

RESPONSIVENESS BLOOPERS GUIB CH 7

REKIK ZIKU

2 OVERVIEW

Understand Responsiveness



Bloopers



Reasons



Avoiding Bloopers – Design Principle



Avoiding Bloopers – Techniques

3 WHAT IS RESPONSIVENESS?

- Responsiveness \neq Performance
- Responsiveness is how quickly an app acknowledges and indicates acceptance of user input
- Performance is how quickly the software computes and displays *results*.
 - High-performance = Quick Results
 - Low Performance = Slow
- Can be highly responsive but slow



Copy



Copying 506 items to "Desktop"



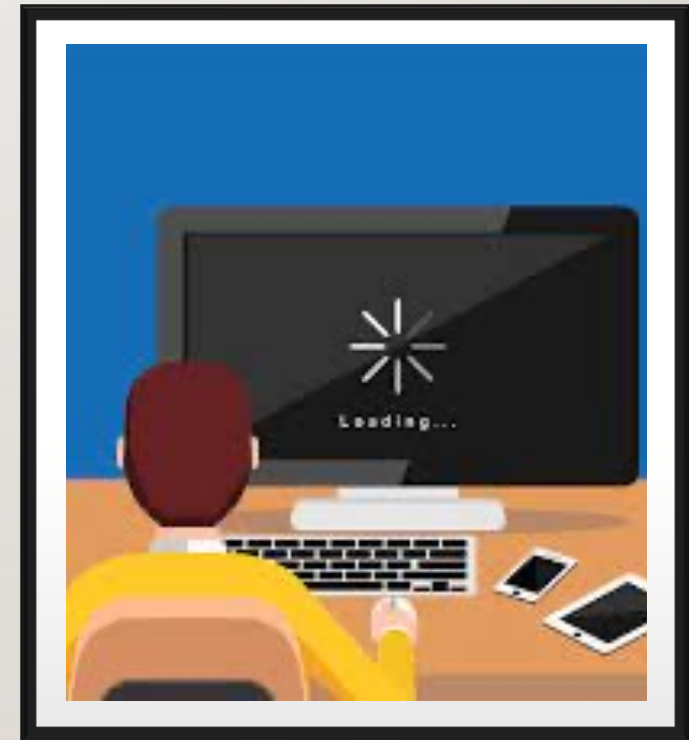
11.9 MB of 35.9 MB – Less than a minute

HIGHLY RESPONSIVE SOFTWARE

- lets you know immediately that your keystrokes, pointing device movements, and clicks were received;
- estimates how long operations will take;
- frees you to do other things while waiting;
- manages queued events intelligently;
- performs housekeeping and low-priority tasks in the background;
- anticipates your requests.

5 BLOOPERS 52 - 57

- Blooper 52: Cursor doesn't keep up with you
- Blooper 53: buttons acknowledge clicks too late or not at all
- Blooper 54: Menus, sliders, and scrollbars lag behind your actions
- Blooper 55: Moving and sizing operations don't keep up with your actions
- Blooper 56: Application doesn't indicate that it is busy; it just ignores you
- Blooper 57: Application occasionally—and unpredictably—is unresponsive while it does internal housekeeping



6 BLOOPERS 58 - 63

- Blooper 58: Long operations don't display progress
- Blooper 59: Long operations provide no way to cancel
- Blooper 60: Application wastes idle time, and when you finally give a predictable command, it takes a long time to finish
- Blooper 61: Application gives no feedback when it hangs, with no indication of what is or is not happening
- Blooper 62: huge images and animations viewable only with a super-high-speed Internet connection
- Blooper 63: Web site always reloads whole pages in response to small edits



7 REASONS FOR POOR RESPONSIVENESS

Developers and development managers don't know how important responsiveness is

UI designers rarely consider responsiveness during design

Programmers believe responsiveness = performance

Programmers treat user input like machine input

Developers use simple implementations

GUI software tools, components, and platforms are inadequate

Managers hire GUI programmers who lack the required skill

8

AVOIDING RESPONSIVENESS BLOOPERS

DESIGN PRINCIPLES

9 RESPONSIVENESS \neq PERFORMANCE!!!

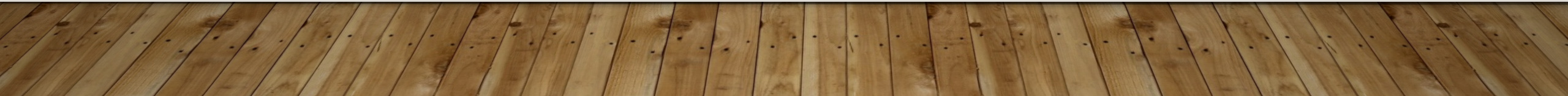
PROCESSING RESOURCES ARE ALWAYS LIMITED

THE USER INTERFACE IS A REAL-TIME INTERFACE

10 ALL DELAYS ARE NOT EQUAL: SOFTWARE NEED NOT DO EVERYTHING IMMEDIATELY

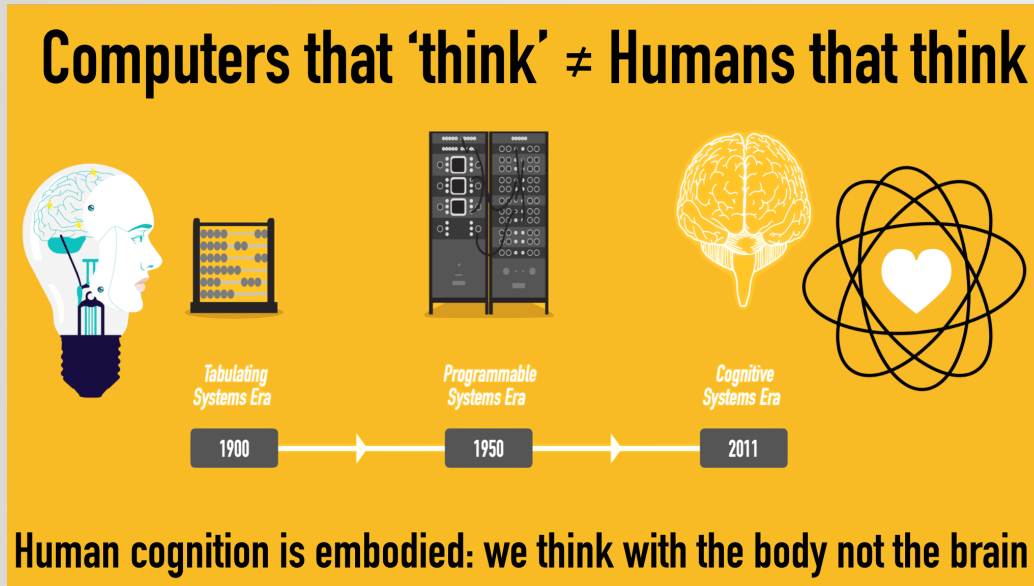
SOFTWARE NEED NOT DO TASKS IN THE ORDER IN WHICH THEY WERE REQUESTED

SOFTWARE NEED NOT DO EVERYTHING IT WAS ASKED TO DO





HUMAN USERS ARE NOT COMPUTER PROGRAMS



PEOPLE ARE NOT SINGLE-CHANNEL
INPUT/OUTPUT DEVICES. PEOPLE CAN DO
SEVERAL THINGS IN PARALLEL

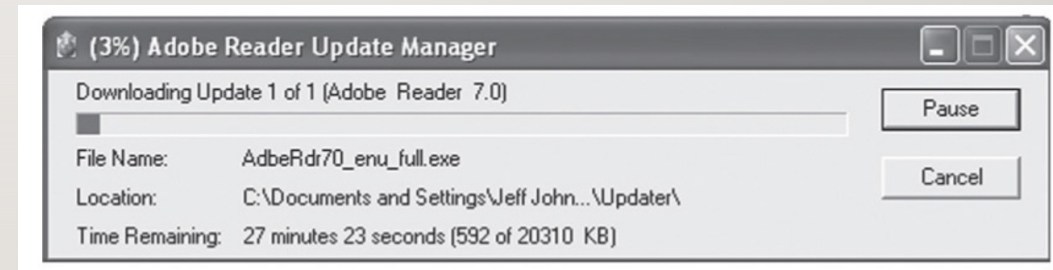
12

AVOIDING RESPONSIVENESS BLOOPERS

TECHNIQUES

13 TIMELY FEEDBACK

- **Acknowledge user input immediately**
- **Provide busy indicators**
- **Display progress indicators for long operations**



14 PARALLEL PROBLEM SOLUTION



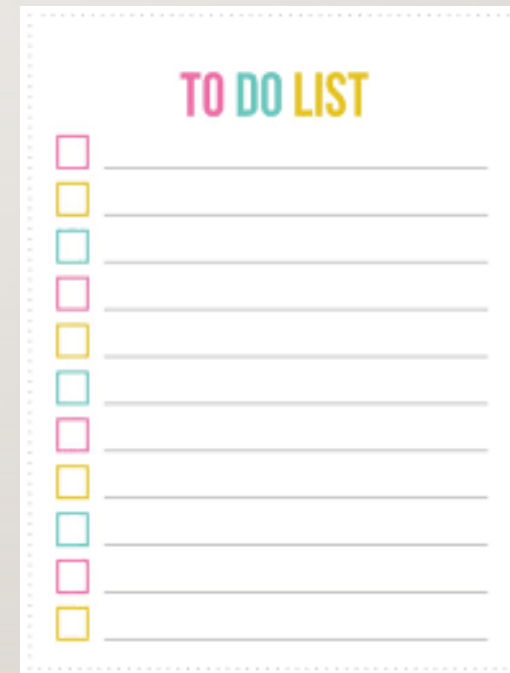
**DELAY NONCRITICAL
WORK**



WORK AHEAD

15 QUEUE OPTIMIZATION

- review the to-do list periodically to decide what tasks to do and in what order



16 DYNAMIC TIME MANAGEMENT

- **Monitor event queue**
- **Monitor time compliance**
- **Predict completion time**
- **Predict time compliance**



Copy



Copying 506 items to "Desktop"



11.9 MB of 35.9 MB – Less than a minute

HIGHLY RESPONSIVE SOFTWARE

- lets you know immediately that your keystrokes, pointing device movements, and clicks were received;
- estimates how long operations will take;
- frees you to do other things while waiting;
- manages queued events intelligently;
- performs housekeeping and low-priority tasks in the background;
- anticipates your requests.

18

THANK YOU!

ANY QUESTIONS?

