People we care!!

- Indian Orthodox Church
- Bunko players
- Black Engineers Society
- University of Maryland Surf Club
- University of Maryland's Wushu Club
- Residents of The University View building
- President of Student Educators of Young Children
- Active Minds
- Church of the Ascension
- UMD triathletes
- Community Portal
- KSP Russian Folk Music Festivals
- Video gamers
- Boy Scout Troop 444
- Maryland Ale and Lager Technicians
- Muslim Youth Soccer League A
- Student Community for Outreach, Retention, and Excellence (SCORE)
- South Asian American Fusion A Capella group
- Computer Programming Club
- Department of Resident Life student employees
- Women's Ultimate Frisbee Team
- University of Maryland Navigators
- UMCP Language house- Chinese cluster
- Tutor finding service
- Literacy Council of Northern Virginia
- Army ROTC
- IEC Mosque
- Ultimate Frisbee Team
- Sigma Beta Rho
- Public Interest Research Group
- Fountain Baptist Church
- WMUC Sports Radio
- Association for Women in Computing
- Elementary school educators
- Muslim Student Association's Brothers Sports and Social group
- Meditation group
- Gamer Symphony Orchestra
- Southern Maryland Chapter of Jack and Jill
- Ethiopian Students Association of the University of Maryland College Park
Lecture 9: Learnability

March 1
Today’s topics

• Learnability
• Design for learnability
• Evaluate learnability
Definitions

• Allowing users to reach a reasonable level of usage proficiency within a short time [Nielsen]

• The time it takes users to learn how to use the commands relevant to a set of tasks [Shneiderman]

• Ease at which new user can begin effective interaction and achieve maximal performance [Dix]
Scope

- **Initial learnability**
  - User’s initial performance with the system

- **Extended learnability**
  - User’s improved performance with the system over time
First encounter

What was it like when you first
• did a particular task?
• used a particular tool for that task?
First encounter: tasks

• Everyday tasks
  – Send a mail
  – Buy a product
  – Talk to someone

• Computer tasks
  – Run a virus scanner
  – Write a program to compute something
First encounter: tools

• Software tools
  – Email
  – Web browser
  – Course Wiki
  – Eclipse
  – Flex

• Hardware tools
  – Mouse
  – Keyboard
  – Wii remote
  – Guitar hero
Initial learning

• I need to get X done using a computer.
• I don’t know how to do it.
  – Which tool?
  – How to operate the tool?
• I need to learn.
• I need to choose a learning strategy.
• I use the strategy to learn.
• I apply what I learned.
• I complete my task.
  – If not, goes back to an earlier step
Extended learning

• What else (other tasks) can I do with this tool?
• Is there a better way to use this tool?
• Is there a better tool for this task?

• Better: faster, easier to remember, fewer errors, more enjoyable
How do you learn?

- Ask someone?
- Read manual?
- Take courses?
- Watch videos?
- Follow examples?
- Explore, trial and error
What kinds of learner are you?

- Visual learners
- Auditory learners
- Tactile/Kinesthetic learner
- Logical leaner
- Social leaner
- Solitary learner
I understand something better if I

a) watch someone else did it
b) listen to someone’s explanation
c) try it out myself
d) think it through
e) talk about it with someone
Learnability depends on individual skill differences

• Level of experience with computers
  – E.g., How computers work, what can be done

• Level of experience with interface
  – E.g., UI hardware, UI widgets

• Quality of domain knowledge
  – E.g., Word processing, graphics editing,

• Experience with similar software
  – E.g., An earlier version, a competing product
Design
Two ways to achieve good learnability through design

• Minimize the needs for learning
  – Ideally: No need to learn
  – Practically: Self-explanatory

• Make learning easy
Design principles

• Minimize the need for learning
  – Visibility
  – Feedback
  – Affordance
  – Natural mapping
  – Mental models
  – Metaphors
  – Consistency
  – Speak the user’s language
  – Platform standards

• Make learning easy
  – Help and documentation
  – Transition from novice to expert
Visibility: Toolbar icons
Visibility

• Relevant parts of system should be visible
• Relevant to users’ tasks
Visibility: Toolbar icons
Affordance: Aliens
Affordance

• Definition
  – Perceived and actual properties of a thing that determine how the thing could be used

• Perceived vs. actual
  – E.g., It looks clickable and is actually clickable
Affordance: Door

How about visibility?
Affordance: iPod
Affordance: camera
Exercise

- How can we make something perceived as clikable?

- perceived as draggable?

This sentence is a link.
Affordance: bad examples

perceived attribute != actual attribute

This sentence is not a link.
Feedback: Remote control
Feedback

• Explore, trial and error
• Actions should have immediate and visible effects
• Feedback allows users to make causal associations between actions and outcome
• If possible, provide preview of an action’s outcome before the action is committed.
Feedback: Nuclear button

Providing feedback is not always possible, especially for irrecoverable and catastrophic actions.
Natural mapping: Stove burners

How about visibility, affordance, feedback?
Natural mapping

• Physical arrangement of controls should match arrangement of functions

• Best mapping is direct, but natural mapping don’t have to be direct.
  – Stove burners, car turn signals
Mental model: Shower controls

How about visibility, affordance, feedback, mapping?
Mental model

- Logical learners
- What users’ think how the system works
  - Parts and how they work together
  - Use model to reason
High-level to low-level models

• User’s mental model
  – what the user actually think

• Interface model
  – what we want users to think

• System model
  – how the system actually works

• Ideally: Interface model == User model
Example: Mechanical Turk

Interface model

System model
Example: changing channels using a remote

• System model
  – Change frequency

• Interface model
  – Adding the channel number by 1?
  – Scrolling down a list?

• Mental model
  – Adding the channel number by 1?
  – Scrolling down a list?
Forming mental models

• How do people form a mental model?
  – Based on what’s visible
  – Based on affordance
  – Based on causal associations established through exploration and feedback
What is your mental model of

• Remote control
• Optical mouse
• Sleep vs. hibernation mode
• Remote desktop
Reasoning using system models

• Advanced users

• Software
  – Text editor: a single string, or an array of strings
    • Explains why some strings are wrapped when copy and paste across different editors
  – Pixels vs. vector graphs:
    • Affects how to copy and paste

• Hardware
  • Solid-state drive?
Designing interface models

• Simple
  – Hide the complexity of system models from users

• Appropriate
  – Allow users to reason effectively for their tasks

• Well-communicated
  – Allow users to form the intended mental models
Wrong mental models

• Usually,
  – Wrong model leads to wrong reasoning
  – Wrong reasoning leads to wrong outcome
  – E.g., Thermostat as a valve

• But sometimes simpler but inaccurate mental models can simplify reasoning
  – E.g., Electricity as water
Metaphor: Desktop
Metaphors

• Explain unfamiliar things in terms of familiar things
• Hook into user’s existing mental model
• Use frequently in education
• Allow users to infer other functionality
Metaphors: Examples

- Unix pipe
- Java, Java Bean, Jar,
- Desktop
- Window
- Slide
- Email
- Tablet
- Tabletop
- Avatar
Metaphors: Tips

• Don’t stretch
• Don’t be too literal (bend rules when necessary)
  – E.g., recycle bins are “on” the desktop in GUI.
• Don’t be constrained by the metaphor
  – E.g., folders can be resized
• Don’t translate existing bad designs
• Be aware of users who stretch their imagination too far
  – E.g., sitting on the desktop
Consistency

• Principle of least surprise
  – Similar things look and act similarly
  – Different things look and act differently

• Kinds of consistency
  – Internal
  – External
  – Metaphorical
Consistency
Speak the user’s language

• Use common words
• Avoid technical jargons
  – E.g., “unhandled exception error”
• Whether or not domain-specific terms are appropriate depend on the target users
  – E.g., medical terms for doctors
Follow platform standards

• Goal: improve external consistency
• Follow each platform’s standard
  – Apple Human Interface Guidelines
  – Windows Vista User Experience Guidelines
  – GNOME Human Interface Guidelines
  – KDE User Interface Guidelines
  – Java Look and Feel Design Guidelines
• Draw “inspiration” from popular programs
Help and documentation

• Users don’t like to read manuals
  – They prefer learning while making progress toward their goals
• Should not be an excuse for poor design
• Task-oriented vs. function-oriented
Types of help

• Tutorials
• Getting started manuals
• Reference manuals
• “Show me” videos
• Wizards
• Tooltips
• Context-sensitive help
• Stencils-based tutorials
Transition to experts

• User perform “cost/benefit” to decide whether to try a new functionality
  – Is it worth spending time learning a new trick?

• Make actions reversible
  – Will I mess something up?

• Increase user’s motivation to transition to efficient behaviors
  – computer games do this well
Transition to experts: Techniques

- Training wheel
- Tip of the day
- Success stories
- Cookbook
- Intelligent coach
Evaluation
Formative vs Summative: Objectives

• Formative learnability evaluation
  – Expose learnability issues

• Summative learnability evaluation
  – Assess overall learnability
  – Check if requirements are met
Formative vs Summative: Techniques

• Formative learnability evaluation
  – Think-aloud (initial learning)
  – Diary keeping (extended learning)
  – Coaching (initial learning, extended learning)

• Summative learnability evaluation
  – Think-aloud + quantitative analysis
  – Train, test, and score
  – Train, test, and survey
Metrics

• Task
• Command
• Mental
• Subjective
• Documentation
• Other
Task

• Percentage of users who
  – complete a task without any help
  – complete a task optimally
  – complete a task within a time frame

• Time it takes for users to
  – complete a certain task successfully
  – complete a set of tasks

• Improvement made over certain time interval
Command metric

- Success rate of commands after being trained
- Increase in commands used over certain time interval
- Increase in complexity of commands over time interval
- Percentage of commands known to user
- Percentage of commands used by user
Mental metrics

- Decrease in average think time over certain time interval
- EEG patterns
Subject metrics

• Number of learnability related user comments
• Learnability Likert statements
Documentation metric

• Decrease in help commands used over certain time interval
• Time taken to review documentation until starting a task
• Time to complete a task after reviewing documentation
Other

• Comparing usability for novice and expert users
• Number of rules required to describe the system
Summary

• Learnability design principles
• Learnability evaluation techniques
Activity: Learnability

• Brainstorm ideas for improving the learnability of the UI your team is designing