UX DESIGN RULES

CART 416

RILLA KHALED

USABILITY

USABILITY ACCORDING TO JAKOB NIELSEN

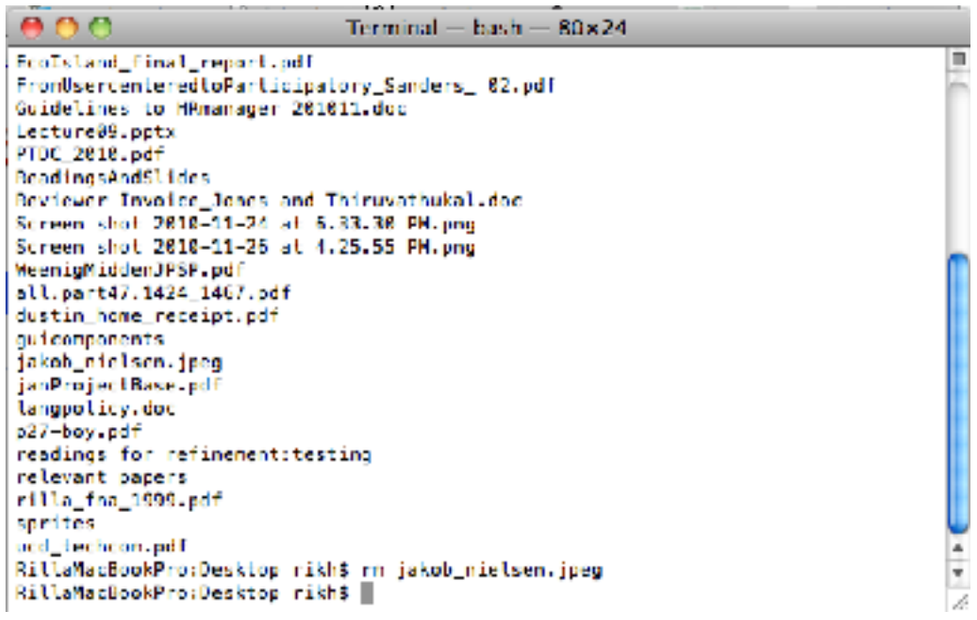
- Usability is a quality attribute that assesses how easy user interfaces are to use.
- Usability is different from playability (which is specific to games)
- Usability is defined by five quality components.

LEARNABILITY

"How easy is it for users to accomplish basic tasks the first time they encounter the design?"

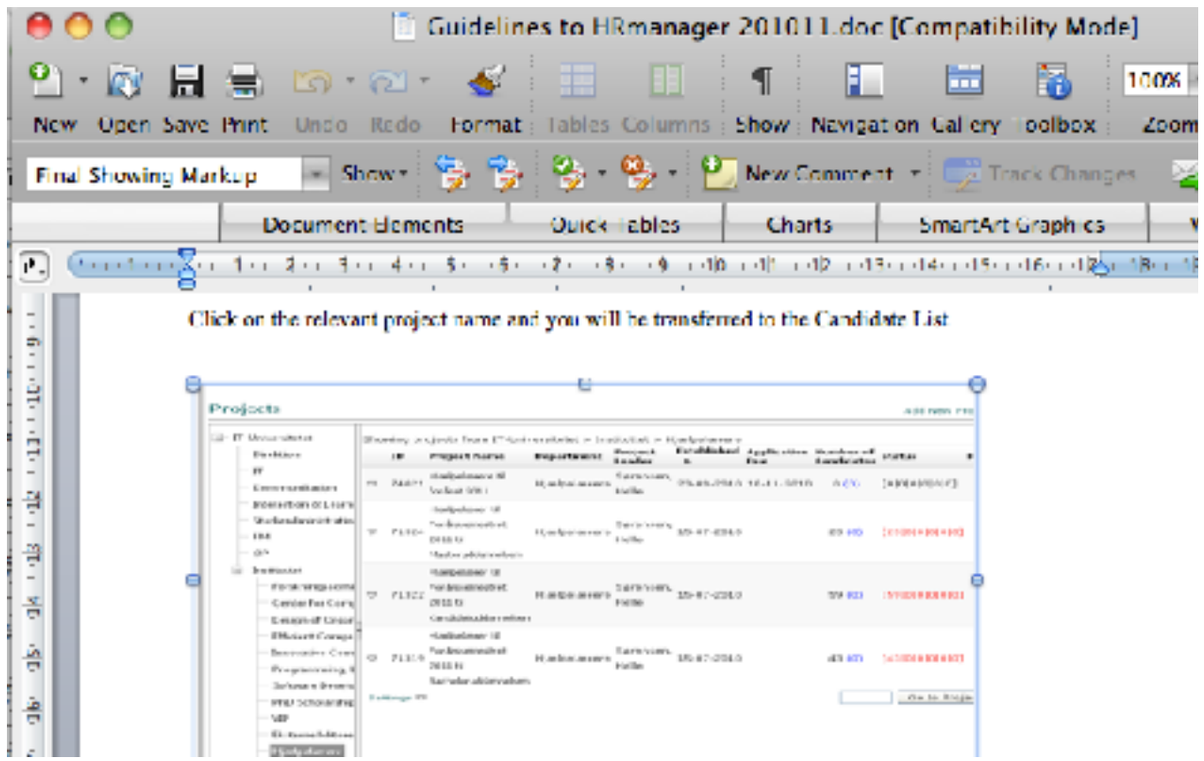
EFFICIENCY

"Once users have learned the design, how quickly can they perform tasks?"



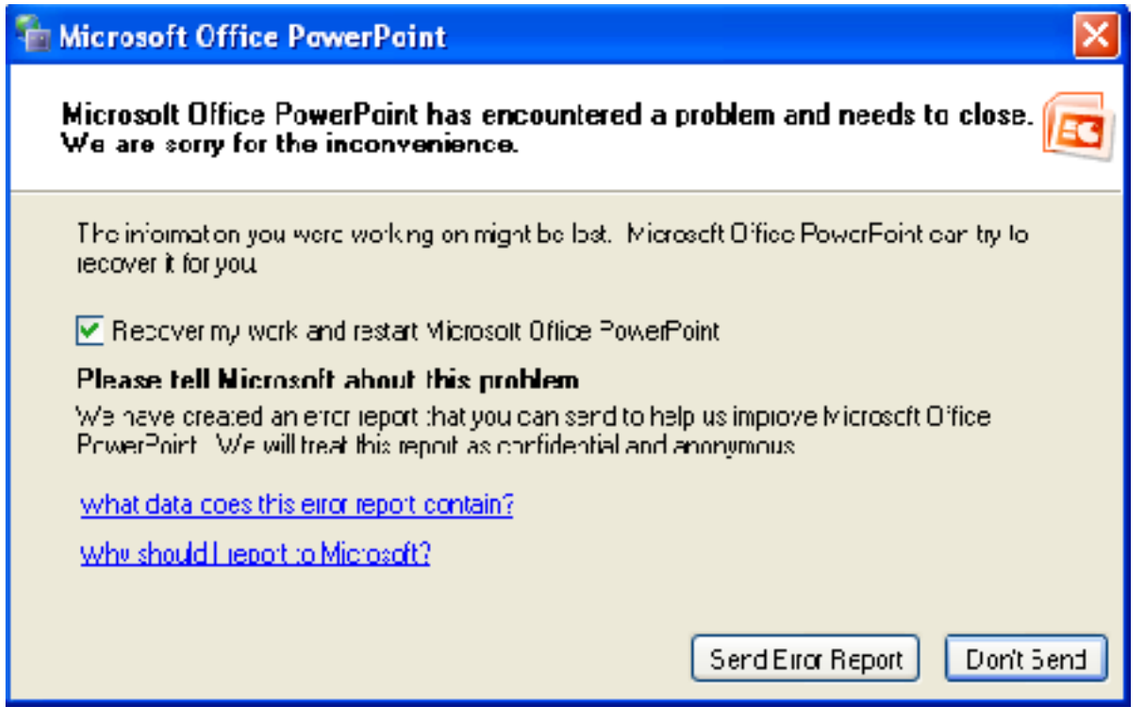
MEMORABILITY

"When users return to the design after a period of not using it, how easily can they reestablish proficiency?"



ERRORS

“How many errors do users make, how severe are these errors, and how easily can they recover from the errors?”



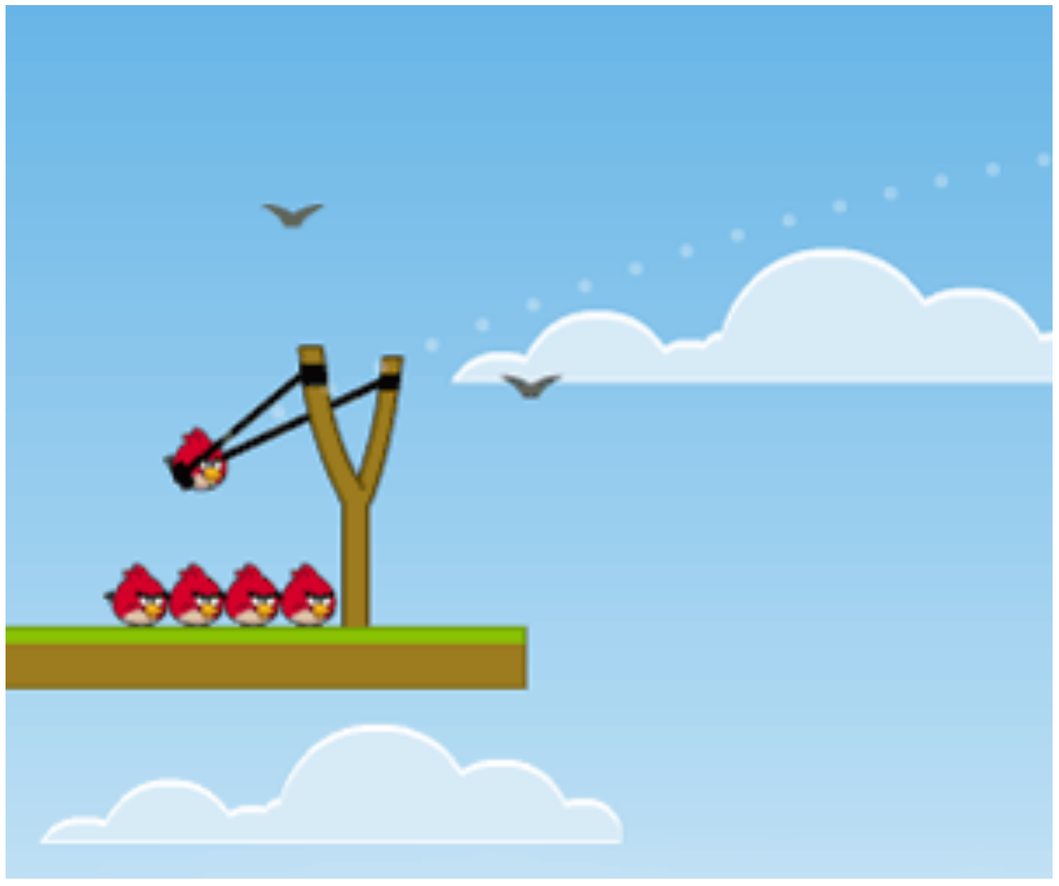
SATISFACTION

“How pleasant is it to use the design?”

(SOME) INTERACTION DESIGN PRINCIPLES

FEEDBACK AND FEEDFORWARD

- feedback: every action should be accompanied by appropriate acknowledgement of that action (e.g. moving the mouse should move the cursor)
  - too little feedback means users didn't know their action was successful
  - too much feedback can be annoying
- feedforward: knowing what will happen before you perform an action
  - allows users to act confidently
  - various forms:
    - message boxes
    - avatar outlines



Angry Birds feedforward

FITTS'S LAW

- the time that it takes to move from a starting position to a target is determined by:
  - distance to the target
  - size of the target
- models the act of pointing
  - finger
  - mouse
  - controllers
- implications:
  - bigger targets means faster speed
  - screen edges and corners are very useful: you can't overshoot
  - controls next to user's regularly frequented locations reduce distance

MAGICAL NUMBER SEVEN RULE

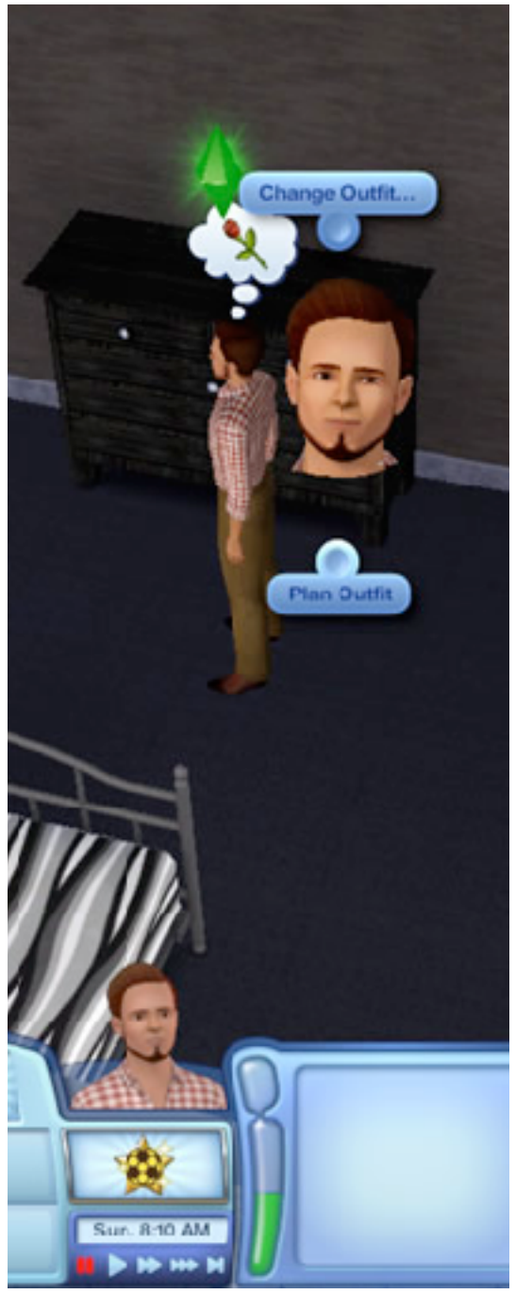
- the human mind is best able to retain information in short term memory in chunks of 7 “plus or minus two”
- e.g. after 5 – 9 numbers, we start to make errors
- not a blanket rule about interface elements though
  - applies to that which needs to be kept in short term memory or visualised
  - if information is displayed on screen, it doesn’t need to be retained in short term memory

POKA-YOKE PRINCIPLE

- mistake proofing:
  - avoiding (yokeru)
  - inadvertent errors (poka)
- put constraints on products to remove/prevent errors
- lots of ways to do this:
  - signs
  - procedures
  - enabling/disabling interaction
  - affordances

TESLER'S LAW OF THE CONSERVATION OF COMPLEXITY

- some complexity is inherent in every process
- beyond a certain point, all you can do is shift complexity from one place to another
- implications:
  - all processes have elements that cannot be made simpler
  - is there a reasonable place to move the complexity to? Can burdensome parts or frequently repeated parts of tasks be automated?



A GENERAL GAME UI DESIGN PROCESS

① WHAT ARE THE INPUTS AND OUTPUTS?

List all the actions the player performs in the game, as well as all game-state information.

② PRIORITISE.

Note next to each input and output whether the player uses it constantly or only occasionally. The ease of accessing each item should be proportional to how often it must be accessed.

③ FIND THE AFFORDANCES.

For each player action, consider what controls best affords that action. If there are several actions that seem to map to same control, consider context-dependent actions. If the best input is unclear, ask the target player audience.

For outputs, consider whether info needs to be shown and where.

④ GIVE IMMEDIATE FEEDBACK FOR YOUR INPUTS.

Give the player feedback whenever s/he does something – visual, auditory, tactile, a combo. It will improve the responsiveness (and juiciness) of the game.

⑤ REDUCE EVERYTHING!

Think of ways to reduce and collapse the controls. Make every game screen, menu, and subscreen fight for it's life. Most game actions – especially frequently used ones – should only be a single button press away.

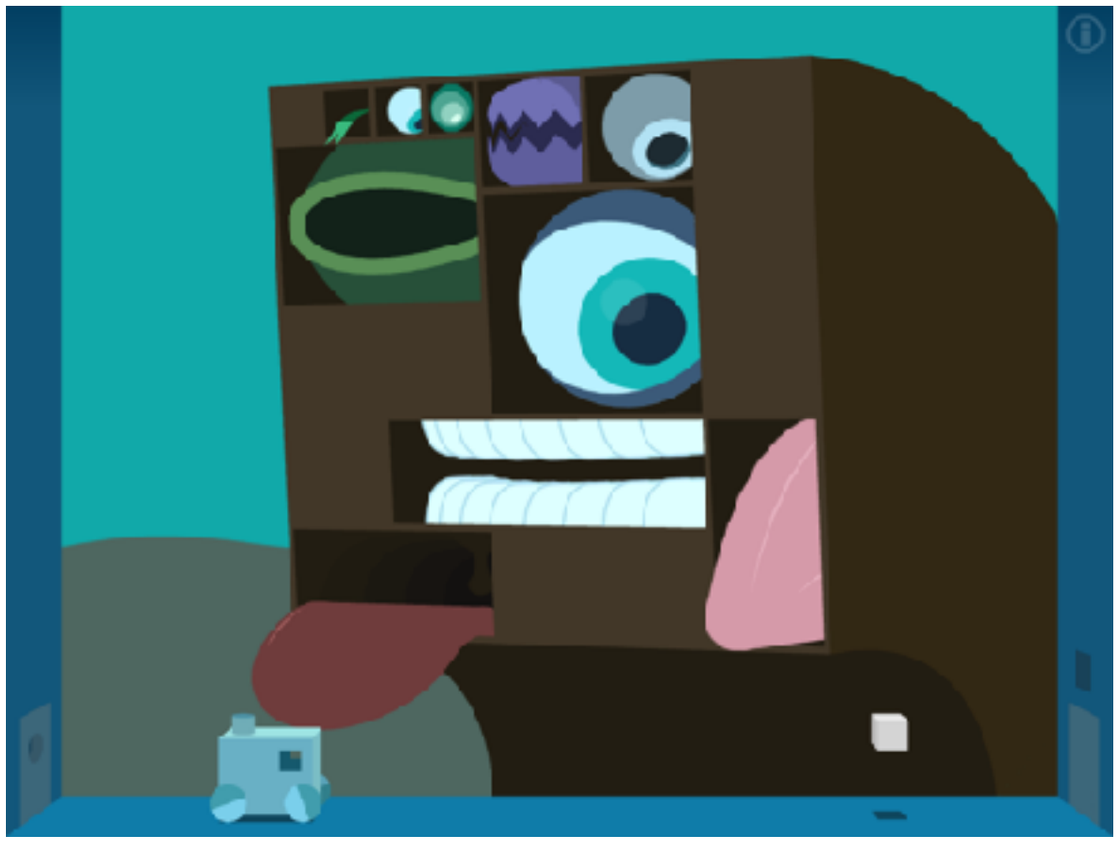
CHECK YOUR FEEDBACK LOOPS BY ASKING YOURSELF:

- What do players need to know at this moment?
- What do players want to know at this moment?
- What do you want players to feel at this moment? How can you give feedback that creates that feeling?
- What do the players want to feel at this moment? Is there an opportunity for them to create a situation where they will feel that?
- What is the player's goal at this moment? What feedback will help them towards that goal?

Jesse Schell. 2008. The Art of Game Design: A Book of Lenses. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

MAKE JUICY INTERACTIONS

- Represent second order motion, i.e. motion that is derived from the action of the player
- “Juicy systems” give the player additional control and reward the player in many ways at once.
- Is your second order motion powerful and interesting?



WINDOSILL
VECTORPARK

Jesse Schell. 2008. The Art of Game Design: A Book of Lenses. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.