CS 352
Introduction to Usability Engineering

Foundations & Strategies

Surprise-Explain-Reward

Attention Investment

- Usual computer science view:
  - "If we build it they will come"
- But why should they?
  - Cost of new feature: learning it + cost of interacting with it
  - Benefit of new feature: not clear without incurring the cost
  - Risks: wasted time (cost), getting environment into an odd state from which they can't easily recover

How to Interest Them:

Arouse Curiosity

- Research in curiosity also suggests –
  - Showing them the presence of an “information gap” makes them curious

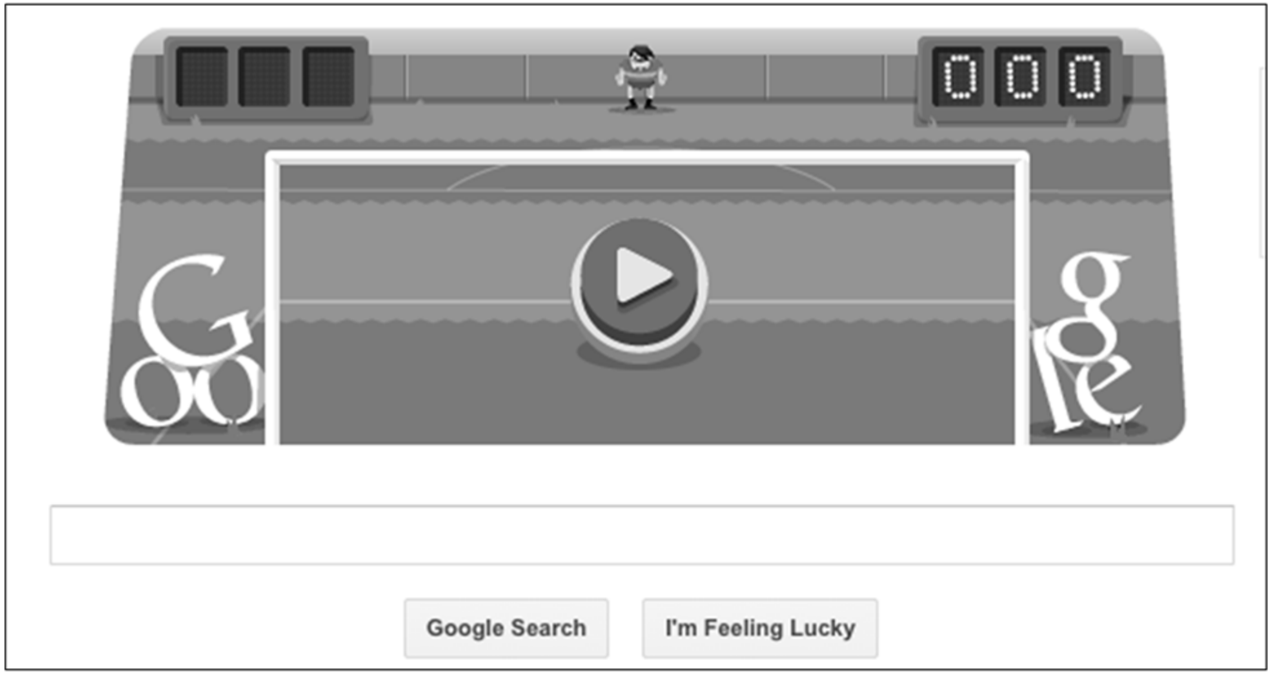
Our Strategy:

Surprise-Explain-Reward

- Surprise: shows them the presence of information gap, to arouse curiosity
- Explain: user seeks explanation to close the information gap
  - Self-directed learning (a key attribute)
  - Suggests the actions we want them to do
- Reward: make clear the benefits of taking those actions early

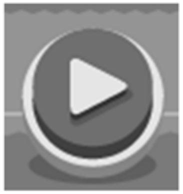
Example 1: Google

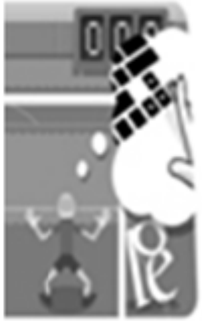
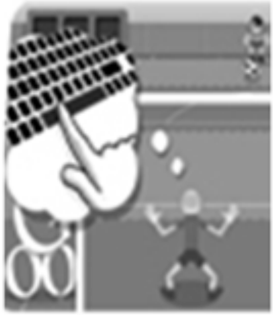
- Imagine my surprise when I saw this:



Example 1: Google

- When I did the only thing I could do to seek an explanation:



- …it said  and then 

- These “explanations” told me:
  - It was a game (implies the reward)
  - How to play it

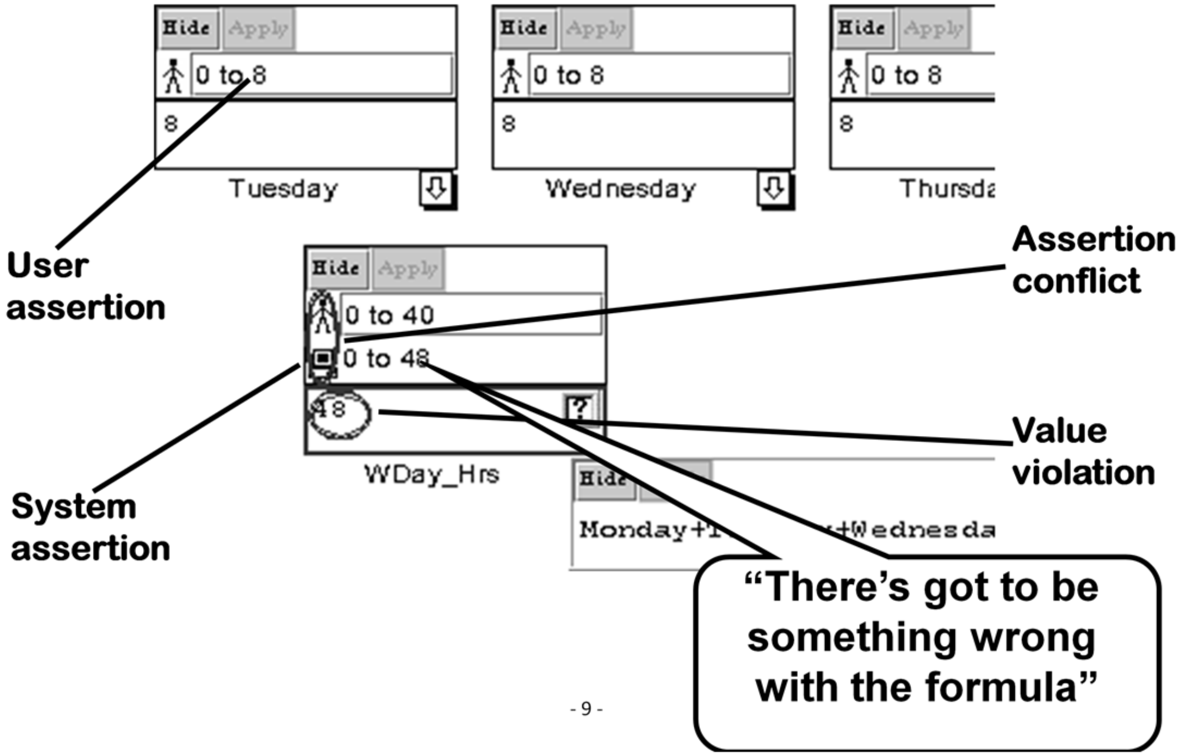
Example 1: What to Notice

1. Did not interrupt me
   - Surprised me only when I asked it to do something
2. Surprise was effective at arousing my curiosity
   - I did seek an Explanation
3. Explanation was effective at making reward clear and telling me how to get it
   - I played the game and it made me smile

Example 1: Reflection

- Google continuously replaces their logo with a themed logo, sometimes interactive, occasionally a game
- I've heard people say it's tiresome and they don't pay attention
- Question-
- Does repetition reduce the curiosity?
  - – Think back to our discussion of perception…

Example 2: Assertions in Spreadsheets



User assertion

System assertion

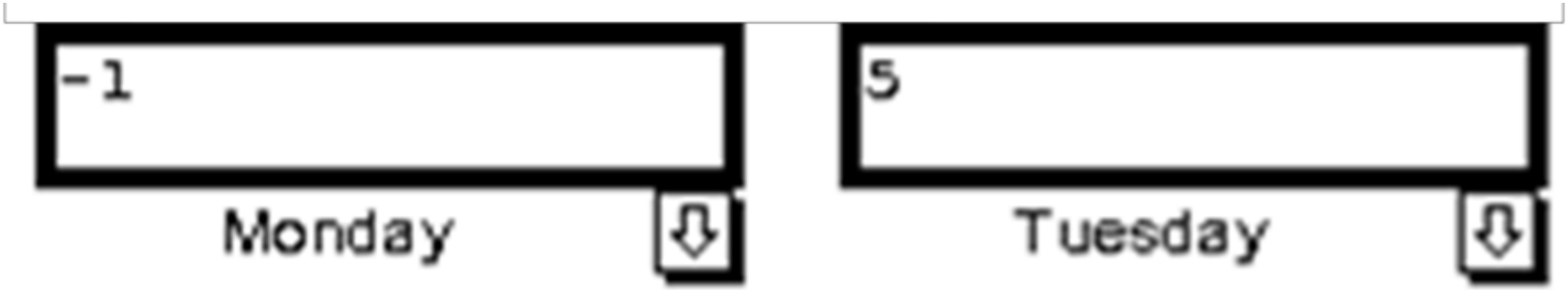
Assertion conflict

Value violation

"There's got to be something wrong with the formula"

Example 2 (cont)

1. User invokes “help me test” to get it to suggest test values
2. Surprise: Assertions tool
   - Reveals the information gap
   - These are deliberately bad guesses

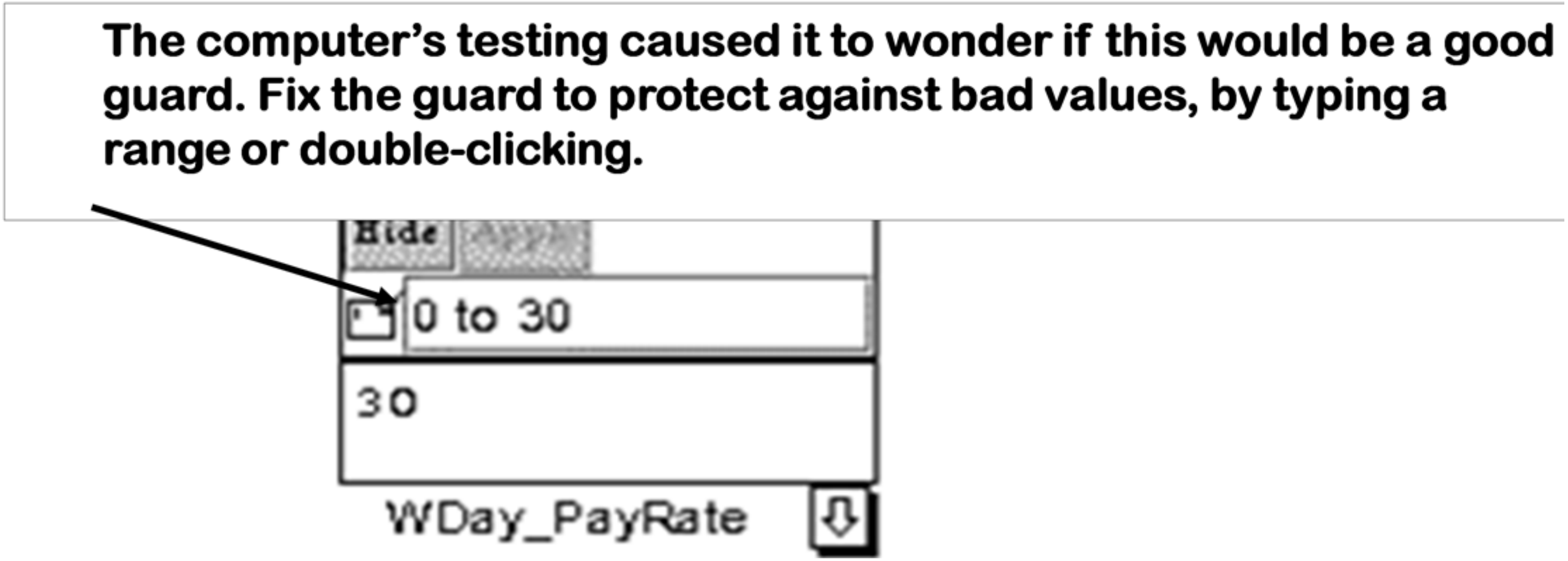


Example 2: Surprises

- All surprises use negotiated style interruptions, not immediate style
  - Surprises only when user has asked it to display something
  - Attempts to earn user's attention but does not require it

Explanation System Principles

- Semantics, reward, suggested action
  - – With enough info to succeed at the action



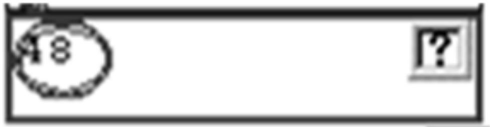
The computer's testing caused it to wonder if this would be a good guard. Fix the guard to protect against bad values, by typing a range or double-clicking.

Example 2: Explanation System

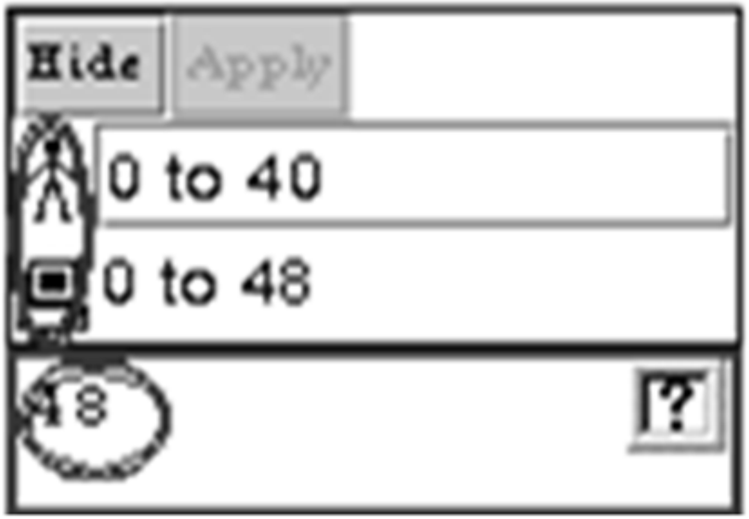
- Why explanations' viewpoint is via tool tips
  - Psychology: users seek explanation from the surprising object
- Why suggested action
  - Minimalist learning: get users to take action
- Why reason/reward
  - Attention investment: must make rewards clear

Example 2: Rewards

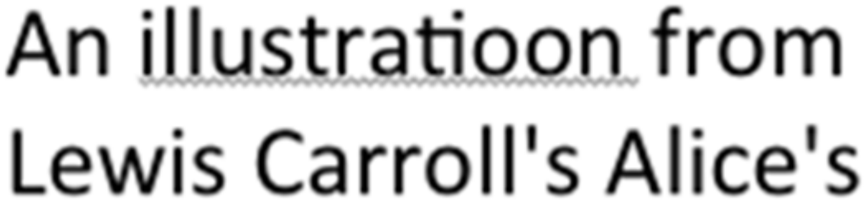
- Red circles around values indicate either bugs or incorrect assertions



- Computer-generated assertions might “look wrong”
- Red circles around conflicting assertions
- Are first surprises, then rewards



Example 3: in Word Processing



An illustratioon from
Lewis Carroll's Alice's

- Surprise and not interrupting me
- Explain: none, but it's clear how to fix it and what reward will be
- Reward: spelling errors gone!

For Your Projects?

1. Find circumstances where you can introduce without interrupting
2. Present a surprise to arouse curiosity
3. Explain should be available at the surprising object, and make obvious:
   - Why to proceed (what's the reward?)
   - How to proceed
4. There should be a genuine reward (saved time, etc.) not just a gold star