COMP140: Creative Computing Hacking

Interface Design & Evaluation

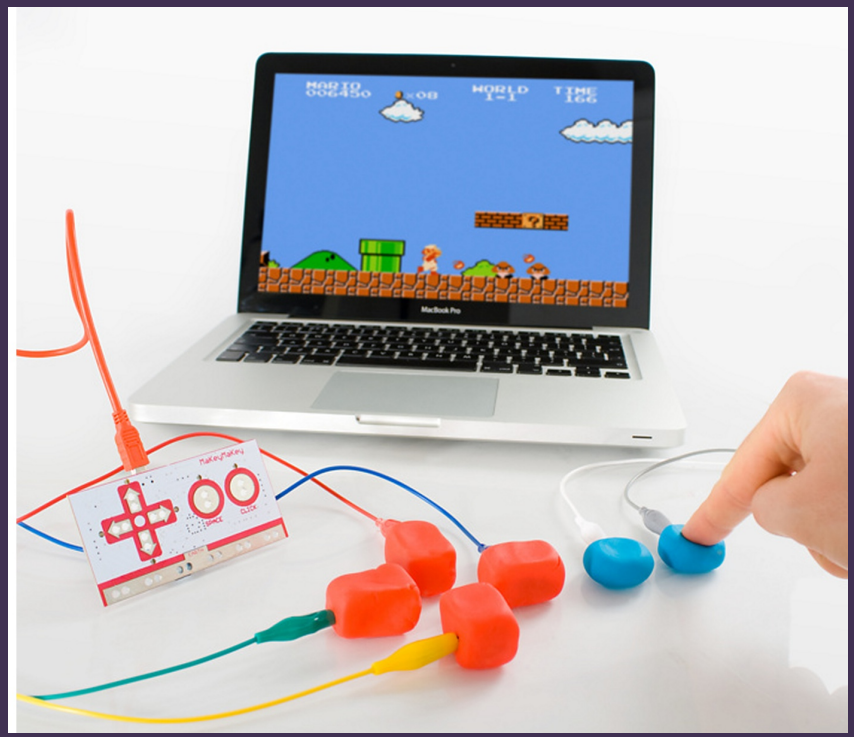
Lecture Objectives

Today's lecture will build upon the practical design of your game controller, focusing on:

- Exploring the nature of input, output, and interaction styles
- Examining the role of prototyping in design

This will be followed up by a practical in which you will identify heuristics and apply them to a peer's game interface.

Important Notice



Remember to bring your *Makey Makey* kit and associated materials to these lectures for practical support toward the end of each of these sessions.

FALMOUTH
UNIVERSITY

Input, Output, and Interaction Styles

Learning Outcomes

In this section you will learn how to...

- **Explain** the role of input and output in systems design
- **List** and **describe** a variety of input and output devices, giving examples of situations where each may be appropriate
- **Explain** what interaction styles are, while **critically evaluating** their respective advantages and disadvantages
- **Discuss** the role of direct manipulation in interacting with current computer systems

Further Reading

- Shneiderman, B. (1998) *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. 3rd Edition. Addison Wesley.

Input and Output Technologies

- **input**: the process that occurs as data from the players mind (or from the environment) is transformed into data that computers can use.
- **output**: the process of re-representing computer data into a form the player can perceive, comprehend, and make use of.

Input and Output Technologies

When developing and/or selecting an input device for a game, designers are faced with several design trade-offs:

- no single optimal device for all tasks
- form of data (e.g. selection verus alphanumeric)
- variety of players each with different characteristics

Activity

You are making a 2D-platformer for blind players.

- Self-organise into pairs.
- **Discuss** the challenges of this scenario on Slack.
- **Research** the characteristics of 2D platformers and blind players, **posting** your key findings on Slack.
- **Design** and/or **identify** appropriate an input device to support players' jump accuracy.

Time: 10-minutes.

Socrative JBYPC3BBY

- **Summarise** your choice and/or design of input device.

Input and Output Technologies

Similar challenges arise when designing and/or selecting output devices for a game:

- no single optimal device for all tasks
- form of data (e.g. attention verus conversant)
- no single optimal sense to engage
- variety of players each with different characteristics

Activity

You are making an RTS game for deaf players.

- Self-organise into pairs.
- **Discuss** the challenges of this scenario on Slack.
- **Research** the characteristics of RTS games and game players, **posting** your key findings on Slack.
- **Design** and/or **identify** appropriate an output device to direct players' attention to units in conflict.

Time: 10-minutes.

Socrative JBYPC3BBY

- **Summarise** your choice and/or design of output device.

Interaction Styles

- **interaction style**: a term used to describe different approaches to communication between players and computer games.
- Includes things such as:
  - command-line entry;
  - menu;
  - form-fill;
  - natural language;
  - WIMP;
  - direct manipulation.

Socrative `JBYPC3BBY`

- In pairs.
- Quietly discuss what you think is meant by the acronym `WIMP' for 2-minutes.
- **State** the meaning of WIMP.

Command Line Entry

Advantages:

- Functionally powerful.
- Quick to Use.

Disadvantages:

- Requires player to remember commands and syntax.
- Little feedback, *or* far too verbose.

Menus

Advantages:

- Facilittes information navigation.
- Restricts potential actions—safe for novices.
- Reduces memory load—knowledge in the world.

Disadvantages:

- Restricts functionality and freedom.
- Can be made too complex—difficult to find functions.

Form Fill-In

Advantages:

- Paper as a metaphor.
- Simple and intuitive.

Disadvantages:

- Minimally interactive.
- Requires an effective supporting layout.

Natural Language

Advantages:

- Intuitive and potentially powerful.
- Works effectively for simple interactions.

Disadvantages:

- Technological limitations—accents for example.
- Ambiguitiy in language interpretation.
- Unsuitable for `twitch' contexts.

Socrative `JBYPC3BBY`

- In pairs.
- Quietly discuss the advantages of‘WIMP’ for 2-minutes.
- **State TWO** advantages of WIMP.

Socrative JBYPC3BBY

- In pairs.
- Quietly discuss the disadvantages of‘WIMP’ for 2-minutes.
- **State TWO** disadvantges of WIMP.

Direct Manipulation

"The central tenets of direct manipulation are visibility of the objects of interest, actions being performed through the rapid, reversible, incremental behaviours and actions performed directly on screen objects"

(Perry, 2006, p. 33)

Direct Manipulation

Examples:

- Interactive Page Animations
- Desktop Icons
- Scrollbars

Direct Manipulation

Direct manipulation aims to address two interface challenges (from Normal and Draper, 1986):

- Gulf of Execution
- Guld of Evaluation

Gulf of Execution

- **Gulf of execution**: the 'distance' between a player's goal and the means of achieving it.
- "One measure of this gulf is how well the system allows the person do the intended actions directly, without extra effort" (Norman, 1988, p. 51)

Gulf of Evaluation

- **Gulf of evaluation**: the `distance' between the state of the game and the player's ability to assess it through perceiving representations.
- "The gulf is small when the system provides information about its state in a form that is easy to get, is easy to interpret, and matches the way the person thinks of the system" (Norman, 1988, p. 51)

Direct Manipulation

Shneiderman (1998) suggests several advantages:

- Novices learn functionality quickly

Direct Manipulation

Shneiderman (1998) suggests several advantages:

- Novices learn functionality quickly
- Experienced users can define new functions and features

Direct Manipulation

Shneiderman (1998) suggests several advantages:

- Novices learn functionality quickly
- Experienced users can define new functions and features
- Casual or intermittent users can retain operational concepts

Direct Manipulation

Shneiderman (1998) suggests several advantages:

- Novices learn functionality quickly
- Experienced users can define new functions and features
- Casual or intermittent users can retain operational concepts
- Built-in constraints mean that all user actions are legal

Direct Manipulation

Shneiderman (1998) suggests several advantages:

- Change is incremental, with immediate feedback to players

Direct Manipulation

Shneiderman (1998) suggests several advantages:

- Change is incremental, with immediate feedback to players
- Reduces anxiety because the system is comprehensible and actions reversible

Direct Manipulation

Shneiderman (1998) suggests several advantages:

- Change is incremental, with immediate feedback to players
- Reduces anxiety because the system is comprehensible and actions reversible
- Gain confidence through mimesis and predicting actions

Socrative JBYPC3BBY

- In pairs.
- Quietly discuss examples of direct manipulation in games for 2-minutes.
- **State TWO** examples of direct manipulation.

FALMOUTH
UNIVERSITY

Prototyping

Learning Outcomes

In this section you will learn how to...

- **Explain** the role of prototyping in game interface design
- **Compare** different approaches to prototyping
- **Select** an appropriate prototyping method for particular usability challenges

Further Reading

- Jensen, S. (2002) *The Simplicity Shift*. Cambridge University Press.

The Value of Prototyping

- Jenson (2002) describes the value of prototyping: "to fail—and fail fast!"
- Only by learning from our mistakes can we develop truly usable designs.
- It is very rare, if ever, we get things right the first time and this is especially true of complex systems—such as those in interaction design.

Socrative JBYPC3BBY

- In pairs.
- Quietly discuss how prototyping game designs has help you develop those designs, for 2-minutes.
- **Explain** the benefits of prototyping your own words.

Searching the Design Space

Design can be conceptualised as serching for an *acceptable* design within an infinite design space. This perspective provokes several ideas:

- Designers may not search effectively
- Designers may not recognise an *acceptable* design
- Designers may converge on a local maxima in the design space: a bad design

Searching the Design Space

Prototyping, therefore, becomes:

- An effective search and evaluation method
- A means to communicate design information

Searching the Design Space

"In practice, prototyping allows designers to **conceptualise** their products, to better understand the kinds of task that the users do and to support them with the appropriate technology"

(Perry, 2006, p. 50)

Searching the Design Space

"Importantly, prototyping forces the (game designers) to visualise all of the steps in (game) software (even beyond the interface), and how well the interface will operate in practice."

(Perry, 2006, p. 50)

Socrative `JBYPC3BBY`

- In pairs.
- Quietly discuss the consequence of **not** prototyping and play-testing for 2-minutes.
- **Explain ONE** of these consequences.

Approaches to Prototype Development

Prototypes can be used at a number of levels:

- **Game conceptualisation**: developing the game concept into a game design.
- **Task-level prototyping**: how a particular game mechanic and/or task for the player meshes with player expectation and their attempts to fulfill their goals.
- **Menues and HUDs**: the form and placement of data input and output in specific contexts.

Approaches to Prototype Development

Different methods facilitate these levels:

- **Requirements animation**: demonstrating potential functionality and use-cases as animations that can be easily assessed by players.

Approaches to Prototype Development

Different methods facilitate these levels:

- **Requirements animation**: demonstrating potential functionality and use-cases as animations that can be easily assessed by players.
- **Rapid prototyping**: intensively collecting information on requirements and modelling them as small prototypes that can be easily assessed by players.

Approaches to Prototype Development

Different methods facilitate these levels:

- **Evolutionary prototyping**: developing an initial model that is evaluated and adapted until it 'evolves' into an improved end-product.

Approaches to Prototype Development

Different methods facilitate these levels:

- **Evolutionary prototyping**: developing an initial model that is evaluated and adapted until it 'evolves' into an improved end-product.
- **Incremental prototyping**: step-wise development of large prototypes, such as vertical slices, in phases to avoid delays between specification and delivery.

Full Prototype

- complete version of the intended system
- may be a model or a roughly assembled throw-away

Paper Prototype

- no functionality
- used to talk through a design and demonstrate interfaces

Horizontal Prototype

- complete coverage of the all interface elements
- little to no functionality

Vertical Prototype

- incomplete coverage of the interface elements
- high level of functionality in restricted areas

Low Fidelity Prototype

- little resemblence to the final `look and feel'
- cheap and fast to develop

High Fidelity Prototype

- much resemblence to the final `look and feel' (may even be better i.e., pre-rendered vs real-time in engine)
- expensive and time-consuming

‘Wizard of Oz’ Prototype

- no functionality at all—simulated through intervention by a hidden person
- requires operator to have key knowledge of system states and interactions

Socrative JBYPC3BBY

- In pairs.
- Quietly discuss which prototyping appeach is appropriate for early-stage battle interface design for an RPG for 5-minutes.
- Prototyping methods are not exclusionary (e.g. could combine high-fidelity with vertical for a pitch).
- **State ONE** prototyping methods **and justify** your answer.

Socrative `JBYPC3BBY`

- In pairs.
- Quietly discuss which prototyping appeach is appropriate for *late*-stage battle interface design for an RPG for 5-minutes.
- Prototyping methods are not exclusionary (e.g. could combine high-fidelity with vertical for a pitch).
- **State ONE** prototyping methods **and justify** your answer.

Best Practices and Pitfalls

Where is prototyping most effective?

- Poorly defined or uncertain requirements
- Cost of system rejection is high
- High-stakes (think, e.g. nuclear power station)
- Assess impact of changes through requirements to implementation (i.e., contract scope)

Best Practices and Pitfalls

What are the limitations?

- May introduce uneccessary constraints early in design process
- Development has little, if any, direct input by stakeholders (i.e., expert is implementing the prototype)
- Consumes time—a trade-off with production schedule
- Little data on safety, reliability, response time, and so on—may lead to impractical designs

Socrative JBYPC3BBY

- In pairs.
- Quietly discuss which situations are appropriate for prototyping a game component for 2-minutes.
- **Explain ONE** such situation.

FALMOUTH
UNIVERSITY

Practical Activity

Heuristic Analysis Task

- **Review** the heuristics at `https://www.nngroup.com/articles/ten-usability-heuristics/`.
- Self-organise into pairs.
- **Setup** your game and novel game controller.
- **Demonstrate** the prototype to a peer.
- **Conduct** a heurstic analysis of your peer's game interface, following the guidence at: `https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/`

Coursework Progress

- **Prepare** for the final sprint review to take place next week.
- **Develop** the final draft version of the prototype game controller.
- **Ensure** that you are ready to conduct heuristic analyses of your peers' work.