Quiz 4

Victoria Deng

September 2020

Input Mechanisms

Keyboards:

- The original typewriters had jamming issues from keys being too close to each other.
- The QWERTY keyboard was developed to prevent these jams. Although there are other alternatives as new devices develop, it became the standard as everyone was already used to the layout.

QWERTY: Used as it is considered the industry standard. There are however, faster alternatives, and QWERTY has issues with keeping the placement of non-alphanumeric characters consistent. Eg. Pound isn't needed in aus, but both pound and dollar would be needed in the UK. QWERTY also favours your left hand, whereas most people are right handed.

Dvorak: Places the most used letters in the alphabet in the home row (middle or easiest to reach row). The least common letters are placed on the bottom row (hardest to reach). Dvorak also favours your right hand.

Colemak: Similar to QWERTY but with only 17 differences. It's also designed so that the more frequently used keys are positioned in the home row and reduces finger movement.

Limited Space Keyboards: Includes things like nokia phones, where 3 letters are attached to a numeric button, or split keyboards.

Game Design (Keyboards):

- Consider the mental models already in place when defining a control scheme.
- Eg. using arrows or WASD for moving, M for map, space for jumping.
- Avoid using the right hand side of the keyboard if the player requires a mouse.

Mouse: Evolution from a ball in a box to using a laser pointer to control movement.

- Typically comes with a left and right button.
- Don't expect players to have more than two mice buttons.
- Can make considerations for player to rebind keys to a mouse button if they exist (but they might not)

Controllers: Also evolved to a more sleek design. Typically has two movement/camera joysticks, as well as basic button functions. For the purposes of generic game design, we can assume a player won't have this (unless we're designing for a specific console)

Eye-tracking:

- Control the interface by eye gaze direction eg.looking at an item menu to select it
- Uses a laser beam reflected off retina
- Potential for hands free control
- Cheaper and lower accuracy devices sit under the screen like a small webcam

Touch

- Speed: Faster in many operations when comapred to a traditional keyboard or mouse
- Easy to use: Intuitive, and requires less coordination
- Accessibility: Helps users with physical limitations when they interact with devices
- Device size: Saves space and allows more area to interact (eg. iphone compared to nokia phone)
- Touch interfaces are increasing becoming the norm for mobile devices. Normally touch interaction should be the primary expected input methods (for phones), and we can provide visual feedback, optimize touch target size etc.
- Unity also features an Input.GetTouch method, which returns a position, change in position, tap count, phase and fingerID.
- Through touch and the inputs given, we can calculate gestures such as pinch and zoom, or tapping, holding, dragging etc.

Fitt's Law: Tests the effectiveness of different input mechanisms in target acquisition tasks (eg. selecting a button) as well compare different types of users.

$$ID = log_2(\frac{2D}{W})$$

Where:
ID = index of difficulty (bits)
D = distance to the centre of the target
W = width of the target

$$MT = a + b * ID$$

Where:
MT = Movement time (seconds)
a, b = empirically determined constances, which are device dependent

$$TP = \frac{ID}{MT}$$

Where:
TP = Throughput (bits per second)

Infinite Edges The easiest positions to reach are the corners of a screen, as they have infinite bounds, and so don't require deceleration to reach it.

Sensors: May also exist in mobile games, like GPS or compass. Helpful for games like pokemon go.

Accelerometer: Measures acceleration along the three principle axis. Unity handles changes in orientation, but other game developing frameworks may not.

Gyroscope: Senses angular velocity produced by the sensor's own movement. Eg. shaking the phone.

Human Cognition

Design: Should prevent ambiguity and human error.

Attention: Interfaces should exploit the use of effective principles of design, such as sound, colour, font etc.

Cues: Should always let a player know what their next goal should be, or where they should go.

Progressive disclosure: Should avoid overwhelming a player with too much infomation at once, so we can hide parts that aren't immediately necessary.

Colour: Typically has some preconceptions/emotions associated with it

- Blue: Used in large areas and not thin lines, also a calming colour for the brain.
- Red, Green: Can be used in the centre of the field of view, as the edges of the retina are not sensitive to these colours.
- Black, White, Yellow: Used in peripheral vision
- In general avoid using too much colours simultaneously (unless you know what you're doing like fall guys)
- Keep a consistent colour scheme/aesthetic during the game.
- Avoid eye fatigue colours, eg. complimentary colours, or colours that don't naturally go together
- Also make considerations for colour blindness, the most common of which is red-green colour blind.

Perceptions: Different ways of presenting information, each which may be the same thing but emphasize different information.

Mental Models: Ensuring that it is consistent and easily understandble, for example the floppy disk icon is associated with saving data.

Sensory Memory: Lasts about 1-2 seconds, and is the immediate perception of stimuli in the environment. Can be dismissed or stored in short/long term memory.

Short Term Memory: Reduces or eliminates the need to memorize and recall things. Should not expect users to memorise commands or concepts are showing them something once.

Long Term Memory: Design to move important and frequently used commands and concepts to long term memory. If there is a command that is not frequently used, then we can consider refreshing a user's memory. We should also allow users to hid memory aids as they please.

Mnemonics: Used to improve a person's long term memory.

Learning: A good interface or game usuall allows a user to learn by doing. Avoid apps that require a manual beforehand.

Evaluation Techniques

Evaluation:

- Tests usability and functionality of a system
- Occurs in a laboratory, field or in collaboration with users
- Evaluates both design and implementation
- Should be considered at all stages in the design life cycle

Laboratory Studies:

- Advantages: Specialist equipment available, uninterrupted environment
- Disadvantages: Lack of context, difficult to observe several users cooperating
- Appropriate: If a system location is dangerous or impractical for constrained single user systems to allow controlled manipluation of use. I.e if the equipment is super new and requires a complicated set up.

Field Studies:

- Advantages: Natural environment, retains context, longitudinal studies possible
- Disadvantages: Distractions
- Appropriate: When we want to maintain context, and do studies over a long period of time.

Cognitive Walkthrough:

- Evaluates a design on how well it supports a user in learning tasks
- Usually performed by an expert in cognitive psychology
- Walk an expert through the design and they point out potential problems

Heuristic Evaluation:

1. Visibility of system status: should always keep users informed about what is going on
2. Match between system and real world: The system should use language familiar to the user, and follow real world conventions.
3. User control and freedom: Supporting undo and redo states, as users can often make mistakes in selection. Should allow them to go back to their previous state easily.
4. Consistency and standards: Users should not have to wonder whether different words, situations or actions mean the same thing.
5. Error Prevention: Prevent an error from occuring in the first place, instead of error messages. Include a warning or confirmation option to prevent users from committing to errors.
6. Recognition: Instead of making a user memorize things, keep instructions, actions, options visible and easily retrievable.
7. Flexibility: Allow users to control their own speed that they play the game, eg. skip cutscene, or tailoring frequent actions.
8. Aesthetic: Dialogue should not contain irrelevant information. Too much info obscures the important ones.
9. Error support: Error messages should be easy to read (no code), indicate a problem and suggest a solution.
10. Help and documentation: Ideally should be able to be used without, but if necessary should be easy to search and have instructions to solve a problem.

Review Based Evaluation:

- Results are used to support or refute parts of design
- Should be able to ensure that the results are transferable to the new design
- Model-based evaluation

Think Aloud:

- User observed performing task
- User commentates as they are doing a task, and their thoughts and opinions about said task
- Advantages: Simple, requires no expertise, can provide useful insight and show how a system is used
- Disadvantages: Subjective, selective and describing while doing a task may affect the task performance

Cooperative Evaluation:

- User and evaluator can ask each other questions as they proceed through the game
- Similar to think aloud
- Additional Advantages: less contraints and easier to use, user is encourages to criticize a system, clarification is possible.

Post Task Walkthrough

- User reacts on action after the event, used to fill in intention
- Advantages: Analyst has time to focus on relevant incidents
- Disadvantages: Lack of freshness, may be an after interpretation of events

Eye tracking:

- Head or desk mounted equipment tracks the position of the eye
- Captures eye movement and reflects the amount of cognitive processing a display requires
- Measures eye movements, and targets

Physiological measurements:

- Measures emotional responses linked to physical changes
- Measurements include heart activity, blood pressure, sweat, muscle and brain activity

Interviews:

- Analyst asks a user pre-prepared questions on a one-to-one basis.
- Advantages: Can be varied to suit context, issues can be explored in depth, can elicit user views and identify unanticipated problems.
- Disadvantages: Very subjective and can be time consuming.
- Has two types of questions: closed questions (yes, no) and open questions. Generally avoid long questions, and questions that make assumptions (about game, gender, knowledge etc)

Questionares:

- Set of fixed questions given to users
- Advantages: Quick and reaches a large user group, can be analyzed more rigorously
- Disadvantages: Less flexible and probing