UI design

10 rules of good ui design

1. Make Everything the User Needs Readily Accessible
2. Be Consistent
3. Be Clear
4. Give Feedback
5. Use Recognition, Not Recall
6. Choose How People Will Interact First
7. Follow Design Standards
8. Elemental Hierarchy Matters
9. Keep Things Simple
10. Keep Your Users Free & In Control

Reading

https://www.joelonsoftware.com/category/uibook/

summary of mobile ui design

Challenges

- small screens
- fat fingers
- poor text entry

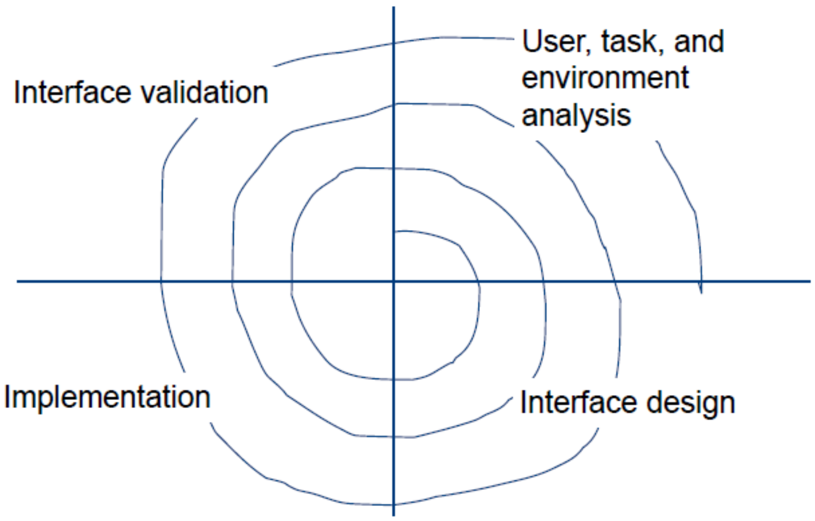
simplify

- follow design patterns
- use touch gestures where possible

Golden rules

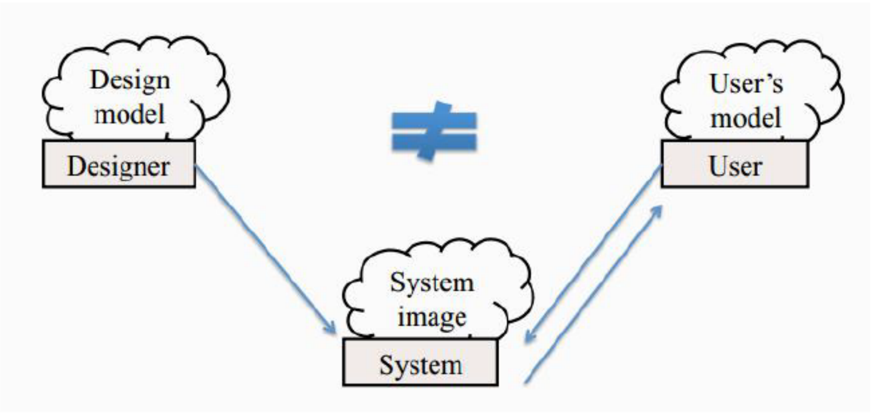
- let the user be in control
    - undo
    - macros
    - direct manipulation
    - modes
        - use a new window instead of a new node
        - make modes visible (signed in/logged out) 看看ppt p.28
- reduce the user's memory load
    - 询问保存密码，保存偏好
    - reduce demand on short-term memory
    - establish meaningful defaults
    - define intuitive shortcuts
    - use real-world metaphors
    - speak user's language
    - let user recognize, not remember
- be consistent
    - If you are using a shortcut in one program, keep it consistent in the other programs
    - use visual interface standards
        - for OS
        - for organization
        - for product or set of products
    - keep user from getting lost
    - system will explain itself

The UI design process



Models

- design model - what the designer thinks about the system
- user model – what the user thinks about the system
- system image – interface, manuals, training material, website



Design

初期

- What are users like?
- What do they think the system should be like?
- What is a single, consistent model of the system that can satisfy all the users?

后期

Design

- What should system be like?
- How can we make the users understand it?
- For each aspect of the system, design the system image to match the desired user model

Validation

- Does user model match our goal?

Task analysis and modeling

- what tasks will a user of the system perform
- high level – why people use the system
- low level – tasks involved in using the system

Tasks and use cases

- Use cases are high-level tasks
    - decompose high-level ones into low-level ones
    - find ones that are missing
    - simplify by generalizing
- UI design requires more detail than use case analysis usually provides

在我看来，tasks就是拆解成一个个的小的功能p50

Design model相当于定义对象以及对象的行为？p51

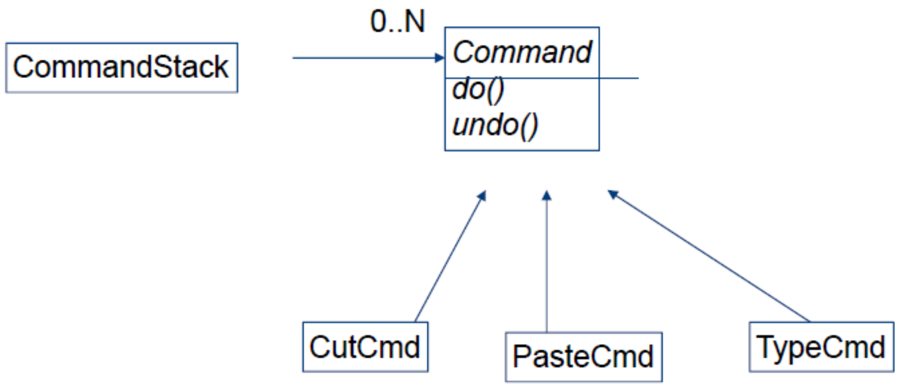
User interface design

- UI communicates with the user
- ps
    - needs feedback and iteration
    - there are standard ways of making a UI
    - Great UIs are rare and require creativity

low-level design

- Map task into actions that can be directly implemented by standard widgets
- Use consistent labels across tasks
- Use consistent widgets across tasks

command design pattern



response time

- 0.1s – limit for "instantaneous"
- 1 s – limit for "doesn't interrupt flow"
  - consider progress indicator
- 10s - limit for "keeping attention focused on dialog"
  - consider making it a bg task

evaluating ui–ui metrics

- size of written specification
- # of user tasks
- # of actions per task
- # of system states
- # of help messages

size of evaluation

- 统计上有效的样本数量应该是20-100
- 最常见的数量：5

Design alternatives

- novice users (初学者)
  - menus
  - make it look like something else
  - simple
- expert users
  - commands
  - specialize to make users efficient
  - powerful

Implementation concerns

- simplicity
  - tradeoff between # of features and simplicity
  - don't compromise usability for function
  - a well-designed interface fades into the background
  - basic functions should be obvious
  - advanced functions can be hidden
- safety
  - make actions predictable and reversible
  - each action does one thing
  - effects are visible
    - user should be able to tell whether operation has been performed
  - undo
- use standard lib/toolkits
  - don't build on your own
    - When to build you own

        - you are a platform provider
        - you have special needs and a lot of money
        - you are not in a hurry
        - you know what you are doing
    - provide familiar controls
    - provide consistency
    - reduce cost of implementation
    - lib designers probably better UI designers than you are
  - separate UI from application
    - MVC
    - 3-tier(层)：presentation, application, data
    - UI 和 应用独立变化（互不影响）
    - UI 和 应用是不同人开发的
    - Web page has just enough code to call the actual application logic
    - 优点
      - write automatic tests for application objects, not for UI
      - 写前端的不需要知道如何很好地编程
      - programmers don't need to be good UI designers
    - 缺点
      - application objects generate HTML
      - code tends to creep into web pages
    - 结果
      - Easier to test （自动化测试，冒烟测试）
      - Easier to change

Summary

- UI设计是个难题
  - 要理解用户
  - 要清楚问题
  - 要了解技术
  - 要懂如何去评价
- UI设计很重要
  - 用户看见的
  - 对于软件的成功与否起到关键性的作用