投票 最多可选1项

设置

# Are you ready ?

A Yes

B No


提交

# Review Ch14 Component Design

- What is component ?
- Component design principles
  1. TheSingle responsibility principle (SPR)
  2. The Open-Closed Principle (OCP)
  3. The Liskov Substitution Principle (LSP)
  4. The Dependency Inversion Principle (DIP)
  5. The Interface Segregation Principle (ISP)
- Component design steps
  - elaborate all classes (attributes and methods)
  - message / data
  - activity diagram/ flowchart /state diagram
  - deployment
- Coupling , Cohesion, Complexity

# Software Engineering

## Part 2
## Modeling

## Chapter 15
## User Interface Design

# Contents

# 15.1 Interface Design

**Easy to learn?**

**Easy to use?**

**Easy to understand?**

**interaction design ≠ art designing**

**interaction design = Interface Design based on HCI based**

**UI: User Interface**
**HCI : Human Computer Interaction**

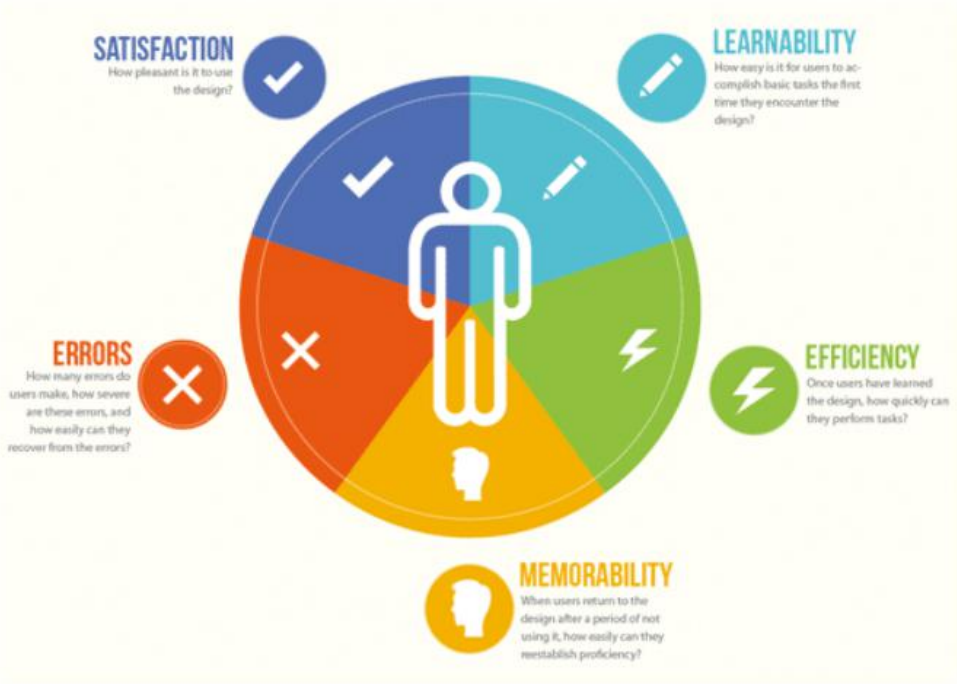
# 15.1 Interface Design

## Typical Design Errors

- lack of consistency
- too much memorization
- no guidance / help
- no context sensitivity
- poor response
- Arcane/unfriendly

# 15.1 Interface Design

**Usability: easy to learn, effective to use and provide an enjoyable experience**




What usability consists of

https://userbrain.net/blog/find-and-fix-usability-problems

# 15.1 Interface Design

- Place the user in control
- Reduce the user's memory load
- Make the interface consistent

# 15.1.1 Place the User in Control

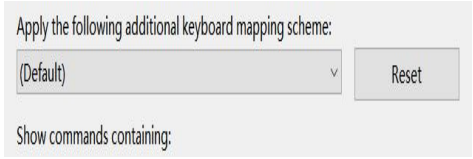

- Define interaction modes in a way that does not force a user into unnecessary or undesired actions.

- Provide for flexible interaction(keyborad, mouse, cmd, voice).
- Allow user interaction to be interruptible and u able.
- Streamline interaction as skill levels advance and allow the interaction to be customized (macro).
- Hide technical internals from the casual user.
- Design for direct interaction with objects that appear on the screen (drag into trash).

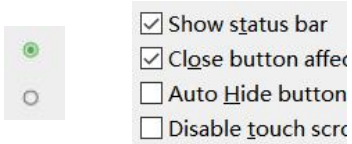
# 15.1.2 Reduce the User's Memory Load

the more a user has to remember,
the more error-prone the interaction will be

- Reduce demand on short-term memory.
- Establish meaningful defaults (reset).




- Define shortcuts that are intuitive (alt+p = print).
- The visual layout of the interface should be based on a real world metaphor.




- Disclose information in a progressive fashion. (organized hierarchically.)

# 15.1.2 Reduce the User's Memory Load

## *Violating a UI Golden Rule*

**The scene:** Vinod's cubicle, as user interface design begins.

**The players:** Vinod and Jamie, members of the *SafeHome* software engineering team.

**The conversation:**

**Jamie:** I've been thinking about the surveillance function interface.

**Vinod (smiling):** Thinking is good.

**Jamie:** I think maybe we can simplify matters some.

**Vinod:** Meaning?

**Jamie:** Well, what if we eliminate the floor plan entirely. It's flashy, but it's going to take serious development effort. Instead we just ask the user to specify the camera he wants to see and then display the video in a video window.

**Vinod:** How does the homeowner remember how many cameras are set up and where they are?

**Jamie (mildly irritated):** He's the homeowner; he should know.

**Vinod:** But what if he doesn't?

**Jamie:** He should.

**Vinod:** That's not the point . . . what if he forgets?

**Jamie:** Uh, we could provide a list of operational cameras and their locations.

**Vinod:** That's possible, but why should he have to ask for a list?

**Jamie:** Okay, we provide the list whether he asks or not.

**Vinod:** Better. At least he doesn't have to remember stuff that we can give him.

**Jamie (thinking for a moment):** But you like the floor plan, don't you?

**Vinod:** Uh huh.

**Jamie:** Which one will marketing like, do you think?

**Vinod:** You're kidding, right?

**Jamie:** No.

**Vinod:** Duh … the one with the flash … they love sexy product features … they're not interested in which is easier to build.

**Jamie (sighing):** Okay, maybe I'll prototype both.

**Vinod:** Good idea … then we let the customer decide.

# 15.1.3 Make the Interface Consistent

- Allow the user to put the current task into a meaningful context. (e.g. window title, icon, color)
- Maintain consistency across a family of applications (product line).
- If past interactive models have created user expectations, do not make changes unless there is a compelling reason to do so (ctrl+s).

# 主观题 10分

设置

Please describe some golden rule application points in software interface design?

正常使用主观题需2.0以上版本雨课堂

作答

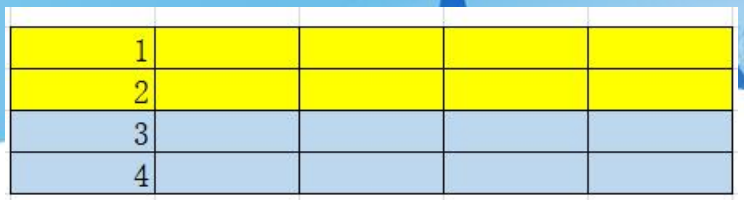
# 15.1.4 Design Principle

WYSIWYG: What You See Is What You Get

GUI elements：
- W(window)
- I(Icon)
- M(Menu)
- P(Pointing device)


https://www.bilibili.com/video/BV1Q741157ve?p=74 (tsinghua university SE)

# 15.1.4 Visibility

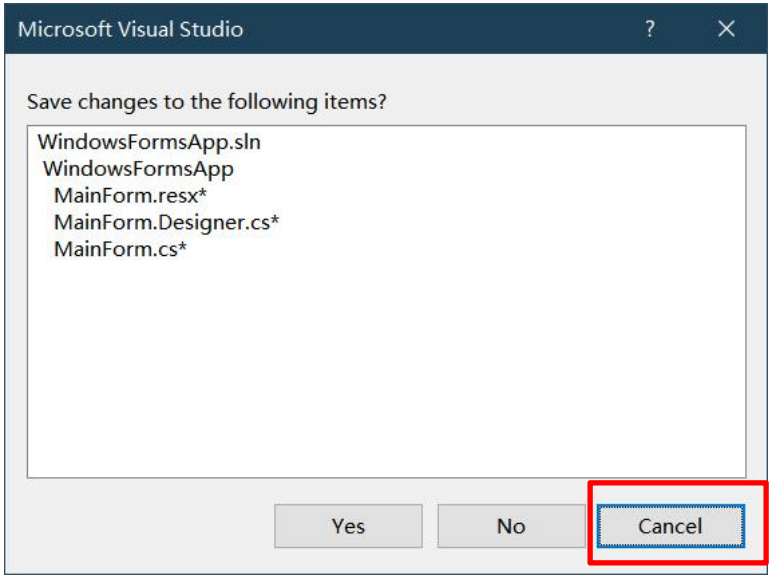


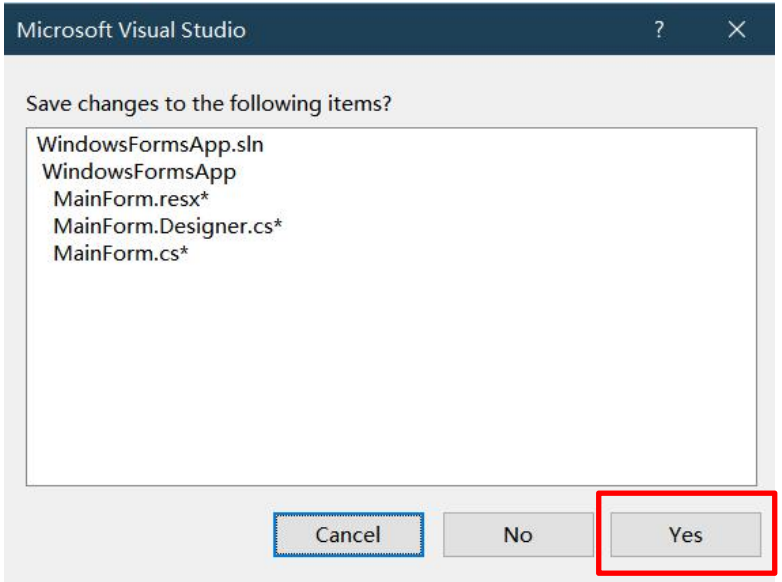

- grouping: grouping of items
  - ✓ logically together ⇒ physically together
  - ✓ Color can also be used for grouping
- sorting: order of items: match screen order
- alignment of items: read/scan, search, compare
- decoration
  - ✓ colors, fonts, boxes, animation
- white space between items
  - ✓ space to separate

https://www.bilibili.com/video/BV1Q741157ve?p=74 (tsinghua university SE)

# 15.1.4 Consistent

- Design interfaces to have similar operations and use similar elements for similar tasks.
- Main benefit is consistent interfaces are easier to learn and use

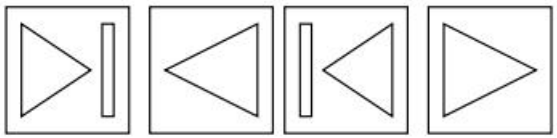


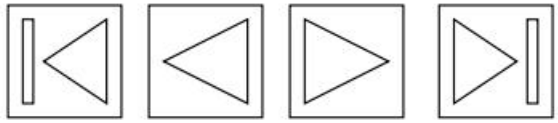
# 15.1.4 Mapping

Relationship between controls and their movements and the results in the world





A



B

https://www.bilibili.com/video/BV1Q741157ve?p=74

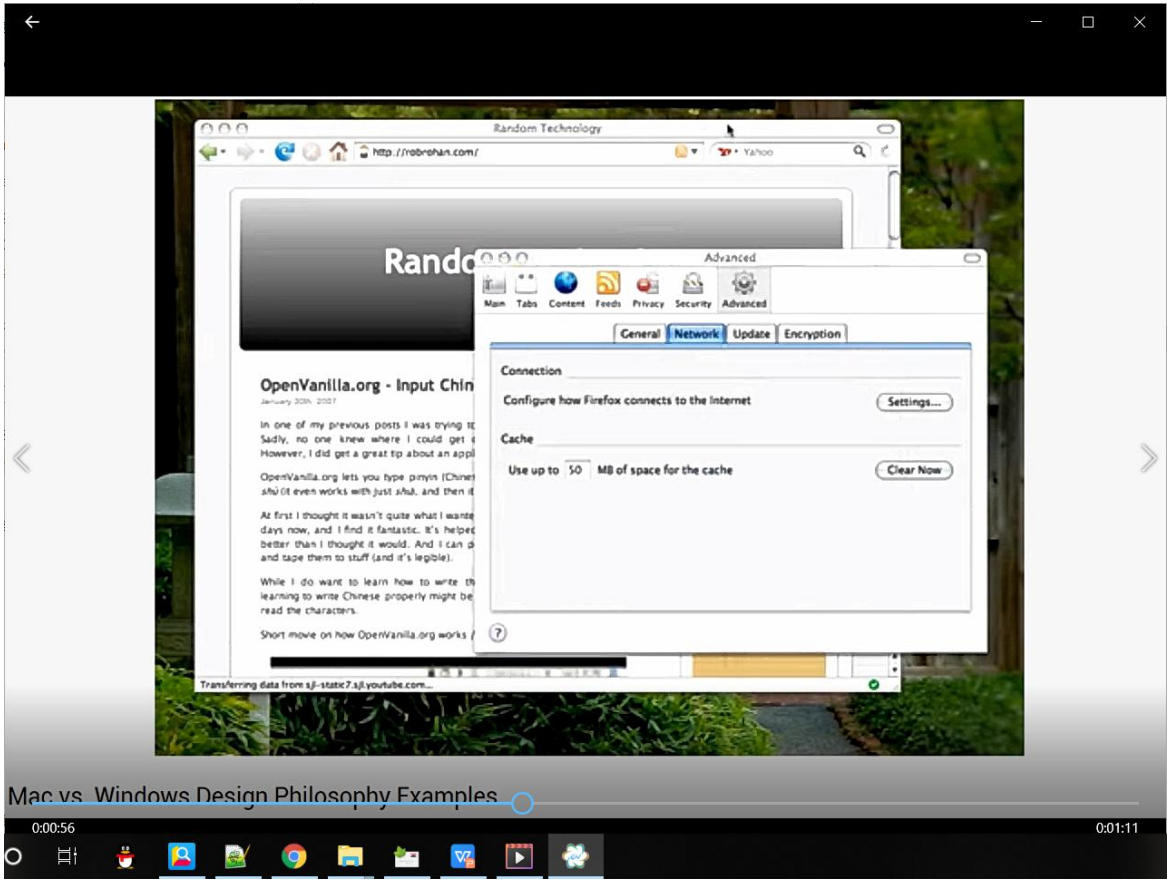
# 15.1.4 Feedback

**Sending information back to the user about what has been done**


https://www.bilibili.com/video/BV1Q741157ve?p=74

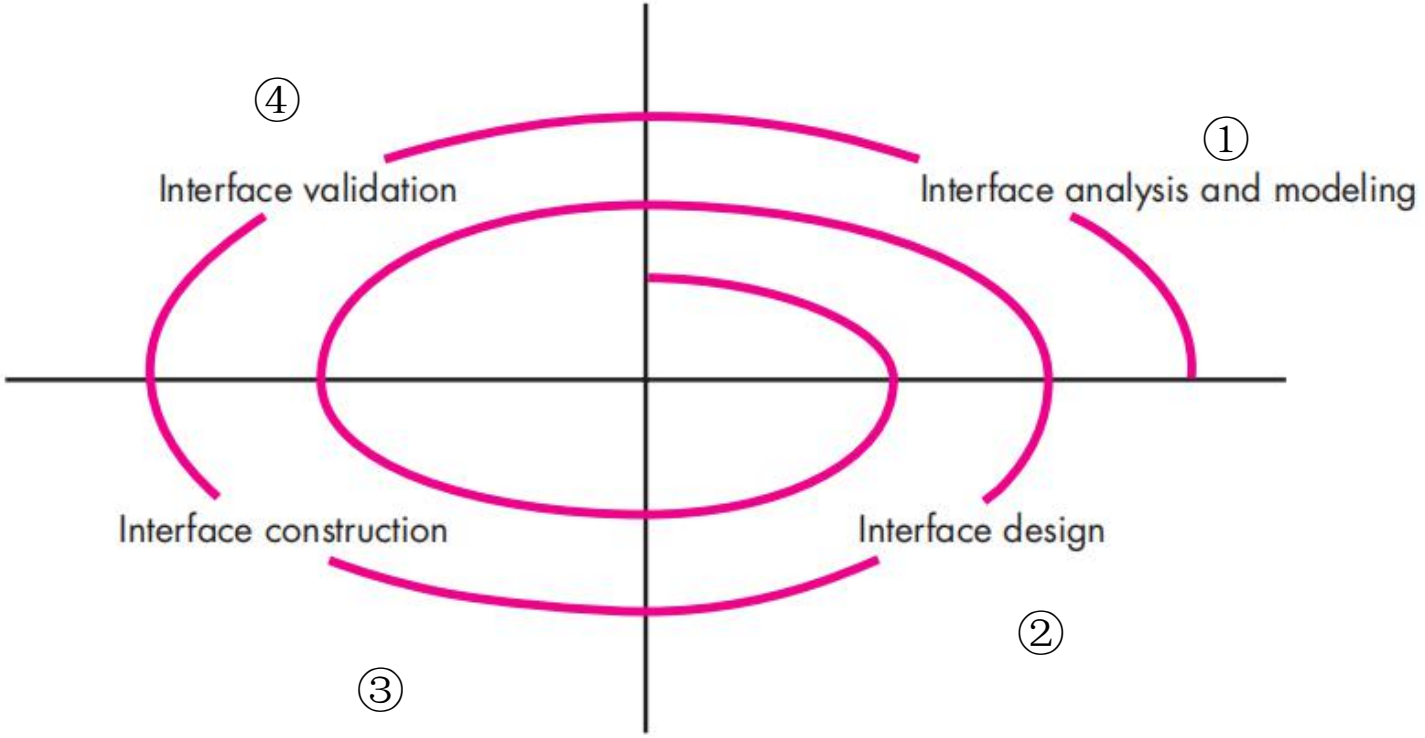
# 15.1.4 Design principle

Let's watch! What kind of principles are used?




https://www.youtube.com/watch?v=vdCvVVFJdns

# 15.2 User Interface Design Process

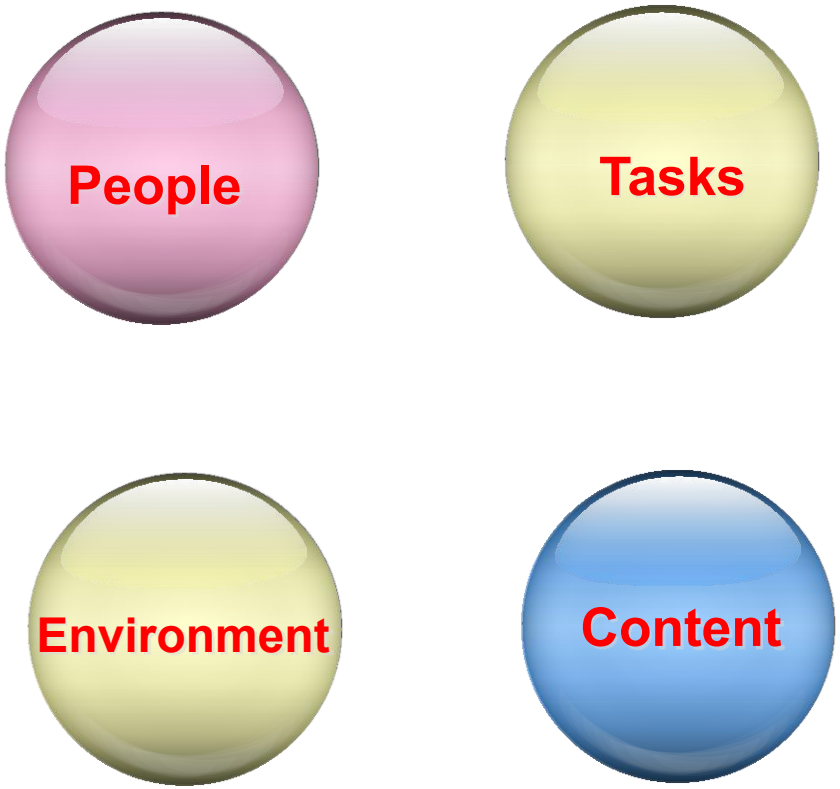



Spiral model

Goal: Define a set of interface objects and actions (and their screen representations) that enable a user to perform all defined tasks in a manner that meets every usability goal defined for the system.

# 15.3 Interface Analysis

Interface analysis **means** understanding



# 15.3.1 User Analysis

- Are users trained **professionals**, technician, clerical, or manufacturing workers?
- What level of **formal education** does the average user have?
- Are the users capable of learning from written materials or have they expressed a desire for classroom training?
- Are users expert typists or **keyboard** phobic?
- What is the **age** range of the user community?
- Will the users be represented predominately by one **gender**?
- How are users **compensated** for the work they perform?
- Do users work normal **office hours** or do they work until the job is done?
- Is the software to be an **integral part** of the work users do or will it be used only occasionally?
- What is the primary spoken **language** among users?
- What are the **consequences** if a user makes a **mistake** using the system?
- Are users experts in the subject matter that is addressed by the system?
- Do users want to know about the **technology** the sits behind the interface?

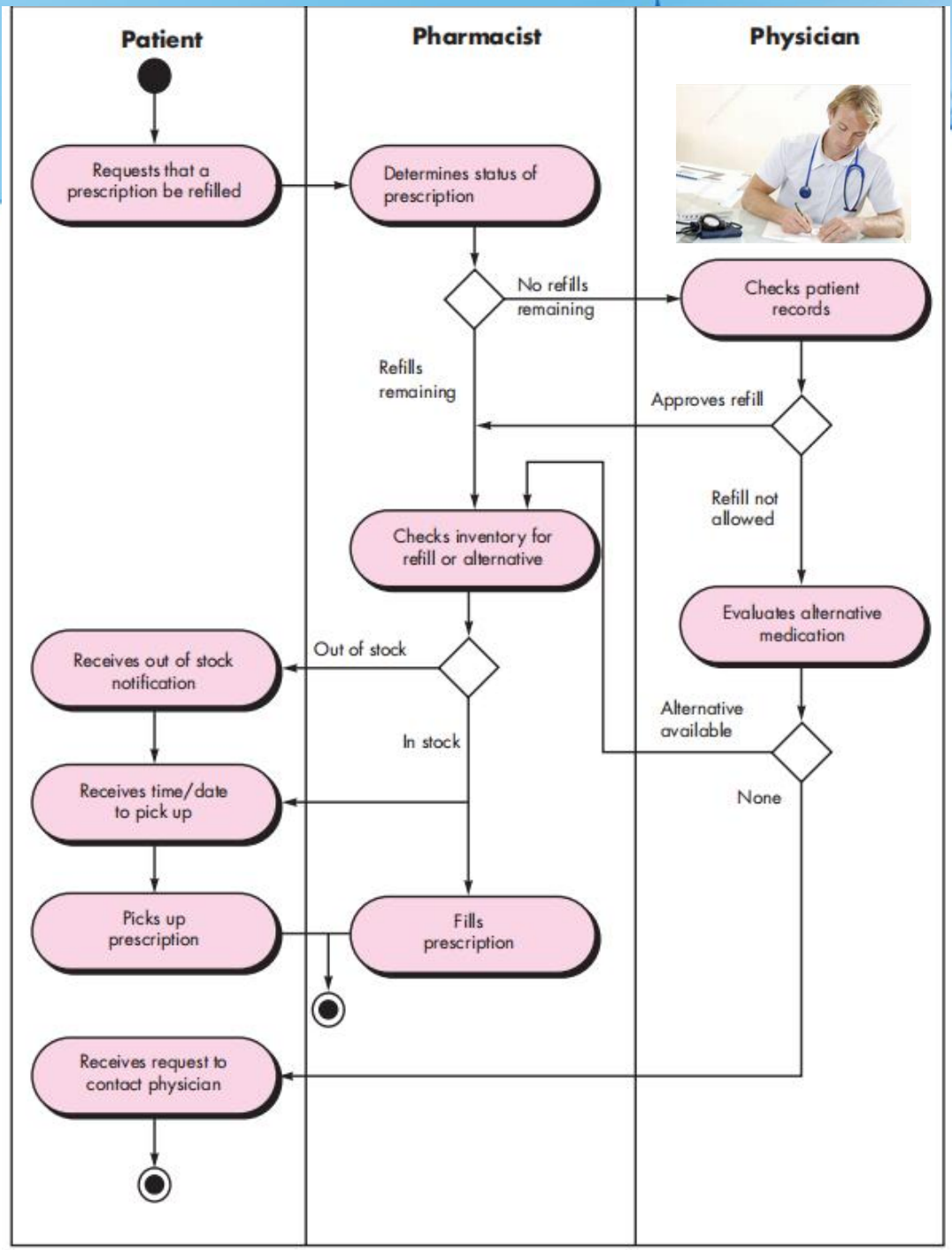
# 15.3.2 Task Analysis and Modeling

- The goal of analysis is to answer the following questions
  - What work will the user perform in specific circumstances?
  - What tasks and subtasks will be performed as the user does the work?
  - What specific problem domain objects will the user manipulate as work is performed?
  - What is the sequence of work tasks—the workflow?
  - What is the hierarchy of tasks?
- Use-cases define basic interaction
- Task elaboration refines interactive tasks
- Object elaboration identifies interface objects (classes)
- Workflow analysis defines how a work process is completed when **several people (**and roles) are involved**.**
- Hierarchical Representation define a task hierarchy.

# 15.3.2 Swimlane Diagram

**User task: Requests that a prescription be refilled**

- Provide identifying information.
- Specify name.
- Specify userid.
- Specify PIN and password.
- Specify prescription number.
- Specify date refill is required



# 15.3.3 Content analysis
## format and aesthetics

- Are different types of data assigned to consistent geographic locations on the screen (e.g., photos always appear in the upper right hand corner)?
- Can the user customize the screen location for content?
- Is proper on-screen identification assigned to all content?
- If a large report is to be presented, how should it be partitioned for ease of understanding?
- Will mechanisms be available for moving directly to summary information for large collections of data.
- Will graphical output be scaled to fit within the bounds of the display device that is used?
- How will color to be used to enhance understanding?
- How will error messages and warning be presented to the user?

# 15.4 Interface Design Steps

1. Using information developed during interface analysis, define interface objects and actions (operations).
2. Define events (user actions) that will cause the state of the user interface to change.
3. Depict each interface state as it will actually look to the end-user.
4. Indicate how the user interprets the state of the system from information provided through the interface.

# 15.4 Interface Design Steps

- Preliminary use case

**Preliminary use case:** I want to gain access to my *SafeHome* system from any remote location via the Internet. Using browser software operating on my notebook computer (while I'm at work or traveling), I can determine the status of the alarm system, arm or disarm the system, reconfigure security zones, and view different rooms within the house via preinstalled video cameras.

To access *SafeHome* from a remote location, I provide an identifier and a password. These define levels of access (e.g., all users may not be able to reconfigure the system) and provide security. Once validated, I can check the status of the system and change the status by arming or disarming *SafeHome*. I can reconfigure the system by displaying a floor plan of the house, viewing each of the security sensors, displaying each currently configured zone, and modifying zones as required. I can view the interior of the house via strategically placed video cameras. I can pan and zoom each camera to provide different views of the interior.

# 15.4 Interface Design Steps

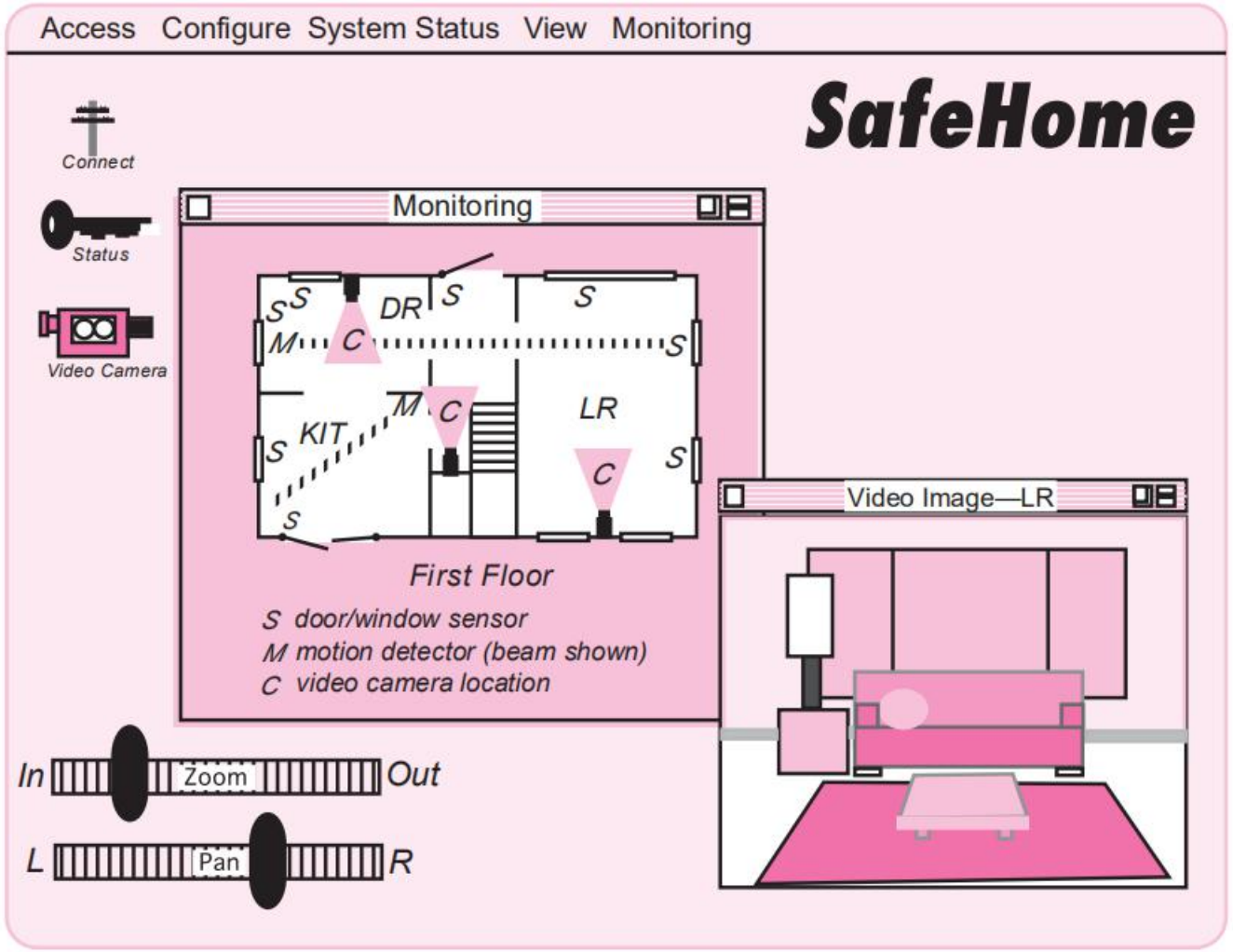
- Preliminary use case

Based on this use case, the following homeowner tasks, objects, and data items are identified:

- *accesses* the *SafeHome* system
- *enters* an **ID** and **password** to allow remote access
- *checks* **system status**
- *arms* or *disarms SafeHome* system
- *displays* **floor plan** and **sensor locations**
- *displays* **zones** on floor plan
- *changes* **zones** on floor plan
- *displays* **video camera locations** on floor plan
- *selects* **video camera** for viewing
- *views* **video images** (four frames per second)
- *pans* or *zooms* the **video camera**

# 15.4 Interface Design Steps

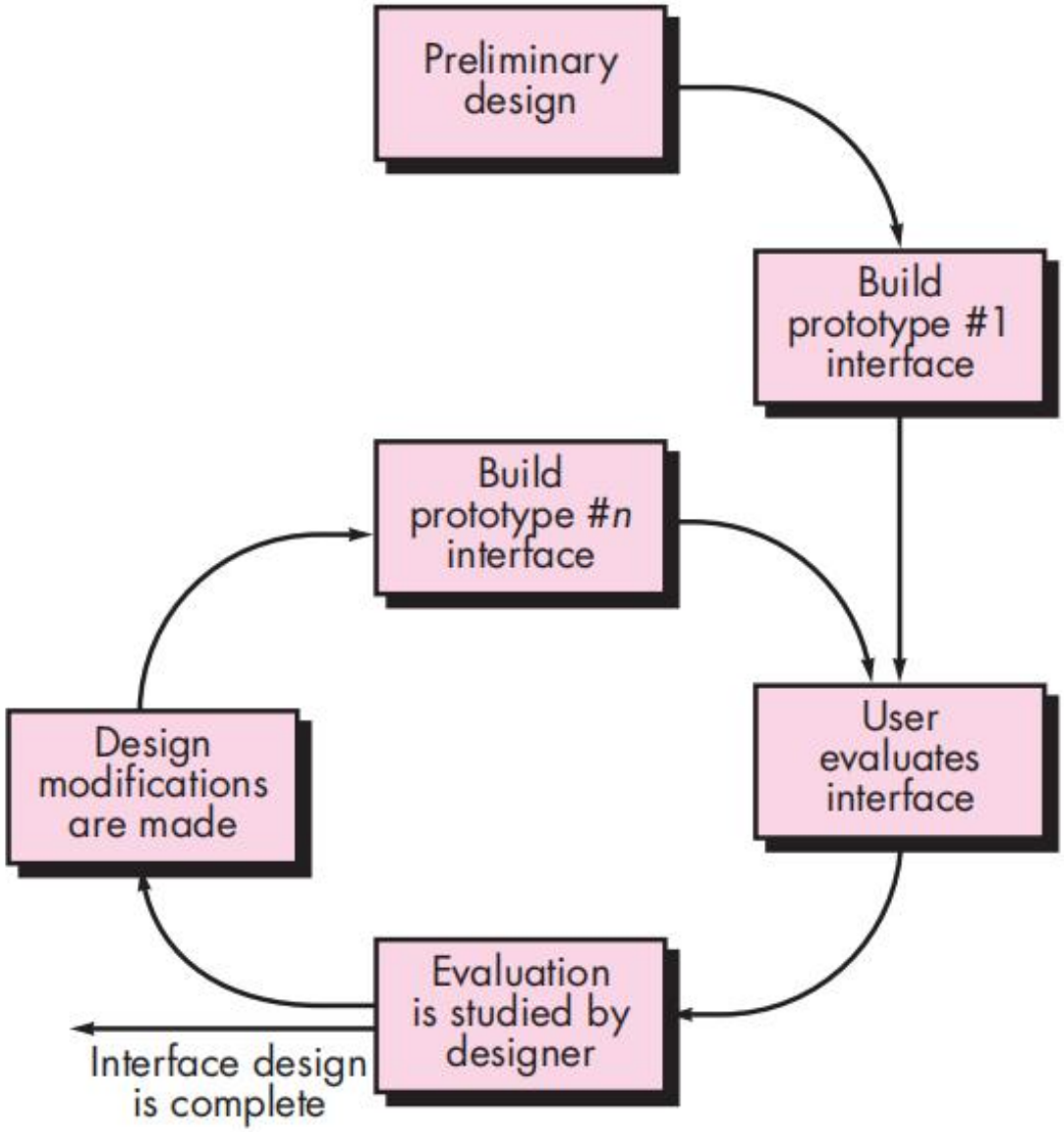
Preliminary sketch



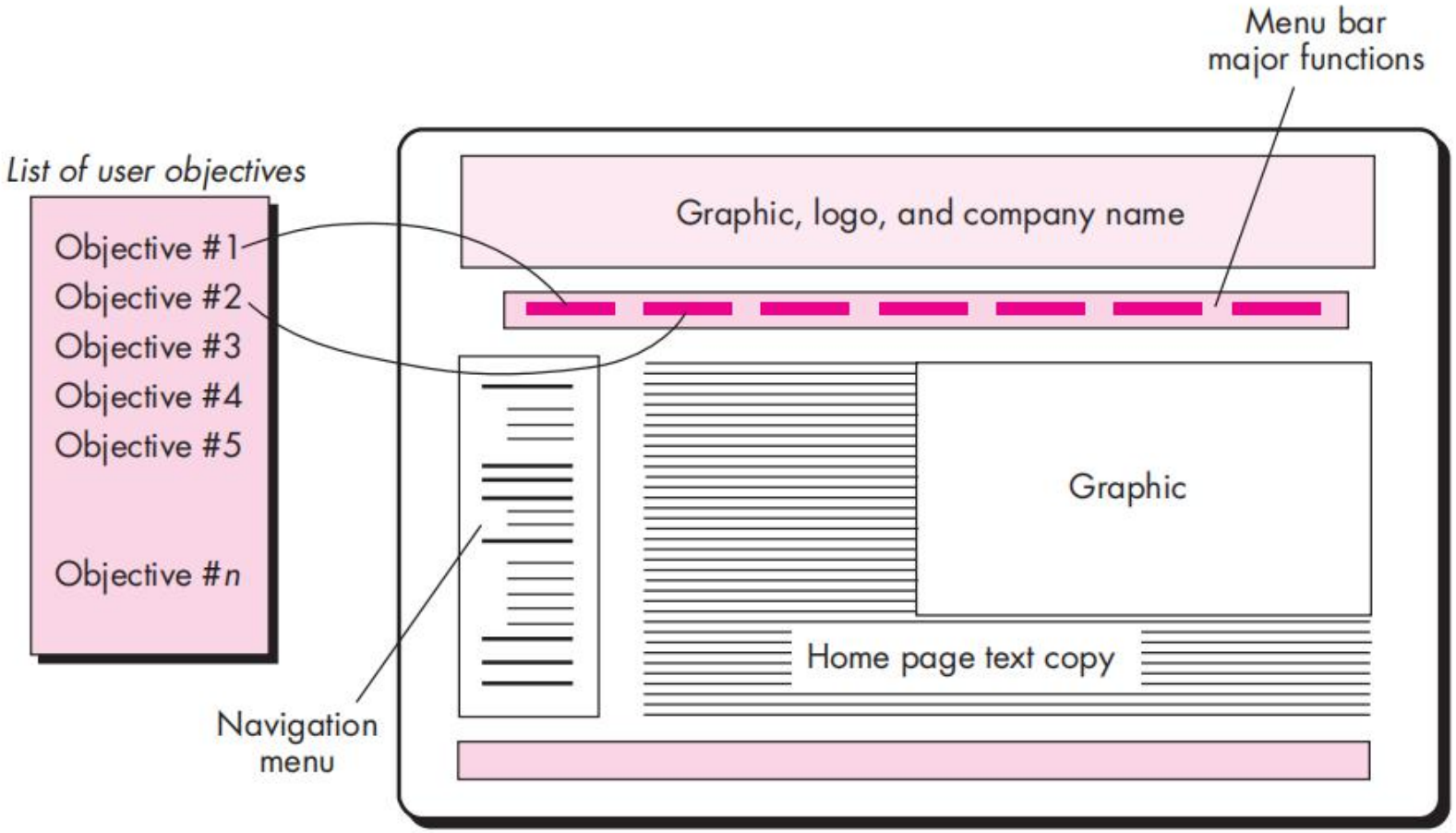
# 15.4.3 Design Issues

- Response time
- Help facilities
- Error handling
- Menu and command labeling
- Application accessibility
- Internationalization

# 15.5 Design Evaluation Cycle



# 15.5 Mapping User Objectives



# Take a break



## Five minutes

# 15.6 KLM (Keystroke Level Model)

**Measure how long will a user take to perform a particular operation**

- ✓ **Keystroke-Level Model**
  - **Notations**
  - **Interface Timings**
  - **Calculation Rules**
  - **An Example**
- ✓ **The fastest possible interface?**


KLM:
https://www.xuetangx.com/learn/THU08091000367/THU08091000367/5883555/video/9213416 (Chinese - tsinghua university SE)

# 15.6 KLM Notaions

Tap: pressing and releasing a key

Click: positioning the GID and then tapping the GID button

Drag: pressing the GID button at one location and then moving the GID before releasing the GID button at a new location

Double click: positioning the GID and then tapping the GID button twice quickly

# 15.6 KLM Notaions

## Denote keyboard operations

- ↓ immediately after the name of a key to indicate that the key has been pressed and to be held: Shift↓
- ↑ immediately after the name of a key to indicate the release of the key: Shift↑
- t↓ t↑ , t↓↑ , t
- A space separates the notation of consecutive actions
- A tap of the space bar is represented by the notation Space
- s p a c e, s↓s↑p↓p↑a↓a↑c↓c↑e↓e↑
- Shift↓ n Control↓ k Shift↑ w Control↑

# 15.6 KLM: Timing

- **Interface Timings**
  - The time it takes the user-computer system to perform a task is the sum of the times the system to perform the serial elementary gestures that the task comprises

| Naming | Typical value | Meaning |
|---|---|---|
| Keying, K | 0.2s | Tap a key on the keyboard |
| Pointing, P | 1.1s | Point to a position on a display |
| Homing, H | 0.4s | User's hand move from the keyboard to Mouse or other GID(Graphical Input Device) or from GID to keyboard |
| Mentally preparing, M | 1.35s | Prepare mentally for the next step |
| Responding, R | | Wait for a computer to respond to input |

# 15.6 KLM: Rules

| Naming |
| --- |
| Keying, K |
| Pointing, P |
| Homing, H |
| Mentally preparing, M |
| Responding, R |

- **Calculation Rules**

  **Figuring out at what points the user will stop to perform an unconscious mental operation (M)**

**Rule 0 Initial insertion of candidate Ms**

- Insert Ms in front of all Ks
- Insert Ms in front of all Ps that select command
  (but not for any Ps that point to arguments of those commands)

**Rule 1 Deletion of anticipated Ms**

If an operation symbol following an M is fully anticipated just previous to that M, delete the M. Pointing and clicking (you move the GID with the intent of tapping the GID button when you reach the target), PMK, PK

PMK: (moving)pointing -> ~~mentally preparing~~ -> keying
PK: (moving)pointing -> keying

# 15.6 KLM: Rules

KKKKKKK -> MKMKMKMKMKMKMK
-> MKKKKKKK

- **Calculation Rules**

**Rule 2 Deletion of Ms within cognitive units**

If a string of MKs belongs to a cognitive unit, delete the all the Ms but the first. a command name, an argument to a command: typing 4567.89

**Rule 3 Deletion of Ms before consecutive terminator**

If a K is a redundant delimiter at the end of a cognitive unit (such as the delimiter of a command immediately following the delimiter of its argument), delete the M in front of it.

| Naming |
| --- |
| Keying, K |
| Pointing, P |
| Homing, H |
| Mentally preparing, M |
| Responding, R |

Linux command: ls - a
("-" is a delimiter, so we can delete M before K)

# 15.6 KLM: Rules

Select * from s ; -> select * from s

- **Calculation Rules**

**Rule 4 Deletion of Ms for the terminators of commands**

If a K is a delimiter that follows a constant string (such as command name or any typed entity that is the same every time you use it), delete it. But if the K is a delimiter for an argument string or any string that can vary, then keep the M in front of it.

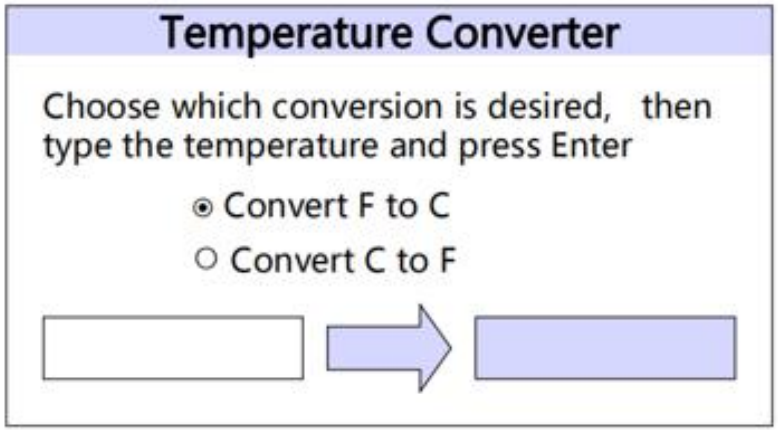
**Rule 5 Deletion of overlapped Ms**

Don't count any portion of an M that overlaps an R

| Naming |
| --- |
| Keying, K |
| Pointing, P |
| Homing, H |
| Mentally preparing, M |
| Responding, R |

# 15.6 KLM: Example

- H (move hand to GID)
- HP (point to the desired radio button)
- HPK(click the radio button)
- HPKH (move hand back to keyboard)
- HPKHKKKK (type 4 characters, e.g: 23.5)
- HPKHKKKKK (tap Enter)
- Rule 0: HMPMKHMKMKMKMKMK
  (Insert M before K,P)
- Rule 1、2、4: HMPKHMKKKKMK

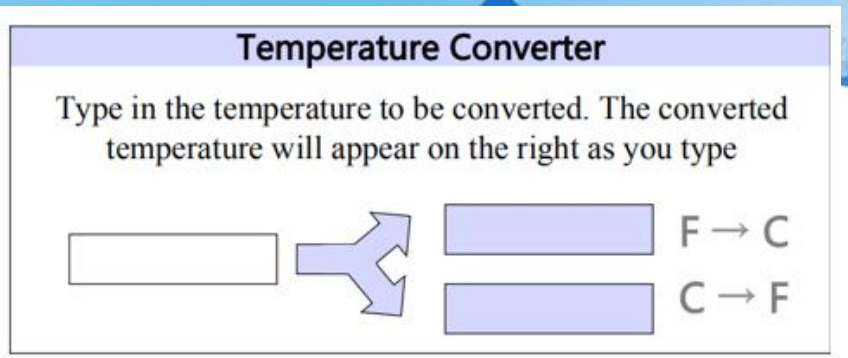



Task: C->F
Init: keyboard

- ∑=0.4+1.35+1.1+0.2+0.4+1.35+4X(0.2)+1.35+0.2=7.15
- : MKKKKMK = 3.7 Task: F->C

(7.15+3.7)/2=5.4 second

# 15.6 KLM: Example




**The fastest possible interface?**

The amount of data conveyed by a communication is the information an user has to provide to complete the special task, which has a lower bound.

This minimal amount is ***independent*** of the design of the interface.

***Consider the information required for a task***

- if the possible message forms (the value of the temperature) of the temperature converter are, -.dd, -d.d, .ddd, d.dd and dd.d
- achieving an interface that required 4 keystrokes, will give us the **highest efficiency**

- **In KLM, MKKKK=2.15s**

# 15.6 Fitts law

$T = a + b\log_2(D/S + 1)$ **describes the time taken to hit a screen target:**

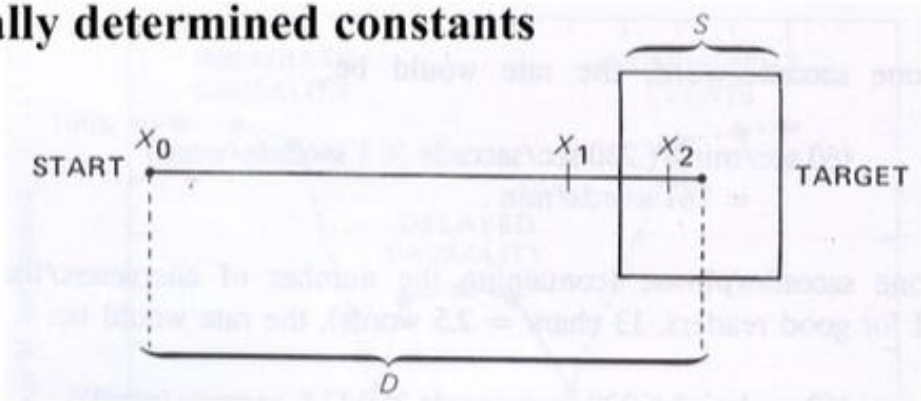
**where:** **a and b are empirically determined constants**

**a=50, b=150**

**T is movement time**

**D is Distance**

**S is Size of target**



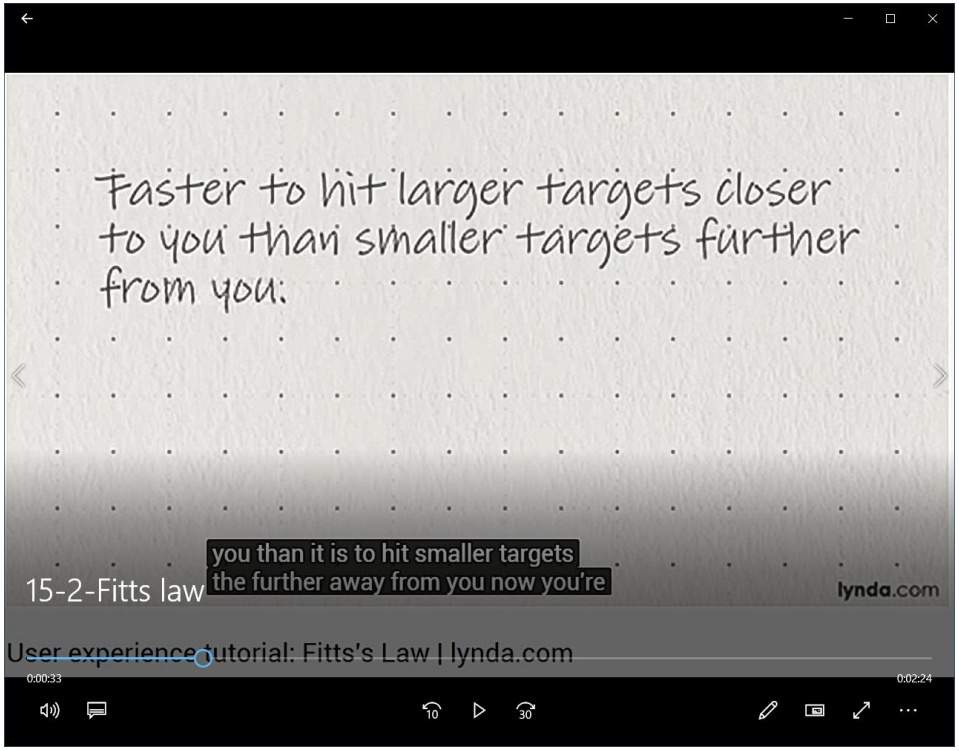

⇒ **targets as large as possible**

**distances as small as possible**

**http://fww.few.vu.nl/hci/interactive/fitts/**

# 15.6 Fitts law

Let's watch!




https://www.youtube.com/watch?v=95RoKSFyQ_k

# 15.6 Fitts law

$T = a + b\log_2(D/S + 1)$ **describes the time taken to hit a screen target:**

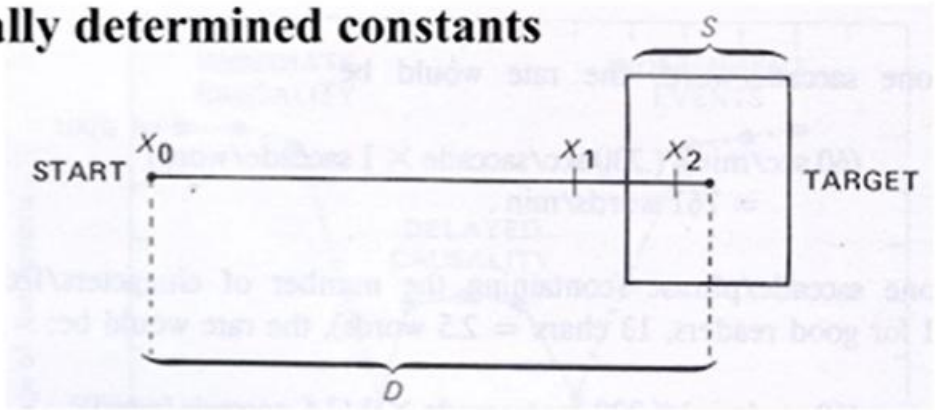
**where:** **a and b are empirically determined constants**

**a=50, b=150**

**T is movement time**

**D is Distance**

**S is Size of target**




⇒ **targets as large as possible**

**distances as small as possible**

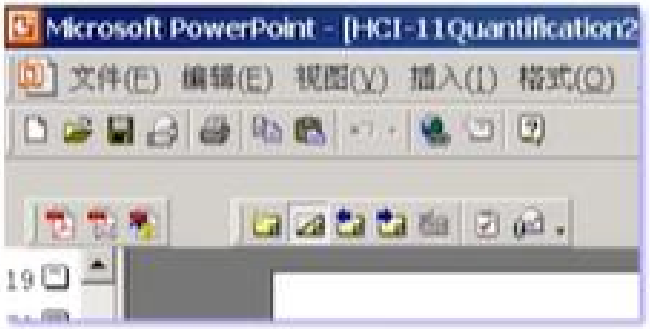
https://www.xuetangx.com/learn/THU08091000367/THU08091000367/5883555/video/9213419

# 15.6 Fitts law

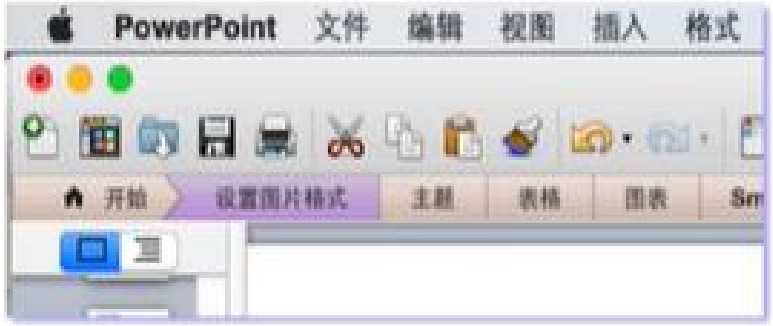
Task: select “文件”（file）menu => which design is more efficient?

$$T = a + b\log_2(D/S + 1)$$

D: moving distance
S: the size of target




s=5




s=50

$50+150\log_2(80/50+1)=256ms$

$50+150\log_2(80/5+1) =663ms$

✓ **Fitts’ Law based GUI enhancement**

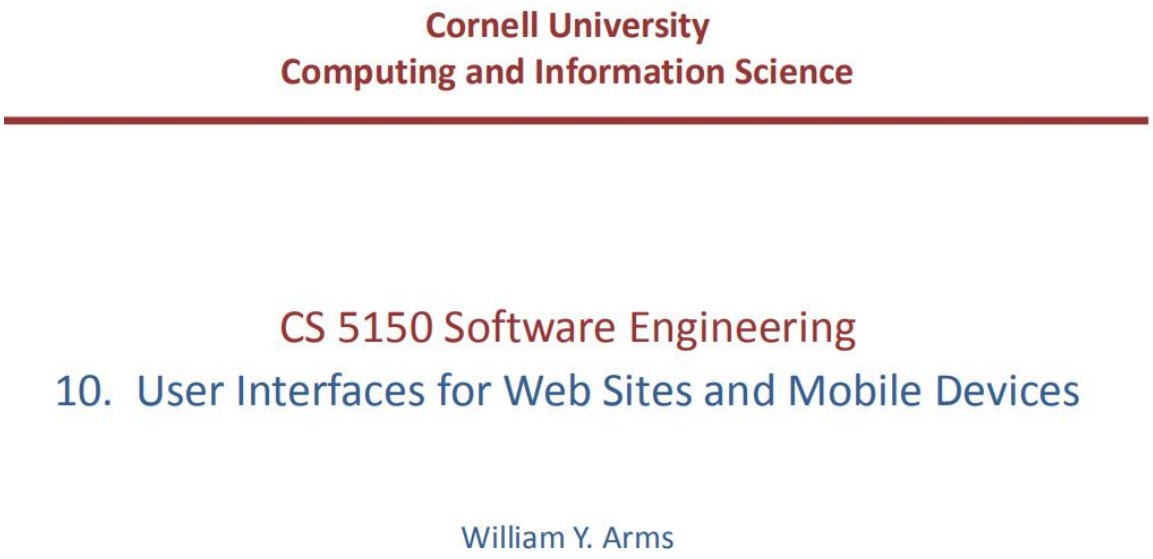
- Decreasing D
- Increasing S
- Decreasing D and Increasing S

https://www.xuetangx.com/learn/THU08091000367/THU08091000367/5883555/video/9213419

# Reading

CS5150 Software Engineering(Cornell University)

- ch9. The User Experience
- ch10. User interfaces for Web Sites and Mobile
- ch11. Evaluation and User Testing




https://www.bilibili.com/video/BV1aW411E7yr?p=15

# Summary

- Interface design golden rules
  1. Place the user in control
  2. Reduce the user's memory load
  3. Make the interface consistent
- Interface design principles
  - visibility / consistent/ mapping/ feedback
- Interface analysis and design step
  - People / Task / Enviroment / Content
  - interface objects and actions / event / state
- Design evaluation: KLM, Fitts

# Practice

Please use three GUI design golden rules and efficient modes(KLM/Fitts law) to analysis one GUI you have done before.

Share your analysis with us next time

# THE END