Crafting Well-Designed Sites

Jake Zimmerman

November 5, 2016

Overview

Before we begin

Typography

Information Hierarchy

Case Study: The Joys of the Craft

Resources & Recommended Readings

Workshop

I have a confession to make.

I’m not a designer;

I’m an engineer.

But, I design.

We'll be looking at some **very practical tips** you can put use to make your sites well-designed.

In particular, we're going to be looking heavily at

- typography, and
- information hierarchy

and how they come together to yield "good design."

Acknowledgements

80% of the content here comes from:

- Practical Typography, by Matthew Butterick
- 05-391 Interaction Design Overview, by Karen Berntsen & Jenna Date

Overview

Before we begin

Typography

Information Hierarchy

Case Study: The Joys of the Craft

Resources & Recommended Readings

Workshop

Typography and grid dominate web design

Everything else on the web is supported by typography and grid:

- Color palettes
- Iconography
- User interfaces
- Experience design
- Interaction design

Typography is the visual component of the written word

Good design is both visual **and** verbal.

Consider: would you go to your first interview in sweat pants?

- No, you'd dress presentably.
- It's not **just** about what you say at the interview.

Practical Typography

We're going to look at Matthew Butterick's *Typography in Ten Minutes*:

Five simple rules to capture 70% of good typography.

Rule 1

The four most important typographic choices you make in any document are

- point size,
- line spacing,
- line length,
- and typeface,

because those choices determine how the body text looks.

Rule 2

Point size should be

- **10–12 points** in printed documents,
- **15-25** pixels on the web.

Rule 3

Line spacing should be **120–145%** of the point size.

Rule 4

The average line length should be

- **45–90 characters** (including spaces), or
- **2-3 alphabets**.

For example:

abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabc

Rule 5

The easiest and most visible improvement you can make to your typography is to **use a professional font**.

Don’t know where to start?

- I recommend Typewolf,
  - especially, his recommendation lists.

Overview

Before we begin

Typography

Information Hierarchy

Case Study: The Joys of the Craft

Resources & Recommended Readings

Workshop

Thoughtful information hierarchy makes navigation sensible

***progressive disclosure**:*
an interaction design technique to help maintain the focus of a user's attention by reducing clutter, confusion, and cognitive workload.

Where should they look…

- first?
- second?
- last?

Practical Information Hierarchy

We're going to look at Karen Berntsen and Jenna Date's three rules for designing information hierarchy.

For clarity, I'm going to number them 6, 7, & 8.

Rule 6

Every design must have a clear **point of entry**.

Rule 7

Careful attention to **progressive disclosure** can make even the most complex content easier to navigate.

Rule 8

Using whitespace intelligently joins the visual & verbal elements of your design.

Designing the *negative* spaces of your design is as crucial as designing the actual content.

Overview

Before we begin

Typography

Information Hierarchy

Case Study: The Joys of the Craft

Resources & Recommended Readings

Workshop

The Joys of the Craft

We're going to take a look at "The Joys of the Craft," an excerpt from Fred Brooks' book *The Mythical Man-Month*.[1]

Specifically, we're going to focus on the **typographic** and **hierarchical** elements of this piece of body text.

[1] Read the full excerpt or the book.

Rule 2: Point size

The Joys of the Craft
Frederick P. Brooks Jr.

Why is programming fun? What delights may its practitioner expect as his reward?

First is the sheer joy of making things. As the child delights in his mud pie, so the adult enjoys building things, especially things of his own design. I think this delight must be an image of God's delight in making things, a delight shown in the distinctness and newness of each leaf and each snowflake.

Second is the pleasure of making things that are useful to other people. Deep within, we want others to use our work and to find it helpful. In this respect the programming system is not essentially different from the child's first clay pencil holder "for Daddy's office."

Third is the joy of always learning, which springs from the nonrepearing nature of the task. In one way or another the problem is ever new, and its solver learns something: sometimes practical, sometimes theoretical, and sometimes both.

Finally, there is the delight of working in such a tractable medium. The programmer, like the poet, works only slightly removed from pure thought-stuff. He builds his castles in the air, from air, creating by exertion of the imagination. Few media of creation are so flexible, so easy to polish and rework, so readily capable of realizing grand conceptual structures. (As we shall see later, this very tractability has its own problems.)

Rule 3: Line height

The Joys of the Craft
Frederick P. Brooks Jr.

Why is programming fun? What delights may its practitioner expect as his reward?

First is the sheer joy of making things. As the child delights in his mud pie, so the adult enjoys building things, especially things of his own design. I think this delight must be an image of God's delight in making things, a delight shown in the distinctness and newness of each leaf and each snowflake.

Second is the pleasure of making things that are useful to other people. Deep within, we want others to use our work and to find it helpful. In this respect the programming system is not essentially different from the child's first clay pencil holder "for Daddy's office."

Third is the joy of always learning, which springs from the nonrepearing nature of the task. In one way or another the problem is ever new, and its solver learns something: sometimes practical, sometimes theoretical, and sometimes both.

Finally, there is the delight of working in such a tractable medium. The programmer, like the poet, works only slightly removed from pure thought-stuff. He builds his castles in the air, from air, creating by exertion of the imagination. Few media of creation are so flexible, so easy to

Rule 4: Line length

The Joys of the Craft
Frederick P. Brooks Jr.

Why is programming fun? What delights may its practitioner expect as his reward?

First is the sheer joy of making things. As the child delights in his mud pie, so the adult enjoys building things, especially things of his own design. I think this delight must be an image of God's delight in making things, a delight shown in the distinctness and newness of each leaf and each snowflake.

Second is the pleasure of making things that are useful to other people. Deep within, we want others to use our work and to find it helpful. In this respect the programming system is not essentially different from the child's first clay pencil holder "for Daddy's office."

Third is the joy of always learning, which springs from the nonrepearing nature of the task. In one way or another the problem is ever new, and its solver learns something: sometimes practical, sometimes theoretical, and sometimes both.

Rule 5: Typeface (Proxima Nova)

The Joys of the Craft
Frederick P. Brooks Jr.

Why is programming fun? What delights may its practitioner expect as his reward?

First is the sheer joy of making things. As the child delights in his mud pie, so the adult enjoys building things, especially things of his own design. I think this delight must be an image of God's delight in making things, a delight shown in the distinctness and newness of each leaf and each snowflake.

Second is the pleasure of making things that are useful to other people. Deep within, we want others to use our work and to find it helpful. In this respect the programming system is not essentially different from the child's first clay pencil holder "for Daddy's office."

Third is the joy of always learning, which springs from the nonrepearing nature of the task. In one way or another the problem is ever new, and its solver learns something: sometimes practical, sometimes theoretical, and sometimes both

Rule 5: Typeface (Proxima Nova)

The Joys of the Craft
Frederick P. Brooks Jr.

Why is programming fun? What delights may its practitioner expect as his reward?

First is the sheer joy of making things. As the child delights in his mud pie, so the adult enjoys building things, especially things of his own design. I think this delight must be an image of God's delight in making things, a delight shown in the distinctness and newness of each leaf and each snowflake.

Second is the pleasure of making things that are useful to other people. Deep within, we want others to use our work and to find it helpful. In this respect the programming system is not essentially different from the child's first clay pencil holder "for Daddy's office."

Third is the joy of always learning, which springs from the nonrepearing nature of the task. In one way or another the problem is ever new, and its solver learns something: sometimes practical, sometimes theoretical, and sometimes both.

Finally, there is the delight of working in such a tractable medium. The programmer, like the poet, works only slightly removed from pure thought-stuff. He builds his castles in the air, from air, creating by exertion of the imagination. Few media of creation are so flexible, so easy to polish and rework, so readily capable of realizing grand conceptual structures. (As we shall see later, this very tractability has its own problems.)

Yet the program construct, unlike the poet's words, is real in the sense that it moves and works, producing visible outputs separate from the construct itself. It prints results, draws pictures, produces sounds, moves arms. The magic of myth and legend has come true in our time. One types the correct incantation on a keyboard, and a display screen comes to life, showing things that never were nor could be.

Programming then is fun because it gratifies creative longings built deep within us and delights sensibilities we have in common with all men.

Rule 6: Point of entry

The Joys of the Craft

Frederick P. Brooks Jr.

Why is programming fun? What delights may its practitioner expect as his reward?

First is the sheer joy of making things. As the child delights in his mud pie, so the adult enjoys building things, especially things of his own design. I think this delight must be an image of God's delight in making things, a delight shown in the distinctness and newness of each leaf and each snowflake.

Second is the pleasure of making things that are useful to other people. Deep within, we want others to use our work and to find it helpful. In this respect the programming system is not essentially different from the child's first clay pencil holder "for Daddy's office."

Third is the joy of always learning, which springs from the nonrepearing nature of the task. In one way or another the problem is ever new, and its solver learns something: sometimes practical, sometimes theoretical, and sometimes both.

Finally, there is the delight of working in such a tractable medium. The programmer, like the poet, works only slightly removed from pure thought-stuff. He builds his castles in the air, from air, creating by exertion of the imagination. Few media of creation are so flexible, so easy to polish and rework, so readily capable of realizing grand conceptual structures. (As we shall see later, this very tractability has its own problems.)

Yet the program construct, unlike the poet's words, is real in the sense that it moves and works, producing visible outputs separate from the construct itself. It prints results, draws pictures, produces sounds, moves arms. The magic of myth and legend has come true in our time. One types the correct incantation on a keyboard, and a display screen comes to life, showing things that never were nor could be.

Programming then is fun because it gratifies creative longings built deep within us and delights sensibilities we have in common with all men.

Rule 7: Progressive disclosure

The Joys of the Craft

Frederick P. Brooks Jr.

Why is programming fun? What delights may its practitioner expect as his reward?

First is the sheer joy of making things. As the child delights in his mud pie, so the adult enjoys building things, especially things of his own design. I think this delight must be an image of God's delight in making things, a delight shown in the distinctness and newness of each leaf and each snowflake.

Second is the pleasure of making things that are useful to other people. Deep within, we want others to use our work and to find it helpful. In this respect the programming system is not essentially different from the child's first clay pencil holder "for Daddy's office."

Third is the joy of always learning, which springs from the nonrepearing nature of the task. In one way or another the problem is ever new, and its solver learns something: sometimes practical, sometimes theoretical, and sometimes both.

Finally, there is the delight of working in such a tractable medium. The programmer, like the poet, works only slightly removed from pure thought-stuff. He builds his castles in the air, from air, creating by exertion of the imagination. Few media of creation are so flexible, so easy to polish and rework, so readily capable of realizing grand conceptual structures. (As we shall see later, this very tractability has its own problems.)

Yet the program construct, unlike the poet's words, is real in the sense that it moves and works, producing visible outputs separate from the construct itself. It prints results, draws pictures, produces sounds, moves arms. The magic of myth and legend has come true in our time. One types the correct incantation on a keyboard, and a display screen comes to life, showing things that never were nor could be.

Programming then is fun because it gratifies creative longings built deep within us and delights sensibilities we have in common with all men.

Rule 8: Meaningful whitespace

The Joys of the Craft

Frederick P. Brooks Jr.

Why is programming fun? What delights may its practitioner expect as his reward?

First is the sheer joy of making things. As the child delights in his mud pie, so the adult enjoys building things, especially things of his own design. I think this delight must be an image of God's delight in making things, a delight shown in the distinctness and newness of each leaf and each snowflake.

Second is the pleasure of making things that are useful to other people. Deep within, we want others to use our work and to find it helpful. In this respect the programming system is not essentially different from the child's first clay pencil holder "for Daddy's office."

Third is the joy of always learning, which springs from the nonrepearing nature of the task. In one way or another the problem is ever new, and its solver learns something: sometimes practical, sometimes theoretical, and sometimes both.

Finally, there is the delight of working in such a tractable medium. The programmer, like the poet, works only slightly removed from pure thought-stuff. He builds his castles in the air, from air, creating by exertion of the imagination. Few media of creation are so flexible, so easy to polish and rework, so readily capable of realizing grand conceptual structures. (As we shall see later, this very tractability has its own problems.)

Yet the program construct, unlike the poet's words, is real in the sense that it moves and works, producing visible outputs separate from the construct itself. It prints results, draws pictures, produces sounds, moves arms. The magic of myth and legend has come true in our time. One types the correct incantation on a keyboard, and a display screen comes to life, showing things that never were nor could be.

Programming then is fun because it gratifies creative longings built deep within us and delights sensibilities we have in common with all men.

Original

The Joys of the Craft
Frederick P. Brooks Jr.

Why is programming fun? What delights may its practitioner expect as his reward?

First is the sheer joy of making things. As the child delights in his mud pie, so the adult enjoys building things, especially things of his own design. I think this delight must be an image of God's delight in making things, a delight shown in the distinctness and newness of each leaf and each snowflake.

Second is the pleasure of making things that are useful to other people. Deep within, we want others to use our work and to find it helpful. In this respect the programming system is not essentially different from the child's first clay pencil holder "for Daddy's office."

Third is the joy of always learning, which springs from the nonrepearing nature of the task. In one way or another the problem is ever new, and its solver learns something: sometimes practical, sometimes theoretical, and sometimes both.

Finally, there is the delight of working in such a tractable medium. The programmer, like the poet, works only slightly removed from pure thought-stuff. He builds his castles in the air, from air, creating by exertion of the imagination. Few media of creation are so flexible, so easy to polish and rework, so readily capable of realizing grand conceptual structures. (As we shall see later, this very tractability has its own problems.)

Yet the program construct, unlike the poet's words, is real in the sense that it moves and works, producing visible outputs separate from the construct itself. It prints results, draws pictures, produces sounds, moves arms. The magic of myth and legend has come true in our time. One types the correct incantation on a keyboard, and a display screen comes to life, showing things that never were nor could be.

Programming then is fun because it gratifies creative longings built deep within us and delights sensibilities we have in common with all men.

Final

The Joys of the Craft

Frederick P. Brooks Jr.

Why is programming fun? What delights may its practitioner expect as his reward?

First is the sheer joy of making things. As the child delights in his mud pie, so the adult enjoys building things, especially things of his own design. I think this delight must be an image of God's delight in making things, a delight shown in the distinctness and newness of each leaf and each snowflake.

Second is the pleasure of making things that are useful to other people. Deep within, we want others to use our work and to find it helpful. In this respect the programming system is not essentially different from the child's first clay pencil holder "for Daddy's office."

Third is the joy of always learning, which springs from the nonrepearing nature of the task. In one way or another the problem is ever new, and its solver learns something: sometimes practical, sometimes theoretical, and sometimes both.

Finally, there is the delight of working in such a tractable medium. The programmer, like the poet, works only slightly removed from pure thought-stuff. He builds his castles in the air, from air, creating by exertion of the imagination. Few media of creation are so flexible, so easy to polish and rework, so readily capable of realizing grand conceptual structures. (As we shall see later, this very tractability has its own problems.)

Yet the program construct, unlike the poet's words, is real in the sense that it moves and works, producing visible outputs separate from the construct itself. It prints results, draws pictures, produces sounds, moves arms. The magic of myth and legend has come true in our time. One types the correct incantation on a keyboard, and a display screen comes to life, showing things that never were nor could be.

Programming then is fun because it gratifies creative longings built deep within us and delights sensibilities we have in common with all men.

Overview

Before we begin

Typography

Information Hierarchy

Case Study: The Joys of the Craft

Resources & Recommended Readings

Workshop

These links are excellent for learning more

- Practical Typography
  - Please just go read this
- Zell Liew on Grids
  - for a good start: Why Vertical Rhythm
- Typewolf
  - Best reference on typefaces and pairings on the Internet
- If you can, take an HCI class!
  - 05-391 Interaction Design Overview
  - 05-392 Designing Human-Centered Systems

Overview

Before we begin

Typography

Information Hierarchy

Case Study: The Joys of the Craft

Resources & Recommended Readings

Workshop

Sketching allows fast iteration of ideas

- Designers like to **iterate** a ton
- Iteration enables exploration of the design space
- We're looking for the best design solution

So we represent

- type with horizontal lines,
- text alignment with vertical lines,
- and **bold** text with thicker lines,

which

- captures information hierarchy,
- and enables rapid iteration.

Digital Designs

Some commonly used software for creating digital designs:

- Sketch
- Illustrator
- Photoshop
- InDesign

Pick your poison. I like Sketch. It's $50 one time for students.

Let’s hop to it!

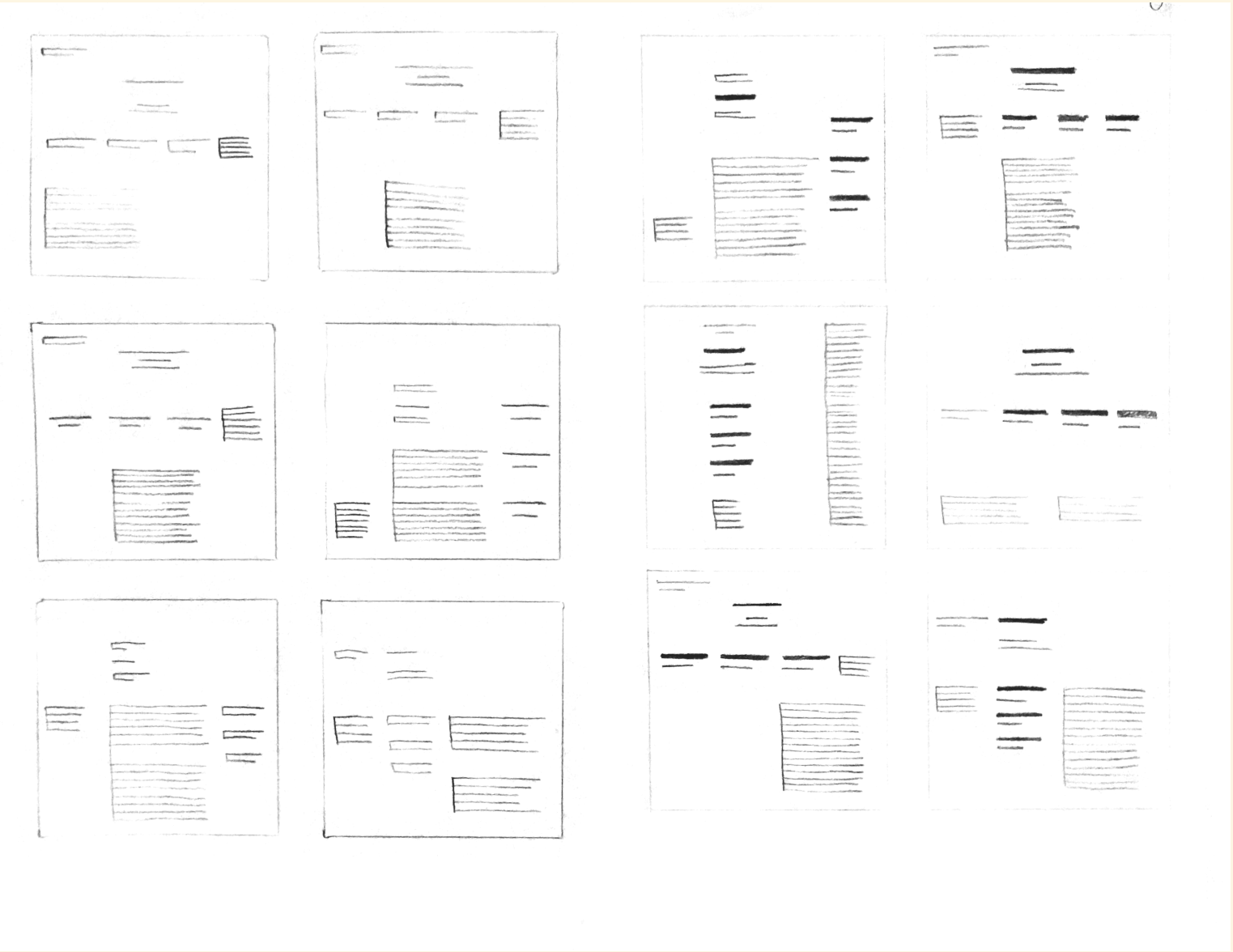
Agenda

- Thumbnail pencil sketches
  - … with critique
- Digital designs
  - … with critique

Designing a Homepage/Poster for WDW

- Only change the typography and hierarchy
- Leave the content alone

Sample thumbnail sketches



Content

Web Dev Weekend
a weekend packed with web development workshops

1:30 p.m. – 3:00 p.m.
HTML & CSS Primer

3:00 p.m. – 4:30 p.m.
Frontend JavaScript Primer

4:30 p.m. – 6:00 p.m.
Crafting Well-Designed Sites

Web Dev Weekend is a series of lectures, workshops, and exercises that introduce you to the fundamentals of web development.

If you want to make websites outside of the classroom, get a head start on later electives, or attend hackathons, Web Dev Weekend will give you the foundational knowledge you need to get started.