CS/INFO 4154: Analytics-driven Game Design

Lecture 4:
Learnability and User Interfaces
The story so far…

Game Mechanics

Brainstorming

Paper Prototyping

**Nuts and Bolts**
- Learnability and User Interfaces
- Level Design and Progressions
- Architecture and Organization
- Artificial Intelligence
- Procedural Content Generation
- Art and Design

**Analytics**
- Internet Telemetry
- Data Analysis
- Data Visualization
- Large-scale Experimentation
- A/B Testing
- Multivariate Testing
Game design document

- Converge on one idea
- Plan for the development cycle
- Identify unknowns in the design
- Due Tuesday, September 15th, 11:59pm
Throwaway Prototype

- No pressure
- “Hello World!” of your game
- Doesn’t need to be playable
- Doesn’t need to be integrated
- Pick *some piece* of your game and build it
  - Avatar moves/jumps on flat land
  - Hexagonal grid with nothing on it
  - Background artwork
- In class on **Tuesday, September 22nd**
Pong / Asset Creation Graded

- 2 = complete
- 1 = partially complete
- 0 = no submission / didn’t work at all
- Resubmit by Thursday, September 17th, 11:59pm
Late Policy

- Some things cannot be turned in late
  - Releases
  - Presentations

- Other things: -25% per day and a guilt trip
  - Game Design Document
  - Revised Plans
  - Peer Evaluations
The goal: happy players
<table>
<thead>
<tr>
<th>Weapon Type</th>
<th>Damage</th>
<th>Range</th>
<th>Accuracy</th>
<th>Range Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Club</td>
<td>3</td>
<td>8</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>Goblin Shaman Staff</td>
<td>188</td>
<td>10</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>Iron Arrow</td>
<td>1</td>
<td>0.1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Rusty Iron Bow</td>
<td>10</td>
<td>8</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>Rusty Iron Dagger</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>Rusty Iron War Axe</td>
<td>4</td>
<td>12</td>
<td>1</td>
<td>100</td>
</tr>
</tbody>
</table>

Total Damage: 152/200
Learnability

the capability of a software product to enable the user to learn how to use it
How do you learn?
How do you learn?
How do you learn?
How do you learn?
What makes a game learnable?
The old days
SUPER MARIO 3
BROS.

INSTRUCTION BOOKLET
NEW TECHNIQUES!

Holding a shell
( Holding the B Button)

Running with a shell
( Holding the B Button)

Kicking the shell
(Releasing the B Button)

Breaking a block

When Mario has a tail

Accelerating
Power Meter going up

More acceleration
Meter full, (P) starting to flash

Take off
Press the A Button repeatedly

Mario can only fly for a short time.
Tap on a seed packet to pick it up!
Pick up the laser divider.
What is a user interface?

the means by which the user and a computer system interact
What makes a UI “good”? 
Critique this UI.
Critique this UI.
Critique this UI
Key Lesson of this Class #1

nobody reads and nobody listens
people can only keep track of a few things
Nielsen’s heuristics for UI design

1. Make system status visible
2. Match the real world
3. Provide control and freedom
4. Be consistent
5. Prevent errors when possible
Nielsen’s heuristics for UI design

6. Facilitate recognition rather than recall
7. Be flexible and efficient
8. Use minimalist design
9. Help users recognize and recover from errors
10. Provide help and documentation
1. Make system status visible
1. Make system status visible
1. Make system status visible
2. Match the real world
2. Match the real world
2. Match the real world
2. Match the real world
6. Recognition rather than recall

- User should *recognize* what’s going on
- Shouldn’t have to *recall* a tutorial
6. Use recognition rather than recall
6. Use recognition rather than recall
3. Provide control and freedom
3. Provide control and freedom
3. Provide control and freedom

Even from back here, I can tell that the ledge ahead is too high for me to jump.
3. Provide control and freedom

Even from back here, I can tell that the ledge ahead is too high for me to jump.
3. Provide control and freedom
3. Provide control and freedom
8. Use minimalist design

- Avoid clutter
- Provide *only* critical information
8. Use minimalist design
8. Use minimalist design
8. Use minimalist design
5. Prevent errors
5. Prevent errors
5. Prevent errors
5. Prevent errors
9. Help users recover from errors
9. Help users recover from errors

I remember that the R key will let me reset the current level to try again.
NEW TECHNIQUES!

Holding a shell → Running with a shell → Kicking the shell → Breaking a block
(Holding the B Button) (Holding the B Button) (Releasing the B Button)

When Mario has a tail

Accelerating → More acceleration → Take off → Mario can only fly for a short time.
Power Meter going up Meter full, (P) starting to flash Press the A Button repeatedly
Nielsen’s heuristics for UI design

1. Make system status visible
2. Match the real world
3. Provide control and freedom
4. Be consistent
5. Prevent errors when possible
6. Facilitate recognition rather than recall
7. Be flexible and efficient
8. Use minimalist design
9. Help users recognize and recover from errors
10. Provide help and documentation
Group Work

- Pick one of your game ideas
- Pick two mechanics in your game
- Brainstorm a plan for how the user will learn these mechanics
- Design a level with a UI (and tutorials?) that will support this learning
- Show to another team