CPSC 427
Video Game Programming

User Experience

GLASS ( ) LETS LIGHT THROUGH, BUT NOT OBJECTS

What is User Experience (UX)?

- Understanding the way humans and technology interact
- Designing products/systems/services that provide best user experience
  - *Best on what metric?*

Even Big Companies Get UX Wrong

- Easy & expensive to get UX wrong

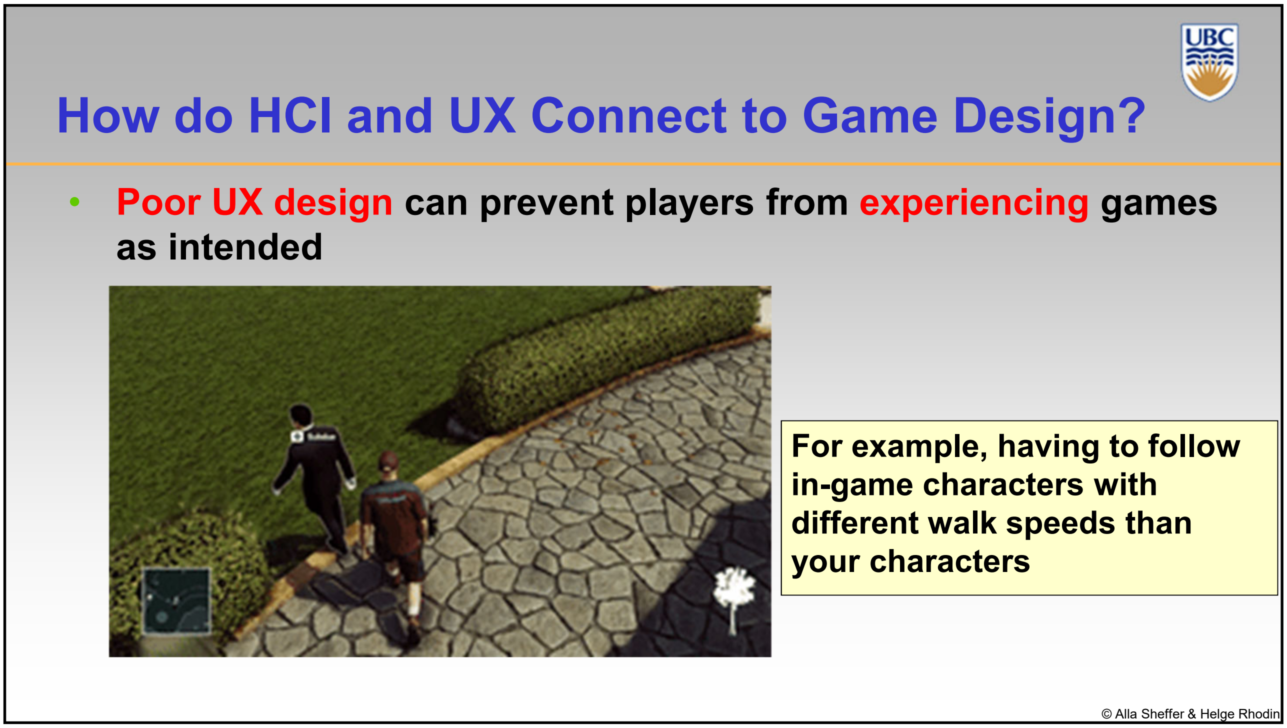
Google Glass failed in the market because it wasn't clear why people should need it

Connection to Game Design

- Impact of design on ease of use & engagement

In Wind Waker, the direction Link looked indicated to the player something of interest was there

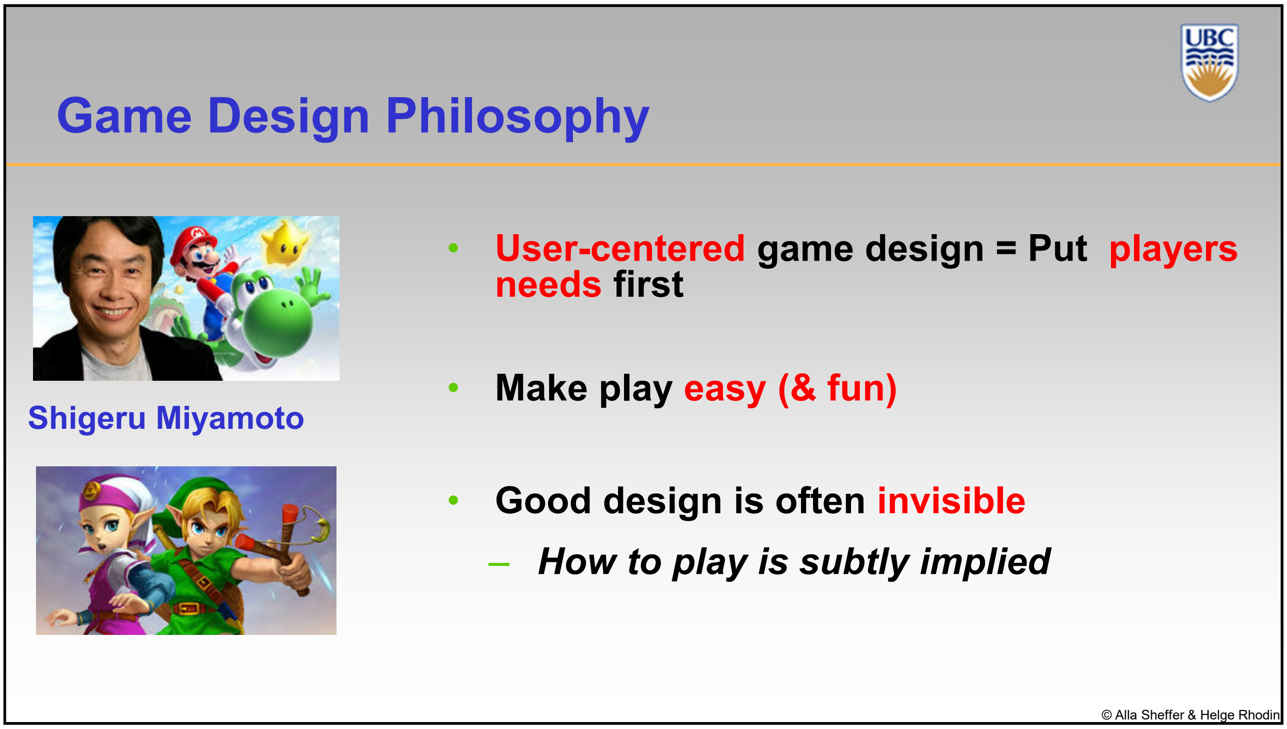
- Design applications & philosophies are interconnected



How do HCI and UX Connect to Game Design?

- **Poor UX design** can prevent players from **experiencing** games as intended

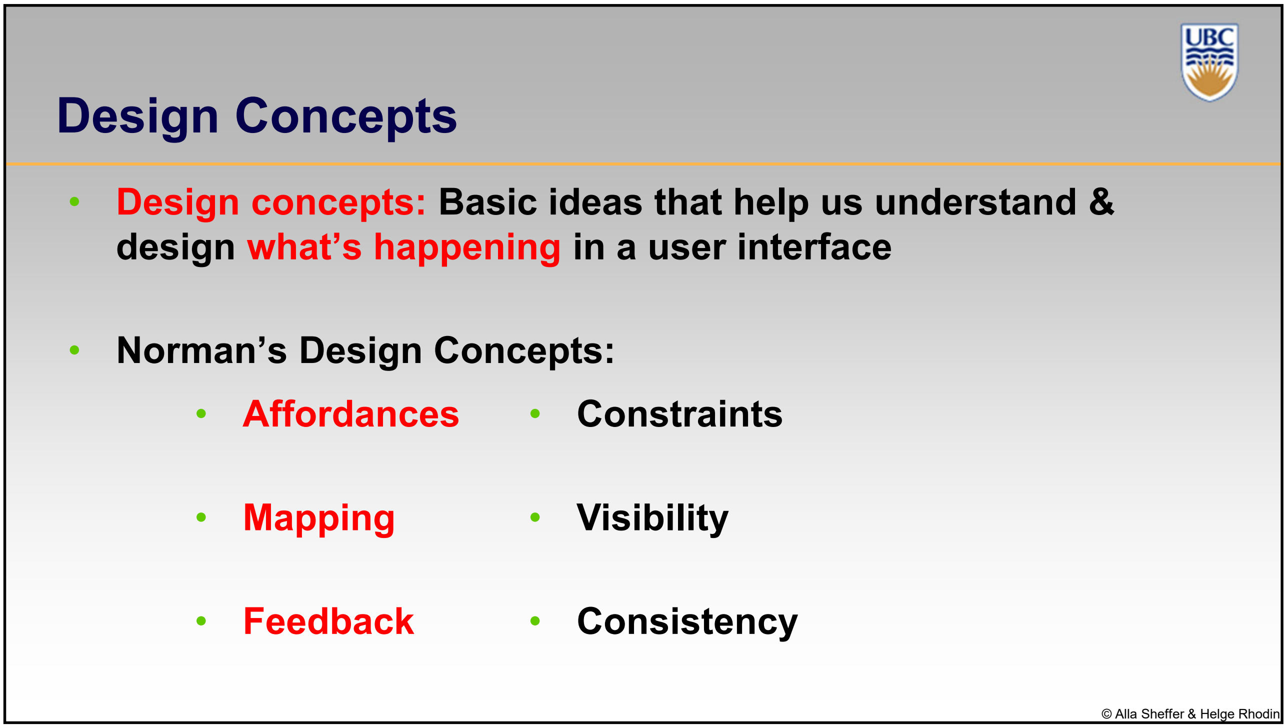
For example, having to follow in-game characters with different walk speeds than your characters



Game Design Philosophy

Shigeru Miyamoto

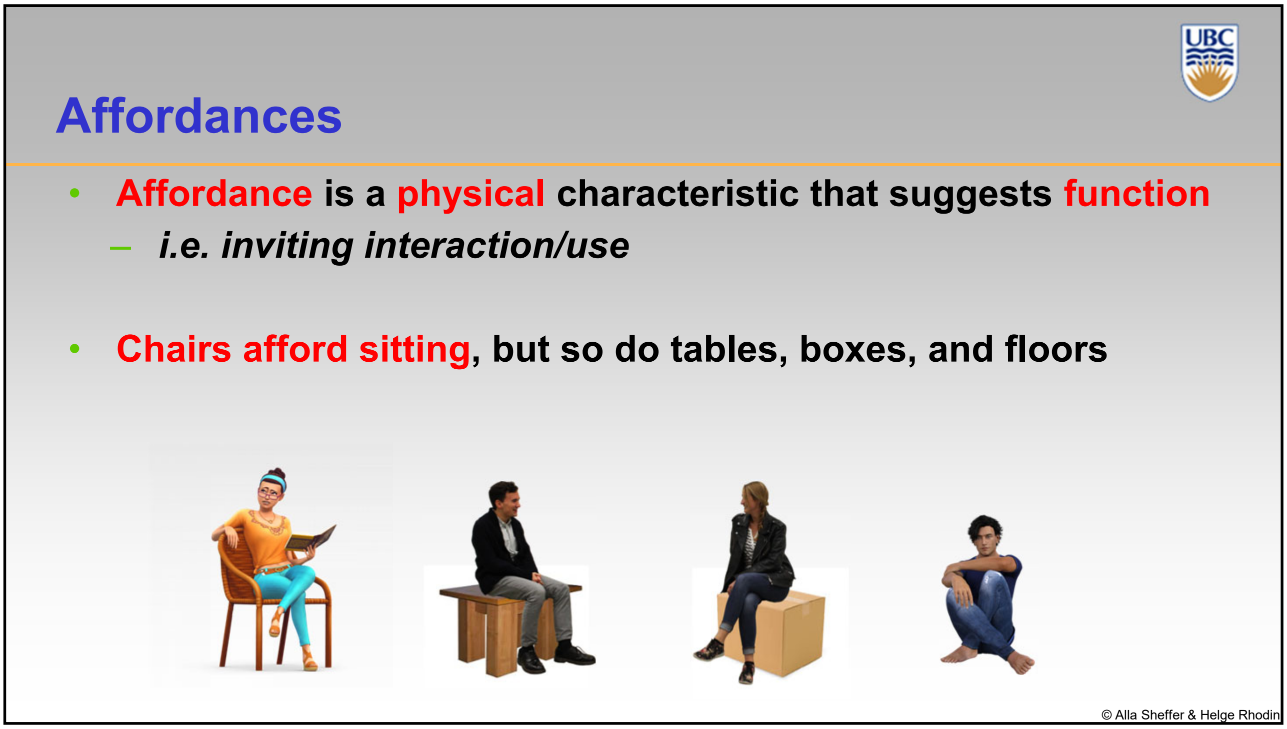
- **User-centered** game design = Put **players needs** first
- Make play **easy (& fun)**
- Good design is often **invisible**
  - *How to play is subtly implied*



Design Concepts

- **Design concepts:** Basic ideas that help us understand & design what’s happening in a user interface
- Norman’s Design Concepts:

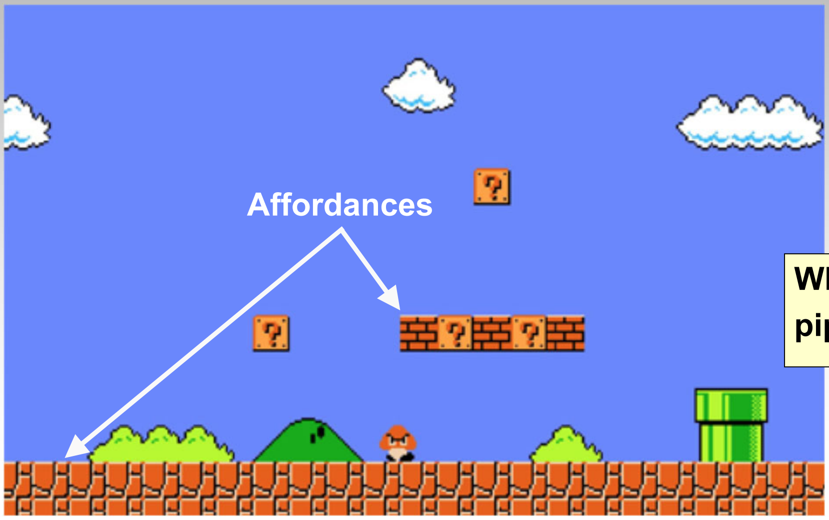
Affordances	Constraints
Mapping	Visibility
Feedback	Consistency



Affordances

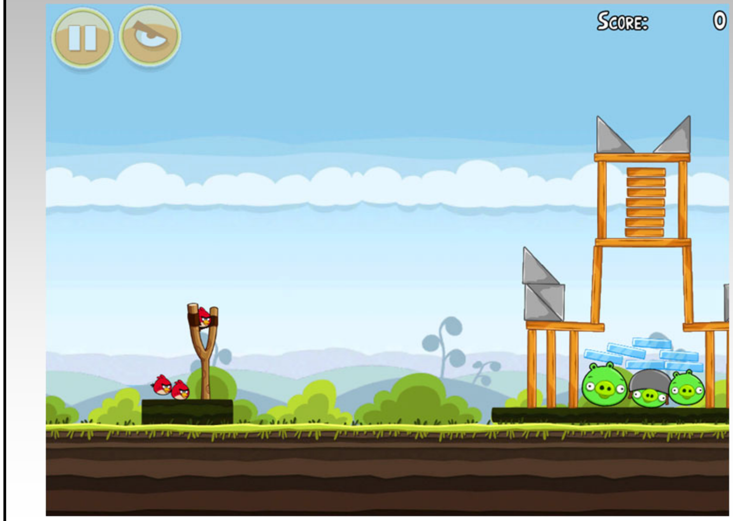
- Affordance is a physical characteristic that suggests function
  - *i.e. inviting interaction/use*
- Chairs afford sitting, but so do tables, boxes, and floors

Example of Affordances in Games

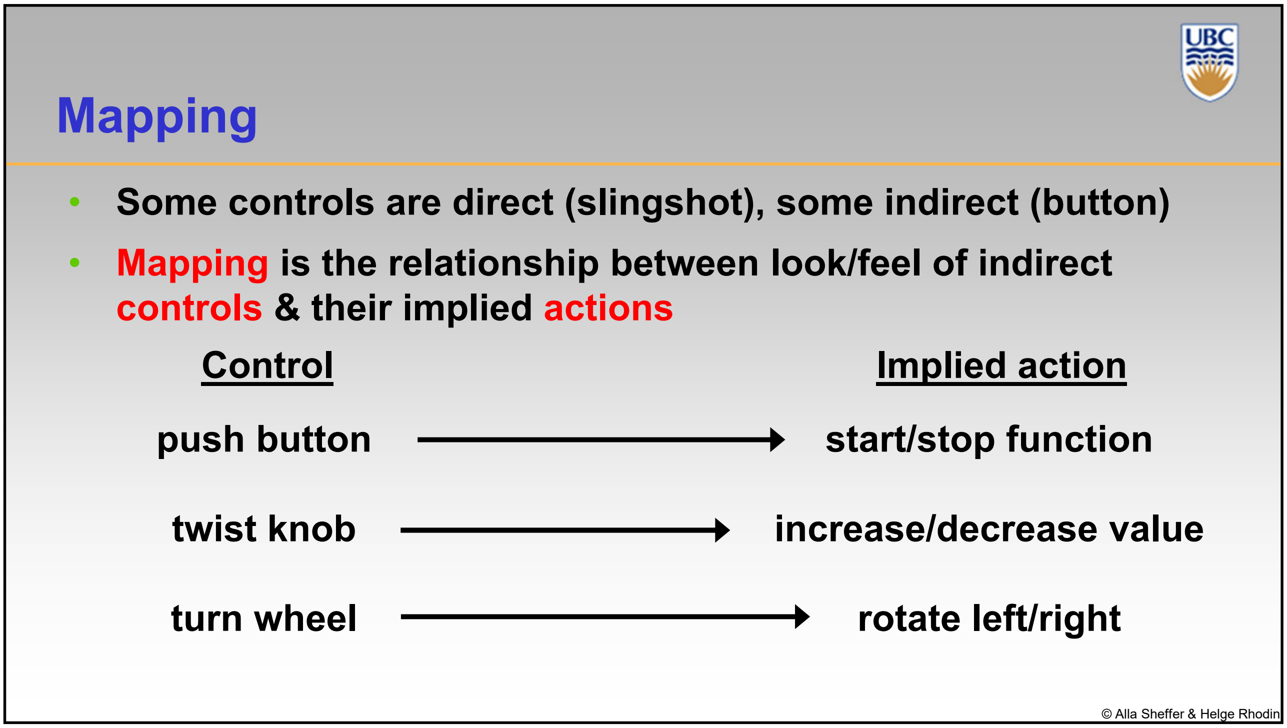


What does the pipe afford?

Example of Affordances in Games



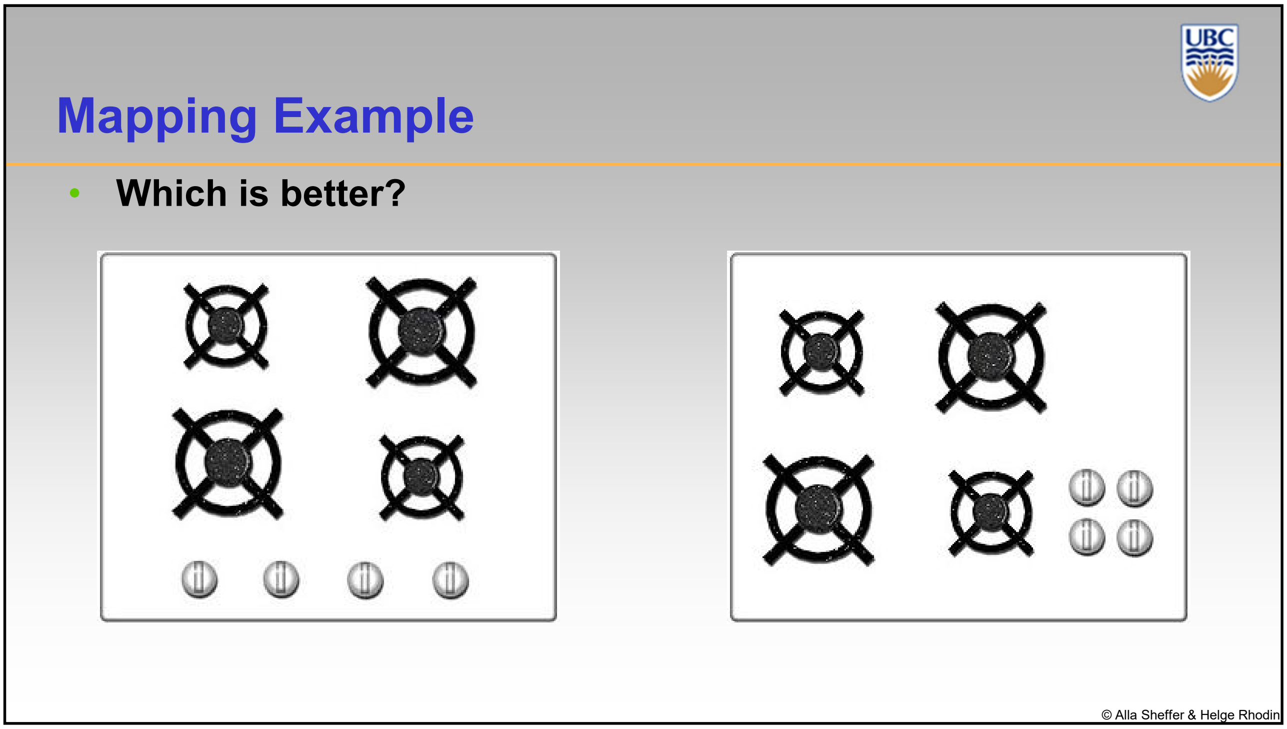
- What does the slingshot afford here?
- What do the blocks afford?
- What does the (pause) button afford?



Mapping

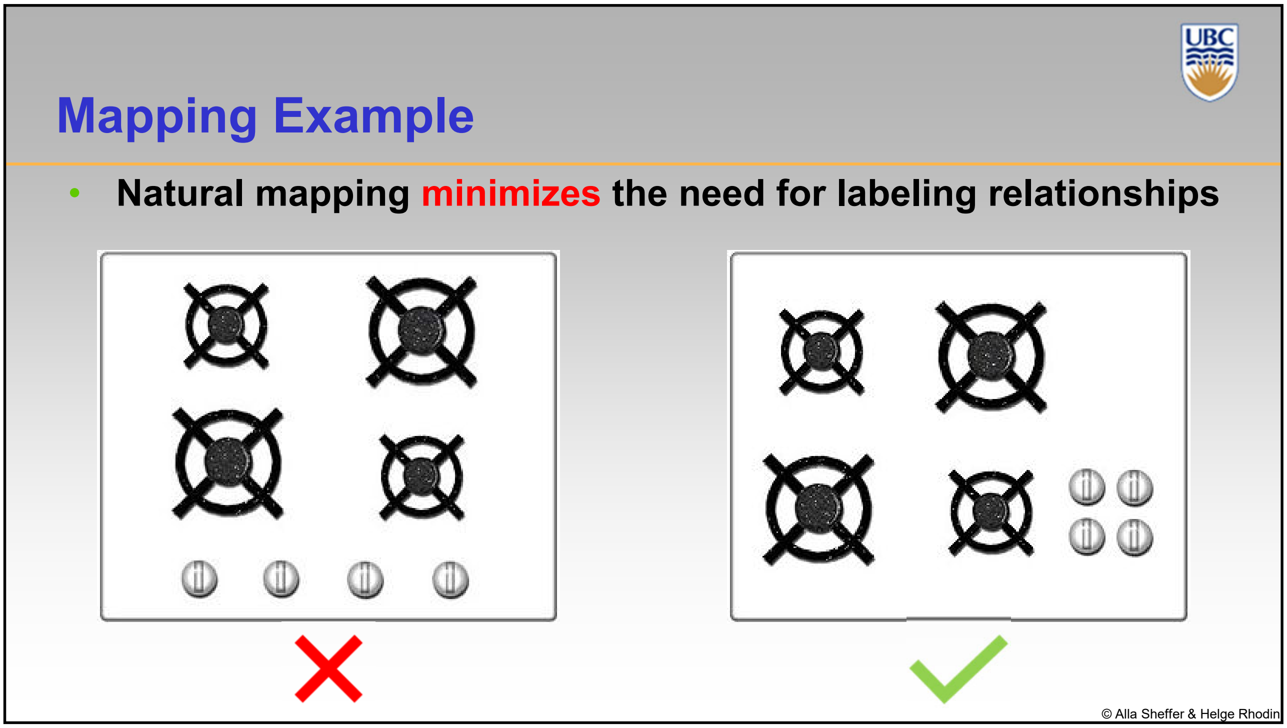
- Some controls are direct (slingshot), some indirect (button)
- **Mapping** is the relationship between look/feel of indirect **controls** & their implied **actions**

<u>Control</u>		<u>Implied action</u>
push button	→	start/stop function
twist knob	→	increase/decrease value
turn wheel	→	rotate left/right



Mapping Example

- Which is better?

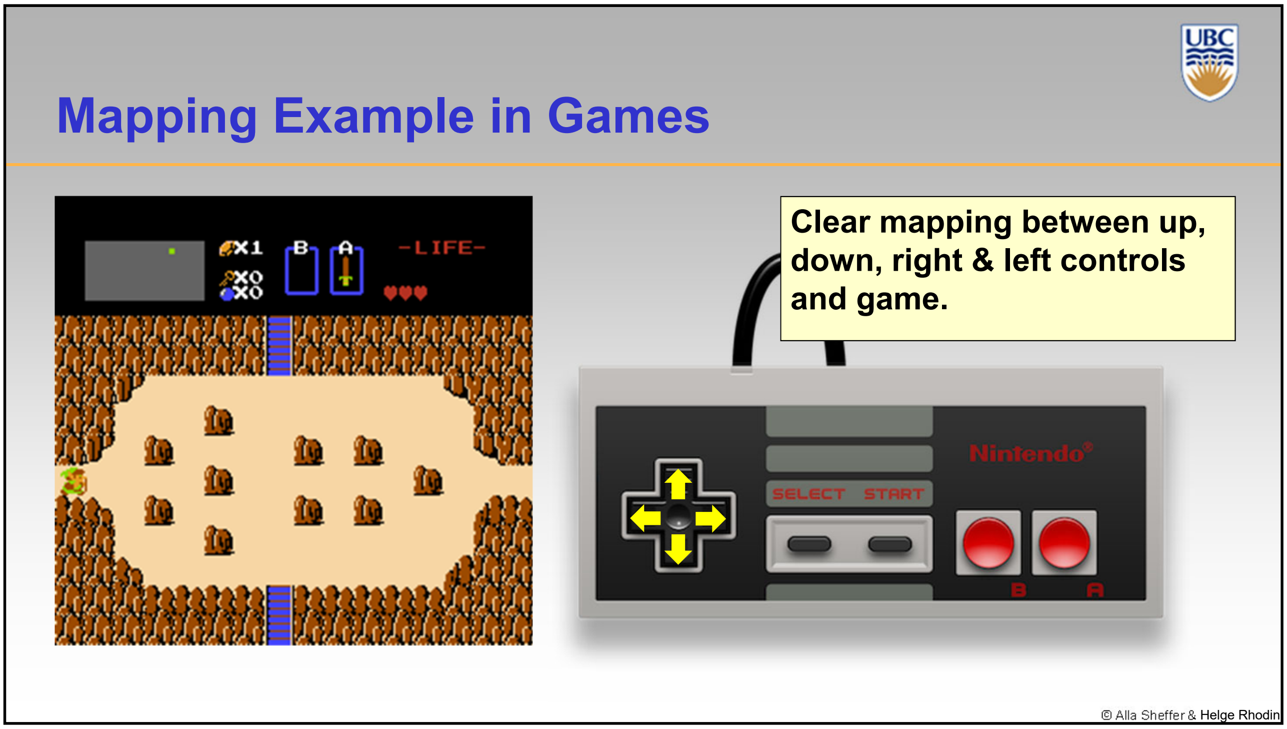


Mapping Example

- Natural mapping **minimizes** the need for labeling relationships

✗ ✓

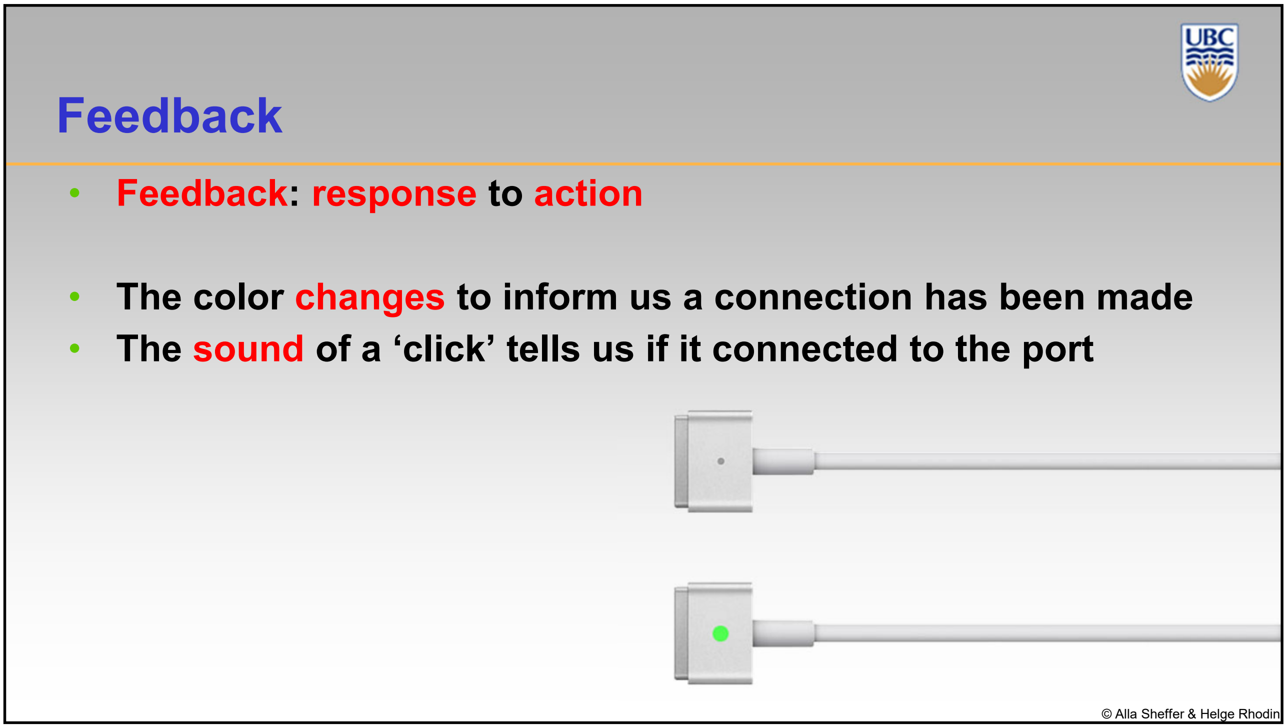
© Alla Sheffer & Helge Rhodin



Mapping Example in Games

Clear mapping between up, down, right & left controls and game.

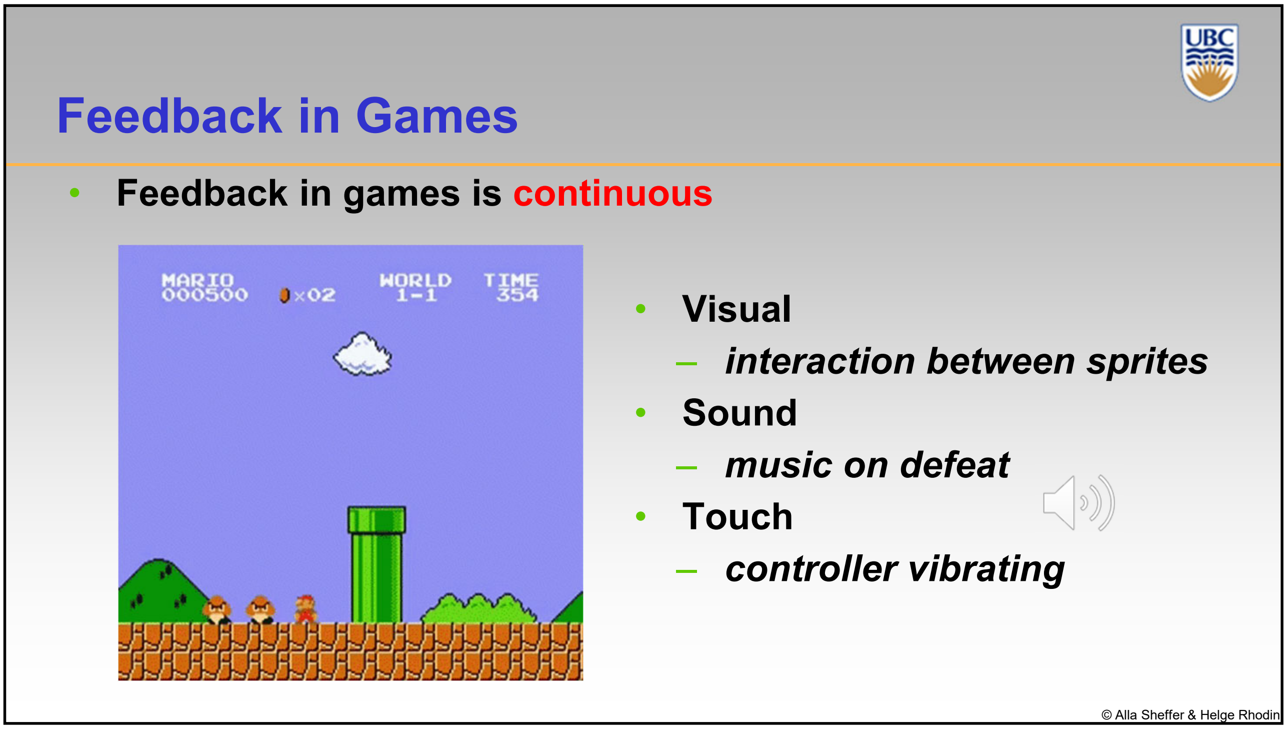
© Alla Sheffer & Helge Rhodin



Feedback

- Feedback: response to action

- The color changes to inform us a connection has been made
- The sound of a ‘click’ tells us if it connected to the port



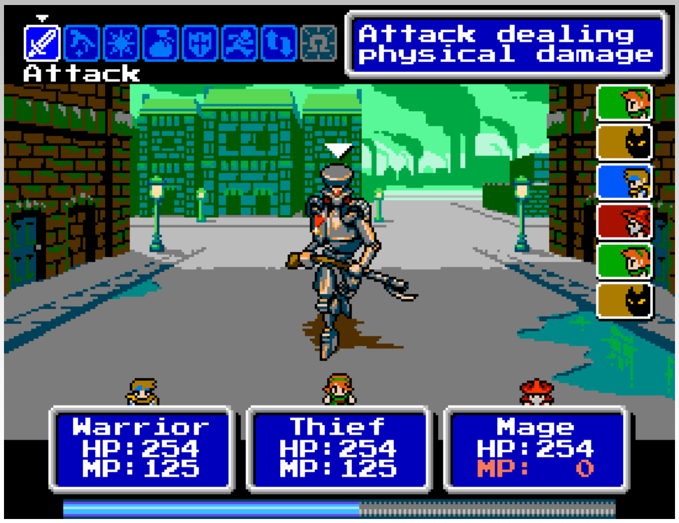
Feedback in Games

- Feedback in games is continuous

MARIO
000500
×02
WORLD
1-1
TIME
354

- Visual
  - *interaction between sprites*
- Sound
  - *music on defeat*
- Touch
  - *controller vibrating*

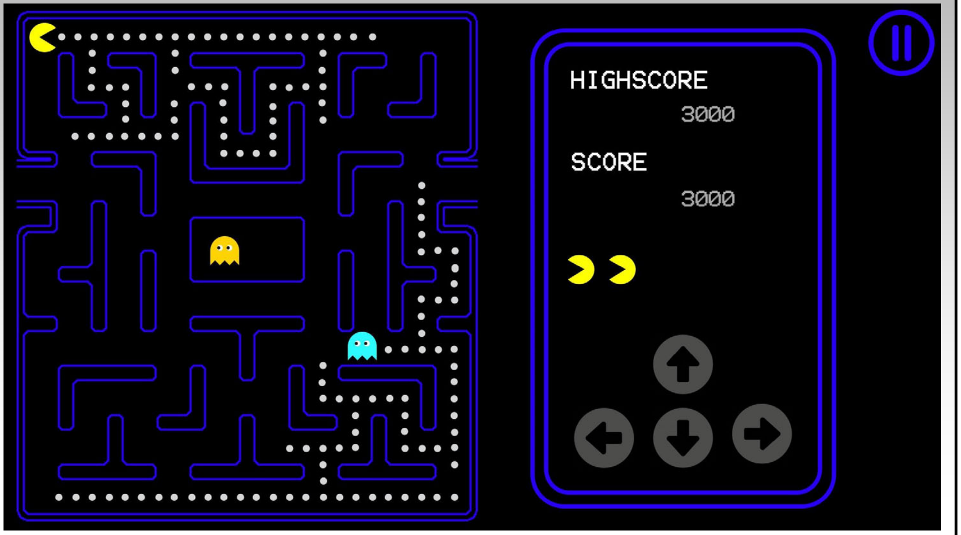
Design Principles Example in Games

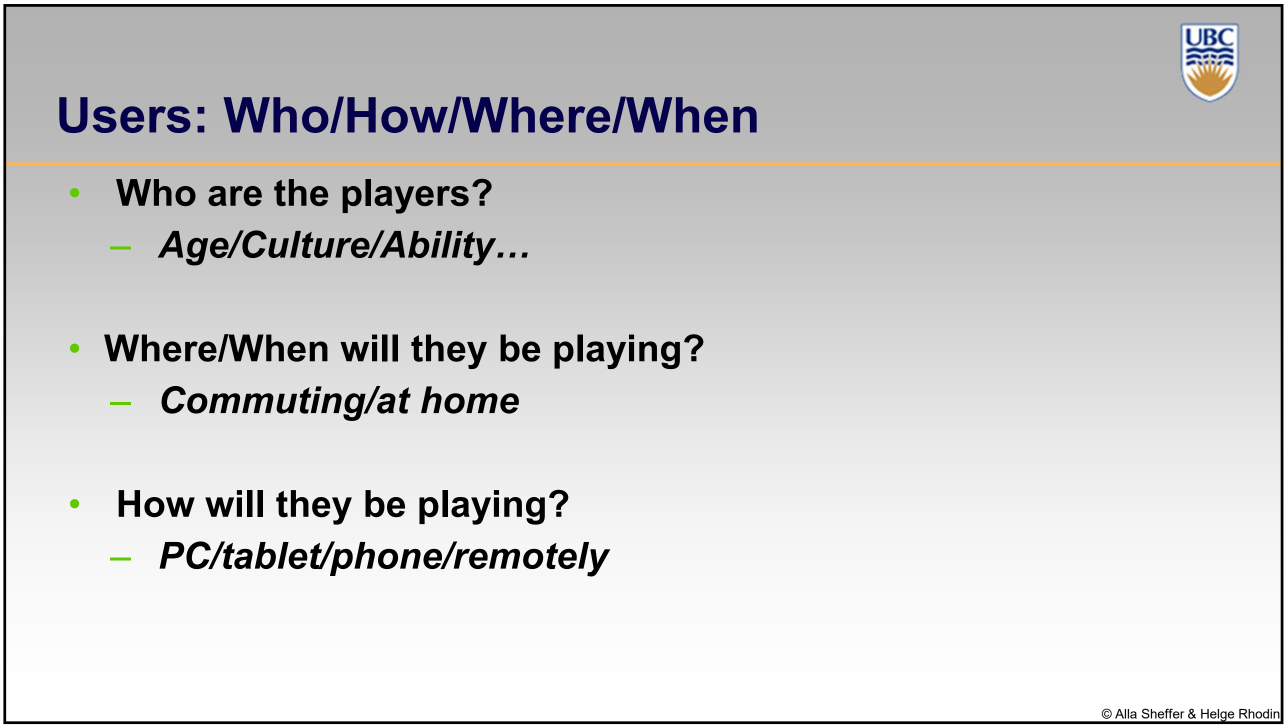


- Affordances?
- Mappings?
- Feedback?

Design Principles

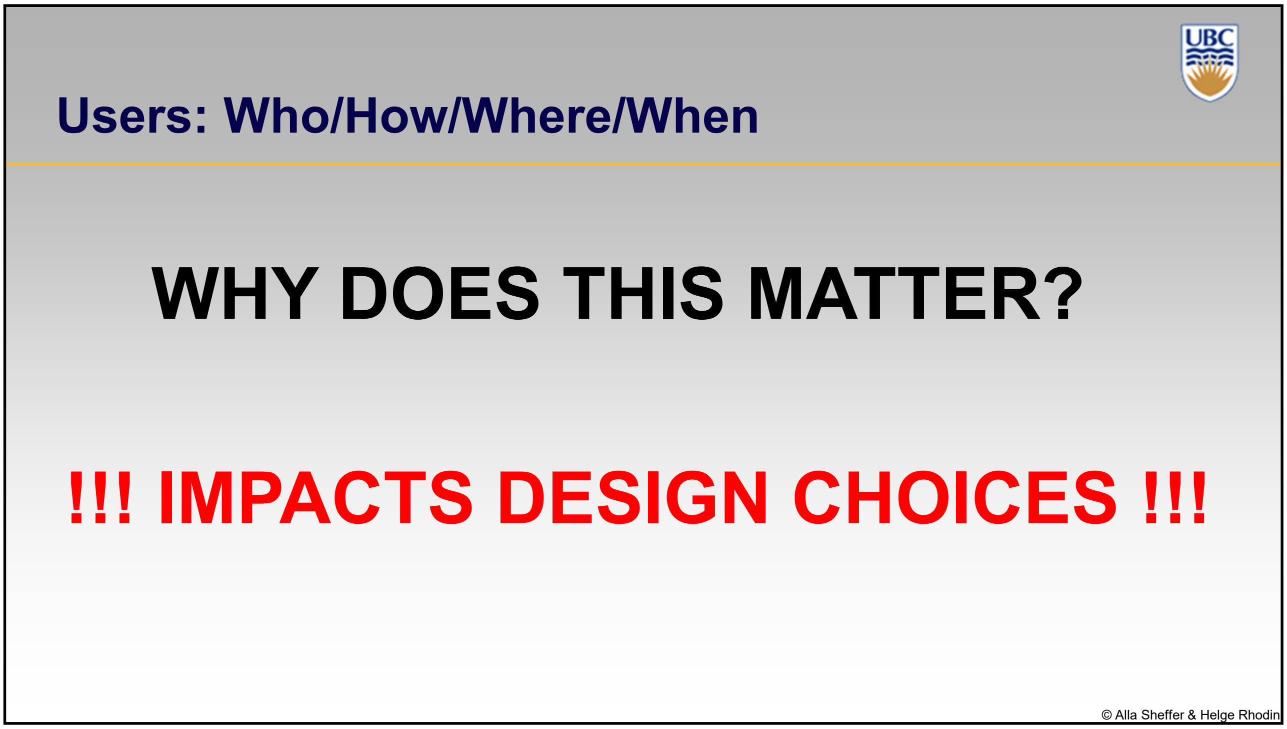
- Affordances
- Mapping
- Feedback





Users: Who/How/Where/When

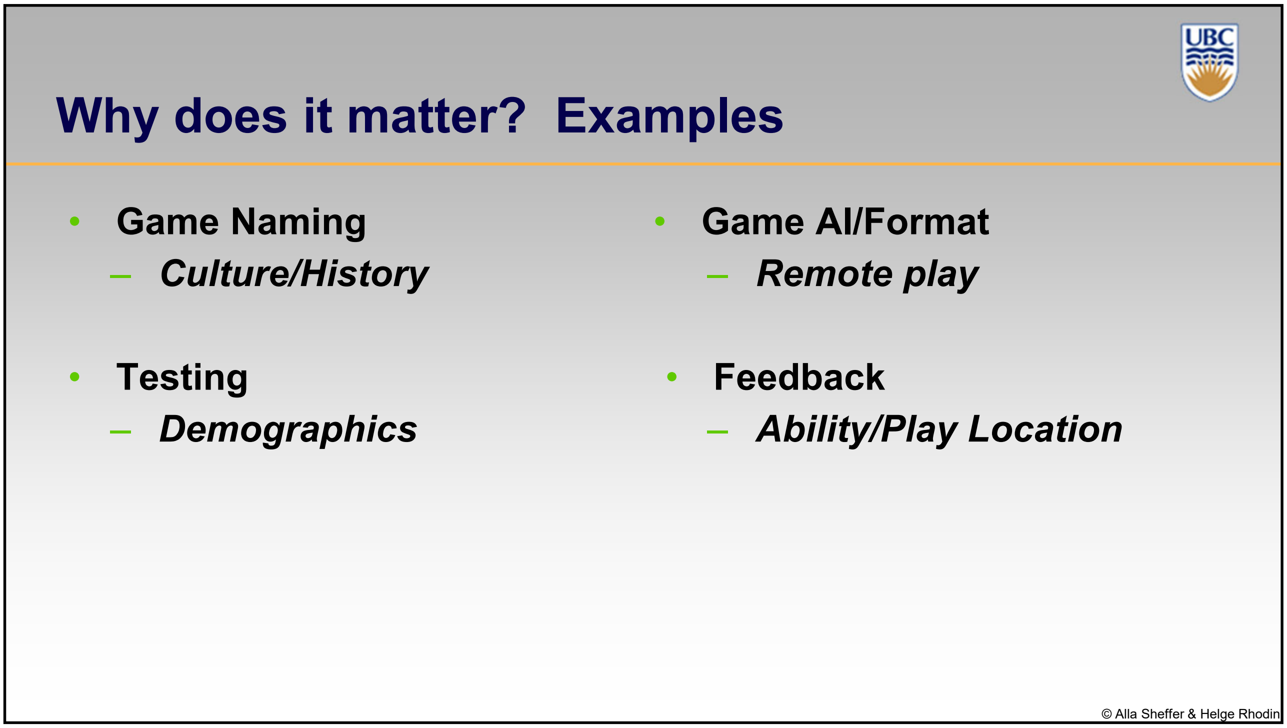
- Who are the players?
  - *Age/Culture/Ability…*
- Where/When will they be playing?
  - *Commuting/at home*
- How will they be playing?
  - *PC/tablet/phone/remotely*



Users: Who/How/Where/When

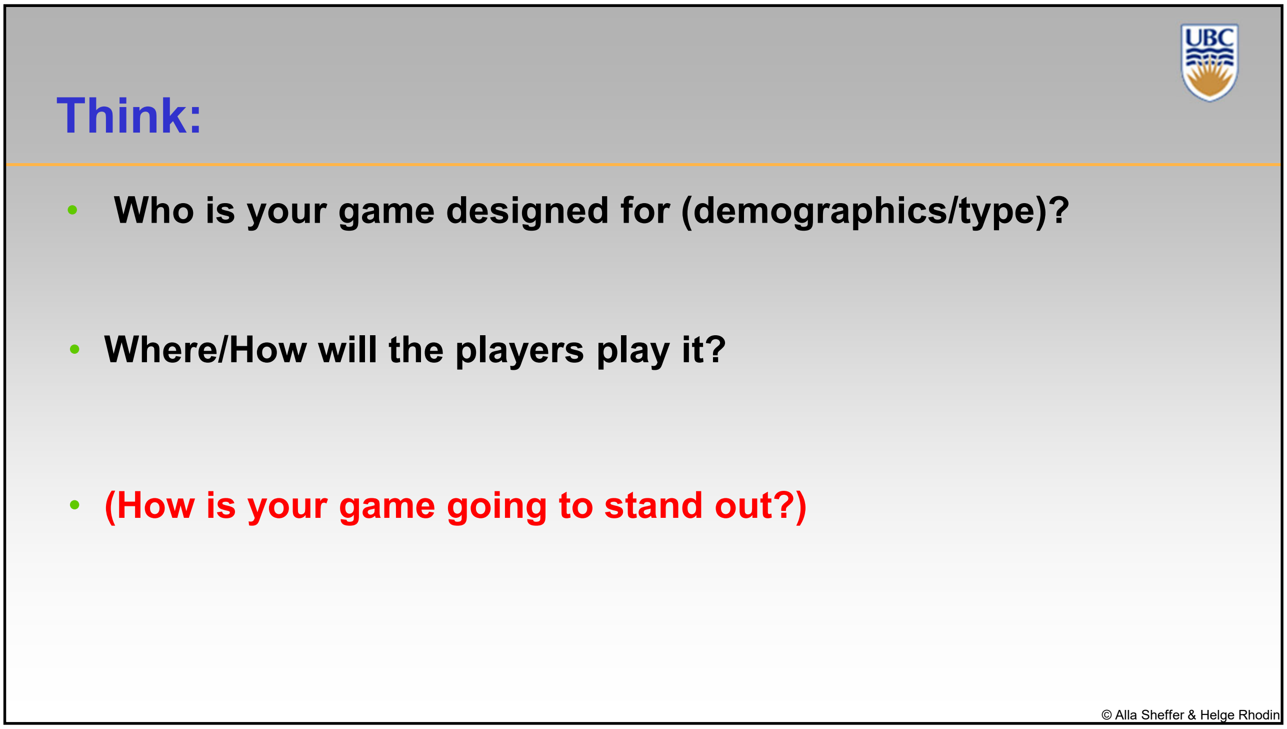
WHY DOES THIS MATTER?

!!! IMPACTS DESIGN CHOICES !!!



Why does it matter? Examples

- Game Naming
  - *Culture/History*
- Testing
  - *Demographics*
- Game AI/Format
  - *Remote play*
- Feedback
  - *Ability/Play Location*

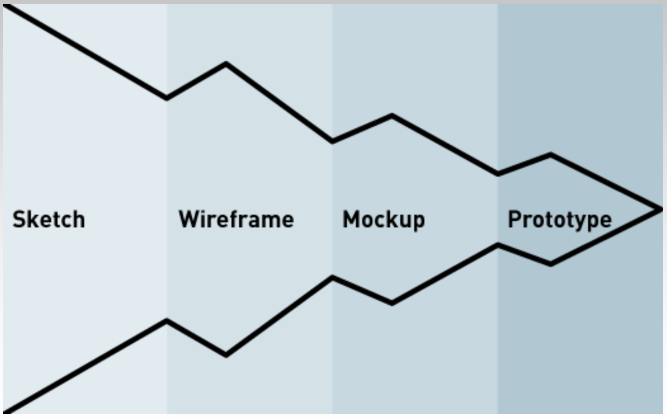


Think:

- Who is your game designed for (demographics/type)?
- Where/How will the players play it?
- (How is your game going to stand out?)

The Design Process

Brainstorming



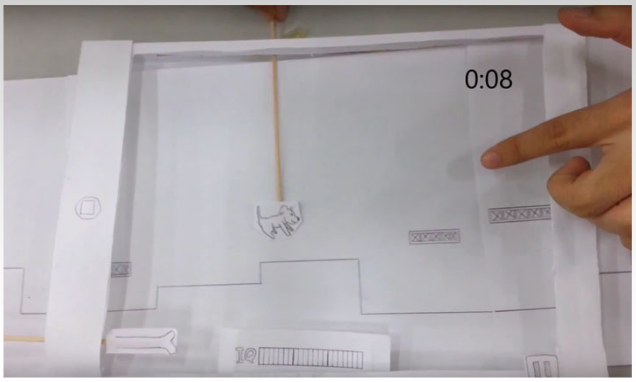
Release

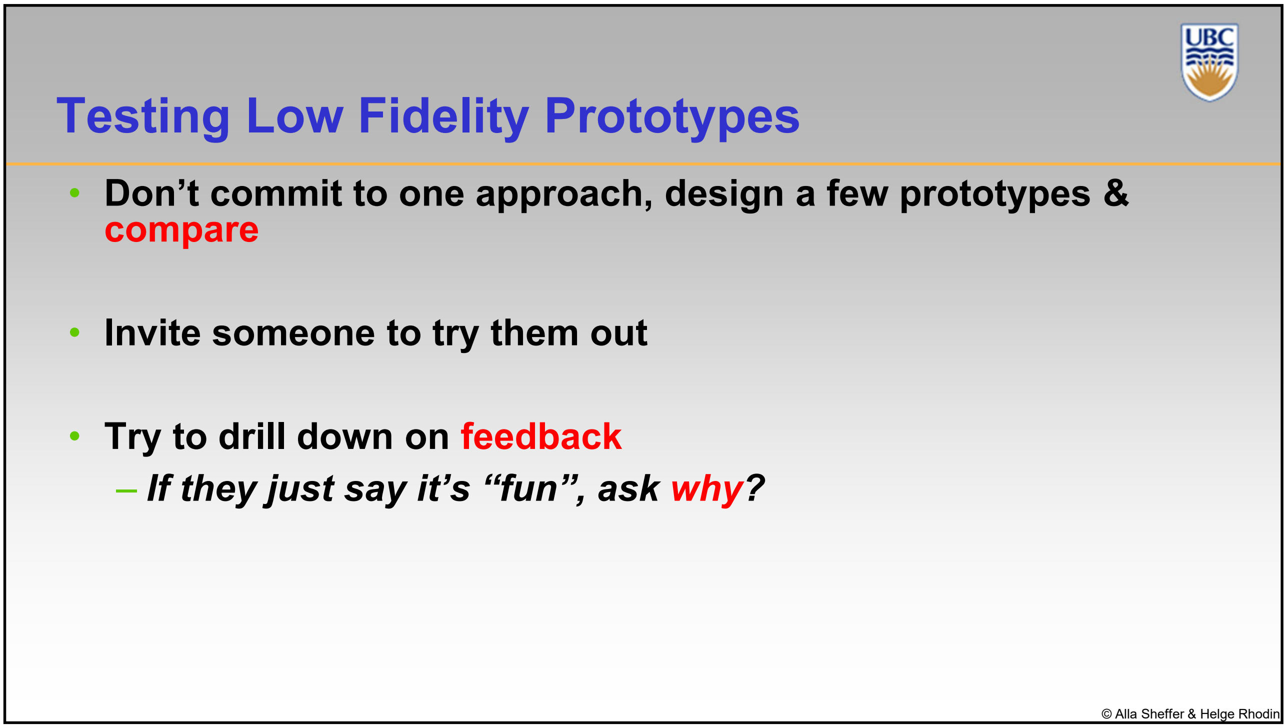
Low fidelity prototyping

High fidelity prototyping

Low Fidelity Prototyping

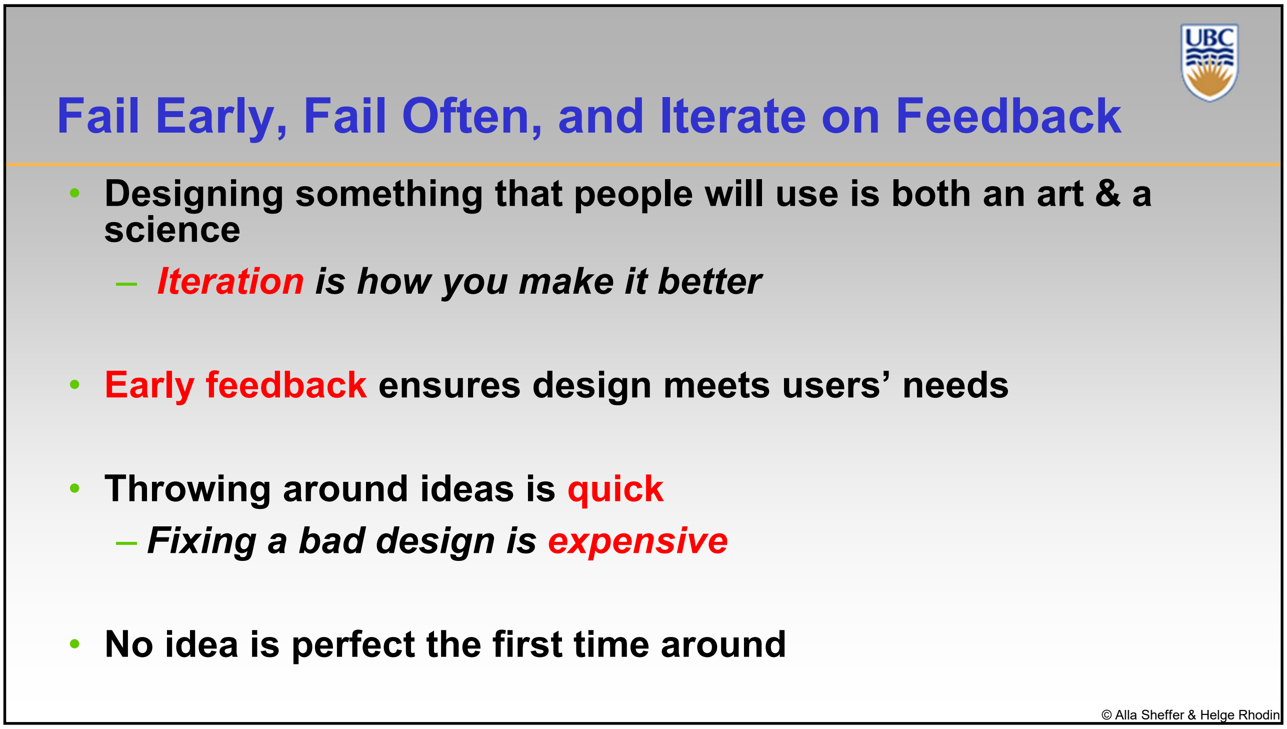
- Used for early stages of design
  - *Quick & cheap to deploy*
  - *Easy to test*
- Iterate on story and core gameplay mechanics
- Sketches are a great way to start designing





Testing Low Fidelity Prototypes

- Don’t commit to one approach, design a few prototypes & **compare**
- Invite someone to try them out
- Try to drill down on **feedback**
  - *If they just say it’s “fun”, ask **why**?*



Fail Early, Fail Often, and Iterate on Feedback

- Designing something that people will use is both an art & a science
  - ***Iteration** is how you make it better*
- **Early feedback** ensures design meets users’ needs
- Throwing around ideas is **quick**
  - *Fixing a bad design is **expensive***
- No idea is perfect the first time around

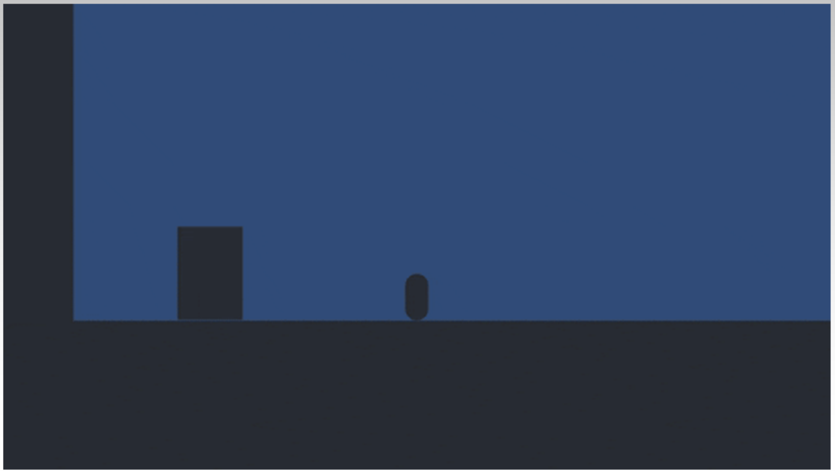
Medium Fidelity Prototyping

- Use medium fidelity prototyping for the early to middle stages of design
  - *Identify questions before coding*
  - *Be selective with what gets built*
  - *Get it right in black and white first*
- Iterate on tone & feel of game
  - *Supplementary game mechanics*
  - *Rough visuals & audio*
  - *Feedback*

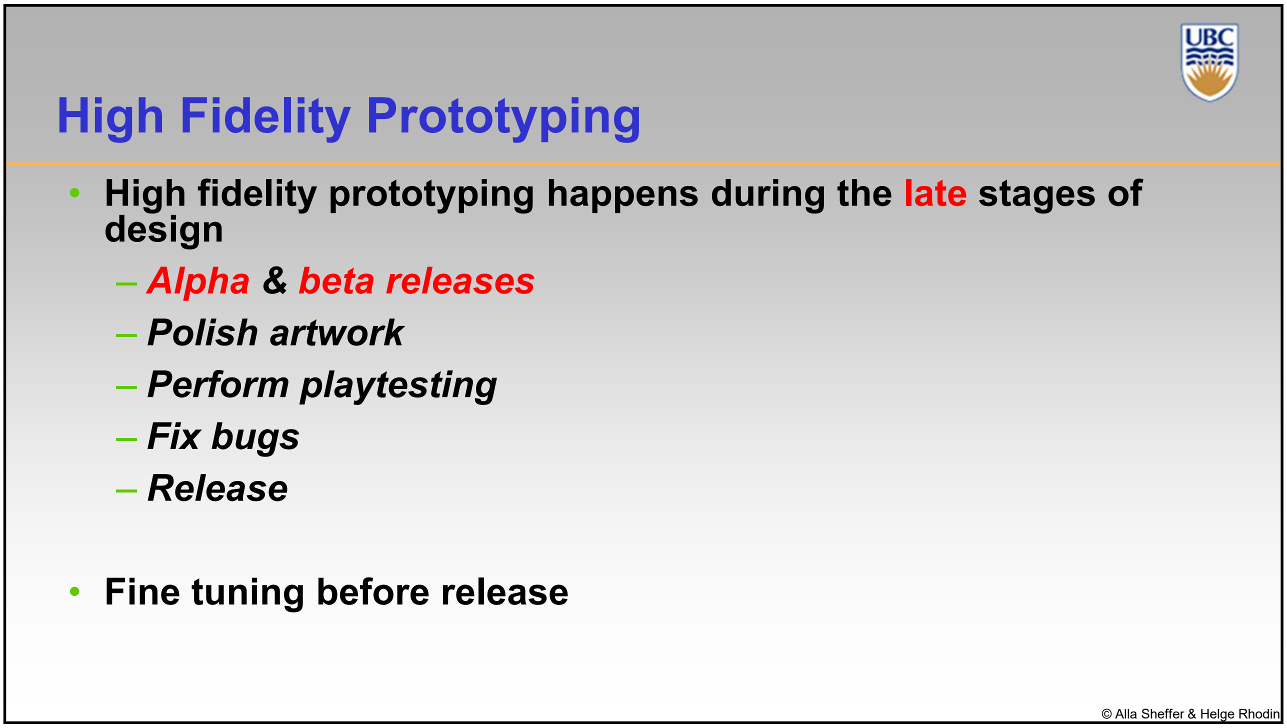
© Alla Sheffer & Helge Rhodin

Greyboxing

- Greyboxing blocks out all elements as shapes to test gameplay



© Alla Sheffer & Helge Rhodin



High Fidelity Prototyping

- High fidelity prototyping happens during the late stages of design
  - *Alpha* & *beta releases*
  - *Polish artwork*
  - *Perform playtesting*
  - *Fix bugs*
  - *Release*
- Fine tuning before release