


# Design & Usability

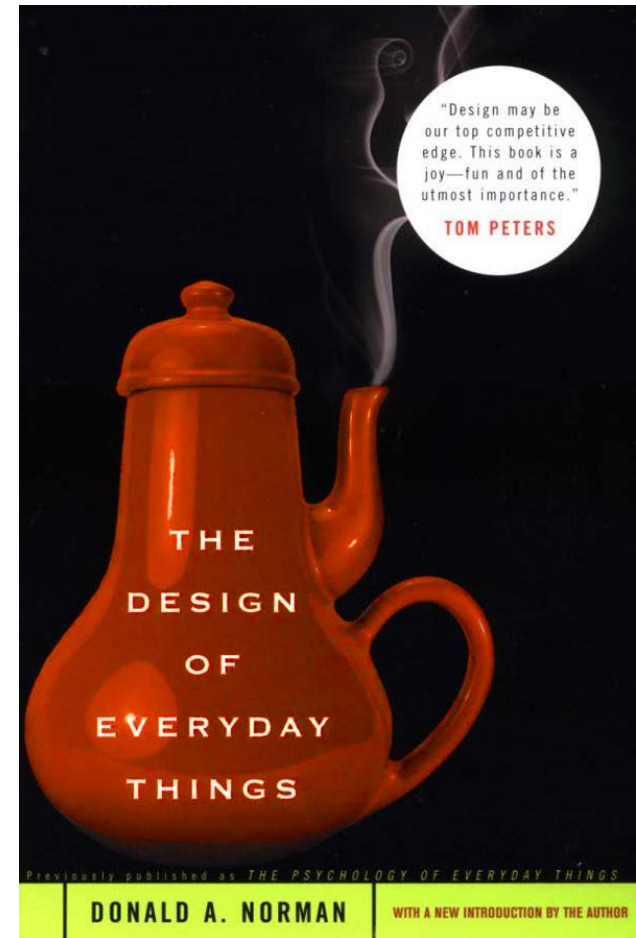
Dr Brendan Cassidy  
CO1111 Computing Skills

# Hello

- ▶ Brendan Cassidy
  - ▶ Office: CM015
  - ▶ Email: [bcassidy1@uclan.ac.uk](mailto:bcassidy1@uclan.ac.uk)
  - ▶ Tel: 3265
- 

# If you are interested in Design

- ▶ If you like the content of this lecture or are interested in design in general
- ▶ The Design of Everyday Things
  - By Don Norman
  - ISBN-10: 0465067107
  - ISBN-13: 978-0465067107
- ▶ First published in 1988 & still recommended reading for employees of design studios around the world



# Design

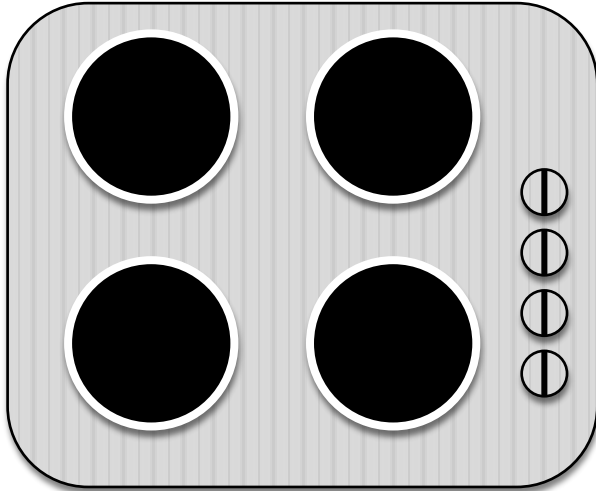
- ▶ Why Design?
  - If the user can't use it, it doesn't work
- ▶ Having a product that functions is not enough
- ▶ Provides a competitive edge for your product
  - design makes products better
    - Makes objects comfortable
    - Makes equipment understandable
    - Makes environments friendly
    - Makes packaging clear
    - Makes websites/apps useful

# Designing for the user

- ▶ There are lots of factors that influence design
  - Aesthetics, Cost, Time, Safety
- ▶ We will focus on designing for the end user
  - Need to understand how people will use items
- ▶ We will look at four ways to improve a design for the user
  - Mapping
  - Affordance
  - Feedback
  - Constraints

# Mapping

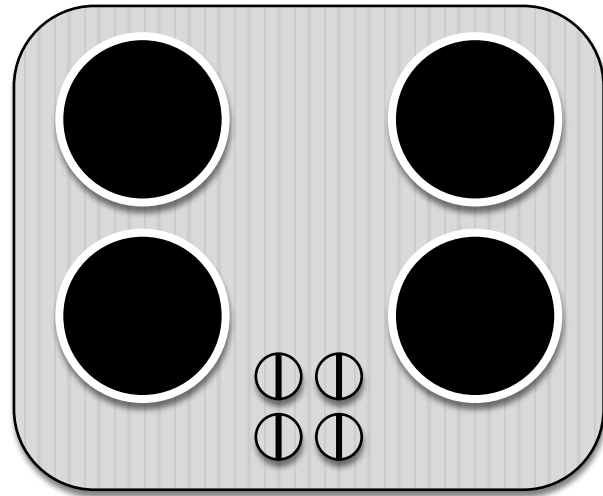
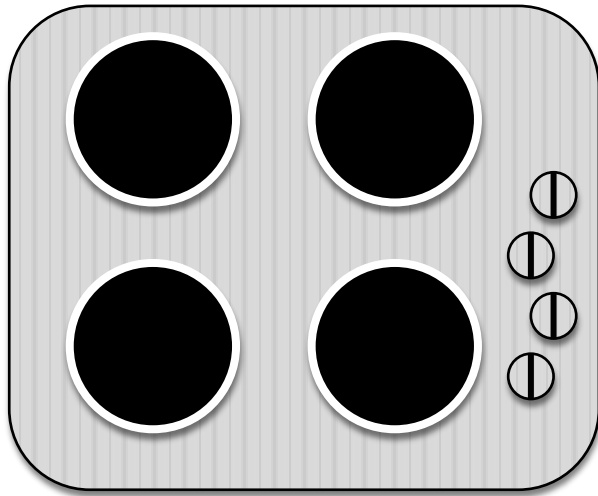
- ▶ Where possible a products controls should map to their function
- ▶ Which knob operates which hob on this cooker?



Source:  
<http://www.baddesigns.com/ranges.html>

# Mapping 2

- ▶ How about these?



# Mapping 3

- ▶ An example from the labs (CM33)





# What are you going on about?

- ▶ Why are you telling me about cookers and light switches?
  - The 4 key principles of mapping, affordance, feedback & constraints apply to all artefacts that are going to be used by people
  - Products & Technologies change rapidly
  - People do not



# Affordance

- ▶ A products design should give clues about how to use it
- ▶ Affordance is a quality of an object that allows an individual to perform an action
  - Must allow an action the user is aware of
- ▶ When you see an object you have never seen before do you know how to use it
- ▶ Appearance of an object should provide clues to its operation
  - The way it works
  - What actions can be taken



# Affordance 2

- ▶ An example from the labs (CM33 Door)



# Affordance 3

- ▶ Do you pull or push this door?



# Bad Affordance



<http://www.flickr.com/photos/cennydd/3191424928/> Picture by Cennydd Bowles



<http://www.flickr.com/photos/immason/2093621345/> Picture by Jeffrey Mason



# Dr Nick App

- ▶ Is it clear without the text that you can press Doctor Nick?



# Affordance & Mapping

- ▶ An elevator example..
- ▶ Do rows of 3 map well?
- ▶ What affords pressing more?
  - The buttons?
  - The labels?
  - Notice the second floor label has nearly been worn away



Source: Picture by Joseph Nicolia

<http://www.flickr.com/photos/24885050@N00/2295018167>

# Affordance & Mapping 2

- ▶ That's Better





# Feedback

- ▶ We need to know if something has happened
  - Give an action an immediate and obvious effect
- ▶ We also need to know if something is happening



Is the bottom right switch on or off? How do we know?



Washing machine dial turns to tell us where it is in the cycle

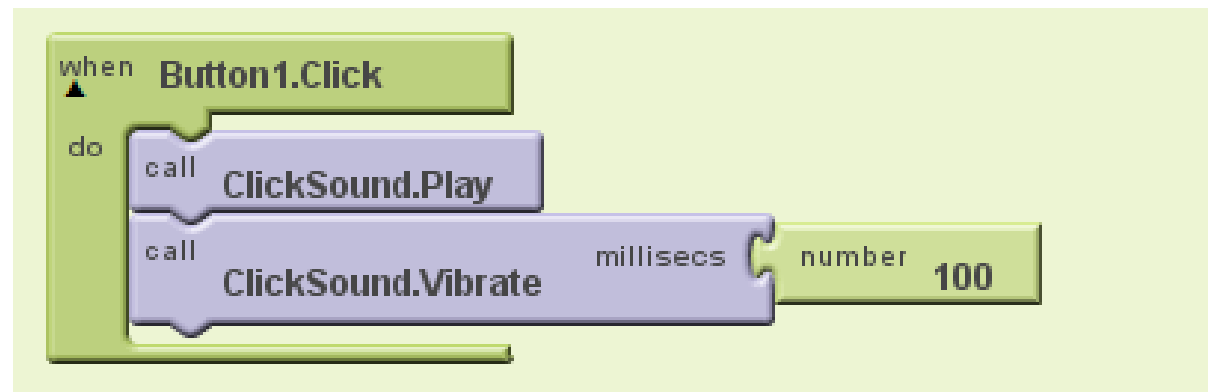
# Feedback (Phone Context)

- ▶ Buttons provide a resistance and a 'click'
- ▶ Touch screens do not
  - Need to simulate this
    - Artificial 'click' sound
    - Vibro-tactile feedback on button press
- ▶ On some resistive touch screen phones the whole screen actually 'clicks'

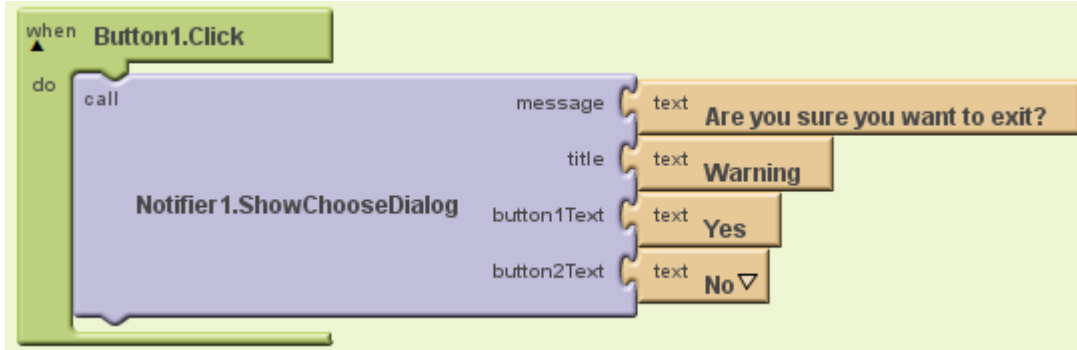


# Using Feedback in App Inventor

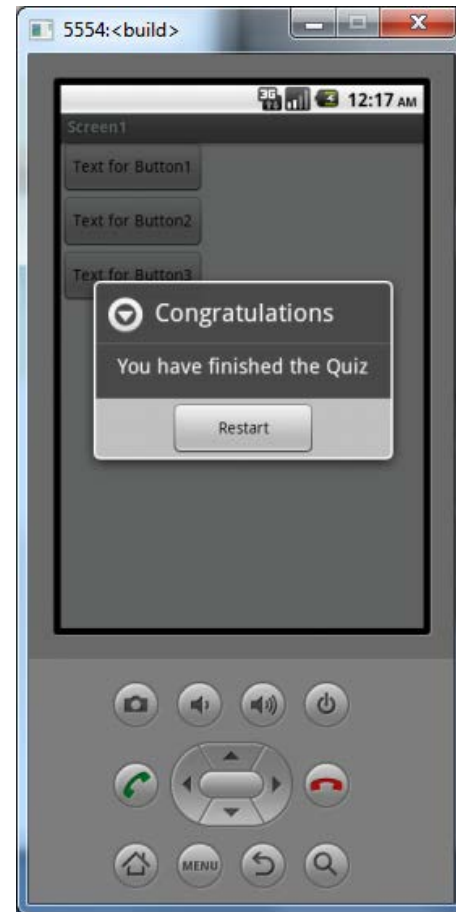
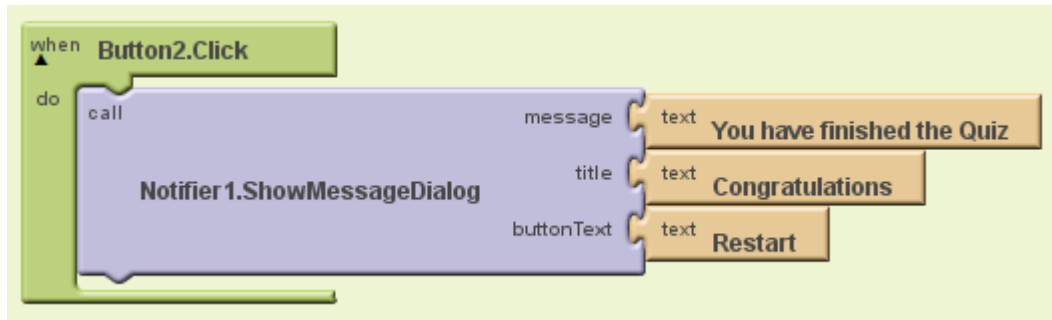
- ▶ Vibrate is a procedure belonging to the sound component
- ▶ It requires a number to specify the duration in milliseconds the phone should vibrate
- ▶ Experiment to see what duration is the most suitable



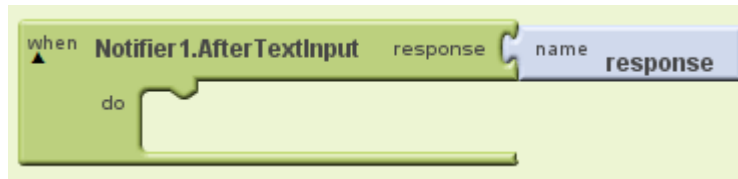
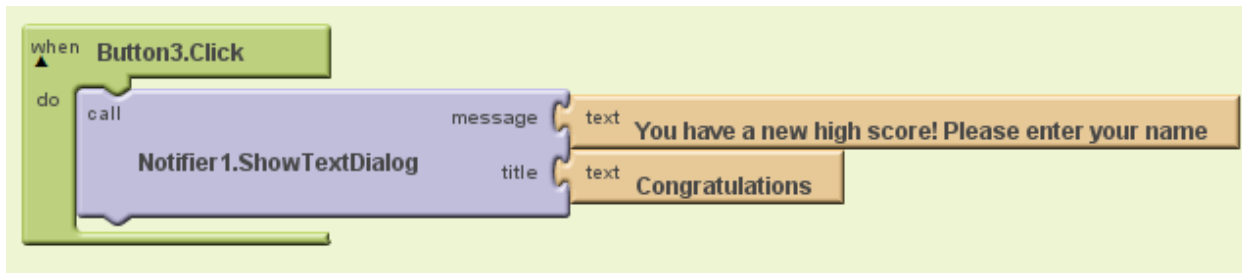
# Feedback 2.1 (Notifications)



# Feedback 2.2 (Notifications)



# Feedback 2.3 (Notifications)



# Constraints

- ▶ Make it difficult for the user to make mistakes
- ▶ General constraints
  - Push plate on a door
    - You can only push it
    - Its positioning is also a constraint
      - Tells you where to push the door
  - House door key
    - There are two possible orientations of the key i can try to open the door with
    - Only one works
  - Car door key
    - Possible actions are restricted as it will work with both orientations



# Constraints 2

- ▶ Constraints in computing
  - Network Cables
    - Not possible to work with both orientations so a clear physical signal is provided
  - Iphone unlock screen
    - With only one operation it is clear what the user must do in order to unlock the phone





# Colour

- ▶ I Want people to be paying careful attention to their choice of colour and size of text available to them
- ▶ Remember the context of use for your application
  - Where is it going to be used?
  - When is it going to be used?
  - How is it going to be used?
- ▶ Pay attention to colour scheme you are using
  - Is it legible
  - Will it cause problems in different contexts
  - E.g. in direct sunlight

# Colour

- ▶ I Want people to be paying careful attention to their choice of colour and size of text available to them
- ▶ Remember the context of use for your application
  - Where is it going to be used?
  - When is it going to be used?
  - How is it going to be used?
- ▶ Pay attention to colour scheme you are using
  - Is it legible
  - Will it cause problems in different contexts
  - E.g. in direct sunlight

# Colour

- ▶ I Want people to be paying careful attention to their choice of colour and size of text available to them
- ▶ Remember the context of use for your application
  - Where is it going to be used?
  - When is it going to be used?
  - How is it going to be used?
- ▶ Pay attention to colour scheme you are using
  - Is it legible
  - Will it cause problems in different contexts
  - E.g. in direct sunlight

# Context of Use

- ▶ What is context of use?
  - The circumstances in which an interactive product will be expected to operate
    - Physical Environment
    - Social Environment
    - Organisational Environment
    - Technical Environment



- ▶ Tells us how and when a product will be used
  - How interactions fit into our daily life
- ▶ Tasks the system does not support




# Physical Environment

- ▶ Is it to be used inside or outside
- ▶ How much lighting is there
- ▶ How much noise is there
- ▶ Will users be wearing specific clothing
  - E.g. Gloves
- ▶ For Example an ATM operates in a very public physical environment
  - Speech interaction may be problematic
  - May operate on a noisy high street
  - People may be wearing gloves


# Social Environment

- ▶ How many people are to use the system
- ▶ Do these people have to be in the same place
- ▶ Will people be using the system at the same time
- ▶ Will data need to be shared
- ▶ Does the sharing have to be synchronous
  - E.g. Google Docs, multiple authors
- ▶ In an ATM context security is paramount

# Organisational Environment


- ▶ More geared towards industry
  - ▶ How good is user support likely to be
  - ▶ How easily can it be obtained
  - ▶ Are there facilities for training
  - ▶ How efficient or stable is the communications infrastructure
  - ▶ How hierarchical is the management structure
- 

# Technical environemt

- ▶ What technologies will the product run on
    - What size screen
    - Input devices
  - ▶ What will it need to be compatible with
    - Older systems that may limit functionality
    - What systems will it need to communicate with
  - ▶ What technical limitations will be relevant
    - Is there a limit on processor speed
    - Or memory
    - If a product is networked what is the network infrastructure like
- 



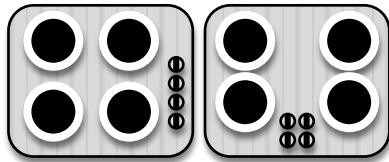
# This Weeks Lab

- ▶ Continue with your Quiz App
  - ▶ Redesign your app considering
    - Feedback
    - Mapping
    - Affordance
    - Constraints
    - Context
  - ▶ Incorporate the use of:
    - Vibrations
    - Notifications
    - Audio Feedback
- 

# Summary

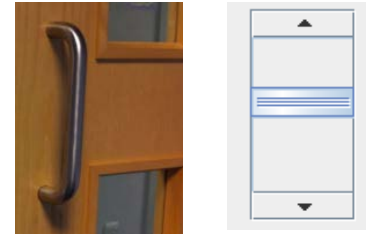
## Mapping

Making a design layout support its operation



## Affordance

providing clues as to how an interface is used



## Constraints

making it difficult for the user to make mistakes



## Feedback

letting the user know what has happened/is happening

