

# Principles of GUI Design and Programming

Design Principles

# Contents

- Interaction design principles
- Interaction design patterns

# Design principles

- Design principles are
  - A set of generalized rules for what makes a good interface
  - A set of commonalities shared among good interfaces
  - A set of ideas you should think about when you are designing

# Design Principles

- Visibility – can I see it?
- Feedback – What is it doing now?
- Affordance – does its look tell me how to use it?
- Mapping – Do I know what this control does?
- Constraint – A limit on what I can do.
- Consistency – do I do it the same way everywhere?

# Visibility

- Things should be readily visible to the user
  - The controls to do tasks should be visible
  - You should give feedback on the state of the system
  - You should make the history of how people got to this state visible

# Traditional dashboard



# Tesla Model 3



# Canon 5D Mark iv





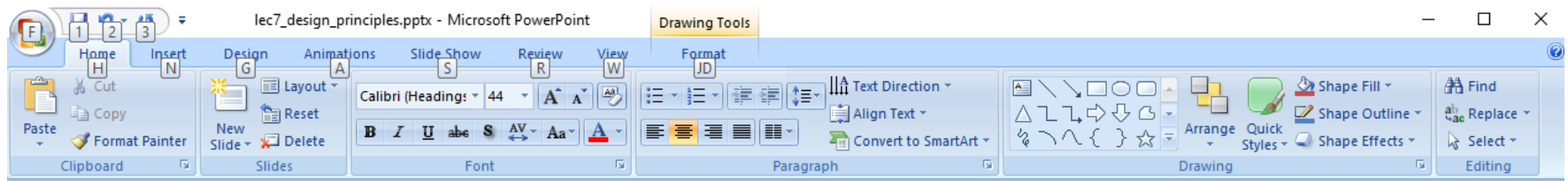
# Fuji Xt2



# Sony A7r iii



# PowerPoint



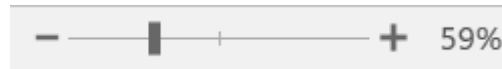
- controls are split into sections
- once a section is expanded, the controls are obvious
- they are in groups with pull downs
- the most used functions are available without using the pulldown
- this is a way to deal with the limited screen space

# Visibility

- The less visible controls are
  - The less likely people will find them
  - The less likely they will know how to use the device
- Use buttons rather than menus
  - If you run out of space, use modifier keys
  - Provide feedback on what the buttons do

# Mapping

The control should resemble the action it performs



The document resize button has a percent and a slider with indicators for the direction of resizing.

# Constraints

- Constraints restrict the user interaction in some way
  - Grey out menu items which cannot be selected
  - Alter the maximum and minimum values of sliders to limit the values which can be input



# Consistency

- Things which look similar should do similar things
- Things which work one way in one application should work the same in all applications
- Consistent interfaces have a set of rules which are always followed
- Consistent interfaces are easier to learn and easier to use

# Affordance

- Affordance refers to when an object give you a clue how to use it
  - Buttons beg to be pushed
  - Sliders beg to be dragged
- The goal is to design interfaces that make it obvious how to use them





# Design patterns

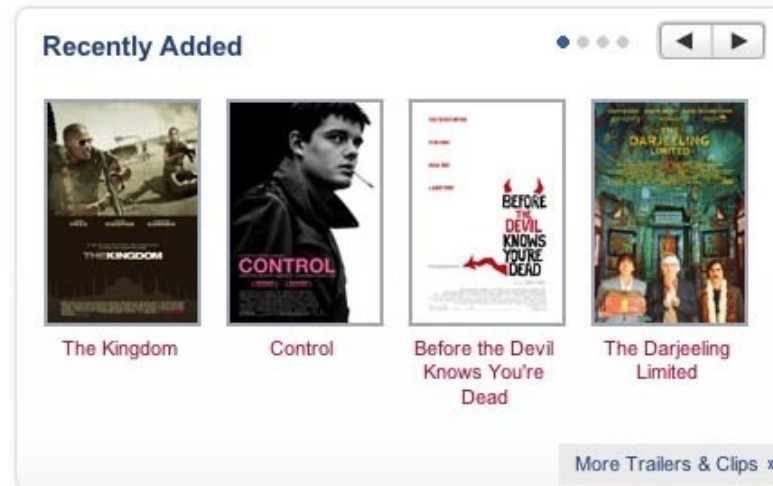
- Design patterns are well known solutions to common problems
- Design patterns
  - Are ideas
  - Not solutions to your specific problem
- Design patterns state
  - A problem
  - A solution
  - When it is appropriate to use that solution

# Design patterns

- One design pattern is
- **NAME:** Swiss Army Navigation
  - **PROBLEM:** Users want to interact with the application not be distracted by navigation controls
  - **SOLUTION:** to have navigation controls slide in from the side when needed and slide out of view when not needed
  - Variations are to have arrows which display and hide control panels at the side of the screen

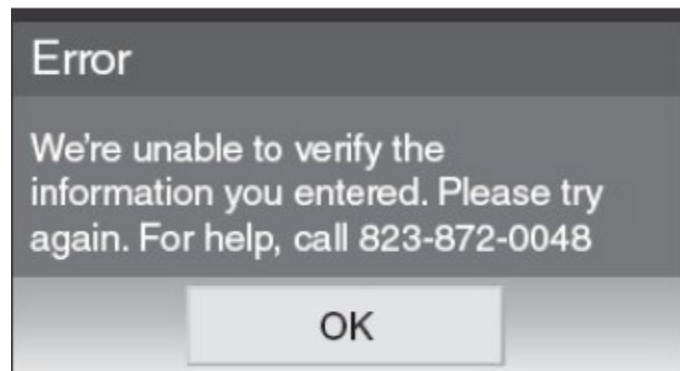
# Carousel Pattern

- Problem
  - User needs to browse a set of items and possibly select one
- Solution
  - Show the items as a scrolling, rotating list that the user can select an item from



# Anti-patterns

- Anti patterns are examples of how not to do something
- Often, they result from moving a desktop application to a smart phone
- Consider
  - You are upgrading phone software
  - Something goes wrong
  - It display a phone number to call for help
  - But you cannot use the phone because it is being upgraded...



# More Design patterns

- The website
  - <http://ui-patterns.com>
- Has a wealth of UI design patterns

# The User Experience

- The user experience is the total experience interacting with the
  - Company
  - Its services
  - Its products
- It is
  - How people feel about a UI
  - Their experience using it

# Factors affecting the UX

- Usability
- Functionality
- Aesthetics
- Content
- Look and feel
- Sensual and emotional appeal
- Fun
- health
- Social capital  
(developing social networks)
- Cultural identity
- Age, gender, race, disability, occupation, education

# Technology as experience framework

- It is difficult to quantify the user experience since it changes continually
- McCarthy and Wright propose a framework based on 4 core threads that make up our experiences
- This framework provides ideas/concepts you should consider when designing a user experience



# Technology as experience framework

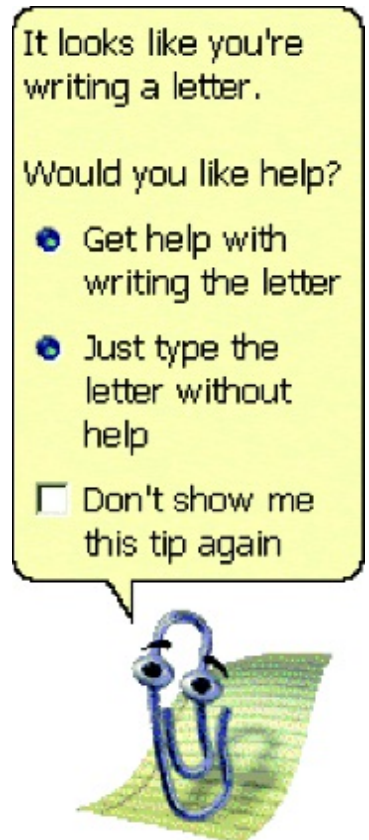
- Sensual
  - Our sensory engagement with the UI. High engagement typical of games, smartphones or chat rooms.
- Emotional
  - Does the UI cause sorrow, joy, happiness or anger
  - People get angry when the computer does not do what they want
  - People get frustrated when they cannot figure out how to accomplish a goal.

# Technology as experience framework

- Compositional
  - This is how the user performs steps to achieve a goal
  - Can the user put items in the shopping cart, checkout and pay without frustration?
- Spatio-temporal
  - The effect of using the interface on our time and space or perception of it

# Emotional Software

- Clippy was designed to aid users of MS office
- It was
  - Almost always wrong
  - Showed up at the worst times
  - Really, really annoying
- Everyone wanted Clippy dead!



# Design patterns on the web

- Mobile
  - <https://ptrns.com>
  - <http://www.mobile-patterns.com/>
- Web & desktop
  - <http://uigarage.net/>
  - <http://ui-patterns.com/>
  - <http://www.welie.com/>
  - <https://www.smileycat.com/category/elements-of-design/>