Interaction

The purpose of interaction is to accomplish a goal within an application domain.

- Domain – area of expertise
- Goal – desired output
- Tasks – operations to be performed to manipulate the domain
- Task Analysis – identification of the problem space for the user in terms of domain, goals, intentions, and tasks.

When dealing with how a user interacts with a computer, these are some of the commonly used terms:

- System – refers to the computer-based system
- Core Language – language of the computer-based system
- User – refers to the individual using the system
- Task Language – the language of the user

Interaction in HCI take place between the user and the system. Interface must translate their communication between them. Translation can fail at several points.

Assessing Interaction

- Frameworks are meant to be means of judging overall usability of an interactive system.
- All analysis is dependent on the current task.
- It is only in attempting to perform a task that we can determine if our tools are adequate.

Controller Display Relationship

- Displays provide operators with information about the operational status of a system.
- Control devices allow the operators to take action to influence the state of the systems.
- Sometimes called "Mappings."
- Relationship between what a user does and how the system responds.
- Should be natural, seamless, efficient and intuitive.

Types of Mappings

- These are the four types of mappings:
- Spatial
- Gain and Transfer functions
- Latency
- Property Sensing

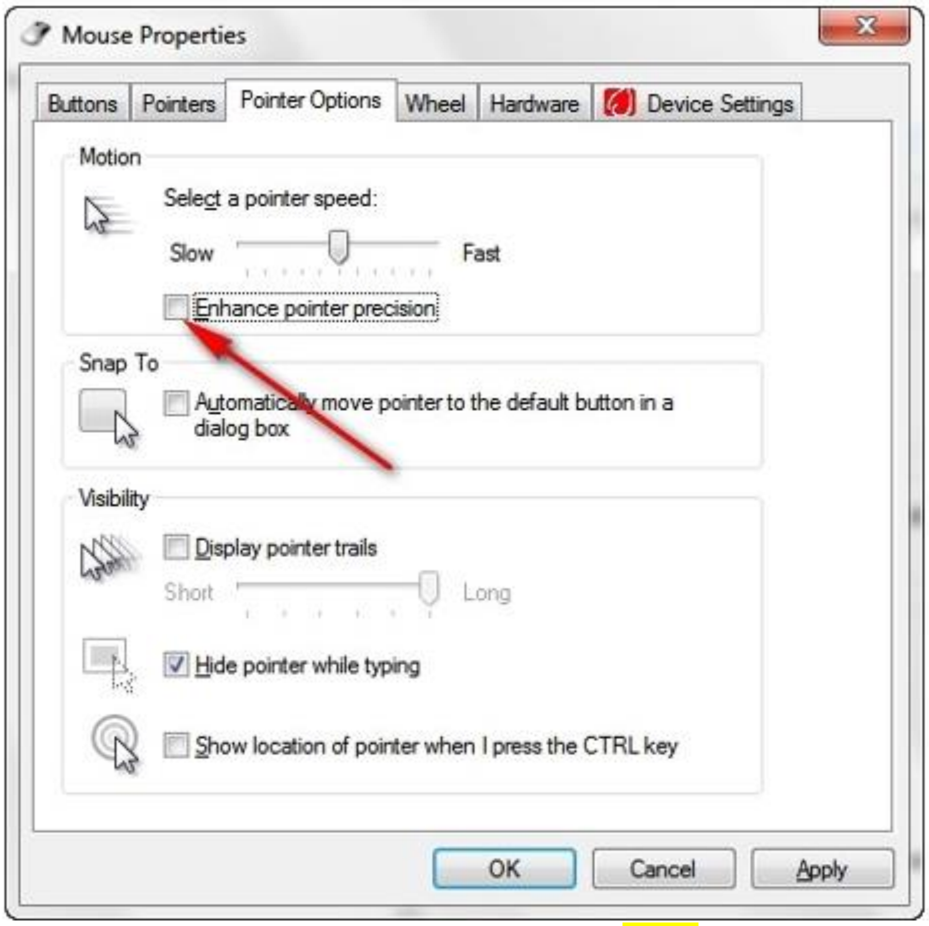
Spatial Mapping

- Spatial memory is the ability to learn the location of objects of interest by interacting with them repeatedly.
- In using a computer, spatial memory allows users to develop a level of "automaticity" when accessing data.
- In other words, this means that the user should be able to make use of their cognitive processes to quickly understand and use a device.
- Example: A mouse is used to move a pointer. Moving the mouse right should move the pointer to the right.

- Spatial mapping allows a user to make use of a mouse and other controller devices to control their computer.
- Spatial mapping is even more important with the rising popularity of VR and AR applications.

Gain and Transfer Mapping

- This refers to the ratio between movement of the controller and movement of the display.
- There is a nonlinear relationship between how you use your controller, and how the display reacts to your movement.



Mouse Properties

Buttons | Pointers | Pointer Options | Wheel | Hardware | Device Settings

Motion

Select a pointer speed:

Slow — Fast

☐ Enhance pointer precision

Snap To

☐ Automatically move pointer to the default button in a dialog box

Visibility

☐ Display pointer trails

Short — Long

☑ Hide pointer while typing

☐ Show location of pointer when I press the CTRL key

OK | Cancel | Apply

- This allows you to adjust different settings for how much force, movement or keys need to be pressed for an action to occur.
- This also allows you to adjust the sensitivity of your device's controllers.

Latency

- This is the delay between input action and the corresponding display or output.
- This is measured in milliseconds.
- This is normally associated with internet connectivity but also occurs with input devices such as keyboards, mice and wireless headphones.

ULTRA-LOW LATENCY CONNECTION

The wireless earbuds have an extremely low 60ms input latency, which means its audio stays synced with your device and will never stutter, providing a competitive gaming advantage and a more immersive experience for videos and music.

- Latency is usually caused by your network hardware, your server's location and the internet routers located between your server and your devices. Some types of internet connections, such as satellite internet connections have high latency in almost every condition. Servers that are located far away from your network can also increase your chances of latency.

Property Sensing

- Property sensing is sensing the force given by the user and transforming that into input for a device.
- This differs from "Gain and Transfer Mapping" as that is adjusting how your devices interpret your inputs.
- Property sensing is focused more on how the users adjusts their own input to get different results from their device.

- This includes adding new sensors to existing devices to detect human actions – touching, tapping, grasping, moving, pushing, flicking, and squeezing.

Mobile Technologies

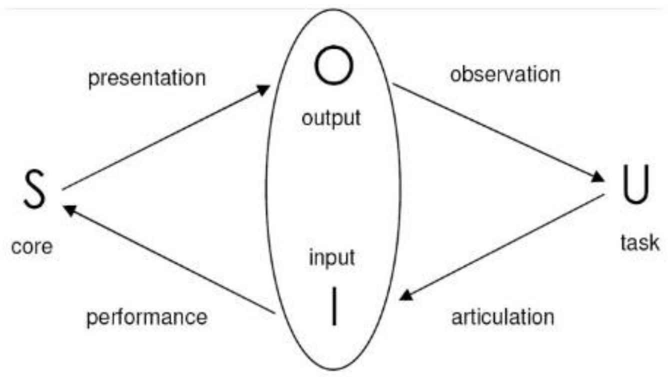
- Mobile Technologies have introduced many different forms of interactions that were not previously available for desktops.
- This has led to new applications and features being created to take advantage of these interactions.

- Some interaction technologies now commonly found on mobile devices are touchscreens and multitouch, accelerometers, fingerprint scanners, face scanners, and even gyroscopes.

Abowd and Beales Framework v. Norman Execution Evaluation Cycle

Abowd and Beale's interaction framework

- Identify system and user components which communicate via the input and output components
- The user formulates a goal and a task to achieve that goal.

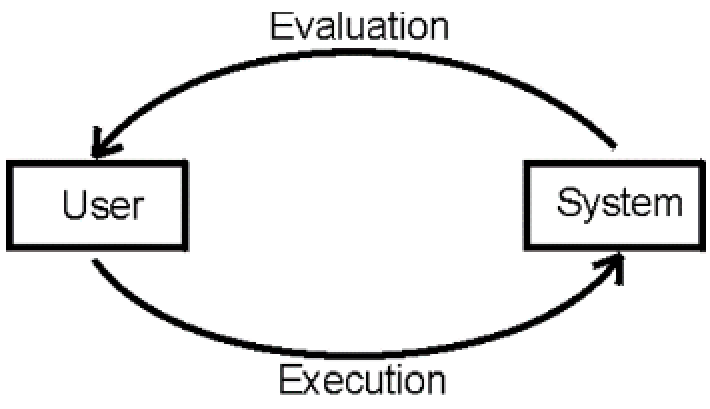


- The task must be articulated within the input language.
- Input language from the user is translated to the system's core language as operations to be performed.
- The system transforms itself into a new state based on the operations that are done.
- The system renders the new state and presents it to the user.

Advantages of using the framework

- It is not restricted to electronic computer systems.
- Identifies all major components involved in interaction.
- Allows comparative assessment of systems.
- Provides an abstraction of interaction.

Norman's Execution-Evaluation Cycle



Norman's Interaction Cycle

- This is the model that is composed of two parts execution and evaluation.
- In this theory, human-computer interaction is divided into seven steps:
    - User establishes a goal
    - User formulates their intention
    - User decides on action on the interface
    - User executes the action
    - User perceives the system state
    - User interprets system response
    - User evaluates system with respect to the goal

Advantages of Norman Execution Evaluation Cycle

- This is a useful means of understanding the interaction clearly and intuitively.
- Allows more detailed work to be placed within a common framework.
- Considers the user's view.

Scenario: You are reading at night. You decide that you need more light.
Step 1: Establishing the Goal
To get more light for reading.
Step 2: Forming the Intention
To switch on the desk light.
Step 3: Specify the Actions Required
Look for the lamp.
Reach over to the lamp.
Press the light switch.
Step 4: Execute the Action
Carry out the actions you specified in step 3.
Step 5: Perceive the System State
Look at the lamp.
Look around the room.
Is the light on?
Step 6: Interpret the System State
Is the light on?
Is the light off?
If it is off, what could be wrong?
Step 7: Evaluate the system in context of the original goal
Have you achieved your goal?

Introduction to Design Principles

Context in HCI

- Also known as paradigm
- Focuses on how the HCI will be formed

- Looks into effective strategies for building interactive, usable systems
    - User experience
    - User interface
    - Graphic design elements and principles
    - Design principles

Takes into consideration several items:

- Other components of HCI
    - Human perceptual system, emotions, processing, and memory
    - Computer input and output devices
    - Interaction models
- Organizational setting
    - How will it be used?
    - Who will use it?
    - Who will design it?

User Interface / User Experience

- Both design ideas and processes assist in the development of the design of the system:
    - How it answers the problems presented
    - How it looks and feels
    - Flow of the events
    - Ease of use
    - Overall experience with the product / service
- These two terms are used interchangeably, and are used in tandem with one another; but are different

User Interface (UI)

- Focuses on the look, feel, presentation, and interactivity of the system
    - Graphic design elements and principles
    - Layouting of the parts of the system dependent on the hardware
    - Use of highly recognizable icons and designs
- Set of processes and principles that are done exclusively digitally
    - Has a lot more tangible processes to achiever good UI
    - Tends to be more technical than UX
- Goal: Allow for intuitive use of the system
- UI is closely tied with graphic design (GD)
    - **Graphic design** – art, science, and profession of presenting information using visual elements
- Utilizes elements and principles of graphic design
- **Some Elements of GD**
    - Color and texture
    - Line, shape, and form
    - Space and size
    - Typography
- **Some Principles of GD**
    - Balance
    - Proximity and alignment
    - Contrast
    - Repetition and movement
    - White space

User Experience (UX)

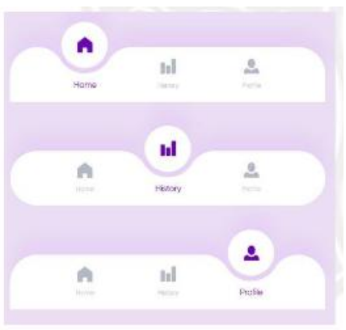
- Focuses on how the user interacts with the system
    - How easy it is to go to a specific page
    - Ease of navigating between pages
    - Using the system without the assistance of anyone else
- Can be used for both digital and non-digital products
    - Tools – How easy it is to hold it? Using one hand? Using a non-dominant hand?
- Goal: Allow for a pleasant experience on using the system
    - Mostly revolves around ideas and concepts

User Interface / User Experience

- UI is a part of UX
- UI manifests the ideas and principles found in UX

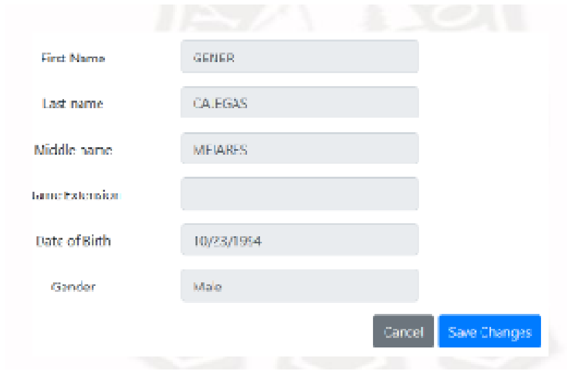
    - It is technically possible to create something without UI, but not without UX, though this is not recommended
    - UX is most effective with the implementation of UI
- UI and UX possesses different skills
    - **UI – graphic design**; presenting information aesthetically
    - **UX – system analysis**; identification of features and constraints; quality assurance

Example: Bottom Navigation



- **User Experience**
    - Identify most used actions
    - Make actions easily accessible
    - Actions can't be accidentally pressed
- **User Interface**
    - Use icons that best represents actions, with additional text
    - Small sizing, but can be tapped

Example: Common Form



First Name	GENER
Last name	CA.EGAS
Middle name	MEIARES
Name Extension	
Date of Birth	10/23/1954
Gender	Male

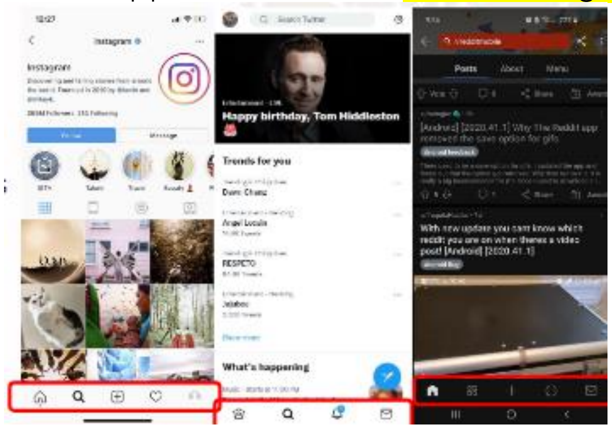
- **User Experience**
    - Textbox is one line to indicate short answers (vs. text area)
    - Sample answers are provided
    - Steps are arranged from top to bottom
- **User Interface**
    - Equal width of textbox
    - Label at the left side of textbox
    - Blue button to indicate "Save"

Overview of Design Principles

- Widely applicable fundamental design considerations for systems and products
    - Applied under discretion (applicability of principle, conflict of principle to other principles, etc.)
    - Can be added or modified on depending on current trends
- Three main principles:
    - Learnability
    - Flexibility
    - Robustness
- Each principle contains other principles under it

Learnability

- Ease where new users can begin effective interaction and achieve maximal performance
- The better the learnability of an application, the less training and time it takes for a person to use it
- Vital that users can pick up how to use the application quickly
- Common designs are the ones familiar with users
    - A lot of web, desktop / laptop, and mobile apps share design methods
    - Focuses on recognition instead of recall
    - Examples: Most mobile applications have a bottom navigation

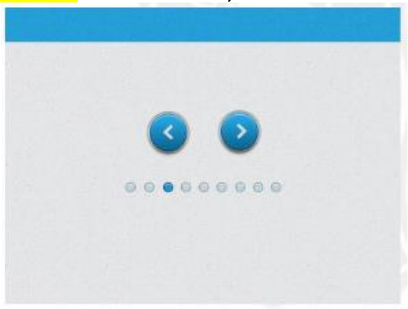


- Factors That Affect Learnability
    - Predictability
    - Synthesizability
    - Familiarity

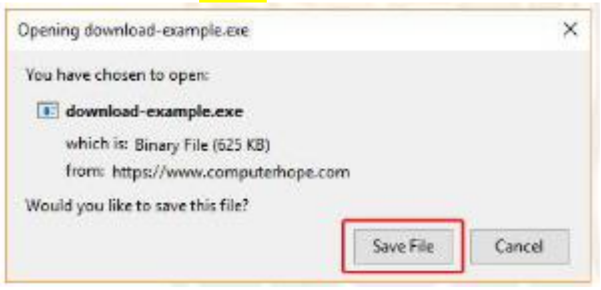
    - Generalizability
    - Consistency

Predictability

- Focuses on the user's ability to foresee operations that can be performed next
    - User must be able to determine the outcome of the present or future interaction with the system
    - Requires a user's knowledge of interaction to be sufficient
- Often referred to as operation visibility
    - Concerned with showing the availability of operations
    - If an operation can be performed, then there should be a clear indication of this to the user
    - User actions should be matched by a response
- Example:
    - Navigation buttons:
        - Items (slides / images / videos) would move from left to right and vice versa
        - Indicates the number of items by the number of dots



    - When saving a file from a web browser, the prompt comes up
        - Shows file name, type of file, and source
        - Shows options to either save the file or cancel



Opening download-example.exe

You have chosen to open:

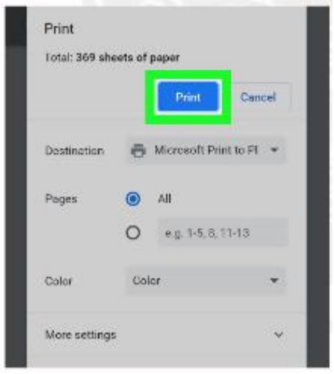
download-example.exe

which is: Binary File (625 KB)

from: https://www.computerhope.com

Would you like to save this file?

    - Blue buttons
        - Used for Google apps (mobile app frameworks, Google Chrome)
        - Indicates that the button is the next step
        - Commonly done step, but not all of the time



Print

Total: 369 sheets of paper

Destination: Microsoft Print to PI

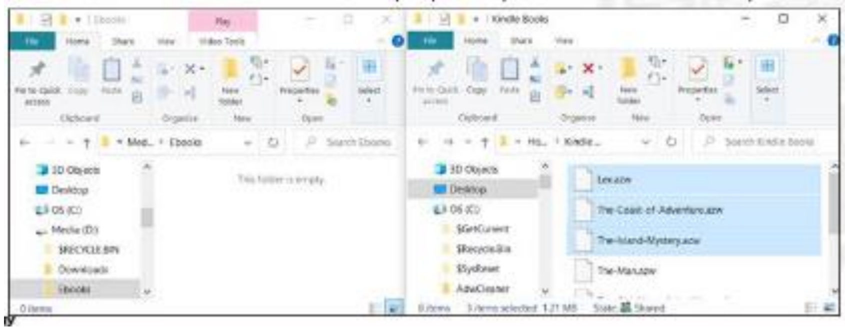
Pages: All / e.g. 1-5, 8, 11-13

Color: Color

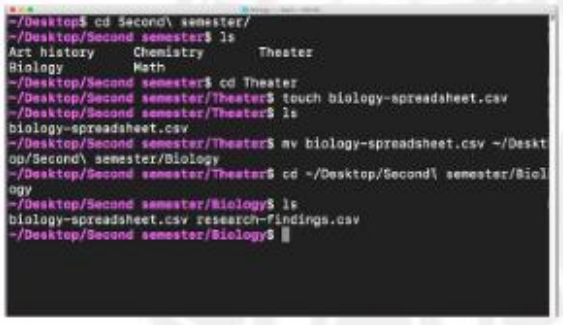
Synthesizability

- Provides support for the user to assess the effect of past operations on the current state
- Two aspects of synthesizability:
    - Immediate honesty
    - Eventual honesty
- Aspects of Synthesizability
    - **Immediate Honesty**
        - Provides user an observable and informative notification about the changes in the system
        - User will be able to immediately tell that their actions have caused observable changes
        - Example: Transferring files from one folder to another

            - ❖ Highlighted items on the later folder are the ones pasted
            - ❖ Notification "This folder is empty" is present on the previous folder



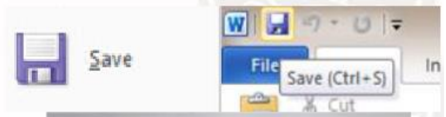
    - **Eventual Honesty**
        - User tries to discover the changes that they made in the system on their own
        - A user not familiar with the system's operations may have trouble synthesizing the consequences of the operations
        - Example: Using Command Prompt
            - ❖ Requires specific knowledge on the different command
            - ❖ No feedback (notification, text, sound, etc.) that would indicate success or failure
            - ❖ User must double check if the action was successfully done



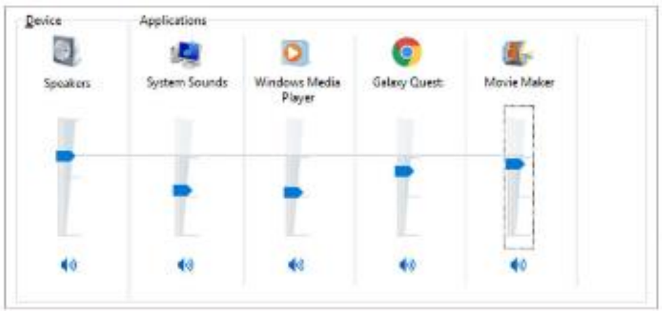
```
~/Desktop$ cd Second\ semester/
~/Desktop/Second semester$ ls
Art history     Chemistry       Theater
Biology         Math
~/Desktop/Second semester$ cd Theater
~/Desktop/Second semester/Theater$ touch biology-spreadsheet.csv
~/Desktop/Second semester/Theater$ ls
biology-spreadsheet.csv
~/Desktop/Second semester/Theater$ mv biology-spreadsheet.csv ~/Deskt
op/Second\ semester/Biology
~/Desktop/Second semester/Theater$ cd ~/Desktop/Second\ semester/Biol
ogy
~/Desktop/Second semester/Biology$ ls
biology-spreadsheet.csv research-findings.csv
~/Desktop/Second semester/Biology$ 
```

Familiarity

- Measures how prior knowledge and experience of the user can be applied to a new system
- Use of appearance can improve the familiarity
- Appearance of an object provides familiarity with its behavior
- Concepts in Familiarity
    - **Guessability**
        - Matching user's expectations with how objects in the real-world functions
    - **Metaphor**
        - Provide another avenue for the user to better understand the system
    - **Affordance**
        - Interaction design principle where the appearance of objects in a system suggests how they can be manipulated
- Example: Save icon
    - Most save icons are based on the floppy disk or diskette – initially used for storage

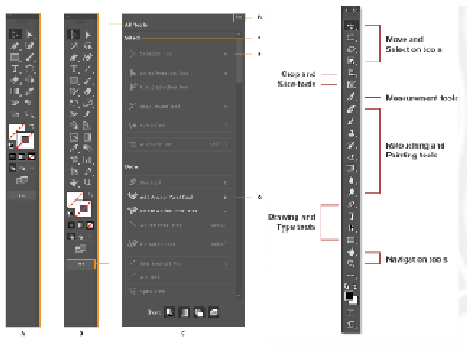


- Example: Soundboard
    - Volume sliders in soundboards are used digitally

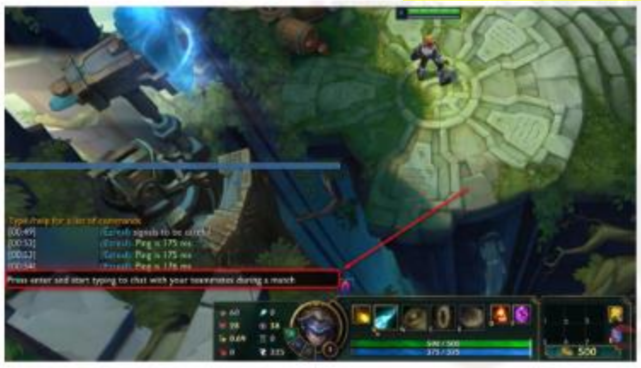


Generalizability

- Form of consistency among different applications
- Extending specific interaction knowledge to new situations
    - Determine how easy is it to learn to perform new tasks, given a user's current experience of a system or a UI
    - UI standards and guidelines assist / enforce generalizability
- Allows users of one application to easily adopt to another similar application
- Example: Tools in Adobe software
    - Adobe Photoshop (raster / pixel-based) and Adobe Illustrator (vector) have different uses
    - Have similar tools so that users of both software can use them interchangeably

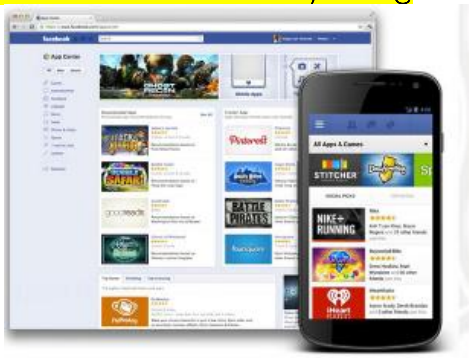


- Example: Enabling chat
    - Most MMORPGs would have the user press "Enter" to enable the chat

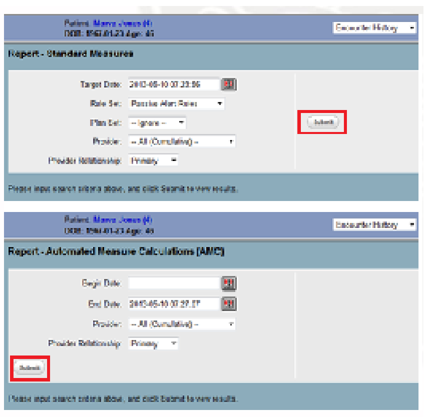

Type /help for a list of commands
Press enter and start typing to chat with your teammates during a match

Consistency

- Likeness in input / output behavior arising from similar situations or task objectives
    - "Sameness" should be applied to the use of terminology, formatting, and input / output behavior
    - Same icons and labels should mean the same thing
- Used to support generaliability
- One of the most widely applied design principles in user interface design
- Example: Layouting for applications
    - The navigation in Facebook for both mobile and desktop applications are both at the top
    - Uses the same color scheme and similar layouting



- Example: Placement of elements
    - The example (right) has the "Submit" button on different places
    - Can cause confusion to the user even if it's on the same application


Patient: Marva Jones (4)
DOB: 1961-01-23 Age: 45

Encounter History

Report - Standard Measures

Target Date: 2012-05-10 07:23:56
Rule Set: Possible Alert Rules
Plan Set: -- Ignore --
Provider: -- All (Cumulative) --
Provider Relationship: Primary

Submit

Please input search criteria above, and click Submit to view results.

Patient: Marva Jones (4)
DOB: 1961-01-23 Age: 45

Encounter History

Report - Automated Measure Calculations (AMC)

Begin Date:
End Date: 2012-05-10 07:27:17
Provider: -- All (Cumulative) --
Provider Relationship: Primary

Submit

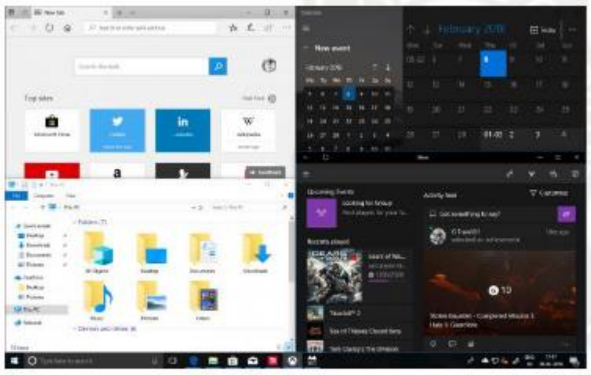
Please input search criteria above, and click Submit to view results.

Flexibility

- Extends the way a user and the system exchanges information
- Allows a system to adapt to different types of users and different uses of a system
- Help improve a system's usability
- Factors That Affect Flexibility:
    - Multithreading
    - Task Migratability
    - Task Substitutivity
    - Customizability

Multithreading

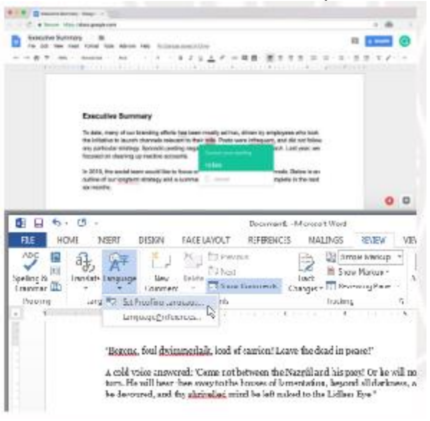
- Ability of a system to support user interaction for more than one task at a time
- Within a user interface, a thread can be considered a specific task is being performed
- Can be described as concurrent or interleaved
- Types of Multithreading
    - **Interleaved System**
        - Permits work on a single task at a given time
        - Example: A word processor allows multiple documents to be open, but only one can be worked on at any instant
        - Example: Multiple windows
            - Multiple windows can be open at the same time
            - Cannot actively access them at the same time



    - **Concurrent System**
        - Allows multiple tasks to be actioned at a given time
        - Example: In most operating systems, a document can be edited in MS Word, while music is playing in the background and a file is being downloaded
        - Example: Discord
            - VoIP best known for gaming (channels, calls, etc.)
            - Can use the VoIP option to talk to other users while playing games

Task Migratability

- Passing responsibility of execution of tasks between user and system
- Provides a system the ability to help the user without directly making changes to a file or document
- Example: Spell Checker
    - Checks words against its own dictionary
    - Not recommended to allow a spell-checker program to carry out this task without the user's assistance
    - Gives suggestions, but does not automatically correct spelling mistakes

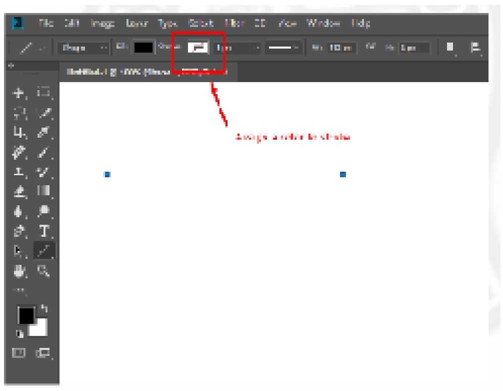


'Begone, foul dwimmerlaik, lord of carrion! Leave the dead in peace!'

A cold voice answered: 'Come not between the Nazgûl and his prey! Or he will not slay thee in thy turn. He will bear thee away to the houses of lamentation, beyond all darkness, where thy flesh shall be devoured, and thy shrivelled mind be left naked to the Lidless Eye.'

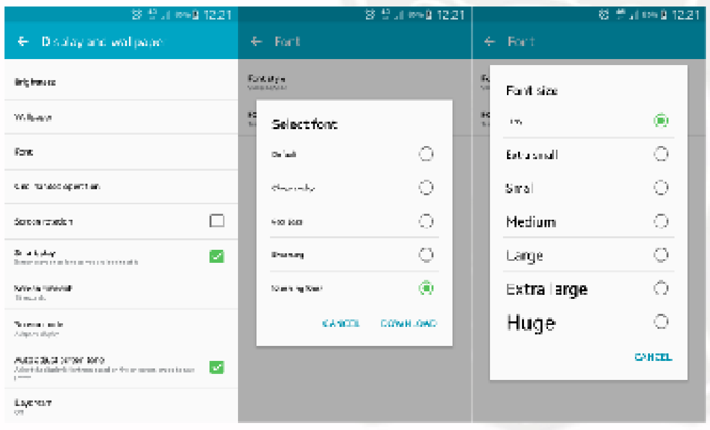
Task Substitutivity

- Allows equivalent values of input and output to be substituted for each other
- Can blur the distinction between output and input
- Example: Drawing a line
    - Drawing application may allow start and end co- ordinates of a line to be specified
    - Can also allow the line to be drawn first, and the system indicates the end point co-ordinates



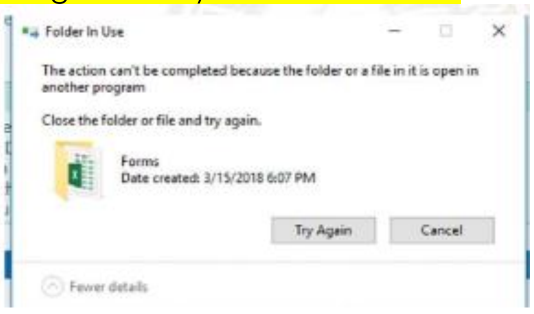
Customizability

- Modifiability by the user (adaptability) or the system (adaptivity)
- User can modify the user interface
- UI should be able to support individual preferences
- Some ways of customizing UI:
    - Themes and colors
    - Notifications
    - Account information
- Example: Phone settings
    - Can change the display settings such as wallpaper, brightness, font style, and font size



Robustness

- Level of support provided to the user
- Determines successful achievement and assessment of goal-directed behavior
- Focused on supporting users so that they can achieve their goals efficiently
    - Allows the user to navigate the system on their own



Folder In Use

The action can't be completed because the folder or a file in it is open in another program

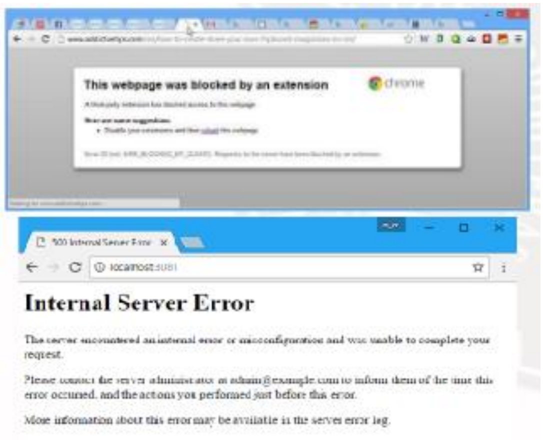
Close the folder or file and try again.

Forms
Date created: 3/15/2018 6:07 PM

- Factors Affecting Robustness:
    - Observability
    - Recoverability
    - Responsiveness

Observability

- User should be able to evaluate and understand the internal state of the system from its perceivable representation
    - If the user cannot understand the internal state of the system, then the user's confidence may be very low
- Common example: Time-consuming operation
    - Current status of operation should be displayed
    - This is done by a text, graphic element, or both
- Example: Webpage errors would describe why the webpage did not load
    - 3xx – client must take additional action to complete the quest
    - 4xx – client errors
    - 5xx – server errors



This webpage was blocked by an extension

[illegible]

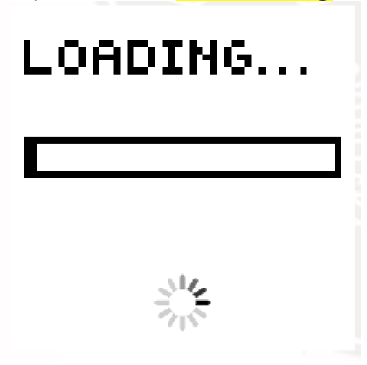
Internal Server Error

The server encountered an internal error or misconfiguration and was unable to complete your request.

Please contact the server administrator at admin@example.com to inform them of the time this error occurred, and the actions you performed just before this error.

More information about this error may be available in the server error log.

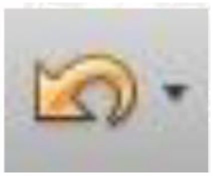
- Example: UI elements for loading
    - Reassures the user that the process is still working and has not crashed or failed



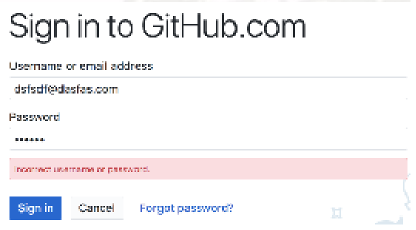
Recoverability

- User should be able to reach a desired goal after recognition of errors in previous interaction
- **Error Recovery Ways:**
    - **Forward (Negotiation) –** User accepts the current state of system and negotiating from the present state towards the required state
    - **Backward (Undo) –** User removes the undesired outcome of the previous interaction by returning to a previous state

- Example: The undo button in most processor documents is an example of a backward error recovery



- Example: An example of forward error recovery
    - The username is retained when the password is rejected



Sign in to GitHub.com

Username or email address

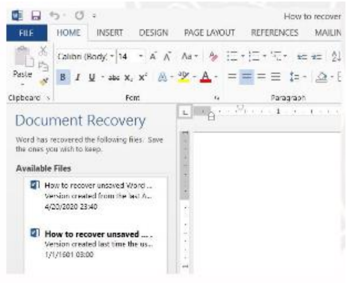
dsfsdf@dasfas.com

Password

••••••

Incorrect username or password.

- Example: An example of backwards error recovery
    - Microsoft Word has a document recovery feature
    - Recovers documents that have been autosaved



Document Recovery

Word has recovered the following files. Save the ones you wish to keep.

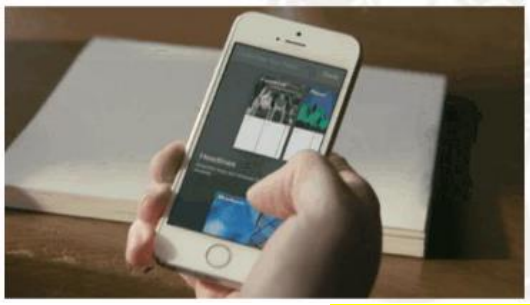
Available Files

How to recover unsaved Word ...
Version created from the last A...
4/20/2020 23:40

How to recover unsaved ....
Version created last time the us...
1/1/1601 03:00

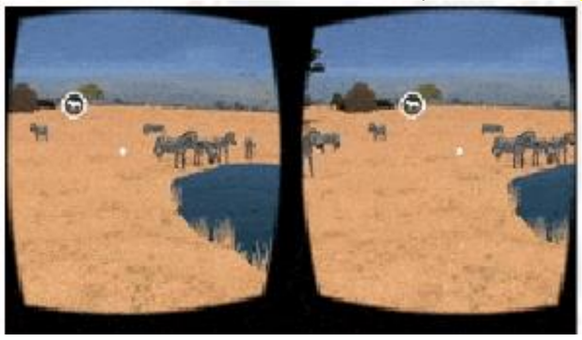
Responsiveness

- Measure of the rate of communication between the user and the system
    - Indicating changes of states within the system is important
    - Short duration of instantaneous response time is more desirable
- Example: When doing touch functions, it should do the necessary animations
    - Done mostly for mobile, but also for touchpads, mouses, etc.



- Example: In first person shooter (FPS) games, the precision and accuracy for pointing, shooting, and navigating is important

- Example: In virtual reality, when the user moves their head, then the output device should move too
    - The precision of the LCD screen in VR is also important for responsiveness



Usability Testing

Introduction to Usability Testing

- **Testing –** Set of procedures to ensure that the system is functional based on certain requirements
- Testing in HCI mostly looks into whether or not the design of the system is easy to use and intuitive to the target user
- **Software Testing –** Focuses on the functionality of the features, operations, and constraints of a system

Usually done through usability testing

Testing in HCI	**Software Testing**
Tests how the user interacts with the system	Tests how the system works in the computer
Done through asking the user questions regarding experience	Done (mostly) through running the system in controlled, programmed tests
Requires surveys/guide question	Requires high-level rechnology

Objectives of Testing in HCI

- Testers can complete tasks successfully and independently
- Asses testers' performance and mental state while completing tasks
- See how much users enjoy using it
- Identify user-centric problems and their severity
- Find solutions to identified problems

Considerations on Testing in HCI

- Objectives of testing
- Role of end-user
- Target performance goals
- Evaluation time constraints
- Hardware device used

What is Usability?

- Component of user experience design
- Measure of how well a specific user in a specific context can use a product/design to achieve a defined goal
- Goal must be achieved effectively, efficiently and satisfactorily
- Design usability is measured throughout the development process

Elements of Usability

- **Effectiveness / Useful**
  - Supports users in completing actions accurately
- **Efficiency / Usable**
  - Users can perform tasks quickly through the easiest process
- **Engagement / Used**
  - Users find it pleasant to use and appropriate for its topic
- **Error Tolerance**
  - Supports a range of user actions
  - Only shows an error in genuine erroneous situations
  - Done by finding out the number, type and severity of common errors users make
- **Ease of Learning / Learnability**
  - New users can accomplish goals easily and even more easily on future visits

What is Usability Testing?

- Practice of testing how easy a design is to use with a group of representative users
- Involves repeated observation of users completing tasks through different types of designs

Objectives of Usability Testing

- Determine whether testers can complete tasks successfully and independently
- Assess user performance and mental state while completing tasks
- See how much users enjoy using it
- Identify problems and their severity
- Find solutions

Steps in Usability Testing

- Planning
- Define what is needed to be tested
- Decide how to conduct the test
- Set user tasks
- Prioritize most important tasks
- Clearly define tasks
- Create scenarios where the design can be used naturally
- Recruit testers
- Facilitate / moderate testing

Usability Testing Methods

- **In-Person**
    - Formal, live testing of users
    - Requires empathetic moderator to note testers' experiences
- **Remote**
    - Users using the system on their own environments
    - Allows for more accurate field insights
- **Guerilla**
    - Informal testing on passers-by / colleagues
    - Faster process, but higher possibility of inaccurate data

System Usability Scale (SUS)

- Survey tool to measure the usability of a system
- Can be used for a wide array of systems
- Ten-questions and five responses
- **Benefits:**
    - Very easy scale to administer to participants
    - Can be used on small sample sizes with reliable results
    - Effectively differentiates between usable and unusable systems
- **Considerations:**
    - Scoring system is somewhat complex
    - Interpreted as numbers, and not percentages
    - The best way to interpret your results involves "normalizing" the scores to produce a percentile ranking
    - SUS is not diagnostic

Participants must score the following 10 items with one of five responses that range from Strongly Agree to Strongly disagree:

- I think that I would like to use this system frequently.
- I found the system unnecessarily complex.
- I thought the system was easy to use.
- I think that I would need the support of a technical person to be able to use this system.
- I found the various functions in this system were well integrated.
- I thought there was too much inconsistency in this system.
- I would imagine that most people would learn to use this system very quickly.
- I found the system very cumbersome to use.
- I felt very confident using the system.
- I needed to learn a lot of things before I could get going with this system.

Interpreting Scores

- **Data Treatment Steps:**
    - Participant's scores for each question are converted to a new number
    - Numbers are added together
    - Sum is multiplied by 2.5 to convert the original scores of 0-40 to 0-100
- **Interpretations:**
    - SUS > 68 == Above average
    - SUS < 68 == Below average
    - Best way to interpret your results involves "normalizing" the scores to produce a percentile ranking

Tips During Testing

- Allow users to speak their mind out loud
- Let users struggle; don't over moderate

- Use pauses and silence if needed
- Allow user to take breaks or take time
- Be responsive to the user

INTRODUCTION TO METRICS

What is Metrics?

- General term for the measurement/criteria during testing
- Used to determine whether or not something is considered successful or not

Types of Metrics

- Depending on what needs to be tested, the metrics change as well:
  1. **Usability Metrics**
     - **What is Usability?**
       - Measure of how well a specific user in a specific context can use a product/design to achieve a goal
       - Goal Must be achieved effectively, efficiently and satisfactory
       - Design usability is measure throughout the development process
     - **What is Usability Metrics?**
       - How easy a design is to use with a group of representative users
       - Involves repeatedly observation of users completing tasks through different types of designs.

  2. **Performance Metrics**
     - **What is Performance Metrics?**
       - Used to determine:
       - Magnitude of a specific usability issues
       - How many people are likely to encounter the same issues
     - **Basic Measure for Performance**
       - **Task Success**
         - Most common performance metric
         - Looks whether participants can complete a task or not
           - ✓ If the participant cannot complete the task, then something needs to be fixed
           - ✓ Requires a clear end-state
         - Ways to measure task success:
           - ✓ Participant verbally articulates the answer to the task
           - ✓ Participant provides answers in a more structured way
       - **Time on Task**
         - Also known as task completion time or task time
         - Time it takes to perform a task
           - ✓ Faster is not always better
           - ✓ Important when tasks are performed repeatedly
         - Ways to measure time on task:
           - ✓ Logging on start time and end time
           - ✓ Manual or automated timers
           - ✓ Rules for turning the timer on or off
       - **Errors**
         - Possible outcome of a performance issue
         - Incorrect action on the part of the user
         - Any action that prevents the user from completing a task efficiently
         - Ways to measure error:
           - ✓ Know what is the correct action / sequence of actions
           - ✓ Single or multiple opportunities for error
       - **Efficiency**
         - Amount of effort required to complete a task
           - ✓ **Cognitive –** deciding what action to task
           - ✓ **Physical –** activity required to complete the action
         - Steps to collect and measure efficiency:
           - ✓ Identify actions to be measured
           - ✓ Define start and end of action
           - ✓ Count the actions
           - ✓ Actions must be meaningful
           - ✓ Look only at successful tasks

- **Learnability**
    - How much time and effort are need to become proficient
    - Steps to collect and measure learnability:
        - ✓ Time-on-task
        - ✓ Errors
        - ✓ Task successes per unit time
        - ✓ Track these over time

3. **Issue-Based Metrics**
    - **What is Issue-Based Metrics?**
        - Identifying:
        - Understand what is and is not an issue
        - Know both the product and the usability questions that arise
        - Being very observant about participant behavior
    - **Possible Issues**
        - Preventing task completion
        - Takes the user "off-course"
        - Confusion / misinterpretation of content / navigation
        - Production of an error
        - Visibility of items that should not be noticed
        - Assumption of utilizing something correctly
        - Assumption of task completion
        - Conduct of the wrong action
    - **Identifying Issues**
        - **In-Person Studies**
            - Requires interaction with the users
            - Think-aloud protocols
            - Observations
        - **Automated Studies**
            - Allow participants to enter comments at the task level
            - Looks into success, time, ease of use ratings, and verbatim comments
    - **Biases in Metrics**
        - **Bias –** prejudice in favor or against a thing, person, or group compared to another
        - Bias when identifying metrics:
            - Systematic error in development and testing
            - Skews results of testing
            - Issues may not be fully resolved
            - Limits the types of users who can use the system
    - **Types of Biases**
        - **Participant**
            - Demographics
            - Role of the participant
            - Capabilities of the participant
            - Number of participants
            - Level of feedback from the participant
        - **Task**
            - Clearly defined v. open ended
            - Self-generated v. manually operated
        - **Method**
            - Input and output devices
            - Data gathering procedure
            - Algorithms used
            - Features and constraints focused
        - **Artifact**
            - Hardware specifications
            - Software specifications
            - System performance
        - **Environment**
            - Equipment in conjunction to the room
            - Size of the room
            - External forces
        - **Moderator**

            - Establishment of rapport
            - Comfort of the user

4. **Self-Reported Metrics**
    - **Why Self-Reported Metrics?**
        - Important information about user's perception of the system and their interaction with it
        - How users feel about the system
        - Tasks:
            - Collection of self-reported data
            - Analysis of self-reports
            - Post-task ratings
    - **Collection of Self-Reported Data**
        - **Collection Tools**
            - **Rating scale**
                - ✓ **Likert scale –** statement to which the response is the level of agreement
                - ✓ Presenting pairs of bipolar adjectives
            - **Open-ended questions**
                - ✓ Contains a set of premade questions
                - ✓ Helps guide the participant to deliver a relevant answer
                - ✓ Not meant to derive an answer the moderator wants to hear
        - **Collection Methods**

Method	Pros	Cons
Oral	Easiest method for quick ratings	Observer has to take notes
Written	Good for quick ratings and longer surveys	Need to be manually entered
Online	Good for quick ratings and longer surveys	Needs computer access

    - **Biases in Self-Reported Data**
        - **Selection bias**
            - Selection of participants with specific attributes for testing
            - Does not demonstrate a truly randomized sample
        - **Reporting bias**
            - Selective revealing or suppression of information
        - **Social desirability bias**
            - Tendency to provide more positive feedback in person or on the phone
        - **Avoiding Bias**
            - Person collecting should be different from the test monitor
                - ✓ Prevents observer-expectancy effect – collector of results can subconsciously influence the people participating in an experiment
            - Make the survey anonymous
                - ✓ Prevents prejudices and selection bias
                - ✓ Examples of prejudices: ageism, classism, racism, sexism, etc.
    - **Analyzing Self-Reports**
        - **Quantitative Data**
            - Assign a numeric value, compute for averages
            - Focus on top 2 and bottom 2 responses
        - **Qualitative Data**
            - Summarizing responses from open-ended questions
            - Identify keywords through apriori coding
            - Group keywords into themes
    - **Types of Post-Task Ratings**
        - Ease of use
        - After-scenario questionnaire
        - Expectation measure
        - Usability magnitude estimation

5. **Behavioral and Psychological Metrics**
    - **Observing and Coding Overt Behaviors**
        - **Verbal behaviors**

            - Usually expressed by the user during oral testing
            - Accompanied with non-verbal behaviors
        - **Non-verbal behaviors**
            - Usually described by affective states
            - Requires observation and specialized equipment
- **Affective States**
    - Boredom
    - Confusion
    - Delight
    - Frustration
    - Engaged concentration (flow)
    - On-task and off-task solitary
    - On task receiving and giving answers
- **Data Capture Equipment**
    - **Facial expressions**
        - Eye tracking
            - ✓ Proportion of users looking at specific element
            - ✓ Time to notice an element
        - Scan paths
        - Skin conductance
    - **Posture**
        - Pressure
        - Brain signal
        - Motion
- **Other Concepts**
    - Incidence of affective states
    - Persistence of affective states
    - Likelihood to games

Topic	Number of Items	Number of Points
Interaction frameworks	5	10
Assessing interactions	5	10
Design principles	20	40
Usability Testing	10	20
Metrics	10	20
Total	**50**	**100**