - easy to recover from
- Who is the ~~user~~ **human**?
  - The World
  - Company/Team
  - Future You
- Terms
  - **Human-Centered Design**
  - Design
    vs. Art
  - Human
    vs. User
    - limited
    - "lazy"
    - has complicated memory
    - visual
    - requires a mental model
- Goals
  - Define design
    (and related terms)
  - Show the benefits of design
    (and the costs of bad design)
  - Share concrete applications
    (that will improve your career)
- Who Am I?
  (and why should you care?)
  -
  - Zack Jordan
    - **designer**
    - author
    - developer
    - composer
    - entrepreneur
    - etc.
- Resources
  - The Design of Everyday Things
    Donald Norman
  - Don't Make Me Think
    Steve Krug
- What can I do??
  - Style
    - Naming
      - examples
        - doThing()
          not 'thing()
        - isVisible
          (not negative!)
        - hasProperty
        - 4 * 5 * 8
          (no magic values!)
      - accomplishes
        - mental load
        - consistency
  - Code Reviews
    - User Advocates!
    - injectData(5, 5, 100, true, 'init', 15, 105, 127, 0, 0, 0, 0, 170, 1.8, false, false, true, 'RUD');
  - Test-Driven Development
    (resistance is futile)
    - ```
      if (Class == Null || Jess->Length() == 0) && (title->Endwith(@"\") || title->Endwith(@"\"))
      {
        title->RemoveChardAtIntle->Length()-1);
        Jess = data->GetLocationsPerWord(title);
      }
      else if (Class == Null || Jess->Length() == 0) && title->Length() > 0)
      {
        wrong = title->Chardatitle->Length()-1);
        if (!) == '\' && ( == '\')
        {
          title->Append(@"\");
        }
        else
        {
          title->Append(@"\");
        }
        Jess = data->GetLocationsPerWord(title);
      }
      ```
- What's a non-HCD world like?
  - real world
    -
      - WTF doors
    -
      - Norman doors
    - VCRs
    - git
      (forgive me)
    - ???
- Why don't we design well?
  - The Curse of Knowledge
  - No User Advocate
      ```