

C

CLEAR

Be discernible quickly

Make the state of the app more apparent

pg 85: Ensure that users can tell if the system is ready • Make modes more obvious • Give a clear indication of which item will be acted on • Give users a clear sense of progress

Give obvious and timely feedback

pg 90: Give feedback as quickly as you can • Be sure to send the right message • Leverage signal strength to make feedback clearer • Use previews as a form of advance feedback

Provide a distinct sense of context

pg 95: Use easily discernible structure and indicators • Ensure that users know where they are, what's ahead, and how to get back

U

UNOBSTRUCTED

Get out of the way

Reduce unnecessary delays

pg 103: Eliminate blocking transitions • Hunt down stutters and pauses • Let users leave the guided tour • Avoid controls that restrict a user's speed

Improve speed and comfort during unavoidable delays

pg 108: Leverage batching, queueing, watchers, and schedulers • Let users work in parallel during blocking tasks • Let users entertain themselves

Work with human movement

pg 113: Accommodate for physical limitations • Speed things up with multiple simultaneous inputs • Help users avoid switching input methods • Let users keep their eyes on their work

Make direct paths

pg 118: Make it possible to take shortcuts • Ensure that the important stuff is visible • Connect the dots to what users really want • Let users recycle their work

P

PREDICTABLE

Make it possible to anticipate things

Create stable targets

pg 127: Use consistent locations for actions • Eliminate sudden layout shifts • Ensure that you don't intercept your users' inputs

Make inputs more consistent

pg 132: Treat user input the same every time • Use more reliable commands • Make use of physical controls

Produce dependable outcomes

pg 137: Offer paths to reliable results • Warn before removing capabilities

I

INDULGENT

Accept sloppier input

Leverage Fitts's Law

pg 144: Use larger targets • Put things closer • Simplify inputs and gestures

Create more tolerant inputs

pg 148: Infer what users mean • Watch out for time-limited interactions • Let users work with and around constraints • Allow for mind-changing

D

DIGESTIBLE

Reduce mental strain

Let users stay mentally calibrated

pg 154: Keep things in context to avoid "virtual doorways" • Let users keep their momentum • Keep an eye out for distractions • Watch out for signal overload

Cut down on the need to remember

pg 160: Allow for and provide contextual clues • Make mappings familiar • Offer direct comparisons • Watch out for behavior that defies expectations

Do the mental heavy lifting

pg 165: Allow for sorting and filtering • Summarize with context

Help users avoid pain

pg 169: Keep mental states in mind • Introduce friction as "guardrails" where necessary • Free your users from fear

Tickle the mind with comfort

pg 174: Improve aesthetics and accommodate individual preferences • Provide flexibility where you can • Help users feel oddly satisfied

How to use this

This *CUPID Cheat Sheet* helps you find sources of effort that stick around permanently in an interface.

Think of this as a pre-flight checklist. Go one column at a time asking "how might we" questions. For example, for the 2nd column (U), you could start by asking "How might we reduce unnecessary delays?"

Beneath this, you'll find a list of various tactics for achieving this objective and a page # for more info.