Structure

01.07.

Prolog

https://janice-beck.github.io/hacking/exp-00.html

Introduction

This is about 1. Graphic Design, 2. Hacking

Wat am I looking for?

Problem: Habits, design education feels fixed when the world is not everything is changing so fast. no technical literacy. Overwhelming amount of tools, still relying on the ones i learned (Adobe) when there are new ones every day? i cant code! is this bad? own tools? how do i relate to my tools? i dont feel technically literate. Then the education, not much has changed and in practice, applied, no time to adapt or to

Goal: proposing A new role for designers. i want to make beautiful things, but in a way that feels independent, myself but without expressing myself too much (not to personal, authorship omg). but i also want to find out things (designer as researcher) to explain things, from a different point of view. i want to be technically literate, know how and when to use my tools and not be used by them or the companies that produce them. i want to find or at least to try out a new role: one that is somehow applicable in a field of production that hasnt adapted yet, that is not adapting itself as fast as it should. that is playful not taking itself so serious but serious enough that you might get the feeling we as designers could really change something.

Hacking as Philosophy and Method

What is a system

Hacking as Framework

Goals

Trying out a new Role

What is Hacking & Experimental Setup

Definitions

Experimental Framework

Framework, Values, Dokumentation

Introducing the three core values that guide every experiment:

- Playfulness – risk, humor, joy in breaking things
- Learning – each hack should teach me (and others) something
- Transparency – all steps, resources, and failures are visible

How Hackers Work

Experiments Phase I: Systems outside of me

Typeface

Grid & Margin

Conclusion Experiments I

Fonts, grids, and images.
Understanding their inner workings, question their conventional uses
Experiments 1.1 – 1.4

The Politics of Tools
Experiments Phase II: Systems i am Inside of – Tools & Workflows

Langdon Winner
Lev Manovich
(?Marshall McLuhan/ Tool as extension?)
A.I. -> Age of specialization is over (?)

The designer and her Tools

software critique: tools are not neutral
Design software as opinionated systems that promote efficiency, conformity…
- Defaults as invisible ideology
- Tools shape aesthetics and behavior
- Open source (?)

Workflows and Habits

Workflows are also Systems

Experiments Phase II

systems and habits that shape my everyday design work/working behavior
how my tools and workflows embody invisible values.
Expose and subvert the underlying logic of productivity-driven design culture

Hacking as Intervention: Political and Artistic activism
Experiments II: Hacking the Institution

Connect to broader activist and artistic hacking traditions.
Positioning Hacking as cultural critique, resistance
Examples:

- Tactical Media: temporary, strategic disruptions, activist media aesthetics
- !Mediengruppe Bitnik: poetic hacks that reveal/critique systems from within

Experiments:
- bending institutional formats (written thesis or mentoring and midterm presentation, deadlines…),
- positioning the thesis as a meta-reflection on the Master's program itself

What would it mean to not just work within institutions, but on them?
And what might be possible when we see institutional systems as just another layer of design—structured, editable, and hackable?

Hacking the Self: Ego, (Authorship), Self-Image
Experiments III: I am a system

If institutions shape what is considered legitimate, valuable, or "serious" design work, they also shape how designers see themselves: as authors, as professionals, as brands.
Hacking
Beliefs around originality, productivity, mastery, authorship, and self-worth.
The narrative of the solo genius designer / modernism -> HGK Study?
Possible references: Kenneth Goldsmith, Hito Steyerl

A turn inward
Experimenting with ego by remixing, copying, stealing — breaking the myth of the "original designer"
- Eliminating the author: Co-Authorship / Collaboration
- Reflection: How this affected my self-image as a designer / How and why did it feel that way?

Conclusion
Proposing New Role for designers:
One that finds entering Points in seemingly fixed Systems
No specialization in Tool or Skill
Process Orientation – Detachment from Output
No Ego-centrism

Hacking as Design strategy

Idea: This thesis should not only be about hacking, but a hack in itself. Trying to question / disrupt systems in design but also the structure and norms of the (written) thesis in our institution → An Intertwined, Self-Reflective, Hacked Framework/ Structure.

27.05.2025

Generell Stil? Adaption Hacking sprache/ Bericht, stark Prozess erzählen?

0. Prologue: Manifesto / The first Experiment

(https://janice-beck.github.io/hacking/exp-00.html)

As a first, actually preliminary Experiment in my Research, I edited “How to Become a Hacker” (maybe the most known and influential text by a Hacker, about hacking) just by changing the word "hacker" to "designer" in-browser. This minimal, almost lazy technical intervention led to several realizations: it reveals parallels between hacking and designing in terms of attitude; it shows the instability of authorship and the potential to manipulate meaning through small interventions; and it positions text as a system—structured but hackable.

Erklären: Arbeite m Praxis des hacking; Text (bzw. alles) = System ist methodische Annahme des Ansatzes -> was heisst das, was man annimmt?

1. Introduction

Building up on the line of thought I opened in the Prologue, the Introduction could start with questions resulting out of the first experiment, also guiding for the whole process: If text is a system, and systems are everywhere, what does it mean to design by hacking them? / If everything is a system—text, tools, workflows, education...—
then what exactly is the designer’s role within them?

Further contents of the introduction:

- Haltung / Interesse: neue Rolle ausprobieren (Designer as serviceprovider, as Author, as Researcher..) die die ich kenne langweilen mich (?) // System ändern
- was will ich herausfinden?
    - Presenting hacking as a method for understanding, reconfiguring, subverting systems
    - Explaining that the thesis itself will be a hack/experiment — not just about hacking = hacking as both, subject and strategy
    - Positioning myself; more interested in philosophical, ethical…than in technical aspects
    - Introducing my Method as a loop: theory → experiment → reflection → theory
- Stating core research question: What can I hack as a designer and why should I? How and what kind of knowledge can designers generate by adapting hacking strategies? Warum lohnt sich hacking für Designer & worauf Zielt es ab?
- (?) Preview the progression of my experiments:
  From systems outside of me (fonts, tools) → to systems I am inside of (workflows, institutions, expectations) → to systems I am (identity, authorship, ego).

-> STRUKTUR DER GANZEN ARBEIT

2. Definition of Terms / Experimental Setup

2.1 The Hacker
2.2 Hacking as practice
2.3 Hacker ethics

In this section, I define the specific understanding of hacking that forms the theoretical basis of this thesis—rooted in the Western, MIT-originated hacker ethic. Hacking is presented not as a purely technical act, but as a creative, playful, and subversive approach to understanding and reconfiguring systems. This sets the values that guide the practical experiments throughout the work. It is a rather technical and value-based definition grounded in history and culture. Bereits in relation zu Design setzen / was ich über Hacking sage, bezieht sich auf Design

2.4. Experimental Setup: Framework, Values, Dokumentation

Introducing the three core values that guide every experiment:

- Playfulness – risk, humor, joy in breaking things
- Learning – each hack should teach me (and others) something
- Transparency – all steps, resources, and failures are visible

Explaining the conceptual Framework and structure I experiment within, as well as the documentation process: the website, protocols, logs, screenshots, source files.
Kurz

3. How Hackers Work: Culture, Process, and Failure

Chapter 2 outlined the values and ethics based on the history of hacking, this chapter zooms in on how these values play out in practice/how hackers actually work, and transfers this to a way of living/philosophical framework. The theoretical basis for this is Pekka Himamens" The Hacker Ethic and the Spirit of the Information Age"

4.1 Trial, Error, and Joy in Breaking Things // Wiederholung vermeiden
4.2 Community and Collaboration
4.3 Hacking as a way of seeing/being in the world
4.4 Comparison to Graphic design / State of the Art example Anja Grooten, Hackers & Designers: https://hackersanddesigners.nl/hacking-designing-paradoxes-of-collaborative-practice-by-anja-groten (?)

4. Experiments Phase I: Foundational Layer

This chapter marks the shift from theory to practice and documents my first set of design experiments. I began to experiment with the foundational elements of graphic design: Fonts, grids, and images. By hacking these systems, I aimed to understand their inner workings and question their conventional uses.
Experiments 1.1 – 1.4

5. Hacking the Workflow/ Experiments Phase II: Tools / Process

Drawing from critical theory and hacker philosophy, this chapter explores tools as ideological devices, the role of habits and defaults, and the political implications of reclaiming autonomy through design processes.

5.1 The politics of tools

Building on Langdon Winner and software critique: tools are not neutral.
Design software as opinionated systems that promote efficiency, conformity…
Defaults as invisible ideology
Tools shape aesthetics and behavior
Open source (?)

5.2 Workflows as systems

Own habits as “invisible code”/workflows are systems too
Critique of “optimization” culture in design practice
Reference Pekka Himanen: hackers follow rhythms of curiosity, not clocks
Work as play, as passion, as rhythm—rejecting the protestant ethic

5.3 Experiments Phase II

This chapter documents the second phase of my experiments, in which I shift focus to the systems and habits that shape my everyday design work/working behavior. I explore how my tools and workflows embody invisible values. These interventions aim to expose and subvert the underlying logic of productivity-driven design culture.
Experiments 2.1 – 3.3

Imagination!!! how tools shape that

5 a reflection: awareness of habits , what we learn.. can i hack myself

6. Hacking as Intervention: Political and Artistic activism / Experiments III: Hacking the Institution

The goal of this section is to Zoom out again and connect to broader activist and artistic hacking traditions. Positioning Hacking as cultural critique (and again, resistance).
Examples:

- Tactical Media: temporary, strategic disruptions, activist media aesthetics
- !Mediengruppe Bitnik: poetic hacks that reveal/critique systems from within

Hacking as Political act / resistance: McKenzie Wark, A Hacker Manifesto: Hacking as opposition to the "vectoralist class" (owners of information)
Hacking as a method for reclaiming agency and resisting professionalization
???? – evtl weglassen

6.2

This section turns toward the institutional structures that frame and constrain design practice.
Here, I begin to treat elements like the written thesis, the midterm presentation, etc as formats that are possible to hack. By seeing these as systems—with their own expectations, defaults, and codes of behavior—I explore how they shape not only the form of design work, but also the role of the designer.
This phase could include experiments in:

- bending institutional formats (like the written thesis or mentoring and midterm presentation, deadlines…),
- positioning the thesis as a meta-reflection on the Master's program itself—its openness, its lack of definition and focus, and its pedagogical ideologies.

Through these hacks, I could try to ask:
What would it mean to not just work within institutions, but on them? And what might be possible when we see institutional systems as just another layer of design—structured, editable, and hackable?

7. Hacking the Self: Ego, (Authorship), Self-Image / Experiments IV: I am a system

If institutions shape what is considered legitimate, valuable, or "serious" design work, they also shape how designers see themselves: as authors, as professionals, as brands. Hacking the institution inevitably exposes the internalized structures that live inside the designer — beliefs around originality, productivity, mastery, authorship, and self-worth.
//The narrative of the solo genius designer / modernism

Possible references: Paul Soulellis / Publishing as Artistic Practice,
Kenneth Goldsmith, Hito Steyerl

—

This is a turn inward — the most radical application of hacking is not on form or tools, but on my own identity and authorship. It wraps together ethics, design culture critique, and method.

- Experimenting with ego by remixing, copying, stealing — breaking the myth of the “original designer”
- Eliminating the author: Co-Authorship / Collaboration
- Reflection: How this affected my self-image as a designer / How and why did it feel that way?

Experiments 1.5.1 – 1.6; 4.1, 4.2

8. Conclusion

:-)
Summary, relate findings to applied graphic design

21.05.25

0. Prologue: Manifesto / The first Experiment

(https://janice-beck.github.io/hacking/exp-00.html)

As a first, actually preliminary Experiment in my Research, I edited “How to Become a Hacker” (maybe the most known and influential text by a Hacker, about hacking) just by changing the word "hacker" to "designer" in-browser. This minimal, almost lazy technical intervention led to several realizations: it reveals parallels between hacking and designing in terms of attitude; it shows the instability of authorship and the potential to manipulate meaning through small interventions; and it positions text as a system—structured but hackable.

1. Introduction

Building up on the line of thought I opened in the Prologue, the Introduction could start with questions resulting out of the first experiment, also guiding for the whole process: If text is a system, and systems are everywhere, what does it mean to design by hacking them? / If everything is a system—text, tools, workflows, education...—
then what exactly is the designer’s role within them?

Further contents of the introduction:

- Presenting hacking as a method for understanding, reconfiguring, subverting systems
- Explaining that the thesis itself will be a hack/experiment — not just about hacking = hacking as both, subject and strategy
- Positioning myself; more interested in philosophical, ethical…than in technical aspects
- Introducing my Method as a loop: theory → experiment → reflection → theory

- Stating core research question: What can I hack as a designer and why should I? How and what kind of knowledge can designers generate by adapting hacking strategies?
- (?) Preview the progression of my experiments: from hacking fonts and forms → to tools → to design processes → my self-image as designer → to the thesis structure(maybe).
  From systems outside of me (fonts, tools) → to systems I am inside of (workflows, institutions, expectations) → to systems I am (identity, authorship, ego).

2. Definition of Terms / What is hacking?

2.1 The Hacker
2.2 Hacking as practice
2.3 Hacker ethics

In this section, I define the specific understanding of hacking that forms the theoretical basis of this thesis—rooted in the Western, MIT-originated hacker ethic. Hacking is presented not as a purely technical act, but as a creative, playful, and subversive approach to understanding and reconfiguring systems. This sets the values that guide the practical experiments throughout the work. It is a rather technical and value-based definition grounded in history and culture.

3. Experimental Setup: Framework, Values, Documentation

Introducing the three core values that guide every experiment:

Playfulness – risk, humor, joy in breaking things
Learning – each hack should teach me (and others) something
Transparency – all steps, resources, and failures are visible

Explaining the conceptual Framework and structure I experiment within, as well as the documentation process: the website, protocols, logs, screenshots, source files.

4. How Hackers Work: Culture, Process, and Failure

Chapter 2 outlined the values and ethics based on the history of hacking, this chapter zooms in on how these values play out in practice/how hackers actually work, and transfers this to a way of living/philosophical framework. The theoretical basis for this is Pekka Himamens" The Hacker Ethic and the Spirit of the Information Age"

4.1 Trial, Error, and Joy in Breaking Things
4.2 Community and Collaboration
4.3 Hacking as a way of seeing/being in the world
4.4 Comparison to Graphic design / State of the Art example Anja Grooten, Hackers & Designers: https://hackersanddesigners.nl/hacking-designing-paradoxes-of-collaborative-practice-by-anja-groten (?)

5. Experiments Phase I: Foundational Layer (Form)

This chapter marks the shift from theory to practice and documents my first set of design experiments. I began to experiment with the foundational elements of graphic design: Fonts, grids, and images. By hacking these systems, I aimed to understand their inner workings and question their conventional uses.
Experiments 1.1 – 1.4

6. Hacking the Workflow: Tools (?)

Drawing from critical theory and hacker philosophy, this chapter explores tools as ideological devices, the role of habits and defaults, and the political implications of reclaiming autonomy through design processes.

6.1 Hacking as Political act / resistance: McKenzie Wark, A Hacker Manifesto: Hacking as opposition to the "vectoralist class" (owners of information)
- Hacking as a method for reclaiming agency and resisting professionalization

6.2 The politics of tools
- Building on Langdon Winner and software critique: tools are not neutral. Design software as opinionated systems that promote efficiency, conformity…
- Defaults as invisible ideology
- Tools shape aesthetics and behavior
- Open source (?)

6.3 Workflows as systems

Own habits as “invisible code”/workflows are systems too
Critique of “optimization” culture in design practice
Reference Pekka Himanen: hackers follow rhythms of curiosity, not clocks
Work as play, as passion, as rhythm—rejecting the protestant ethic

7. Experiments Phase II: Tools / Process

This chapter documents the second phase of my experiments, in which I shift focus to the systems and habits that shape my everyday design work/working behavior. I explore how my tools and workflows embody invisible values. These interventions aim to expose and subvert the underlying logic of productivity-driven design culture.
Experiments 2.1 – 3.3

8. Hacking as Intervention: Political and Artistic activism

The goal of this chapter is to Zoom out again and connect to broader activist and artistic hacking traditions. Positioning Hacking as cultural critique (and again, resistance).
Examples:

- Tactical Media: temporary, strategic disruptions, activist media aesthetics
- !Mediengruppe Bitnik: poetic hacks that reveal/critique systems from within

9. Experiments III: Hacking the Institution

This chapter turns toward the institutional structures that frame and constrain design practice.
Here, I begin to treat elements like the written thesis, the midterm presentation, etc as formats that are possible to hack. By seeing these as systems—with their own expectations, defaults, and codes of behavior—I explore how they shape not only the form of design work, but also the role of the designer.
This phase could include experiments in:

- bending institutional formats (like the written thesis or mentoring and midterm presentation, deadlines…),
- positioning the thesis as a meta-reflection on the Master's program itself—its openness, its lack of definition and focus, and its pedagogical ideologies.

Through these hacks, I could try to ask:
What would it mean to not just work within institutions, but on them? And what might be possible when we see institutional systems as just another layer of design—structured, editable, and hackable?

10. Hacking the Self: Ego, Authorship, and Remixing

If institutions shape what is considered legitimate, valuable, or "serious" design work, they also shape how designers see themselves: as authors, as professionals, as brands. Hacking the institution inevitably exposes the internalized structures that live inside the designer — beliefs around originality, productivity, mastery, authorship, and self-worth.
//The narrative of the solo genius designer / modernism

Possible references: Paul Soulellis / Publishing as Artistic Practice,
Kenneth Goldsmith, Hito Steyerl

11. Experiments IV: I am a system

This is a turn inward — the most radical application of hacking is not on form or tools, but on my own identity and authorship. It wraps together ethics, design culture critique, and method.

- Experimenting with ego by remixing, copying, stealing — breaking the myth of the “original designer”
- Eliminating the author: Co-Authorship / Collaboration
- Reflection: How this affected my self-image as a designer / How and why did it feel that way?

Experiments 1.5.1 – 1.6; 4.1, 4.2

12. Conclusion

:-)
Summary, relate findings to applied graphic design

Fragments

Intro

In this thesis, hacking functions both as subject and as strategy. It is the lens through which I investigate graphic design practice, and the method I use to conduct that investigation. The thesis itself is structured as a series of experiments, each one hacking a different system I interact with as a designer. Following a progression and structure that formed itself naturally during the process, i started my experiments with what i summarize under the term «Systems outside of me». Those are the elements that I, as most Designers, encounter first: For example; type, grids, and images. Then i moved on to systems i am inside of; tools, workflows, institutions – and finally ended up in systems that i am; ego, authorship, identity.

The method I use is intentionally self-reflective and iterative. Each experiment begins with a theoretical idea, which leads to a practical intervention. The outcomes of that experiment are then analyzed and reflected upon—often raising new questions. These questions lead back to theory, which in turn informs the next experiment. In this way, the process forms a series of loops rather than a straight, linear path.

Goal
now that i explained my motivation, methodological assumption and gave an overview about framework and structure of this work/ thesis, also want to formulate the goal or aim i have with this thesis: Warum lohnt sich hacking für Designer & worauf Zielt es ab? again, a loop, to the first part of the introduction. the frustration but also the believe that design as practice has more to offer and in the practice of graphic design itself is room, potential for change–should be. this may seem personal or even self indulgent. but i am the case studs, i am an example, classic biography of young graphic designer

My claim is: that adapting hacking techniques – both methodologically and philosophically – could offer designers new ways to think about : questions in graphic design – like Questions of agency, autonomy, and participation.

Big overall GOAL:
Make designers kind of understand: nothing has to be the way it is. Nothing is fixed. There are possibilities for interventions and they do make a change, even tiny ones are relevant somehow, even if its just in our own practice / field of action .

2.

Definition of Terms / What is hacking? -> situation the project

->

important before i start with definition of the term; this thesis is about graphic design. not about hacking. i already tried to state this in the introduction: hacking serves as a vehicle, a framework, approach. everything i say about hacking i relate already to design.

start with situation in graphic design/whats wrong with graphic design. Apathy regarding intertwinedness with capitalism and its problems (rupen Pater, caps Lock), overwhelmedness wit ever changing tools, ai, branche wird totgesagt, ego problems. depressed designers. beeing a designer is a spectrum from service provider to author. ersonal connection to design output, thats why we exploit ourself->ego. es ist ein circus, netzwerk über meritocracy und instagram, ständig instagram, immer zeigen, poliert. schöne slides und grafik ist so schwer und ich habe 48h nicht geschlafen vor der präsentation. warum????
meines ERACHTENS, what skills do i think graphic designers need for the future, what do we want to teach? adaptability, system thinking, posing questions. imagination.. context to society to politics, not just producing output but also ques

The Hacker

The term *hacker* is highly context-dependent and often ambiguous. It carries a certain fuzziness, shaped by both historical developments and societal perceptions. Popular clichés—heavily influenced by politically charged campaigns of the 1980s—tend to portray hackers primarily as individuals who unlawfully break into computer systems to steal data or cause harm. The Cambridge Dictionary reflects this narrow view, defining a hacker as “someone who gets into other people's computer systems without permission in order to find out information or to do something illegal.” [1].

The foundation of this paper is based on a different understanding of the term *hacker*—one rooted in the self-image and ethos described by Steven Levy in his influential 1984 book *Hackers: Heroes of the Computer Revolution*. In it, Levy traces the origins of hacker culture back to the late 1950s and early 1960s, particularly within the academic and technological environment of MIT. He portrays hackers as passionate tinkerers—driven by curiosity, playfulness, and a desire to understand, repurpose, and improve existing systems. Levy describes how a group of young model railroad enthusiasts, members of the Tech Model Railroad Club (TMRC), would sneak into rooms at night to experiment with the newly installed, government- and military-funded computers, such as the IBM 704 and later the TX-0 and PDP-1. Their goal was not sabotage, but to find creative ways to make their model train systems more efficient and sophisticated. This reimagining of function—seeing potential in alternative uses—is a core element of the hacker mindset. It reflects not only a disregard for rigid rules and formalities but also a deep sense of creative exploration and innovation.

one figure of this mit group is richard stallman. we know him already from prologue. he made jargon file, website where terms in hacker culture, slang are explained. for the term hacker he already has several definitions, : 1. A person who enjoys exploring the details of programmable systems and how to stretch their capabilities, as opposed to most users, who

prefer to learn only the minimum necessary. RFC1392, the *Internet Users' Glossary*, usefully amplifies this as: A person who delights in having an intimate understanding of the internal workings of a system, computers and computer networks in particular.

2. One who programs enthusiastically (even obsessively) or who enjoys programming rather than just theorizing about programming.

3. A person capable of appreciating *hack value*.

4. A person who is good at programming quickly.

5. An expert at a particular program, or one who frequently does work using it or on it; as in 'a Unix hacker'. (Definitions 1 through 5 are correlated, and people who fit them congregate.)

6. An expert or enthusiast of any kind. One might be an astronomy hacker, for example.

7. One who enjoys the intellectual challenge of creatively overcoming or circumventing limitations.

relevant for us are especially 5.-6

Hacking as practice

So what, then, is hacking? According to Steven Levy's perspective, hacking is about *playing*, *tinkering*, and *experimenting*. It is not inherently tied to computers—far from it. Hacking is a mindset, a way of engaging with the world that involves curiosity, creativity, and the urge to push boundaries.

One of the most influential figures in this context is Richard Stallman, founder of the Free Software Movement and described by Levy as one of "the last of the true hackers." In his essay *On Hacking*, Stallman reflects on the nature of hacking and writes:

"It is hard to write a simple definition of something as varied as hacking, but I think what these activities have in common is playfulness, cleverness, and exploration. Thus, hacking means exploring the limits of what is possible, in a spirit of playful cleverness.[2]

For Stallman, hacking is not limited to code—it's a form of expression. He even considers John Cage's musical piece *4'33"* as hacking. It challenges conventional expectations of what music is, in a way that is both clever and thought-provoking. As Stallman puts it, "Playfully doing something difficult, whether useful or not—that is hacking." [3]

While this definition resonates with me and I find it quite appealing, it lacks one essential aspect that I consider fundamental, both to personal understanding of hacking and to the focus of this project, I already mentioned this in the vorhergehenden abbschnitt: Hacking always aims at a system. It is about understanding systems. entering into something that already exists, deconstructing it, and reappropriating it.

As John Draper, the inventor of phreaking, the art of exploiting and manipulating telephone systems—especially to make free long-distance calls—put it:

"I’m learning about a system. The phone company is a system. A computer is a system, do you understand? If I do what I do, it is only to explore a system. Computers, systems—that's my bag. The phone company is nothing but a computer."[4]

This view emphasises that hacking is not just a playful or clever activity; it is deeply rooted in a drive to understand, manipulate and interact with systems at a fundamental level. For me, this exploratory aspect is what makes hacking so fascinating: How do I find vulnerabilities in a system? Where are the entry points for subversion?

Hacker ethics

From the early hacker scene described in the previous section emerged a set of values—a kind of moral code–that was clearly articulated by Steven Levy in his aforementioned book *Hackers*. Levy outlined core principles such as freedom of and unlimited access to information, a deep mistrust of authority and centralized systems (anti-bureaucracy), and a belief in meritocracy[5]. Overall, this ethic reflects a strongly libertarian mindset. While Levy’s generalizing approach has been criticized[6], his formulation remains foundational. It continues to shape the broader subculture that traces its roots back to those early MIT hackers, and its influence is still visible today, for example in the values upheld by groups like the Chaos Computer Club. On their website, they explicitly cite Steven Levy:

«What are the ethical principles of hacking - motivation and limits

- Access to computers - and anything which might teach you something about the way the world really works - should be unlimited and total. Always yield to the Hands-On Imperative!
- All information should be free.
- Mistrust authority - promote decentralization.
- Hackers should be judged by their acting, not bogus criteria such as degrees, age, race, or position.
- You can create art and beauty on a computer.
- Computers can change your life for the better.
- Don't litter other people's data.
- Make public data available, protect private data.[7]”

The last two points are additions made by the Chaos Computer Club, but overall, these principles are a logical product of the hacker culture that emerged at MIT.

However, for my project—especially in the context of graphic design—it was not Levy’s list that proved most relevant, but rather the famous text *"How to Become a Hacker"***[8]** by Eric S. Raymond (ESR), a prominent figure in the open-source movement. Published in the late 1990s, this document describes hacker culture and offers guidance on the skills and attitudes necessary to "become a hacker." While Raymond builds heavily on the ideas established by earlier figures like Stallman, his focus shifts away from computers and technology in a narrow sense.

Particularly the section *"The Hacker Attitude"* was crucial for my work and its later application to graphic design. Raymond's list moves beyond the technical sphere into a more generalized, philosophical dimension, making it broadly applicable to other disciplines. This approach aligns closely with Stallman's previously mentioned understanding of hacking as a creative, exploratory attitude that can be extended to almost any artistic or intellectual activity.

Raymonds idea of the "Hacker Attitude" outlines five key points, which I will present and discuss in the following section.
The first point is: *"The world is full of fascinating problems waiting to be solved."*
Here, Raymond emphasizes that becoming a hacker requires "a lot of effort" and thus strong internal motivation. He writes: *"You have to get a basic thrill from solving problems, sharpening your skills, and exercising your intelligence."* **[9]**
An interesting observation is that this theme recurs frequently: the idea of finding joy in difficulty—taking pleasure in solving a (sometimes self-imposed) challenge–or problem.

The second point, *"No problem should ever have to be solved twice,"* strongly emphasizes the principle of sharing and the ideal of free and open access to information, as already mentioned earlier. Hackers share their knowledge—whether it's code, software, or other insights they have discovered. They document not only what works but also what doesn't. For Raymond, this is a matter of respect toward other hackers: *"Creative brains are a valuable, limited resource. They shouldn't be wasted on reinventing the wheel (...) you have to believe that the thinking time of other hackers is precious — so much so that it's almost a moral duty for you to share information, solve problems, and then give the solutions away so that others can tackle new problems."* **[10]**
It's ultimately a very rational approach, rooted in efficiency and collective progress.

It is also interesting that Raymond frequently links hackers with creativity, suggesting that his principles apply not only to hackers but to creative people in general. As also seen in the section *"Boredom and drudgery are evil,"* where he writes: *"Hackers (and creative people in general) should never be bored or have to drudge at stupid repetitive work."* He justifies this, again, by appealing to efficiency and effectiveness: valuable time should not be wasted on mundane tasks when it could be used to solve new problems. Raymond advocates for automating repetitive work—except when it is done deliberately, for example, to learn a specific skill or gain particular experience.
The fourth point, *"Freedom is good,"* further underlines the strong libertarian spirit that characterizes hacker culture. *"Hackers are naturally anti-authoritarian,"* Raymond writes. Bureaucracy, censorship, secrecy, and any form of institutionalized control that hinders access to information must, in this view, be resisted.

The final point, *"Attitude is no substitute for competence,"* once again emphasizes the meritocratic values that define hacker culture: "Hackers worship competence.". This section also reveals something important about the hacker work ethic. Throughout his text, Raymond repeatedly refers to hacking as "hard work" — "Becoming a hacker will take intelligence, practice, dedication, and hard work" — yet always connects it to pleasure and enjoyment. The idea that intense effort, continuous practice, and complete dedication are experienced as play, forms the foundation of the hacker work ethic.

7///GRAPHI C D ESIG N????

Conclusion 2

The last three sections provided an overview of the definitions and foundations of hacking that I will use and build upon in this work. They outlined how hacking as a practice—when traced back to its origins in the labs at MIT—is rooted in values such as curiosity, playfulness, and the desire to understand, repurpose, and improve the existing. Hacking is not inherently tied to computers; rather, it is about systems and understanding those. Hacking means entering something that already exists and is apparently closed, deconstructing it, and reappropriating it. Therefore, it is deeply rooted in a drive to explore, manipulate, and interact with systems at a fundamental level. Seeing potential and possibilities in alternative uses is a core element of the hacker attitude, making hacking a deeply creative skill.

This is coupled with a strong libertarian outlook—marked by a certain disrespect for rules and authority, and a fundamental rejection of surveillance and censorship. The ethic which emerged out of this subculture, centers around the core values of freedom, meritocracy, subversion and –again–playfulness. Hackers view problem-solving as a form of play, with the joy of overcoming difficult challenges being a central aspect of the hacking mindset.

It must be noted, however, that I focus here on a very specific understanding of hacking—one largely centered in the Western techno-academic tradition. While there have been and continue to be other origins and motivations for hacking, such as in Cuba in the 60s[11], where scarcity and necessity were driving forces rather than play and curiosity, my theoretical framework and practical experiments are fundamentally based on the MIT-originated, Western-centric hacker ethic.

Hacking as philosophy(?)/Philosophical Perspectives on Hacking

The concept of hacking and its associated ethic, as outlined in the previous chapter, has been further developed and formalized within philosophical and cultural discourse. In this section I will exemplary examine the work of Finnish philosopher Pekka Himanen, who in *The Hacker Ethic* (2001) frames hacking as an alternative attitude on living and working— a cultural and philosophical practice, detached from purely technical skills. Building on this foundation, I then turn to the more radical ideas of theorist and writer McKenzie Wark, who conceptualizes hacking not simply as a personal ethos but as a political act — deeply embedded in struggles over information, control, and ownership.

To build on the previous section, where we ended with a short look at the hacker work ethic: Finnish philosopher Pekka Himanen explicitly examines this work ethic and positions it as an alternative model to the dominant Protestant work ethic as shaped by Max Weber[12]. For hackers, motivation for work comes from excitement, intrinsic interest, and joy — as described earlier. This stands in stark contrast to the Protestant work ethic, which Himanen characterizes with three core attitudes: "Work must be seen as an end in itself; at work one

must do one's part as well as possible; and work must be regarded as a duty, which must be done because it must be done" (Himanen, p. 9). I find this excursion into the idea of "work" particularly relevant for my project, especially considering that graphic design itself is a form of labor — and it raises interesting questions about designers' own relationship to their work, on which I will elaborate later.

For hackers, "passion describes the general tenor of their activity" (p. 18).

This passionate relationship also extends to their concept of time: while the Protestant ethic centers life around regular, repeated working hours — an idea rooted in medieval monasteries — hackers reject this structure. Himanen illustrates this difference through the example of Linus Torvalds, the creator of the Linux operating system: "When Torvalds programmed his first version of Linux, he typically worked late into the night and then woke up in the early afternoon to continue. Sometimes he shifted from coding Linux to just playing with the computer or doing something else entirely." Himanen describes this as typical of hackers, who value an individualistic rhythm of life, far removed from the traditional 9-to-5 model that still dominates most working environments. In today's networked society, it is remarkable how persistent these old notions of work still are.

Himanen also discusses the "money ethic." For hackers, social motivation and peer recognition are far more important than financial gain. This links back to meritocracy as a key value of hacker culture: "Why do hackers use their leisure time to develop programs they openly give away to others?" Himanen asks — answering that for hackers, recognition within a community that shares their passion is more important than money (p. 51). However, this recognition must always be the result of passionate, meaningful creation; it cannot substitute for passion itself.

According to Himanen, it is precisely this link between the social and the passionate levels that makes the hacker model of working so powerful (p. 51). Contrary to common stereotypes, hacking is therefore actually a deeply social activity.

---

[1] https://dictionary.cambridge.org/dictionary/english/hacker
[2] https://stallman.org/articles/on-hacking.html (25.04.25)
[3] https://stallman.org/articles/on-hacking.html (25.04.25)
[4] *Secrets of the Little Blue Box*, *Ron Rosenbaum*, *Esquire Magazine* *(October 1971) as republished by Slate*
[5] Levy, S. (1994). *Hackers: Heroes of the Computer Revolution*. Delta. (S.32-33)
[6] most notably by hacker Acid Phreak, who said in 1990: "There is no one hacker ethic. Everyone has his own. To say that we all think that same way is preposterous" [6]Acid Phreak (1990) Quoted in Jack Hitt and Paul Tough, 'Is Computer Hacking a Crime?', Harpers Magazine (March): 48
[7] *CCC | Hacker Ethics*. (o. D.). https://www.ccc.de/en/hackerethics
[8] Eric S. Raymond, *How to Become a Hacker*, n.d., http://www.catb.org/esr/faqs/hacker-howto.html#attitude.
[9] Eric S. Raymond, *How to Become a Hacker*, n.d., http://www.catb.org/esr/faqs/hacker-howto.html#attitude.

[10] Eric S. Raymond, *How to Become a Hacker*, n.d., http://www.catb.org/esr/faqs/hacker-howto.html#attitude.
[11] *How communism turned Cuba into an island of hackers and DIY engineers*. (2015, 7. Januar). PBS News. https://www.pbs.org/newshour/science/isolation-generation-master-inventors-cuba, Additionally: Motherboard. (2013, 20. Juni). *Cuba's DIY Inventions from 30 Years of Isolation* [Video]. YouTube. https://www.youtube.com/watch?v=v-XS4aueDUg

[12] Weber, M. (2017). *Die protestantische Ethik und der Geist des Kapitalismus*. Musaicum Books.

How hackers work

In the previous chapter, I explored the values and ethics of hacking—curiosity, playfulness, autonomy—as historically and culturally developed principles. This chapter now asks: *How do hackers actually work?* and how is this possible to situate in a design context?

Rather than treating hacking as merely a technical skill, Finnish philosopher Pekka Himanen proposes a broader reading in his 2001 book *The Hacker Ethic and the Spirit of the Information Age*. For Himanen, hacking is an alternative work ethic, one that challenges conventional ideas about labor, productivity, and value—ideas that still permeate much of the design world.

Passion Over Productivity

Himanen contrasts the hacker ethic with what Max Weber famously termed the Protestant work ethic—a value system that emphasizes discipline, obligation, and delayed gratification. In that model, work is a duty, a moral responsibility, and an end in itself.

Hackers, on the other hand, are not motivated by obligation but by passion. Himanen writes: "For hackers, *passion describes the general tenor of their activity*" (p. 18). This sense of intrinsic motivation is fundamental: hackers work not because they have to, but because they want to. Joy, excitement, and intellectual curiosity drive their actions.

This mindset stands in stark contrast to the way design work is often structured: fast-paced, client-driven, and deadline-oriented, where passion is frequently secondary to performance. Yet many designers feel this tension—that there *could be more* to the work than endless rounds of polish and presentation. In that sense, the hacker ethic holds up a mirror to our own profession.

Flexible Time, Fluid Work

Himanen also challenges dominant ideas of time and structure. The 9-to-5 model—rooted in industrialization and reinforced by modern work culture—is largely rejected in hacker communities. He describes figures like Linus Torvalds working deep into the night, shifting between coding and aimless tinkering, following energy and interest rather than schedules or productivity metrics.

This non-linear rhythm reflects an important insight: *creative work happens unevenly.* Design work, too, often defies neat scheduling. Yet we continue to measure it through hours billed, deadlines met, or slides prepared.

3.4 Work as Contribution, Not Competition

Recognition within hacker culture is based on contribution and skill, not status or credentials. Himanen calls this a social ethic, where peer respect is earned through sharing knowledge and solving problems in elegant or imaginative ways. It's not about outperforming others but advancing a shared body of work.

In comparison, the design field often operates on different terms—competitive portfolios, individual visibility, carefully polished presentations. Platforms like Instagram turn design into performance, further entrenching comparison and ego-driven output.

START

to get more practical, out of philosophy, hands on design: ofcourse i am not the first designer that connects hackinng with design. one of most dominant/prominent figures in this scene is anja groten, hackers and designers. inn her essay “**Hacking & Designing Paradoxes of Collaborative Practice”**

she also states that the concept of hacking is not discipline specific or exclusive to the field of computer programming. yet she critizises, that the appropriation of hacking terminology designers miss out on addressing the sociality inherent in hacking practices. In an experiment she posits a fictional dialogue between the stereotype of a hacker and a designer. in this dialogue, i felt a biut accused, because i felt i was doing exactly was designer vorgeworfen wird die auch hacken wollen. the hacker doubts thatdesigners “actually understand what hacking means. Hacking is not a method you can first learn and then apply. Neither can you conceptualize hacking by means of design. Designers need to learn how to write, read, and fix code. They need to get literate before they can call themselves hackers. “ the dialogue also talks about the attitude of hackers that is so appealing for design practice. interesting also the fatc, about working with frustration: Hacking might be an attitude towards making. But this attitude is tightly connected to the practice of writing software, debugging, running and maintaining systems, which is—and this is important to acknowledge—continuously frustrating! Hackers are exposed to things not working. The hacking attitude that is so interesting for you designers is a direct result of encountering resistance, over and over again. Hackers have developed a tremendous tolerance to frustration because we are constantly fighting code. It is the thin line between

frustration and pleasure that is important to understand when describing a hacker's mode of production.

i think designers can also relate to that.

but basically anti thesis to mine, fictional hacker says: Hacking is not a method. If you dig a little deeper you will come across complex forms of interactions, which shape what hackers produce and how they produce it. You cannot learn hacking like you would learn a skill, a subject, or a method. Hacking derives from and contributes to an ecology. You need to be embedded in the ecology in order to understand its workings. You designers tend to glorify hacking and forget about a whole lot of dynamics that are at play in hacking culture. Hackers cannot be described as a homogeneous group. There are many tensions and contradictions within hacker communities. Some hackers make money, some are activists, some are criminals—yet they might all work together on the same project.

again, documentation: **:** The constant state of exposure—and along with it, a sustained vulnerability—is enabled only through constant and meticulous practices of documentation. Far from covering up our bugs, we openly acknowledge and even explain them. We don't hide problems.

WORKING:

Hacking is not a method, skill, or toolkit. It cannot be "learned and applied" like a design methodology. It's embedded in lived experience and system interaction.

Attitude of making: Hacking is characterized by a *defiant, playful*, and *resistant* approach. Not just constructing things, but *disrupting*, *repurposing*, *tweaking*, and *messing with* systems.

Frustration as a creative driver: Hackers constantly encounter things not working. Their tolerance to friction is part of their productivity — *the pleasure of solving problems under resistance* is central to their practice.

No glorification of chance: Unlike some art/design discourses, hacking is about precision, iteration, and effectiveness — not randomness or accidental outcomes.

Deep concentration ("deep hack mode"): Getting lost in a technical problem, not unlike creative flow, but grounded in logic and code structure.

Transparency as a principle: Bugs and failures are openly documented, not hidden. Sharing mistakes is part of the method.

Critique of design's surface obsession: Groten indirectly critiques how designers often aim for polished outcomes, in contrast to hacking's iterative, exposed, and messy process.

SOCIAL ASPECT

These are the cultural, ethical, communal, and interpersonal dynamics embedded in hacking practices:

- **Community over individualism**: Hacking is deeply social — even if done in isolation, it builds on shared infrastructures, collective knowledge, and a culture of mutual critique.

- **Meritocracy and recognition**: The value of work is determined by its usefulness and elegance as judged by the community. Praise is when someone reuses or builds on your code.

- **Collaboration amid difference**: Hacker communities are diverse and often contradictory — some are activists, some are entrepreneurs, some criminals. These differences coexist in shared projects.

- **Tensions and hostilities**: Hacker environments are not always inclusive. There's real critique around *non-welcoming*, *aggressive*, or *hostile* cultures, especially for non-male, non-white, or older contributors.

- **Pedagogy and initiation**: Hacking promotes open learning, but also harsh entry rituals — "RTFM" culture, direct critique, and sometimes abrasive feedback loops.

- **Ethics of transparency and openness**: Documenting code is not just technical — it's an ethical practice that enables shared learning, accountability, and vulnerability.

- **An ecology of friction**: Groten suggests that rather than idolizing hackers, designers should learn from the conflicts and dilemmas of hacker cultures. These tensions produce more grounded, exposed, and accountable modes of making.

- **Designers can borrow this ethos**: By embracing exposed processes, incomplete outcomes, and open-endedness, designers might foster their own *"ecology of frictions."*

EGO

Recognition within hacker culture is based on **contribution and skill**, not status or credentials. Himanen calls this a **social ethic**, where peer respect is earned through sharing knowledge and solving problems in elegant or imaginative ways. It's not about outperforming others but advancing a shared body of work. This links back to meritocracy as a key value of hacker culture: "Why do hackers use their leisure time to develop programs they openly give away to others?" Himanen asks — answering that for hackers, recognition within a community that shares their passion is more important than money (p. 51). According to Himanen, it is precisely this link between the social and the passionate levels that makes the hacker model of working so powerful (p. 51). Contrary to common stereotypes, hacking is therefore actually a deeply social activity.

next, hacking is a social activity. its a lot about contributing to something. do something together, and finding meritocracy, bestätigung in that. es geht auch um documentation, how important:

y the way, documentation is crucial for code to be reusable by others. I always make sure my code is clean and beautiful before I publish it. If hackers like my code it means what I made is effective. It's a compliment when someone uses my code.

and beauty! so also designers and hackers have in common this urge to create something beautiful, but hackers are actually proud when they are copied. designers are scared of watch other “stealing ideas” beeing original

In the previous chapter, I explored the values and ethics of hacking as historically and culturally developed principles. This chapter now asks: *How do hackers actually work?* What structures and habits shape their practice, and how can these be understood as a model for creative work more broadly?

Rather than treating hacking as merely a technical skill, Finnish philosopher Pekka Himanen proposes a broader reading in his 2001 book *The Hacker Ethic and the Spirit of the Information Age*. For Himanen, hacking is an alternative work ethic, one that challenges conventional ideas about labor, productivity, and value—ideas that still permeate much of the design world.

Passion Over Productivity

Himanen contrasts the hacker ethic with what Max Weber famously termed the **Protestant work ethic**—a value system that emphasizes discipline, obligation, and delayed gratification. In that model, work is a duty, a moral responsibility, and an end in itself.

Hackers, on the other hand, are not motivated by obligation but by **passion**. Himanen writes: “For hackers, *passion describes the general tenor of their activity*” (p. 18). This sense of

intrinsic motivation is fundamental: hackers work not because they have to, but because they want to. Joy, excitement, and intellectual curiosity drive their actions.

Himanen also challenges dominant ideas of time and structure. The 9-to-5 model—rooted in industrialization and reinforced by modern work culture—is largely rejected in hacker communities. He describes figures like Linus Torvalds working deep into the night, shifting between coding and aimless tinkering, following energy and interest rather than schedules or productivity metrics.

Recognition within hacker culture is based on **contribution and skill**, not status or credentials. Himanen calls this a **social ethic**, where peer respect is earned through sharing knowledge and solving problems in elegant or imaginative ways. It's not about outperforming others but advancing a shared body of work. Himanen also discusses the "money ethic." For hackers, social motivation and peer recognition are far more important than financial gain. This links back to meritocracy as a key value of hacker culture: "Why do hackers use their leisure time to develop programs they openly give away to others?" Himanen asks — answering that for hackers, recognition within a community that shares their passion is more important than money (p. 51). According to Himanen, it is precisely this link between the social and the passionate levels that makes the hacker model of working so powerful (p. 51). Contrary to common stereotypes, hacking is therefore actually a deeply social activity.

ofcourse i am not the first designer that connects hackinng with design. one of most dominant/prominent figures in this scene is anja groten, hackers and designers. inn her essay “

Hacking & Designing Paradoxes of Collaborative Practice”

Rather than viewing hacking as a set of tools or techniques, Groten describes it as a **mentality**—a stubborn, often playful attitude toward systems. In her workshop *Levels of Autonomy*, for instance, participants repurpose remote-control cars into DIY autonomous vehicles using sensors and microcontrollers. But the goal isn’t just technical transformation. The workshop invites a broader reflection on how autonomous systems are built, framed, and justified.

This dual focus—on hands-on making and critical reflection—is central to many hacking practices. As Groten puts it, the hacker’s approach involves “a defiant yet playful attitude to making.” It’s an ethos grounded in curiosity, confrontation, and the desire to understand things by breaking and remaking them.

Groten's dialogue between a stereotypical hacker (H) and a designer (D) surfaces the frictions embedded in hacker culture itself. H emphasizes that hacking isn't simply an aesthetic or mindset—it's a practice rooted in **technical literacy**, constant failure, and deeply social forms of production. As the hacker says, "You cannot learn hacking like you would learn a skill, a subject, or a method. Hacking derives from and contributes to an *ecology*."

Frustration is baked into the process. As Groten writes, hackers develop a "tremendous tolerance to frustration," since the work frequently revolves around broken code, obscure errors, and experimental fixes. The ability to remain in this state—where pleasure and failure are deeply entangled—is itself a form of expertise.

Designers, by contrast, are often trained to **conceal** failure. Smoothness, polish, and visual clarity are prized. Processes are hidden, outcomes idealized. Yet this avoidance of visible friction may limit how designers learn, share, and critique their own work. From this perspective, hacker culture offers an alternative: a space where unfinished work, partial knowledge, and ongoing negotiation are not only accepted, but essential.

3.3 Hacker Ethics (and Its Contradictions)

While hacker culture often celebrates openness, collaboration, and creative freedom, it also contains tensions. Despite the emphasis on access and inclusion, many hacking environments are experienced as hostile—particularly by those who don't fit the archetype of the white, male, hyper-competent coder. Groten cites examples of misogyny and aggressive communication in open-source spaces, reminding us that "confrontational rhetorics" can function both as pedagogy and exclusion.

This contradiction is central to understanding hacker practice: its **radical transparency** can both empower and alienate. Hackers document bugs, publish imperfect code, and invite public scrutiny. These acts demand vulnerability—but also resilience. As the dialogue points out, participation often requires the ability to withstand conflict, which not everyone has equal access to. In this way, hacking is never just technical; it's political, cultural, and personal.

Toward a Designerly Hackerism

For designers, then, the challenge is not to simply borrow hacking jargon, but to grapple with its **messy realities**. To think like a hacker is not just to act subversively or work experimentally. It is to engage in a practice that is embedded in specific infrastructures, histories, and social dynamics. Hacking is not a general metaphor for "being creative." It is a situated way of working—improvised, collaborative, vulnerable, and often contentious.

Designers can learn from this. Rather than seeing hacking as a toolkit to extract inspiration from, we might instead treat it as a **mirror**—a lens through which to reflect on the assumptions, rituals, and blind spots in our own field. this text by groten helped me to ssharpen the very broad definition i had of hacking.here i want again to emphasize that i never would call myself a hacker. i am aware that i am borrowing the practices etc. but still i think i dont have to see it that strict. i derive my methodology from hacking and how i edfinr the term … maybe nont so important

Tools

In design, tools have always been more than mere instruments: they shape what can be made, how it can be made, and even what is imagined as possible. Historically, entire design paradigms were determined by the available tools — from the precision of the printing press to the experimental freedoms enabled by phototypesetting or digital page layout. These tools do not simply execute design; they embed values, standards, and affordances that fundamentally steer creative work. (Manovich, 2013, p. 135) In contemporary practice, software such as Adobe Creative Suite has become almost synonymous with graphic design, its functions and interfaces woven deeply into design education, professional workflows, and aesthetic expectations. ““Mediums” as they are implemented in software are part of distinct cultural histories that go back for hundreds and often thousands of years. ” p. 226, “. These histories influence how we understand and use these media today.” p.226 “A medium, then, is not just a set of materials and tools (whether physical, mechanical, electronic, or implemented in software) and artistic techniques supported by these toolsit is also an imaginary database of all expressive possibilities, compositions, emotional states and dynamics, representational and communication techniques, and “content” actualized in all the works created with a particular combination of certain materials and tools.” p. 226
isbn: 978-1-62356-745-3 978-1-62356-817-7

This raises a crucial question: how much autonomy do designers actually have within these tool environments? When a tool defines the parameters of a layout, a type choice, or a workflow, to what extent are we freely designing, and to what extent are we fulfilling the tool’s embedded logic? There is a growing recognition that software itself can carry politics, biases, and hidden assumptions — not only about aesthetic taste, but also about what design is and how it should function.
do artifacts have politics langdon winner : “at matters is not technology itself, but the social or economic system inwhich it is embedded. This maxim, which in a number of variations is the central premise of a theory that can be called the social determination of technology, has an obvious wisdom. It serves as a needed corrective to those who focus uncritically on such things as "the comput er and its social impacts" but who fail to look behind technical things to notice the social circumstances of their development, deployment, and use. This view provides an antidote to naive technological determinism?the idea that tech nology develops as the sole result of an internal dynamic, and then, unmediated by any other influence, molds society to fit its patterns. Those who have not recognized the ways in which technologies are shaped by social and economic forces have not gotten very far. But the corrective has its own” p. 3

any technical devices and systems important in everyday life contain possibilities for many different ways of ordering human activity. Consciously or not, deliber ately or inadvertently, societies choose structures for technologies that influence how people are going to work, communicate, travel, consume, and so forth over a very long time. In the processes by which structuring decisions are made, different people are differently situated and possess unequal degrees of power as well as unequal levels of awareness. p. 127
“According to this view, the adoption of a given technical system unavoidably brings with it conditions for human relationships that have a distinctive political cast?for example, centralized or decentralized, egalitarian or inegalitarian, re pressive or liberating. This is ultimately what is at stake in assertions like those of Lewis Mumford that two traditions of technology, one authoritarian, the other democratic, exist side by side inWestern history. In all the cases I cited above the technologies are relatively flexible in design and arrangement, and variable in their effects. Although one can recognize a particular result produced in a particular setting, one can also easily imagine how a roughly similar device or system might have been built or situated with very much different political consequences. The idea we must now examine and evaluate is that certain kinds of technology do not allow such fle”
do artifacts have politics? source: deadalus, vol. 109, no. 1 modern technology:problem or opportunity(winter1980)pp. 121-136 published by the MIT press on behalf of american academy of arts and sciences

Hacking as Intervention: Political and Artistic activism

The goal of this section is to Zoom out again and connect to broader activist and artistic hacking traditions. Positioning Hacking as cultural critique (and again, resistance).
Examples:

- Tactical Media: temporary, strategic disruptions, activist media aesthetics
- !Mediengruppe Bitnik: poetic hacks that reveal/critique systems from within

until now i defined what definition of hacking i use, how hackers work and also relationship of tools and graphic designers. hacking in graphic design almost only is focusing on tools like luuse described but also collectives like hackers and designers or varia. mostly focus on the technological aspect. but i think there is more to hacking ethics. and thats why i want to have a look how those values are transferred in the art scene. ill keeo it short, just only outline the parts that where relevant for me/graphic design experiments. i do this ti investigate on hackings potential as cultural critique (and again, resistance).

coming from tools in graphic design. learned that there is systemic perspective and notion in ou tools, more important how we use them, attitude towards it and not which one we use it. remember, attitude towards system is the for me key aspect with which i connect hacking to graphic design or why i use hackinng methodology, inspired by hacking, ti investigate graphic design. now im more interested in political impact of hacking methods, and how they have been adapted in the arts. why? to have an abstraction of hacking inspired techniques, what artists take from it. because designers only take this tool and collective and super left and technological approach. but the political geht verloren, wird übersehen. deshalb i want to zoom in on certain aspects of that in the following section.

!mediengruppe bitnik

tactical media:
interesting: they didnt wanted to be defined/critiques of roles: Definitions also create boundaries. What was once so liquid would become increasingly structured and separated as the movement was theorized and historicized. On the other hand, joy can emerge out of separation that expresses itself as generative difference. There was a feeling of relief that those involved in tactical media could be any kind of cultural hybrid. Artist, scientist, technician, craftsperson, theorist, activist, etc., could all be mixed together in combinations that had different weights and intensities. These many roles (becoming artist, becoming activist, becoming scientist, etc.) contained in each individual and group could be acknowledged and valued. Many Roles!!

it was a molecular intervention. For a brief time there was and continues to be a relief from capital's tyranny of specialization that forces us to perform as if we are a fixed set of relationships and characteristics,

definitions of tactical media: First, tactical media is a form of digital interventionism.* It challenges the existing semiotic regime by replicating and redeploying it in a manner that offers participants in the projects a new way of seeing, understanding, and interacting with a given system. Systems!
collective work: "Specialization does not predetermine action. This is partly why tactical media lends itself to collective efforts, as there is always a need for a differentiated skill base that is best developed through collaboration." p.8 from critical art ensemble, digital resistance, explorations in tactical media.
https://monoskop.org/images/3/3a/Critical_Art_Ensemble_Digital_Resistance_Explorations_in_Tactical_Media.pdf

"Tactical Media are never perfect, always in becoming, performative and pragmatic, involved in a continual process of questioning the premises of the channels they work with. This requires the confidence that the content can survive intact as it travels from interface to interface. But we must never forget that hybrid media has its opposite its nemesis, the Medialen Gesamtkunstwerk. The final program for the electronic Bauhaus."
https://www.nettime.org/Lists-Archives/nettime-l-9705/msg00096.html

Geert Lovink on Fri, 16 May 1997 10:30:25 +0200 (MET DST)

critical enginneer:

The Critical Engineering Working Group Julian Oliver

Berlin, October 2011 Gordan Savičić

Danja Vasiliev

THE CRITICAL ENGINEERING MANIFESTO

0. The Critical Engineer considers Engineering to be the most transformative language of our time, shaping

the way we move, communicate and think. It is the work of the Critical Engineer to study and exploit this

language, exposing its influence.

1. The Critical Engineer considers any technology depended upon to be both a challenge and a threat. The

greater the dependence on a technology the greater the need to study and expose its inner workings,

regardless of ownership or legal provision.

2. The Critical Engineer raises awareness that with each technological advance our techno-political literacy

is challenged.

3. The Critical Engineer deconstructs and incites suspicion of rich user experiences.

4. The Critical Engineer looks beyond the 'awe of implementation' to determine methods of influence and

their specific effects.

5. The Critical Engineer recognises that each work of engineering engineers its user, proportional to that

user's dependency upon it.

6. The Critical Engineer expands 'machine' to describe interrelationships encompassing devices, bodies,

agents, forces and networks.

7. The Critical Engineer observes the space between the production and consumption of technology. Acting

rapidly to changes in this space, the Critical Engineer serves to expose moments of imbalance and deception.

8. The Critical Engineer looks to the history of art, architecture, activism, philosophy and invention and finds

exemplary works of Critical Engineering. Strategies, ideas and agendas from these disciplines will be

adopted, re-purposed and deployed.

9. The Critical Engineer notes that written code expands into social and psychological realms, regulating

behaviour between people and the machines they interact with. By understanding this, the Critical Engineer

seeks to reconstruct user-constraints and social action through means of digital excavation.

10. The Critical Engineer considers the exploit to be the most desirable form of exposure.

https://criticalengineering.org/ce.pdf

Free Art & Technology Lab (F.A.T. Lab)

Hacking as Intervention: Political and Artistic Activism

Until now, I have explored what hacking might mean beyond its purely technical framing, asking how hackers work, what values they embody, and how those values relate to graphic design tools and practices. Much of the current discourse around hacking in graphic design—such as that seen in collectives like Hackers & Designers or Luuse—often foregrounds the technical dimension: open-source tools, collaborative coding, and experimental infrastructures. While these aspects are vital and often inherently critical (e.g., rejecting monopolistic software, advocating for transparency), they can also feel alienating to classically trained or applied designers, for whom coding and critical making are not primary practices.

Yet there is more to hacking than technological or aesthetic gestures. Its ethical stance, its potential for resistance, and its capacity for cultural and institutional critique are equally significant. That is why, in this section, I want to zoom out—tracing how hacking's values have extended into activist and artistic practices that use intervention not merely as a technical tactic, but as a political gesture. These approaches frame hacking as a mode of critique that confronts systems of power, authorship, and control.

By shifting the focus away from tool fetishism and toward systemic questioning, these practices offer models for how designers might adopt hacking strategies that are both critical and accessible—strategies that don't require full technical immersion, but still challenge dominant norms and institutional frameworks.

A useful starting point is the tradition of tactical media, which emerged in the 1990s. Tactical media can be understood as a form of temporary, strategic intervention, working with and around new media. It interrupts existing power structures and semiotic regimes by repurposing them, offering participants fresh ways of seeing, understanding, and interacting with the infrastructures that govern their lives.

My favourite example is the activist duo **The Yes Men**, who became known for spectacular interventions at global trade conferences, posing under false names as representatives of powerful corporations or organizations they deemed exploitative. They engineered public-relations disasters for their targets by grotesquely exaggerating their positions, or by performing sudden moments of "enlightenment" on their behalf.

One of their most striking actions was their *Bhopal news hijacking*: in 2004, posing as a Dow Chemical spokesperson, they appeared live on BBC World and announced that Dow would finally accept full responsibility for the 1984 Bhopal disaster — a gas leak that killed thousands in India — and pay 12 billion dollars in compensation to victims. The BBC, taking this at face value, broadcast the announcement globally, sending Dow's stock price into free fall before the hoax was revealed. In a single stroke, the Yes Men forced the world to confront Dow's ongoing refusal to make reparations, exposing the company's inaction through a carefully constructed fiction.

As Fluter magazine describes, the Yes Men “created an alternative way of thinking. With their work, they show us that the reality around us is not fixed — it can be changed, if we act.” For me, this is crucial: tactical media uses humor, fiction, and provocation as hacks, disturbing dominant narratives and showing that systems are malleable.

Tactical media also resists rigid definitions, refusing to fix identities or roles. Artists, scientists, technicians, activists, designers — all these positions can merge and shift within tactical practices, forming new hybrid constellations. As the Critical Art Ensemble writes, specialization does not predetermine action: collective efforts thrive on differentiated skill sets, always recombining.

Another fascinating reference point is **!Mediengruppe Bitnik**, an artist collective from Switzerland whose works can be read as poetic hacks that reveal and critique systems from within. Rather than simply breaking or bypassing systems, they slip inside them, subverting expectations from the inside out. A well-known example is their *Delivery for Mr. Assange* project: in 2013, they sent a parcel containing a hidden live-streaming camera to Julian Assange inside the Ecuadorian Embassy in London. Over the course of its postal journey, the camera broadcast its own progress in real time, transforming a standard logistics chain into a performative event — and turning the postal system itself into an unwitting stage and medium for artistic intervention.

What is crucial here is that !Mediengruppe Bitnik recognized the postal system as a system in its own right: a network of rules, procedures, and flows that could be appropriated, exploited, and reimagined. They leveraged the system’s predictability and trust to create a moment of radical visibility, exposing how infrastructures can be made to serve unintended functions.

Their work exemplifies how hacking-inspired interventions can probe infrastructures not merely by attacking or dismantling them, but by *inhabiting* them creatively. They open up hidden layers of systems — in this case, postal tracking, international security, and diplomatic asylum — to public reflection.

What resonates with me is how !Mediengruppe Bitnik expand hacking beyond technical exploits. They transform hacking into an aesthetic and critical practice, using the unexpected to produce ruptures in everyday systems of control. In this sense, they invite designers to think of hacking not just as a means to modify tools, but as a broader cultural attitude of revealing, questioning, and shifting the parameters of what is assumed to be stable.

dunne and raby:

https://www.youtube.com/watch?v=dFiXGbRmn2Q

“asking questions rather than providing solutions. what is it worth questioning, how design can even pose questions. and where would this questioning happen beyond an academic context?”

meat experiment. imagination! messy perspective. dragging things into consumer perspective.

“design as an accessible language to open up discussion”

doing things that lead to ozher thing that dont have an actual conclusion.

definition of what critical is: Critique is not necessarily negative; it can also be a gentle refusal, a turning away from what exists, a longing, wishful thinking, a desire, and even a dream. Critical designs are testimonials to what could be, but at the same time, they offer alternatives that highlight weaknesses within existing normality.

experimental jetset

beeing open abt their influences: https://www.youtube.com/watch?v=kIzgqg7zl4M&t=119 showing that something comes from something: as designer you already learn quite early that language is something material. that can bbe shaped in any form you want. (concrete poetry) systemic approach? hacking= beeing transparent abt their sources kind of.

also rooted in theri heritage(duch sociodemoctratic..)

they also show up at 3 persons, as collectif. together (we), single person steps back. vorgriff zu authorship/self

What I find crucial is that these practices foreground **attitude** over tool. Tactical media, critical engineering, and artistic hacking all highlight a systemic awareness: understanding how infrastructures shape us, and how we might tactically disrupt or reimagine them. Designers who adopt hacking as inspiration often focus only on its tools or collective models, missing this political dimension. By revisiting hacking through its activist and artistic lineages, we see its potential as cultural critique, as resistance, and as a generative mode of interference.

This reframing is highly relevant for graphic design. It suggests that what truly matters is not simply *which* tools we adopt, but *how* we position ourselves toward the systems they operate within. Rather than celebrating clever hacks in isolation, we might learn from these activist-artistic practices to treat design itself as a field of intervention: to expose, to question, and to remake. In the following sections, I will outline a few examples and techniques that have been most relevant for me as a designer, helping to reimagine my own practice as a critical, systemic, and political act.

Alongside these activist and artistic hacking traditions, I find the framework of **Speculative and Critical Design (SCD)**, as articulated by Anthony Dunne and Fiona Raby (2013), highly relevant. SCD expands hacking's critical, resistant stance into design culture more broadly. Instead of exploiting technical systems, speculative designers use fictional scenarios, critical prototypes, and design fictions to "hack" cultural assumptions, exposing and questioning hidden social and institutional values. They treat design itself as a site of cultural critique — a practice of making visible what is otherwise taken for granted. This resonates with my own interest in graphic design not merely as a technical discipline, but as a system of codes, infrastructures, and inherited habits that can and should be disrupted.

THESIS_V1

01. Prologue

> To do the Graphic Design philosophy right, you have to be loyal to excellence. You have to believe that graphic design is a craft worth all the intelligence, creativity, and passion you can muster. (…) Graphic design and implementation should be a joyous art, a kind of high-level play. If this attitude seems preposterous or vaguely embarrassing to you, stop and think; ask yourself what you've forgotten. Why do you design graphic instead of doing something else to make money or pass the time? You must have thought graphic design was worthy of your passion once... To do the graphic design philosophy right, you need to have (or recover) that attitude. You need to care. You need to play. You need to be willing to explore.

The Designer Attitude

1. THE WORLD IS FULL OF FASCINATING PROBLEMS WAITING TO BE SOLVED

Being a graphic designer is lots of fun, but it's a kind of fun that takes lots of effort. The effort takes motivation. Successful athletes get their motivation from a kind of physical delight in making their bodies perform, in pushing themselves past their own physical limits. Similarly, to be a graphic designer you have to get a basic thrill from solving problems, sharpening your skills, and exercising your intelligence. If you aren't the kind of person that feels this way naturally, you'll need to become one in order to make it as a graphic designer. Otherwise you'll find your energy is sapped by distractions like sex, money, and social approval. (You also have to develop a kind of faith in your own learning capacity — a belief that even though you may not know all of what you need to solve a problem, if you tackle just a piece of it and learn from that, you'll learn enough to solve the next piece — and so on, until you're done.)

2. NO PROBLEM SHOULD EVER HAVE TO BE SOLVED TWICE

Creative brains are a valuable, limited resource. They shouldn't be wasted on re-inventing the wheel when there are so many fascinating new problems waiting out there. To behave like a graphic designer, you have to believe that the thinking time of other graphic designers is precious — so much so that it's almost a moral duty for you to share information, solve problems and then give the solutions away just so other graphic designers can solve new problems instead of having to perpetually re-address old ones. Note, however, that "No problem should ever have to be solved twice." does not imply that you have to consider all existing solutions sacred, or that there is only one right solution to any given problem. Often, we learn a lot about the problem that we didn't know before by studying the first cut at a solution. It's OK, and often necessary, to decide that we can do better. What is not OK is artificial technical, legal, or institutional barriers (like closed-source code) that prevent a good solution from being re-used and force people to re-invent wheels. (You don't have to believe that you're obligated to give all your creative product away, though the graphic designers that do are the ones that get most respect from other graphic designers. It's consistent with graphic designer values to sell enough of it to keep you in food and rent and computers. It's

fine to use your graphic design skills to support a family or even get rich, as long as you don't forget your loyalty to your art and your fellow graphic designers while doing it.)

3. BOREDOM AND DRUDGERY ARE EVIL

Graphic designers (and creative people in general) should never be bored or have to drudge at stupid repetitive work, because when this happens it means they aren't doing what only they can do — solve new problems. This wastefulness hurts everybody. Therefore boredom and drudgery are not just unpleasant but actually evil. To behave like a graphic designer, you have to believe this enough to want to automate away the boring bits as much as possible, not just for yourself but for everybody else (especially other graphic designers). (There is one apparent exception to this. Graphic designers will sometimes do things that may seem repetitive or boring to an observer as a mind-clearing exercise, or in order to acquire a skill or have some particular kind of experience you can't have otherwise. But this is by choice — nobody who can think should ever be forced into a situation that bores them.)

4. FREEDOM IS GOOD

Graphic designers are naturally anti-authoritarian. Anyone who can give you orders can stop you from solving whatever problem you're being fascinated by — and, given the way authoritarian minds work, will generally find some appallingly stupid reason to do so. So the authoritarian attitude has to be fought wherever you find it, lest it smother you and other graphic designers. (This isn't the same as fighting all authority. Children need to be guided and criminals restrained. A graphic designer may agree to accept some kinds of authority in order to get something he wants more than the time he spends following orders. But that's a limited, conscious bargain; the kind of personal surrender authoritarians want is not on offer.) Authoritarians thrive on censorship and secrecy. And they distrust voluntary cooperation and information-sharing — they only like ‘cooperation’ that they control. So to behave like a graphic designer, you have to develop an instinctive hostility to censorship, secrecy, and the use of force or deception to compel responsible adults. And you have to be willing to act on that belief.

5. ATTITUDE IS NO SUBSTITUTE FOR COMPETENCE

To be a graphic designer, you have to develop some of these attitudes. But copping an attitude alone won't make you a graphic designer, any more than it will make you a champion athlete or a rock star. Becoming a graphic designer will take intelligence, practice, dedication, and hard work. Therefore, you have to learn to distrust attitude and respect competence of every kind. Graphic designers won't let posers waste their time, but they worship competence — especially competence at designing, but competence at anything is valued. Competence at demanding skills that few can master is especially good, and competence at demanding skills that involve mental acuteness, craft, and concentration is best. If you revere competence, you'll enjoy developing it in yourself — the hard work and dedication will become a kind of intense play rather than drudgery. That attitude is vital to becoming a graphic designer.

01. Introduction

What you just read was the first experiment–the first hack– in a series of hacks that i conducted in the course of this master thesis. It emerged spontaneously, out of a moment of play, while i was researching for the theoretical part of this project. I took two fundamental texts from hacker culture—written by Eric S. Raymond, a central figure in the early hacker scene—and replaced every instance of the word *hacker* with *graphic designer*. A simple, almost lazy intervention: two lines of code in the browser console. But this small act led to several realizations.
First, it confirmed an intuition I already had when choosing hacking as the subject for my final project: that there are parallels between the mindset of hackers and that of (to me, ideal) graphic designers. Second, it revealed the instability of authorship—and how meaning can be radically shifted through minimal interventions. And finally, it positioned text not as fixed content, but as a system—structured, yet hackable. Why this matters, I'll elaborate on shortly.

But first: How did I even get here?
The Process that finally led to this thesis started out of a feeling of frustration. Frustration with the roles we, as graphic designers, are expected to fit into. We're often offered fixed identities—service provider, author, researcher, storyteller…
While these labels may help define our function within certain contexts, none of them ever fully captured what I believed design could be. I had the feeling that the practice of graphic design has more potential than just executing briefs, producing outcomes, or explaining things. I started to look for a different perspective—a different way of relating to design. A way that's less about fitting into predefined frameworks, and more about questioning, rethinking, and opening them up.

That's when I turned to hacking.

Hacking as Philosophy and Method

This thesis explores what happens when the mindset and methods of hacking are applied to the field of graphic design. Not hacking in the narrow sense of cybercrime or even just coding, but in a broader, cultural sense: as it was shaped in the MIT hacker scene of the 1960s and 70s. In that context, hacking is not just technical skill—it's a playful, curious, and subversive way of engaging with systems. It's about understanding how things work, so you can make them work differently.
This attitude offered me a new framework, one that felt both more honest and more expansive. It resonated with the kind of relationship I wanted to have with graphic design—not just as a producer of outcomes, but as someone who can question, reconfigure, and resist systems that are taken for granted.

What is a System?

As many times as I've used the word «system» by now, it's worth pausing to ask: what exactly do I mean by system in the context of this thesis?

I return to the first experiment. By replacing the word «hacker» with «graphic designer», I treated the text not as fixed content, but as a structure—something with internal logic, dependencies, and rules. In other words: a system. This small intervention served as a kind of seed for the entire project.

The central methodological assumption of this thesis is this: Everything is a system. This broad definition includes not only texts, but also:

- tools and software,
- workflows, routines, habits,
- professional norms and institutional structures,
- even less tangible things like identity, authorship, or ego.

In short, any structured set of roles, rules, habits, or relationships. If it has a pattern, it can be understood. And if it can be understood, it can be reconfigured, it can be hacked. This perspective allows hacking to become a design method—one that treats existing conditions not as fixed constraints, but as materials to be investigated, questioned, or subverted.

Hacking as Framework

In this thesis, hacking functions both as subject and as strategy. It is the lens through which I investigate graphic design practice, and the method I use to conduct that investigation. The thesis itself is structured as a series of experiments, each one hacking a different system I interact with as a designer. Following a progression and structure that emerged naturally during the process, I began my experiments with what I summarize under the term «systems outside of me». These are the elements that I—as most designers—encounter first: for example, type, grids, and images. Then I moved on to systems I am inside of: tools, workflows, institutions; and finally, I arrived at systems I am: ego, authorship, identity.
The method I use is intentionally self-reflective and iterative. Each experiment begins with a theoretical idea, which leads to a practical intervention. The outcomes of that experiment are then analyzed and reflected upon—often raising new questions. These questions lead back to theory, which in turn informs the next experiment. In this way, the process forms a series of loops rather than a straight, linear path.
This also provides the overall structure of the work—including the one of the written part. A crucial part of my investigation was the constant back-and-forth between thinking and making, between theory and practice. I aim to reflect this interplay within the written thesis itself. For that reason, there is no strict separation between a “theoretical” and a “practical” part. Instead, I directly connect hacking theory to design practice—and then describe the experiments that emerged from that connection. Even this structure is an experiment—an attempt to hack the habits, conventions, and institutional expectations I am familiar with.

Now that I have outlined my motivation, methodological approach, and the framework of this work, I want to articulate the goal of this thesis: Why is hacking relevant for designers—and what does it aim to achieve?
A loop, again, to the beginning of this introduction: the frustration within the limitedness of the profession, this sense of «this can’t be it/das kann es nicht gewesen sein», and the belief that design as a practice holds more potential than it is often allowed to show. I am aware that this might sound personal, even self-indulgent. But I see myself as a case study—an

example of a typical design biography: a creative childhood, a design degree, agency work, followed by disillusionment, boredom. I am not an exception—I am part of a pattern.
My claim is that adopting hacking techniques—both methodologically and philosophically—could offer designers new ways to think about key questions in graphic design: questions of agency, autonomy, authorship, and participation. By framing graphic design as something that operates with, and within systems, hacking becomes a mindset that invites us to question defaults, repurpose structures, and open up alternative paths.
The broader aim of this thesis is to remind designers that systems aren't static. Nothing is fixed. There is always room for intervention and experiment. If you learn that even a typeface or a software preset can be hacked, you might begin to see that larger systems—institutions, workflows, even your own self-image—are open to change as well. Design, then, becomes a tool for reflection, resistance, and transformation. These interventions may not always change the whole system, but they could reshape how we think, how we work, and how we define our practice.

02. What is hacking and Experimental setup

Before diving into the definition of the term, it's important to clarify: this thesis is not about hacking per se. It is about the practice of graphic design. Hacking serves here as a conceptual framework, a lens, a method—one that could offer designers a tool for thinking critically and working experimentally within their practice. Everything discussed in relation to hacking is already situated within the context of design. For that reason, this section does not aim to deliver a comprehensive historical account of hacking. Instead, it focuses on the aspects of hacker culture that are relevant to the structure and methodology of this thesis. The goal is to establish a working definition of hacking—as mindset and method—that directly supports the exploration of systems within graphic design.

The term *hacker* is highly context-dependent and often ambiguous. It carries a certain fuzziness, shaped by both historical developments and societal perceptions. Popular clichés—heavily influenced by politically charged campaigns of the 1980s—tend to portray hackers primarily as individuals who unlawfully break into computer systems to steal data or cause harm. The Cambridge Dictionary reflects this narrow view, defining a hacker as "someone who gets into other people's computer systems without permission in order to find out information or to do something illegal." [1].

The foundation of this thesis draws on a different understanding, one rooted in the ethos described by Steven Levy in *Hackers: Heroes of the Computer Revolution* (1984). Levy traces the origins of the term *hacker* to the MIT Tech Model Railroad Club in the late 1950s and early 1960s, where members would sneak into rooms at night to experiment with the newly installed, government- and military-funded computers, such as the IBM 704 and later the TX-0 and PDP-1. Their goal was not sabotage, but to find creative ways to make their model train systems more efficient and sophisticated. This reimagining of function—seeing potential in alternative uses—is a core element of the hacker mindset. It reflects not only a

disregard for rigid rules and formalities but also a deep sense of creative exploration and thrive for innovation.
One of the most influential figures in this context is Richard Stallman, founder of the Free Software Movement and described by Levy as one of “the last of the true hackers.” In his essay *On Hacking*, Stallman reflects on the nature of hacking and writes:

“It is hard to write a simple definition of something as varied as hacking, but I think what these activities have in common is playfulness, cleverness, and exploration. Thus, hacking means exploring the limits of what is possible, in a spirit of playful cleverness.[2]

For Stallman, hacking is not limited to code—it’s a form of expression. He even considers John Cage’s musical piece *4'33"* as hacking. It challenges conventional expectations of what music is, in a way that is both clever and thought-provoking. As Stallman puts it, “Playfully doing something difficult, whether useful or not—that is hacking.” [3]

Beyond personal essays, ESR also helped compile one of the most influential cultural documents in the hacker community: the Jargon File. Originally a glossary of slang and technical in-jokes among early programmers, the Jargon File eventually grew into a living lexicon of hacker culture.

Among its many entries, the definition of *hacker* lists several overlapping meanings:

1. A person who enjoys exploring the details of programmable systems and how to stretch their capabilities.

2. One who programs enthusiastically or obsessively.

3. Someone capable of appreciating *hack value*—the cleverness or elegance of a solution.

4. A person who is good at programming quickly.

5. An expert or power user of a particular system or tool (e.g., a Unix hacker).

6. An expert or enthusiast of any kind—someone might be an *astronomy hacker*, for example.

7. One who enjoys the intellectual challenge of creatively overcoming or circumventing limitations.

These definitions expand the scope of hacking beyond computers or software. They present hacking as a form of deep engagement, creative exploration, and problem-solving—traits that are equally relevant to design practice. Especially definitions 5 and 6 suggest a more generalized application of the hacker mindset: one rooted in skill, curiosity, and a desire to stretch the boundaries of a medium, system, or discipline.

Hacking is always about systems. It requires entering into something that already exists, deconstructing it, and making it do something else. John Draper, better known as "Captain Crunch," and a pioneer of so-called *phreaking* (hacking telephone networks), puts it plainly: "I’m learning about a system. The phone company is a system. A computer is a system, do

you understand? If I do what I do, it is only to explore a system. Computers, systems—that's my bag. The phone company is nothing but a computer."[4]

This systemic perspective is my key. Whether we're talking about telephony, software, publishing platforms, or design workflows, hacking means critical engagement with structure. It is less about producing finished outcomes and more about asking: How does this work? What else could this be?

From this early hacker scene emerged a set of values—a kind of moral code–that was clearly articulated by Steven Levy in his aforementioned book *Hackers*. Levy outlined core principles such as freedom of and unlimited access to information, a deep mistrust of authority and centralized systems (anti-bureaucracy), and a belief in meritocracy[5]. Overall, this ethic reflects a strongly libertarian mindset. While Levy's generalizing approach has been criticized[6], his formulation remains foundational. It continues to shape the broader subculture that traces its roots back to those early MIT hackers, and its influence is still visible today

THESIS_V2“

02. Prologue

> To do the Graphic Design philosophy right, you have to be loyal to excellence. You have to believe that graphic design is a craft worth all the intelligence, creativity, and passion you can muster. (…) Graphic design and implementation should be a joyous art, a kind of high-level play. If this attitude seems preposterous or vaguely embarrassing to you, stop and think; ask yourself what you've forgotten. Why do you design graphic instead of doing something else to make money or pass the time? You must have thought graphic design was worthy of your passion once... To do the graphic design philosophy right, you need to have (or recover) that attitude. You need to care. You need to play. You need to be willing to explore.

The Designer Attitude

2. THE WORLD IS FULL OF FASCINATING PROBLEMS WAITING TO BE SOLVED

Being a graphic designer is lots of fun, but it's a kind of fun that takes lots of effort. The effort takes motivation. Successful athletes get their motivation from a kind of physical delight in making their bodies perform, in pushing themselves past their own physical limits. Similarly, to be a graphic designer you have to get a basic thrill from solving problems, sharpening your skills, and exercising your intelligence. If you aren't the kind of person that feels this way naturally, you'll need to become one in order to make it as a graphic designer. Otherwise you'll find your energy is sapped by distractions like sex, money, and social approval. (You also have to develop a kind of faith in your own learning capacity — a belief that even though you may not know all of what you need to solve a problem, if you tackle just a piece of it and learn from that, you'll learn enough to solve the next piece — and so on, until you're done.)

2. NO PROBLEM SHOULD EVER HAVE TO BE SOLVED TWICE

Creative brains are a valuable, limited resource. They shouldn't be wasted on re-inventing the wheel when there are so many fascinating new problems waiting out there. To behave like a graphic designer, you have to believe that the thinking time of other graphic designers is precious — so much so that it's almost a moral duty for you to share information, solve problems and then give the solutions away just so other graphic designers can solve new problems instead of having to perpetually re-address old ones. Note, however, that "No problem should ever have to be solved twice." does not imply that you have to consider all existing solutions sacred, or that there is only one right solution to any given problem. Often, we learn a lot about the problem that we didn't know before by studying the first cut at a solution. It's OK, and often necessary, to decide that we can do better. What is not OK is artificial technical, legal, or institutional barriers (like closed-source code) that prevent a good solution from being re-used and force people to re-invent wheels. (You don't have to believe that you're obligated to give all your creative product away, though the graphic designers that do are the ones that get most respect from other graphic designers. It's consistent with graphic designer values to sell enough of it to keep you in food and rent and computers. It's

fine to use your graphic design skills to support a family or even get rich, as long as you don't forget your loyalty to your art and your fellow graphic designers while doing it.)

3. BOREDOM AND DRUDGERY ARE EVIL

Graphic designers (and creative people in general) should never be bored or have to drudge at stupid repetitive work, because when this happens it means they aren't doing what only they can do — solve new problems. This wastefulness hurts everybody. Therefore boredom and drudgery are not just unpleasant but actually evil. To behave like a graphic designer, you have to believe this enough to want to automate away the boring bits as much as possible, not just for yourself but for everybody else (especially other graphic designers). (There is one apparent exception to this. Graphic designers will sometimes do things that may seem repetitive or boring to an observer as a mind-clearing exercise, or in order to acquire a skill or have some particular kind of experience you can't have otherwise. But this is by choice — nobody who can think should ever be forced into a situation that bores them.)

4. FREEDOM IS GOOD

Graphic designers are naturally anti-authoritarian. Anyone who can give you orders can stop you from solving whatever problem you're being fascinated by — and, given the way authoritarian minds work, will generally find some appallingly stupid reason to do so. So the authoritarian attitude has to be fought wherever you find it, lest it smother you and other graphic designers. (This isn't the same as fighting all authority. Children need to be guided and criminals restrained. A graphic designer may agree to accept some kinds of authority in order to get something he wants more than the time he spends following orders. But that's a limited, conscious bargain; the kind of personal surrender authoritarians want is not on offer.) Authoritarians thrive on censorship and secrecy. And they distrust voluntary cooperation and information-sharing — they only like ‘cooperation’ that they control. So to behave like a graphic designer, you have to develop an instinctive hostility to censorship, secrecy, and the use of force or deception to compel responsible adults. And you have to be willing to act on that belief.

5. ATTITUDE IS NO SUBSTITUTE FOR COMPETENCE

To be a graphic designer, you have to develop some of these attitudes. But copping an attitude alone won't make you a graphic designer, any more than it will make you a champion athlete or a rock star. Becoming a graphic designer will take intelligence, practice, dedication, and hard work. Therefore, you have to learn to distrust attitude and respect competence of every kind. Graphic designers won't let posers waste their time, but they worship competence — especially competence at designing, but competence at anything is valued. Competence at demanding skills that few can master is especially good, and competence at demanding skills that involve mental acuteness, craft, and concentration is best. If you revere competence, you'll enjoy developing it in yourself — the hard work and dedication will become a kind of intense play rather than drudgery. That attitude is vital to becoming a graphic designer.

03. Introduction

What you just read was the first experiment–the first hack– in a series of hacks that i conducted in the course of this master thesis. It emerged spontaneously, out of a moment of play, while i was researching for the theoretical part of this project. I took two fundamental texts from hacker culture—written by Eric S. Raymond, a central figure in the early hacker scene—and replaced every instance of the word hacker with graphic designer. A simple, almost lazy intervention: two lines of code in the browser console. But this small act led to several realizations.
First, it confirmed an intuition I already had when choosing hacking as the subject for my final project: that there are parallels between the mindset of hackers and that of (to me, ideal) graphic designers. Second, it revealed the instability of authorship—and how meaning can be radically shifted through minimal interventions. And finally, it positioned text not as fixed content, but as a system—structured, yet hackable. Why this matters, I'll elaborate on shortly.

But first: How did I even get here?
The Process that finally led to this thesis started out of a feeling of frustration. Frustration with the roles we, as graphic designers, are expected to fit into. We're often offered fixed identities—service provider, author, researcher, storyteller…
While these labels may help define our function within certain contexts, none of them ever fully captured what I believed design could be. I had the feeling that the practice of graphic design has more potential than just executing briefs, producing outcomes, or explaining things. I started to look for a different perspective—a different way of relating to design. A way that's less about fitting into predefined frameworks, and more about questioning, rethinking, and opening them up.

That's when I turned to hacking.

Hacking as Philosophy and Method

This thesis explores what happens when the mindset and methods of hacking are applied to the field of graphic design. Not hacking in the narrow sense of cybercrime or even just coding, but in a broader, cultural sense: as it was shaped in the MIT hacker scene of the 1960s and 70s. In that context, hacking is not just technical skill—it's a playful, curious, and subversive way of engaging with systems. It's about understanding how things work, so you can make them work differently.
This attitude offered me a new framework, one that felt both more honest and more expansive. It resonated with the kind of relationship I wanted to have with graphic design—not just as a producer of outcomes, but as someone who can question, reconfigure, and resist systems that are taken for granted.

What is a System?

As many times as I've used the word «system» by now, it's worth pausing to ask: what exactly do I mean by system in the context of this thesis?

I return to the first experiment. By replacing the word «hacker» with «graphic designer», I treated the text not as fixed content, but as a structure—something with internal logic, dependencies, and rules. In other words: a system. This small intervention served as a kind of seed for the entire project.

The central methodological assumption of this thesis is this: Everything is a system. This broad definition includes not only texts, but also:

- tools and software,
- workflows, routines, habits,
- professional norms and institutional structures,
- even less tangible things like identity, authorship, or ego.

In short, any structured set of roles, rules, habits, or relationships. If it has a pattern, it can be understood. And if it can be understood, it can be reconfigured, it can be hacked. This perspective allows hacking to become a design method—one that treats existing conditions not as fixed constraints, but as materials to be investigated, questioned, or subverted.

Hacking as Framework

Before diving deeper, it's important to clarify: this thesis is not about hacking per se. It is about the practice of graphic design. Hacking serves here as a conceptual framework, a method—one that could offer designers a tool for thinking critically and working experimentally within their practice. It is the lens through which I investigate graphic design practice, and the method I use to conduct that investigation.

The thesis itself is structured as a series of experiments, each one examining a different system I interact with as a designer. Following a progression and structure that emerged naturally during the process, I began my experiments with what I summarize under the term «systems outside of me». These are the elements that I—as most designers—encounter first: for example, type, grids, and images. Then I moved on to systems I am inside of: tools, workflows, institutions; and finally, I arrived at systems I am: ego, authorship, identity.
The method I use is intentionally self-reflective and iterative. Each experiment begins with a theoretical idea, which leads to a practical intervention. The outcomes of that experiment are then analyzed and reflected upon—often raising new questions. These questions lead back to theory, which in turn informs the next experiment. In this way, the process forms a series of loops rather than a straight, linear path.
This also provides the overall structure of the work—including the one of the written part. A crucial part of my investigation was the constant back-and-forth between thinking and making, between theory and practice. I aim to reflect this interplay within the written thesis itself. For that reason, there is no strict separation between a “theoretical” and a “practical” part. Instead, I directly connect hacking theory to design practice—and then describe the experiments that emerged from that connection. Even this structure is an experiment—an attempt to hack the habits, conventions, and institutional expectations I am familiar with.

Now that I have outlined my motivation, methodological approach, and the framework of this work, I want to articulate the goal of this thesis: Why is hacking relevant for designers—and what does it aim to achieve?

My claim is that adopting hacking techniques—both methodologically and philosophically—could offer designers new ways to think about key questions in graphic design: questions of agency, autonomy, authorship, and participation. By framing graphic design as something that operates with, and within systems, hacking becomes a mindset that invites us to question defaults, repurpose structures, and open up alternative paths.
The broader aim of this thesis is to remind designers that systems aren't static. Nothing is fixed. There is always room for intervention and experiment. If you learn that even a typeface or a software preset can be hacked, you might begin to see that larger systems—institutions, workflows, even your own self-image—are open to change as well. Design, then, becomes a tool for reflection, resistance, and transformation. These interventions may not always change the whole system, but they could reshape how we think, how we work, and how we define our practice.

04. What is hacking and Experimental setup

This section focuses on how key aspects of hacker culture relate to the structure and methodology of this thesis—situating hacking directly within the context of design. The goal is to establish a working definition of hacking—as mindset and method—that directly supports the exploration of systems within graphic design.

The term hacker is highly context-dependent and often ambiguous, shaped by both historical developments and societal perceptions. Popular clichés—heavily influenced by politically charged campaigns of the 1980s—tend to portray hackers primarily as individuals who unlawfully break into computer systems to steal data or cause harm. The Cambridge Dictionary reflects this narrow view, defining a hacker as "someone who gets into other people's computer systems without permission in order to find out information or to do something illegal." [1].

The foundation of this thesis draws on a different understanding, one rooted in the ethos described by Steven Levy in Hackers: Heroes of the Computer Revolution (1984). Levy traces the origins of the term hacker to the MIT Tech Model Railroad Club in the late 1950s and early 1960s, where members would sneak into rooms at night to experiment with the newly installed, government- and military-funded computers, such as the IBM 704 and later the TX-0 and PDP-1. Their goal was not sabotage, but to find creative ways to make their model train systems more efficient and sophisticated. This reimagining of function—seeing potential in alternative uses—is a core element of the hacker mindset. It reflects not only a disregard for rigid rules and formalities but also a deep sense of creative exploration and thrive for innovation.
One of the most influential figures in this context is Richard Stallman, founder of the Free Software Movement and described by Levy as one of "the last of the true hackers." In his essay On Hacking, Stallman reflects on the nature of hacking and writes:

"It is hard to write a simple definition of something as varied as hacking, but I think what these activities have in common is playfulness, cleverness, and exploration. Thus, hacking means exploring the limits of what is possible, in a spirit of playful cleverness.[2]

For Stallman, hacking is not limited to code—it's a form of expression. He even considers John Cage's musical piece 4'33" as hacking. It challenges conventional expectations of what music is, in a way that is both clever and thought-provoking. As Stallman puts it, "Playfully doing something difficult, whether useful or not—that is hacking." [3]

Beyond personal essays, ESR also helped compile one of the most influential cultural documents in the hacker community: the Jargon File. Originally a glossary of slang and technical in-jokes among early programmers, the Jargon File eventually grew into a living lexicon of hacker culture.

Among its many entries, the definition of hacker lists several overlapping meanings:

8. A person who enjoys exploring the details of programmable systems and how to stretch their capabilities.

9. One who programs enthusiastically or obsessively.

10. Someone capable of appreciating hack value—the cleverness or elegance of a solution.

11. A person who is good at programming quickly.

12. An expert or power user of a particular system or tool (e.g., a Unix hacker).

13. An expert or enthusiast of any kind—someone might be an astronomy hacker, for example.

14. One who enjoys the intellectual challenge of creatively overcoming or circumventing limitations.

In continuity with Stallman and Levy's perspectives, the Jargon File broadens hacking beyond the realm of code. It defines hacking as deep engagement, curiosity, and a desire to stretch the limits of any system—qualities directly transferable to design practice. Especially definitions 6 and 7 suggest a more generalized application of the hacker mindset: one rooted in skill, curiosity, and a desire to stretch the boundaries of a medium, system, or discipline.

Hacking is always about systems. It requires entering into something that already exists, deconstructing it, and making it do something else. John Draper, better known as "Captain Crunch," and a pioneer of so-called phreaking (hacking telephone networks), puts it plainly: "I'm learning about a system. The phone company is a system. A computer is a system, do you understand? If I do what I do, it is only to explore a system. Computers, systems—that's my bag. The phone company is nothing but a computer."[4]

This systemic perspective is my key. Whether we're talking about telephony, software, publishing platforms, or design workflows, hacking means critical engagement with structure. It is less about producing finished outcomes and more about asking: How does this work? What else could this be?

To end this section of definitions, loop back to prologue, “the hacker attitude” by eric s. raymonnd:

1997 essay How to Become a Hacker articulates a broader philosophical framework. Like his predecessors, Raymond frames hacking as an activity that is not merely technical but deeply tied to attitude. In his section on "The Hacker Attitude," he outlines five key points, all of which are directly relevant and applicable to creative practice:

1. **"The world is full of fascinating problems waiting to be solved."**
   Hacking, requires intellectual curiosity and derives joy from difficulty—solving complex problems not because one has to, but because one wants to.

2. **"No problem should ever have to be solved twice."**
   This principle emphasizes openness and the sharing of knowledge—whether it’s code, tools, or process documentation. Hackers are part of a collective knowledge economy, driven by mutual respect and efficiency.

3. **"Boredom and drudgery are evil."**
   Hackers value automation, iteration, and systems thinking—not out of laziness, but because repetition wastes potential. Time spent on mindless tasks could be better spent on creative problem-solving.

4. **"Freedom is good."**
   Hacking carries a fundamentally anti-authoritarian spirit. It mistrusts top-down structures, secrecy, and gatekeeping. It promotes decentralization and autonomy—qualities deeply relevant to any critique of institutionalized or corporate culture.

5. **"Attitude is no substitute for competence."**
   Finally, hacking is a meritocracy. It prizes skill, rigour, and execution over status. Hacking is hard work—but it is work pursued as play, as challenge, as joy.

Together, these points form what Raymond calls “The Hacker Attitude.” For this project, they provide a useful frame for thinking about design as a form of systemic engagement: a way of intervening, reimagining, and repurposing structures, tools, and ideas. Hacking, in this expanded sense, becomes a method—a speculative, critical, and often playful practice of engaging with the world as a system that is always open to change.

Experimental framework and setup

The exploration and familiarization around the term «hacking» laid the foundation for the practical part of this project: the experiments.

The first step was to build a documentation system. Derived from hacker ethics, this meant committing to radical transparency, a principle deeply embedded in hacker culture. Documentation is not an afterthought here; it is central. Every experiment, every mistake is

recorded and made publicly accessible. To enable this, I created a website hosted via GitHub Pages, which functions as a GitHub repository. A repository is an openly accessible directory of files—typically source code—used by developers (and hackers) to collaborate, share, and iterate on projects. One of GitHub's core features is version control, meaning every change is tracked. Most importantly, the code is visible: anyone can view it, copy it, and build upon it.

The website itself consists of basic HTML files, intentionally kept minimal to remain accessible and easily modifiable. The **main content** is the documentation of the experiments. Each experiment includes:

- A **short description**
- A **step-by-step protocol**, detailing every action taken and every source referenced
- The resulting **output**, which may include scripts, PDFs, or other downloadable artifacts

The idea is that every experiment should be **replicable and forkable**—in the same spirit as open-source projects. Anyone can repeat the process, remix it, or take it in a new direction.

IMG / SCREENSHOT WEBSITE

As a starting point, I set three values that each experiment must fulfill. These are drawn from Eric S. Raymond's already introduced text The Hacker Attitude and function as a kind of ethical framework:

- **Playfulness** – A sense of joy and curiosity is central. Hackers are intrinsically motivated; they follow intuition, embrace humor, and aren't afraid to break things—in fact, they often break things on purpose to see what's possible. This mindset encourages risk, surprise, and unplanned directions.
- **Learning** – Each experiment should generate insight. Whether technical, conceptual, or process-related, every hack is an opportunity to understand something new.
- **Transparency** – Nothing is hidden. Every step, source, decision, and failure is documented and made accessible. The process is at least as important as the outcome.

03. How Hackers Work

In the previous chapter, I explored the values and ethics of hacking as historically and culturally developed principles and related them to the concept of my project. This chapter sharpens that definition by focusing more directly on hacking as a way of working. I aim to examine how hackers actually operate in practice, and where their modes of working might overlap with or challenge those of designers. Two texts have been particularly influential in shaping this section. One is The Hacker Ethic and the Spirit of the Information Age by Finnish philosopher Pekka Himanen. Himanen moves beyond the image of hacking as a purely technical activity and instead frames it as an alternative work ethic—one that resists traditional ideas of labor, productivity, and value. His perspective emerges from the same Western techno-academic milieu that many of the early open-source movements did, and it resonates with ongoing conversations in design around autonomy, passion, and purpose in creative work.

Himanen contrasts the hacker ethic with what Max Weber famously termed the Protestant work ethic—a value system that emphasizes discipline, obligation, and delayed gratification. In that model, work is a duty, a moral responsibility, and an end in itself.
Hackers, on the other hand, are not motivated by obligation but by passion. Himanen writes: “For hackers, passion describes the general tenor of their activity” (p. 18). This sense of intrinsic motivation is fundamental: hackers work not because they have to, but because they want to. Joy, excitement, and intellectual curiosity drive their actions.

This passionate relationship also extends to their concept of time: while the Protestant ethic centers life around regular, repeated working hours — an idea rooted in medieval monasteries — hackers reject this structure. Himanen illustrates this difference through the example of Linus Torvalds, the creator of the Linux operating system: "When Torvalds programmed his first version of Linux, he typically worked late into the night and then woke up in the early afternoon to continue. Sometimes he shifted from coding Linux to just playing with the computer or doing something else entirely." Himanen describes this as typical of hackers, who value an individualistic rhythm of life, far removed from the traditional 9-to-5 model that still dominates most working environments. In today's networked society, it is remarkable how persistent these old notions of work still are.

Himanen also discusses the "money ethic." For hackers, social motivation and peer recognition are far more important than financial gain. This links back to meritocracy as a key value of hacker culture: "Why do hackers use their leisure time to develop programs they openly give away to others?" Himanen asks — answering that for hackers, recognition within a community that shares their passion is more important than money (p. 51). However, this recognition must always be the result of passionate, meaningful creation; it cannot substitute for passion itself.
According to Himanen, it is precisely this link between the social and the passionate levels that makes the hacker model of working so powerful (p. 51). Contrary to common stereotypes, hacking is therefore actually a deeply social activity.

Recognition within hacker culture is based on contribution and skill, not status or professional or academic titles. Himanen calls this a social ethic, where peer respect is earned through sharing knowledge and solving problems in elegant or imaginative ways. It's not about outperforming others but advancing a shared body of work. This links back to meritocracy as a key value of hacker culture: "Why do hackers use their leisure time to develop programs they openly give away to others?" Himanen asks — answering that for hackers, recognition within a community that shares their passion is more important than money (p. 51). According to Himanen, it is precisely this link between the social and the passionate levels that makes the hacker model of working so powerful (p. 51). Contrary to common stereotypes, hacking is therefore actually a deeply social activity.

The second text that significantly influenced my thinking was Anja Groten's "Hacking & Designing: Paradoxes of Collaborative Practice." As co-founder of Hackers & Designers, Groten shares a perspective similar to Himanen's in many respects. She understands hacking primarily as a mentality—a persistent, often playful attitude toward systems—rather than a fixed set of tools or methods.

Groten is more critical of designers who aim to adopt hacking methods too casually. In her essay, she stages a fictional dialogue between a stereotypical hacker (H) and a designer (D), which surfaces some of the deep frictions embedded in hacker culture itself. The hacker insists that hacking is not "a method you can first learn and then apply." It isn't simply an aesthetic or a mindset—it's a practice grounded in technical literacy, constant failure, and deeply social forms of production. As the hacker puts it, "You cannot learn hacking like you would learn a skill, a subject, or a method. Hacking derives from and contributes to an ecology."

Groten also highlights how frustration is baked into hacking. Hackers, she writes, develop a "tremendous tolerance to frustration," as their work often involves broken code, obscure bugs, and experimental problem-solving. Learning to navigate this thin line between failure and pleasure becomes a kind of expertise in itself—and that's something many designers, including myself, can relate to.

// what does this mean for designing then/ what can designers learn from it?

intrinsic motivation, common for designers. decision to become a designer is seldom bc of money, ethics, but somehow passion. but we loose tahat passion often times in practiice, applied graphic design. interesting is 9-5 working model. especially uín agencies it works exactly like that, b ut das word mehr und mehr aufgelöst (quellen?)

tools

In graphic design, tools have never been mere neutral instruments; rather, they fundamentally shape what can be made, how it is made, and even what designers imagine as possible. Historically, entire design paradigms have been determined by the constraints and affordances of available tools, from the precision and repeatability of the printing press to the experimental flexibility offered by phototypesetting and later digital page layout systems. As Manovich (2013) argues, contemporary software does not simply execute design but actively embodies values, standards, and cultural assumptions that guide creative practice[1]. Adobe Creative Suite, for example, has become almost synonymous with graphic design itself, its functions and interfaces deeply embedded in design education, professional workflows, and aesthetic conventions. Manovich (2013) situates these tools within broader cultural trajectories, noting that “mediums as they are implemented in software are part of distinct cultural histories that go back for hundreds and often thousands of years,” which continue to shape how we understand and use them today[2]. Thus, a medium is far more than a set of technical materials or tools; it constitutes what Manovich calls an “imaginary database” of expressive, compositional, and communicative possibilities actualized through a particular combination of materials and techniques[3].

Manovich, L. (2013). *Software Takes Command* (p. 135). Bloomsbury Academic.

Manovich, L. (2013). *Software Takes Command* (p. 226). Bloomsbury Academic.

Manovich, L. (2013). *Software Takes Command* (p. 226). Bloomsbury Academic.

This raises a crucial question: how much autonomy do designers actually have within these tool environments? When a tool defines the parameters of a layout, a type choice, or a workflow, to what extent are we freely designing, and to what extent are we fulfilling the tool’s embedded logic? There is a growing recognition that software itself can carry politics, biases, and hidden assumptions — not only about aesthetic taste, but also about what design is and how it should function. Winner (1980) argues that what truly matters is not technology in isolation, but the social or economic system in which it is embedded[1]. This social shaping of technology suggests that technical devices and systems “contain possibilities for many different ways of ordering human activity,” encoding choices that affect how people work, communicate, and live[2]. Such choices, Winner contends, are never entirely neutral, because the adoption of a given technical system inevitably brings conditions for human relationships that have a “distinctive political cast”[3]. In this view, graphic design software cannot be separated from its social and economic context, and its structures — from default templates to built-in hierarchies of tools — shape design practices and constrain creative agency. Recognizing these hidden politics invites a more critical approach to hacking in graphic design, in which designers actively question, resist, or repurpose the embedded logics of their tools.

Winner, L. (1980). Do artifacts have politics? *Daedalus, 109*(1), 121–136.

Winner, L. (1980, p. 127).

Winner, L. (1980, p. 128).

Autonomy and Alternatives

In response to the constraints imposed by mainstream design software, parts of the graphic design community have explored alternative approaches such as open-source tools, creative coding, and self-built workflows. These practices resonate strongly with a hacker ethos, seeking to reclaim autonomy by developing tools rather than merely consuming them. Learning to code — even at a basic scripting level — can empower designers to modify or extend their environments, fostering a sense of agency otherwise denied by closed systems. Initiatives such as Luuse asbl illustrate this post-digital turn, positioning design practice within a political framework of openness, self-determination, and critical awareness. Luuse explicitly strives to develop “alternative methods of editing and publishing,” advancing a thoughtful, curious, and conscious relationship between production tools, designers, and users[1]. By operating through pedagogy, research, and commissioned work, Luuse supports a culture of shared knowledge and the co-creation of commons, challenging proprietary systems with a vision of free culture and open systems[2].

Luuse asbl. (n.d.). *About*. Retrieved from www.luuse.io/

This commitment to openness is deeply connected to a broader open-source movement, which represents a crucial pillar of hacker culture and embodies ideals of transparency, collaboration, and community-driven development. Far from being merely “free as in gratis,” open-source tools, inspired by the GNU philosophy and articulated by figures such as Richard Stallman (2002), emphasize the freedom to study, modify, and share code. This enables designers to break free from proprietary black boxes and engage actively with the tools they use. In doing so, open-source software fosters a culture of collective experimentation, knowledge-sharing, and mutual aid — values long associated with hacker movements. The existence of alternatives like GIMP, Scribus, or FontForge demonstrates that nearly every proprietary tool has a community-built counterpart, challenging the dominance of closed systems and expanding the range of possible practices. Moreover, open-source communities frequently cultivate rich peer-learning environments where designers support one another in developing skills, adapting tools, and experimenting with new workflows.

At the same time, it is important to recognize that open-source software and coding are not universal solutions to the complexities of creative autonomy. While these tools offer greater flexibility and potential for customization, they are also shaped by communities, conventions, and implicit ideologies. Furthermore, not every designer has the time, resources, or inclination to acquire technical expertise — nor should such expertise be a prerequisite for engaging in critical design practices.

This raises a central question: if neither proprietary software nor code alone guarantees true autonomy, what other paths exist? Perhaps what is needed is less about changing the tool, and more about changing the relationship to the tool. Rather than seeking complete mastery or total rejection, designers might cultivate what some call a “hacker mindset” — a playful, curious, critical stance toward familiar interfaces. Gentle misuse, improvisation, or even subtle friction within everyday workflows can challenge the illusion of seamlessness, and reveal hidden assumptions at work.

Ultimately, autonomy may not be a binary — either fully free or fully determined — but a spectrum of negotiation, in which designers remain aware of how tools shape them, and consciously decide when to follow, when to resist, and when to rewrite the rules.

EXPERIMENTS

Systems outside of me

This section documents and reflects on my first set of design experiments—the ones I later grouped under the category *systems outside of me*. I started here intentionally. I have no formal background in coding or programming, only minimal technical skills. Instead, I come from a fairly typical design trajectory: Western design education, a bachelor's degree, followed by agency work and commercial projects that prioritize efficiency, polish, and professionalism. For someone like me, the idea of "hacking"—even after reframing it in a broader, more open sense—can feel intimidating. But, in the spirit of hacking's hands-on imperative, I chose to begin from where I was: with the systems I know best. These are the foundational building blocks of graphic design—elements like type, grids, and images.

These systems are usually treated as fixed and given. They come with built-in rules, functions, and visual expectations. In applied practice, we use them—we lay them out, arrange them, polish them—but rarely question them beyond their aesthetic dimension. That's precisely what I wanted to challenge in this first phase. My aim was to use them differently, make them behave strangely, even "wrong." I wanted to break them open, interfere with their logic, and in doing so, expose their hidden structures.

Typeface

As an example; in one of the early experiments i worked with type. A typeface isn't just a set of letters—it's a system that enforces structure, clarity, and consistency. Behind every typeface is a tightly organized network of rules and relationships between glyphs, carefully designed to ensure readability and aesthetics.

Type is also a particular domain within graphic design. Type design is highly specialized—almost like a closed-off subculture. It's full of conventions, insider knowledge, hierarchies. There are "good" and "bad" fonts, designers often define their taste and status through their font choices. So messing with a perfectly designed, well-kerned, beloved, usable font felt… wrong. Like crossing some unspoken boundary. When I dragged a font file into my editor, it almost felt like as if I were breaking a moral code. This discomfort revealed another layer to the experiments—beyond the technical, they also touched on cultural and emotional dimensions. These aspects became increasingly present and will be explored more deeply in later stages of the work.

From a technical perspective, I focused not only on altering existing glyphs but also and mainly on experimenting with OpenType features. These features can be embedded directly into a font file by code, and are typically used for things like ligatures or alternate characters. I used them to rewrite how the typeface behaved: Letters changed depending on what was typed, where, and how. The font began to react, to misbehave—it responded to its own context. It stopped functioning as a neutral tool and began to act like an agent—bending the very rules it was designed to uphold. It no longer just delivered content, it performed and broke its own logic.

Grid, Margin – analogue

In another experiment i looked at grids. Especially within the Swiss design tradition, the grid holds a near-symbolic status. It is considered the fundamental basis, the very first thing you often do when setting up a design is defining a grid. It stands for order, structure, and organization. In this experiment, I took a contrarian approach by drawing layout grids by hand, analog. For me, this was a counter-practice: very simple, yet unfamiliar. Unfamiliar because I didn't measure anything, working imprecisely and in a way that ran counter to my normal habits. Usually, I work mainly in tools like InDesign, where grids are set up precisely, often tied to strict mathematical ratios. Here, I worked quickly, measurements were intuitive, and I even drew diagonal and uneven lines — something basically impossible in InDesign, which is not made for that kind of freedom.
The outcome was interesting: a completely new structure of grids emerged, one I would never have designed digitally. Yet what I noticed was that even though the grid's *aesthetic* had radically changed, it still retained its *function*: even if the layout appeared off or surprising, the grid still organized.

That observation left me with a question: wouldn't real subversion mean disabling the grid in its very function? Instead of producing order, could it actually create instability?

To explore this question further, I wrote a script using basil.js library that randomized the margins in InDesign, so that each newly added page had an entirely different layout logic. The grid—normally a symbol of precision and order—became an unpredictable element.

This experiment also marked a shift in approach—it was the first time I slipped behind a tool's intended use through code. It felt like I was gaining a new sense of autonomy and broader possibilities. That said, my coding skills alone wouldn't have been enough; tools like ChatGPT played a crucial role in supporting this process. I was still dependent—but in that dependence, something opened up.

analysis /conclusion experiments I

These early experiments led to — and reinforced — several important realizations. Even the most fundamental building blocks of my practice are far from neutral. They carry hidden suppositions, not just about aesthetics, but about functionality and how they are meant to be used. Like looking into a mirror, I became aware of my own habits, shaped by my design education and commercial practice, which taught me to approach and apply these elements in a specific way.

What these experiments also revealed is that adopting a hacking mentality does not have to begin with advanced skills or entirely new tools. It can start on familiar ground, by gently disrupting, creating friction, or rethinking things so close to us, we barely notice them anymore. In that sense, even the most rigid, external systems in graphic design are never fully closed. The assumptions they carry can be bent, misused, or reimagined.

This understanding raised a further question: what about the systems I don't even see? The ones I don't actively choose, but inherit — by opening a program, by repeating a workflow, by simply being a designer?

TOOLS

Systems I am Inside of

Tool and Workflow

The questions that concluded my first set of experiments — what about the systems I don't even see? — led directly into this next phase. These experiments shifted my attention from the systems I use (ouside of me) to the ones I inherit (i am inside of).

On the one hand side, these are systems embedded in the tools themselves: the software defaults, the workflows I repeat almost automatically. This tool dimension has preoccupied me for some time: how tools shape practice, how they become naturalized, how they subtly discipline the designer's thinking. What assumptions do I unknowingly adopt each time I launch a program? What do I accept as "neutral" when it is in fact a highly coded structure? (X)

The very first experiment emerged directly out of the previous phase — from working with randomized margins — and made me question other assumptions embedded in a program like InDesign. I turned my attention to its default settings. These defaults, like Minion Pro 12 pt, 12.7 mm margins, and standard page dimensions, are not neutral. They encode implicit values about hierarchy, legibility, and aesthetic normality. To surface these hidden frameworks, I wrote a script that changes these default settings every time a new document is created. Page size, margins, fonts, font sizes, and swatches all randomize with each new file. In this way, the experiment made these usually invisible defaults visible, critiquing them through their exposure.
From there, I attacked my own workflow more directly. I noticed my habits of perfectionism — endless fine-tuning and polishing. So I built a script (in AppleScript) that exports a PDF and then clears the entire InDesign document every 15 –30 minutes. This intervention sets a hard cutoff, blocking me from endless adjustment and encouraging me to accept the current state as enough — and to move on. I went further: sometimes my design work absorbs me completely, to the point of forgetting time and surroundings. So I introduced another small program, which interrupts me roughly every 30 minutes with a reflective prompt. The goal was not just to pause the work, but to interrupt the flow of optimization. Some interventions were very small, but still revealing. I disabled Cmd+Z in GIMP — no shortcut to undo. Weirdly, this felt like a restriction, but it shifted my entire mode of working: I became more deliberate, paying closer attention to each step.

Finally, I turned to the visual dimension itself: In Figma, I experimented by turning the opacity of my canvas to 0%, removing all visual feedback while still using the familiar tools and interface. The canvas was there, the tools were there — but no preview. It was a radical gesture of trust, or perhaps mistrust, toward my own sense of seeing. In that moment, design no longer relied on constant visual checking.

conclusion experiments II Tools and workflow

All these experiments challenged my habits and made me confront how deeply tools structure design practice: from smoothing interaction, to embedding hidden ideas about what is normal or valuable. They do not simply execute my thinking — they actively co-shape it, disciplining how I see, act, and make decisions as a designer. What mattered was not which tool I used — whether it was code, Photoshop, or GIMP — but recognizing that every tool carries its own assumptions and moral frameworks. Another Thing: along the way, I picked up more coding skills, driven by curiosity rather than obligation, which felt close to hacking's spirit: exploring, questioning, and learning by doing. In the end, these programs and scripts i wrote were not the answer, but more like provocations or jokes — playful ways of thinking about design, and about how to reimagine my attitude towards it.

(X)
Footnote/Loop
A short loop back: I had already explored this line of thinking in part during a project last year. I experimented with fictional design briefs, but forced myself to work by deliberately misusing tools — going against their intended purpose. For example, I designed a book entirely in Photoshop, a poster using only Glyphs, or an entire campaign through Illustrator's path view. This approach was immensely interesting: of course there was a visual dimension, with new and unexpected outcomes, but what fascinated me even more were the shifts in workflow, the new questions that emerged, and the alternative ways of thinking that opened up. Deliberately misusing a tool is incredibly powerful — it makes you sharply aware of your own habits and blind spots. I would recommend trying it.

PROTOCOLS AND INSTITUTIONS

Stand 02.07.

Prologue

To do the Graphic Design philosophy right, you have to be loyal to excellence. You have to believe that graphic design is a craft worth all the intelligence, creativity, and passion you can muster. (…) Graphic design and implementation should be a joyous art, a kind of high-level play. If this attitude seems preposterous or vaguely embarrassing to you, stop and think; ask yourself what you've forgotten. Why do you design graphic instead of doing something else to make money or pass the time? You must have thought graphic design was worthy of your passion once... To do the graphic design philosophy right, you need to have (or recover) that attitude. You need to care. You need to play. You need to be willing to explore. [1]

The Designer Attitude

1. THE WORLD IS FULL OF FASCINATING PROBLEMS WAITING TO BE SOLVED

Being a graphic designer is lots of fun, but it's a kind of fun that takes lots of effort. The effort takes motivation. Successful athletes get their motivation from a kind of physical delight in making their bodies perform, in pushing themselves past their own physical limits. Similarly, to be a graphic designer you have to get a basic thrill from solving problems, sharpening your skills, and exercising your intelligence. If you aren't the kind of person that feels this way naturally, you'll need to become one in order to make it as a graphic designer. Otherwise you'll find your energy is sapped by distractions like sex, money, and social approval. (You also have to develop a kind of faith in your own learning capacity – a belief that even though you may not know all of what you need to solve a problem, if you tackle just a piece of it and learn from that, you'll learn enough to solve the next piece – and so on, until you're done.)

2. NO PROBLEM SHOULD EVER HAVE TO BE SOLVED TWICE

Creative brains are a valuable, limited resource. They shouldn't be wasted on re-inventing the wheel when there are so many fascinating new problems waiting out there. To behave like a graphic designer, you have to believe that the thinking time of other graphic designers is precious – so much so that it's almost a moral duty for you to share information, solve problems and then give the solutions away just so other graphic designers can solve new problems instead of having to perpetually re-address old ones. Note, however, that "No problem should ever have to be solved twice." does not imply that you have to consider all existing solutions sacred, or that there is only one right solution to any given problem. Often, we learn a lot about the problem that we didn't know before by studying the first cut at a solution. It's OK, and often necessary, to decide that we can do better. What is not OK is artificial technical, legal, or institutional barriers (like closed-source code) that prevent a good solution from being re-used and force people to re-invent wheels. (You don't have to believe that you're obligated to give all your creative product away, though the graphic designers that do are the ones that get most respect from other graphic designers. It's consistent with graphic designer values to sell enough of it to keep you in food and rent and computers. It's fine to use your graphic design skills to support a family or even get rich, as long as you don't forget your loyalty to your art and your fellow graphic designers while doing it.)

3. BOREDOM AND DRUDGERY ARE EVIL
Graphic designers (and creative people in general) should never be bored or have to drudge at stupid repetitive work, because when this happens it means they aren't doing what only they can do – solve new problems. This wastefulness hurts everybody. Therefore boredom and drudgery are not just unpleasant but actually evil. To behave like a graphic designer, you have to believe this enough to want to automate away the boring bits as much as possible, not just for yourself but for everybody else (especially other graphic designers). (There is one apparent exception to this. Graphic designers will sometimes do things that may seem repetitive or boring to an observer as a mind-clearing exercise, or in order to acquire a skill or have some particular kind of experience you can't have otherwise. But this is by choice – nobody who can think should ever be forced into a situation that bores them.)

4. FREEDOM IS GOOD
Graphic designers are naturally anti-authoritarian. Anyone who can give you orders can stop you from solving whatever problem you're being fascinated by – and, given the way authoritarian minds work, will generally find some appallingly stupid reason to do so. So the authoritarian attitude has to be fought wherever you find it, lest it smother you and other graphic designers. (This isn't the same as fighting all authority. Children need to be guided and criminals restrained. A graphic designer may agree to accept some kinds of authority in order to get something he wants more than the time he spends following orders. But that's a limited, conscious bargain; the kind of personal surrender authoritarians want is not on offer.) Authoritarians thrive on censorship and secrecy. And they distrust voluntary cooperation and information-sharing – they only like ‘cooperation’ that they control. So to behave like a graphic designer, you have to develop an instinctive hostility to censorship, secrecy, and the use of force or deception to compel responsible adults. And you have to be willing to act on that belief.

5. ATTITUDE IS NO SUBSTITUTE FOR COMPETENCE
To be a graphic designer, you have to develop some of these attitudes. But copping an attitude alone won't make you a graphic designer, any more than it will make you a champion athlete or a rock star. Becoming a graphic designer will take intelligence, practice, dedication, and hard work. Therefore, you have to learn to distrust attitude and respect competence of every kind. Graphic designers won't let posers waste their time, but they worship competence – especially competence at designing, but competence at anything is valued. Competence at demanding skills that few can master is especially good, and competence at demanding skills that involve mental acuteness, craft, and concentration is best. If you revere competence, you'll enjoy developing it in yourself – the hard work and dedication will become a kind of intense play rather than drudgery. That attitude is vital to becoming a graphic designer. [2]

Introduction

What you just read was the first experiment–the first hack– in a series of hacks that I conducted in the course of this master thesis. It emerged spontaneously, out of a moment of play, while i was researching for the theoretical part of this project. I took two fundamental texts from hacker culture–written by Eric S. Raymond, a central figure in the early hacker scene–and replaced every instance of the word hacker with graphic designer. A simple, almost lazy intervention: two lines of code in the browser console. But this small act led to several realizations.
First, it confirmed an intuition I already had when choosing hacking as the subject for my final project: that there are parallels between the mindset of hackers and that of (to me, ideal) graphic designers. Second, it revealed the instability of authorship–and how meaning can be radically shifted through minimal interventions. And finally, it positioned text not as fixed content, but as a system–structured, yet hackable. Why this matters, I'll elaborate on shortly.

What am I looking for?

But first: How did I even get here?
The Process that finally led to this thesis started out of a feeling of frustration. Frustration with the roles we, as graphic designers, are expected to fit into. We're often offered fixed identities–service provider, author, researcher, storyteller…
While these labels may help define our function within certain contexts, none of them ever fully captured what I believed design could be. I had the feeling that the practice of graphic design has more potential than just executing briefs, producing outcomes, or explaining things.
I started to look for a different perspective–a different way of relating to design. A way that's less about fitting into predefined frameworks, and more about questioning, rethinking, and opening them up.
That's when I turned to hacking.

Hacking as Philosophy and Method

This thesis explores what happens when the mindset and methods of hacking are applied to the field of graphic design. Not hacking in the narrow sense of cybercrime or even just coding, but in a broader, cultural sense: as it was shaped in the MIT hacker scene of the 1960s and 70s. In that context, hacking is not just technical skill–it's a playful, curious, and subversive way of engaging with systems. It's about understanding how things work, so you can make them work differently.
This attitude offered me a new framework, one that felt both more honest and more expansive. It resonated with the kind of relationship I wanted to have with graphic design–not just as a producer of outcomes, but as someone who can question, reconfigure, and resist systems that are taken for granted.

What is a System?

As many times as I've used the word «system» by now, it's worth pausing to ask: what exactly do I mean by system in the context of this thesis?
I return to the first experiment. By replacing the word «hacker» with «graphic designer», I treated the text not as fixed content, but as a structure–something with internal logic, dependencies, and rules. In

other words: a system. This small intervention served as a kind of seed for the entire project. The central methodological assumption of this thesis is this: Everything is a system. This broad definition includes not only texts, but also tools and software, workflows, routines, habits, professional norms and institutional structures–even less tangible things like identity, authorship, or ego. In short, any structured set of roles, rules, habits, or relationships. If it has a pattern, it can be understood. And if it can be understood, it can be reconfigured, it can be hacked. This perspective allows hacking to become a design method–one that treats existing conditions not as fixed constraints, but as materials to be investigated, questioned, or subverted.

Hacking as Framework

Before diving deeper, it's important to clarify: this thesis is not about hacking per se. It is about the practice of graphic design. Hacking serves here as a conceptual framework, a method–one that could offer designers a tool for thinking critically and working experimentally within their practice. It is the lens through which I investigate graphic design practice, and the method I use to conduct that investigation.

The thesis itself is structured as a series of experiments, each one examining a different system I interact with as a designer. Following a progression and structure that emerged naturally during the process, I began my experiments with what I summarize under the term «systems outside of me». These are the elements that I–as most designers–encounter first: for example, type, grids, and images. Then I moved on to systems I am inside of: tools, workflows, institutions; and finally, I arrived at systems I am: ego, authorship, identity.
The method I use is intentionally self-reflective and iterative. Each experiment begins with a theoretical idea, which leads to a practical intervention. The outcomes of that experiment are then analyzed and reflected upon–often raising new questions. These questions lead back to theory, which in turn informs the next experiment. In this way, the process forms a series of loops rather than a straight, linear path. This also provides the overall structure of the work–including the one of the written part. A crucial part of my investigation was the constant back-and-forth between thinking and making, between theory and practice. I aim to reflect this interplay within the written thesis itself. For that reason, there is no strict separation between a "theoretical" and a "practical" part. Instead, I directly connect hacking theory to design practice–and then describe the experiments that emerged from that connection. Even this structure is an experiment–an attempt to hack the habits, conventions, and institutional expectations I am familiar with.

What is hacking and Experimental setup

This section situates hacker culture within the context of design, establishing a working definition of hacking to support the exploration of systems in graphic design.
The term *hacker* is highly context-dependent and often misunderstood. Popular clichés—shaped by politically charged campaigns in the 1980s—portray hackers mainly as criminals breaking into computer systems. The Cambridge Dictionary reflects this narrow view, defining a hacker as “someone who gets into other people's computer systems without permission in order to find out information or to do something illegal.”[3]
However, this thesis draws on a different understanding, rooted in Steven Levy’s *Hackers: Heroes of the Computer Revolution* (1984)[4]. Levy traces hacking’s origins to playful, creative experimentation by MIT’s Tech Model Railroad Club in the late 1950s and early 1960s, where members sought innovative ways to enhance model train systems using government-funded computers. This ethos—reimagining functions and exploring alternative uses—is central to the hacker mindset. One of the most influential figures in this context is Richard Stallman, founder of the Free Software Movement. In his essay On Hacking, Stallman reflects on the nature of hacking and writes: “It is hard to write a simple definition of something as varied as hacking, but I think what these activities have in common is playfulness, cleverness, and exploration. Thus, hacking means exploring the limits of what is possible, in a spirit of playful cleverness.” [5] For Stallman, hacking is not limited to code–it’s a form of expression. He even considers John Cage’s musical piece 4'33" as hacking. It challenges conventional expectations of what music is, in a way that is both clever and thought-provoking. As Stallman puts it, “Playfully doing something difficult, whether useful or not–that is hacking.”[6]
Hacking is always about systems. It requires entering into something that already exists, deconstructing it, and making it do something else. John Draper, better known as “Captain Crunch”, and a pioneer of so-called phreaking (hacking telephone networks), puts it plainly: “I’m learning about a system. The phone company is a system. A computer is a system, do you understand? If I do what I do, it is only to explore a system. Computers, systems–that's my bag. The phone company is nothing but a computer.”[7] This systemic perspective is my key. Whether we're talking about telephony, software, publishing platforms, or design workflows, hacking means critical engagement with structure. It is less about producing finished outcomes and more about asking: How does this work? What else could this be?

Finally, hacking is a meritocracy. It prizes skill, rigour, and execution over status. Hacking is hard work–but it is work pursued as play, as challenge, as joy.
Looping back to the prologue, these points form what Raymond calls “The Hacker Attitude.”[8] For this project, they provide a useful frame for thinking about hacking and therefore design, as a form of systemic engagement: a way of intervening, reimagining, and repurposing structures, tools, and ideas. Hacking, in this expanded sense, becomes a method–a speculative, critical, and often playful practice of engaging with the world as a system that is always open to change. Importantly, these attributes of the hacker attitude

also say a lot about character and personality–highlighting qualities such as curiosity, resilience, openness, and a love for challenge–which are as essential to the practice as technical skill.

Experimental framework and setup

The exploration and familiarization around the term «hacking» laid the foundation for the practical part of this project: the experiments. The first step was to build a documentation system. Derived from hacker ethics, this meant committing to radical transparency, a principle deeply embedded in hacker culture. Documentation is not an afterthought here; it is central. Every experiment, every mistake is recorded and made publicly accessible. To enable this, I created a website hosted via GitHub Pages, which functions as a GitHub repository. A repository is an openly accessible directory of files–typically source code–used by developers (and hackers) to collaborate, share, and iterate on projects. One of GitHub's core features is version control, meaning every change is tracked. Most importantly, the code is visible: anyone can view it, copy it, and build upon it.
The website itself consists of basic HTML files, intentionally kept minimal to remain accessible and easily modifiable. The main content is the documentation of the experiments. Each experiment includes: A short description, a step-by-step protocol, detailing every action taken and every source referenced and the resulting output, which may include scripts, PDFs, or other downloadable artifacts. The idea is that every experiment should be replicable and forkable–in the same spirit as open-source projects. Anyone can repeat the process, remix it, or take it in a new direction.

As a starting point, I set three values that each experiment must fulfill. These are drawn from Eric S. Raymond's already introduced text The Hacker Attitude and function as a kind of framework:

- **Playfulness** – A sense of joy and curiosity is central. Hackers are intrinsically motivated; they follow intuition, embrace humor, and aren't afraid to break things—in fact, they often break things on purpose to see what's possible. This mindset encourages risk, surprise, and unplanned directions.
- **Learning** – Each experiment should generate insight. Whether technical, conceptual, or process-related, every hack is an opportunity to understand something new.
- **Transparency** – Nothing is hidden. Every step, source, decision, and failure is documented and made accessible. The process is at least as important as the outcome.

IMG_ScreenshotWEB

How Hackers Work: Culture

In the previous chapter, I explored hacking's values and ethics as historically and culturally developed principles, relating them to my project's concept. This chapter sharpens that understanding by focusing on hacking as a way of working—examining how hackers operate in practice and where their modes of work might overlap with or challenge those of designers.

A key influence here is Pekka Himanen's *The Hacker Ethic and the Spirit of the Information Age*. Himanen moves beyond the common technical image of hacking, framing it as an alternative work ethic that challenges traditional ideas of labor, productivity, and value. Himanen contrasts this hacker ethic with Max Weber's Protestant work ethic, which emphasizes discipline, duty, and delayed gratification. In contrast, hackers are driven by passion rather than obligation. As Himanen writes, "For hackers, passion describes the general tenor of their activity"[9]. This intrinsic motivation is central: hackers work not because they must, but because they want to. Their actions are propelled by joy, excitement, and intellectual curiosity.

This passion also shapes hackers' relationship with time. Whereas the Protestant ethic organizes life around regular, repeated working hours—a legacy of medieval monastic schedules—hackers reject this model. Himanen illustrates this with Linus Torvalds, who famously coded late into the night and woke in the early afternoon to continue working or simply play with his computer. Such rhythms prioritize individual autonomy over fixed schedules, diverging sharply from the dominant 9-to-5 culture, which remains remarkably persistent today despite changes in society.

Himanen also identifies a “money ethic” in hacker culture: social recognition and peer respect within a passionate community outweigh financial motives. Recognition is earned through contribution, skill, and sharing knowledge, making hacking a profoundly social and meritocratic practice[10].

The second text shaping my thinking is Anja Groten’s “Hacking & Designing: Paradoxes of Collaborative Practice.” Groten, co-founder of Hackers & Designers, shares many of Himanen’s perspectives but offers a more critical lens on designers who adopt hacking methods superficially. Through a fictional dialogue between a stereotypical hacker and a designer, she reveals tensions within hacker culture. The hacker insists that hacking is not “a method you can first learn and then apply.” Rather, it is a practice grounded in technical literacy, constant failure, and social collaboration. As the hacker says, “You cannot learn hacking like you would learn a skill, a subject, or a method. Hacking derives from and contributes to an ecology.”
Groten also highlights the essential role of frustration in hacking. Hackers develop “a tremendous tolerance to frustration,” as their work often involves broken code, obscure bugs, and experimental problem-solving. Mastering the balance between failure and pleasure becomes an expertise itself.

Summary

Experiments Phase I:

Systems outside of me

This section documents and reflects on my first set of design experiments—the ones I later grouped under the category *systems outside of me*. I started here intentionally. I have no formal background in coding or programming, only minimal technical skills. Instead, I come from a fairly typical design trajectory: Western design education, a bachelor’s degree, followed by agency work and commercial projects that prioritize efficiency, polish, and professionalism. For someone like me, the idea of “hacking”—even after reframing it in a broader, more open sense—can feel intimidating. But, in the spirit of hacking’s hands-on imperative, I chose to begin from where I was: with the systems I know best. These are the foundational building blocks of graphic design—elements like type, grids, and images. These systems are usually treated as fixed and given. They come with built-in rules, functions, and visual expectations. In applied practice, we use them—we lay them out, arrange them, polish them—but rarely question them beyond their aesthetic dimension. That’s precisely what I wanted to challenge in this first phase. My aim was to use them differently, make them behave strangely, even “wrong.” I wanted to break them open, interfere with their logic, and in doing so, expose their hidden structures.

Typeface

As an example; in one of the early experiments i worked with type. A typeface isn't just a set of letters—it's a system that enforces structure, clarity, and consistency. Behind every typeface is a tightly organized network of rules and relationships between glyphs, carefully designed to ensure readability and aesthetics.
Type is also a particular domain within graphic design. Type design is highly specialized—almost like a closed-off subculture. It's full of conventions, insider knowledge, hierarchies. There are "good" and "bad" fonts, designers often define their taste and status through their font choices. So messing with a perfectly designed, well-kerned, beloved, usable font felt… wrong. Like crossing some unspoken boundary. When I dragged a font file into my editor, it almost felt like as if I were breaking a moral code. This discomfort revealed another layer to the experiments—beyond the technical, they also touched on cultural and emotional dimensions. These aspects became increasingly present and will be explored more deeply in later stages of the work. From a technical perspective, I focused not only on altering existing glyphs but also and mainly on experimenting with OpenType features. These features can be embedded directly into a font file by code, and are typically used for things like ligatures or alternate characters. I used them to rewrite how the typeface behaved: Letters changed depending on what was typed, where, and how. The font began to react, to misbehave—it responded to its own context. It stopped functioning as a neutral tool and began to act like an agent—bending the very rules it was designed to uphold. It no longer just delivered content, it performed and broke its own logic.

Grid, Margin

In another experiment i looked at grids. Especially within the Swiss design tradition, the grid holds a near-symbolic status. It is considered the fundamental basis, the very first thing you often do when setting up a design is defining a grid. It stands for order, structure, and organization. In this experiment, I took a contrarian approach by drawing layout grids by hand, analog. For me, this was a counter-practice: very simple, yet unfamiliar. Unfamiliar because I didn't measure anything, working imprecisely and in a way that ran counter to my normal habits. Usually, I work mainly in tools like InDesign, where grids are set up precisely, often tied to strict mathematical ratios. Here, I worked quickly, measurements were intuitive, and I even drew diagonal and uneven lines — something basically impossible in InDesign, which is not made for that kind of freedom.
The outcome was interesting: a completely new structure of grids emerged, one I would never have designed digitally. Yet what I noticed was that even though the grid's *aesthetic* had radically changed, it still retained its *function*: even if the layout appeared off or surprising, the grid still organized. That observation left me with a question: wouldn't real subversion mean disabling the grid in its very function? Instead of producing order, could it actually create instability?
To explore this question further, I wrote a script using basil.js library that randomized the margins in InDesign, so that each newly added page had an entirely different layout logic. The grid—normally a symbol of precision and order—became an unpredictable element. This experiment also marked a shift in approach—it was the first time I slipped behind a tool's intended use through code. It felt like I was gaining a new sense of autonomy and

broader possibilities. That said, my coding skills alone wouldn't have been enough; tools like ChatGPT played a crucial role in supporting this process. I was still dependent—but in that dependence, something opened up.

Conclusion experiments I

These early experiments led to — and reinforced — several important realizations. Even the most fundamental building blocks of my practice are far from neutral. They carry hidden suppositions, not just about aesthetics, but about functionality and how they are meant to be used. Like looking into a mirror, I became aware of my own habits, shaped by my design education and commercial practice, which taught me to approach and apply these elements in a specific way. What these experiments also revealed is that adopting a hacking mentality does not have to begin with advanced skills or entirely new tools. It can start on familiar ground, by gently disrupting, creating friction, or rethinking things so close to us, we barely notice them anymore. In that sense, even the most rigid, external systems in graphic design are never fully closed. The assumptions they carry can be bent, misused, or reimagined.
This understanding raised a further question: what about the systems I don't even see? The ones I don't actively choose, but inherit — by opening a program, by repeating a workflow, by simply being a designer?

The Politics of Tools & Experiments Phase II : Systems outside of me

In graphic design, tools have never been mere neutral instruments; rather, they fundamentally shape what can be made, how it is made, and even what designers imagine as possible. Historically, entire design paradigms have been determined by the constraints and affordances of available tools, from the precision and repeatability of the printing press to the experimental flexibility offered by phototypesetting and later digital page layout systems. In his 2014 work, media theorist Lev Manovich argues, contemporary software does not simply execute design but actively embodies values, standards, and cultural assumptions that guide creative practice. Adobe Creative Suite, for example, has become almost synonymous with graphic design itself, its functions and interfaces deeply embedded in design education, professional workflows, and aesthetic conventions.
Manovich situates these tools within a broader conception of medium as a cultural-technical system that frames what designers can even think to do. As he observes, "mediums as they are implemented in software are part of distinct cultural histories that go back for hundreds

and often thousands of years”[11] which continue to shape how we understand and use them today. Thus, a medium is far more than a set of technical materials or tools; it constitutes what Manovich calls an “imaginary database” of expressive, compositional, and communicative possibilities actualized through a particular combination of materials and techniques[12].

Hacking as Intervention: Political and Artistic activism

Bibliography

---

[1] Adapted from Raymond, E. S. (2003). The art of Unix programming. Retrieved from http://www.catb.org/esr/writings/taoup/html/ch01s09.html

---

[2] Adapted from Raymond, E. S. (2001). How to become a hacker, section « The Hacker Attitude » Retrieved from http://www.catb.org/esr/faqs/hacker-howto.html#what_is

[3] Hacker. (n.d.). In Cambridge Dictionary. Retrieved July 2, 2025, from https://dictionary.cambridge.org/dictionary/english/hacker

[4] Steven Levy, Hackers: Heroes of the Computer Revolution (New York: Anchor Press/Doubleday, 1984).

[5] Stallman, R. (n.d.). On hacking. Retrieved April 25, 2025, from https://stallman.org/articles/on-hacking.html
[6] Stallman, R. (n.d.). On hacking. Retrieved April 25, 2025, from https://stallman.org/articles/on-hacking.html
[7] Rosenbaum, R. (1971, October). Secrets of the little blue box. Esquire Magazine. Republished by Slate.
[8] Raymond, E. S. (2001). How to become a hacker, section « The Hacker Attitude » Retrieved from http://www.catb.org/esr/faqs/hacker-howto.html#what_is
[9] Pekka Himanen, *The Hacker Ethic and the Spirit of the Information Age* (New York: Random House, 2001), p. 18.
[10] Pekka Himanen, *The Hacker Ethic and the Spirit of the Information Age* (New York: Random House, 2001), p. 51.
[11] Lev Manovich, *Software Takes Command* (New York: Bloomsbury Academic, 2013), p. 226.
[12] Lev Manovich, *Software Takes Command* (New York: Bloomsbury Academic, 2013), p. 226.

Hacking as Intervention

Hacking as Intervention: Political and Artistic activism

Experiments III: Hacking the Institution

Until now, I have explored what hacking might mean beyond its purely technical framing, asking how hackers work, what values they embody, and how those values relate to graphic design tools and practices. Much of the current discourse around hacking in graphic design—such as that seen in collectives like Hackers & Designers or Luuse—often foregrounds the technical dimension: open-source tools, collaborative coding, and experimental infrastructures. While these aspects are vital and often inherently critical (e.g., rejecting monopolistic software, advocating for transparency), they can also feel alienating to classically trained or applied designers, for whom coding and critical making are not primary practices.

Yet there is more to hacking than technological or aesthetic gestures. Its ethical stance, its potential for resistance, and its capacity for cultural and institutional critique are equally significant. That is why, in this section, I want to zoom out—tracing how hacking's values have extended into activist and artistic practices that use intervention not merely as a technical tactic, but as a political gesture. These approaches frame hacking as a mode of critique that confronts systems of power, authorship, and control.

By shifting the focus away from tool fetishism and toward systemic questioning, these practices offer models for how designers might adopt hacking strategies that are both critical and accessible—strategies that don't require full technical immersion, but still challenge dominant norms and institutional frameworks.

Tactical Media

A useful starting point is the tradition of tactical media, which emerged in the 1990s. Tactical media can be understood as a form of temporary, strategic intervention, working with and around new media. It interrupts existing power structures and semiotic regimes by repurposing them, offering participants fresh ways of seeing, understanding, and interacting with the infrastructures that govern their lives.

My favourite example is the activist duo The Yes Men, who became known for spectacular interventions at global trade conferences, posing under false names as representatives of powerful corporations or organizations they deemed exploitative. They engineered public-relations disasters for their targets by grotesquely exaggerating their positions, or by performing sudden moments of "enlightenment" on their behalf.

One of their most striking actions was their *Bhopal news hijacking*: in 2004, posing as a Dow Chemical spokesperson, they appeared live on BBC World and announced that Dow would finally accept full responsibility for the 1984 Bhopal disaster — a gas leak that killed thousands in India — and pay 12 billion dollars in compensation to victims. The BBC, taking this at face value, broadcast the announcement globally, sending Dow's stock price into free fall before the hoax was revealed. In a single stroke, the Yes Men forced the world to

confront Dow's ongoing refusal to make reparations, exposing the company's inaction through a carefully constructed fiction[1].
As Fluter magazine describes, the Yes Men "created an alternative way of thinking. With their work, they show us that the reality around us is not fixed — it can be changed, if we act."[2] For me, this is crucial: tactical media uses humor, fiction, and provocation as hacks, disturbing dominant narratives and showing that systems are malleable.
There are many more compelling definitions and interpretations of tactical media—Geert Lovink's ABC of Tactical Media[3], in particular, is a rich resource for those who want to delve deeper.

What is also crucial for this thesis, however, is how tactical media resists rigid definitions altogether—refusing to fix identities, roles, or disciplines. Artists, scientists, technicians, activists, and designers can all inhabit tactical practices simultaneously.
As the *Critical Art Ensemble* argues, "In either case, such role designations are too restrictive in that the role boundaries exclude access to social and knowledge systems that are the materials for their work."[4] Tactical media challenges these boundaries not by erasing them, but by refusing to let them dictate participation. It privileges access over expertise, encouraging collective and adaptive forms of practice.
What does this mean for design practice today—and for this thesis specifically? To me, this could suggest a move away from predefined specializations like "print designer" or "UX designer," and toward a more fluid, open-ended understanding of what a designer can be and do. Tactical media's refusal to fix roles opens space for a practice that is driven less by job titles and more by context, intention, and collaboration.
Could stepping outside fixed roles allow designers to operate more tactically—moving between institutions, tools, and communities as needed, rather than being bound to a single disciplinary identity? These are the kinds of questions that emerge when design is approached not as a profession with stable boundaries, but as a practice always in flux, shaped by its entanglements with technology, politics, and culture.

Poetic hacks
Another fascinating reference point is !Mediengruppe Bitnik, an artist collective from Switzerland whose works can be read as poetic hacks that reveal and critique systems from within. Rather than simply breaking or bypassing systems, they slip inside them, subverting expectations from the inside out. A well-known example is their *Delivery for Mr. Assange* project: in 2013, they sent a parcel containing a hidden live-streaming camera to Julian Assange inside the Ecuadorian Embassy in London[5]. Over the course of its postal journey, the camera broadcast its own progress in real time ..
What is crucial here is that !Mediengruppe Bitnik recognized the postal system as a system in its own right: a network of rules, procedures, and flows that could be appropriated, exploited, and reimagined. They leveraged the system's predictability and trust to create a moment of radical visibility, exposing how infrastructures can be made to serve unintended functions.

Their work exemplifies how hacking-inspired interventions can probe infrastructures not merely by attacking or dismantling them, but by *inhabiting* them creatively. They open up hidden layers of systems — in this case, postal tracking, international security, and diplomatic asylum — to public reflection.

What resonates with me is how !Mediengruppe Bitnik expand hacking beyond technical exploits. They transform hacking into an aesthetic and critical practice, using the unexpected to produce ruptures in everyday systems of control. In this sense, they invite designers to think of hacking not just as a means to modify tools, but as a broader cultural attitude of revealing, questioning, and shifting the parameters of what is assumed to be stable.

Metahaven
Metahaven is a research-driven design studio founded by Vinca Kruk and Daniel van der Velden in the Netherlands. Emerging from a background in graphic design, their work has evolved into a multidisciplinary practice that blends theory, activism, branding, and filmmaking. From the start, their work has pushed at the boundaries of what design can do—not just how it looks, but how it thinks, intervenes, and constructs meaning.

A key example is their 2010 book Uncorporate Identity, a hybrid of theory, visual essay, and speculative branding. Unlike traditional design monographs, it takes the form of a nonlinear research publication—one that explores the aesthetics and geopolitics of identity in a post-national, post-branding world. What makes *Uncorporate Identity* especially relevant here is its repositioning of design as a method of inquiry rather than a neutral service. It asks: how does design participate in systems of power? And how might it visualize, disrupt, or reconfigure those systems?

Working with case studies such as Wikileaks, Sealand, or the global PR aesthetics of statehood, Metahaven appropriate the language of branding not to sell, but to expose and destabilize. This resonates with key themes of this thesis: the tension between tool and critique, between default aesthetics and the political realities they obscure. Metahaven's refusal of design as a closed profession aligns with hacker values: openness, disobedience, experimentation. They move through roles—researchers, designers, theorists—not by mastering each domain, but by remaining curious and deliberately unspecialized.

In that sense, Uncorporate Identity can be read as a tactical media project in itself. It doesn't "represent" resistance—it performs it, both visually and structurally. Like hacking, it intervenes in existing systems (branding, national identity, network infrastructure), using their logics against themselves.

For my thesis, Metahaven serve as an important example of how graphic design can operate critically without reducing itself to either aesthetic commentary or technical specialization. Their work exemplifies a mode of practice that is research-based, politically engaged, and deeply aware of the infrastructures it inhabits.

making a slight loop back to design. (who is metahaven?)

Captives of the Cloud: Part I

https://www.e-flux.com/journal/37/61232/captives-of-the-cloud-part-i
they cite a lot of specific things, seem to work deep into a field of interest

White Night
Before A Manifesto
May 2008
https://readings.design/PDF/metahaven_whitenight.pdf
p.6: Surface is the reincarnation of neutrality. Default friends, default faces, default desktops, default writing. In the world of surface, the confrontation with harsh realities, such as having no face, or no friends, becomes mediated and softened by the presence of placeholders,

p.6 (loop to tools): Software does precisely what its name spells out: it softens the relationship between man and manufacture. Writing, visiting friends, searching, finding, saving: what once required at least some physical activity becomes extremely light, pleasant and effortless

p.7:when products generate needs, when
needs trigger speculative value and when values are embodied by products, we can no longer speak of pure consumption, as consumption itself becomes a productive force, as was already the case already for Marx.
p.7: Designers – either by marketing or by fiction – perpetually innovate the seductive regime of surface, which stimulates other designers to do the same thing, disconnected from the non-negotiability of the brutal material ground, historical structure and political struggles on which, originally, surface itself was premised
p.12: In all these transformations, an economy of design objects is implied; objects are simultaneously lifted from their origin, tradition, space, time, use-value, and exchange-value, in order to assume maximum agility in the aggregration of new needs.
Design must be invested with the potential, the intelligence and the tools to break down the new borders it has created by being borderless. It must be invested with the energy to break through the seamless surfaces of fictitious virtue which have become the new walls of the free world.
p.15: It places no emphasis on design as a professional
activity but instead pursues mistakes, nights without sleep, uncool work, messy desktops, and laughter.

first things first manifesto: We, the undersigned, are graphic designers, art directors and visual
communicators who have been raised in a world in which the techniques and apparatus of advertising have persistently been presented to us as the most lucrative, effective and desirable use of our talents. Many design teachers and mentors promote this belief; the market rewards it; a tide of books and publications reinforces it.'
https://www.readingdesign.org/first-things-first

p.18: A new common ground for designers and users is provided by the changing links between production and consumption, of which immaterial labour is the ‘interface’. The products of immaterial labour not only materialize ‘needs, the imaginary, consumer tastes, and so forth’, but also generate and produce new needs, imaginaries, and tastes, so that the act of consumption is not the destruction of the commodity but the establishment of a relationship which links production and consumption (read: designer and user) togethe

they loop back to GNU / Hacking!!:
p.19: The GNU
Manifesto, written by Richard Stallman in 1985: ‘I consider that the golden rule requires that if I like a program I must share it with other people who like it. Software sellers want to divide the users and conquer them, making each user agree not to share with others. I refuse to break solidarity with other users in this way. I cannot in good conscience sign a nondisclosure agreement or a software license agreement. ‘

the sprawl
poetic documentation? also pov of designer. its abt images

EXPERIMENTS

After exploring tools and workflows, I found myself wanting a broader frame—one where I could test what hacking might look like beyond the technical or aesthetic. The idea of protocols gave me that. By protocols, I mean the soft and hard norms that shape how we move through institutions, infrastructures and also the profession itself: the unwritten rules, what we are used to in beeing educated as a designer, in what we call ourself a designer.

In this section, I tried to treat these structures like design material—open to misinterpretation, exaggeration, subversion. The scale remained small, personal, and situated, the aim was not to break, but to bend formats that are usually taken for granted and fixed. Could they be reimagined as spaces of intervention? What happens when you treat institutional constraints not just as background, but as a foreground to work on?

The goal wasn’t to critique from the outside, but to operate *within* existing protocols while. They weren’t about proving a point. Instead, they became small rehearsals in reclaiming authorship over the conditions in which design happens. Attempts to rewire familiar systems—not for destruction, but to surface the assumptions they carry, and imagine how things could be otherwise.

The first experiment in this new phase emerged from a kind of blur: an in-between gesture, moving from disrupting habits in tools toward questioning the frameworks that structure design at a deeper level. Still curious about visibility, defaults, and assumptions, I wanted to

design a zine without relying on conventional layout software. Inspired by Luuse's "designing without seeing" approach and by broader hacker practices, I decided to work only with a code editor and browser.

I was not confident coding for print—or even coding a zine at all—but hacking, I had learned, often begins with copying. So, I started with a file I didn't write: a public-domain book downloaded as raw HTML from Project Gutenberg, an online archive of digitized literature. I treated the HTML structure as scaffolding, inserting my own content and editing only the inline CSS. My preview tool was the browser's native print dialog. No layout grid, no canvas, no InDesign or Scribus. Just: insert my content → tweak code → save file → check print preview → repeat. Pagination, margins, page size—all of it was controlled through the browser's print settings. I leaned into what the system offered by default. The experiment wasn't only about making a zine in an unconventional way—it became a moment of stepping outside WYSIWYG, outside the smoothing logic of layout software, and into a mindset that feels closer to scripting, copying, and gently misusing what's already there.

I printed everything directly from the browser, for paper, I used samples from Fischer Papier—a large supplier for our school. You can order up to ten free A4 samples per paper type, so I pieced the whole zine together from these fragments. The final prints came from our school's free (and slightly terrible) laser printer. This low-budget, patchwork setup felt fitting: an experiment not just in designing differently, but in hacking the entire pipeline– also questioning what's "enough" to make a designed object, and where design truly happens.

Next, I shifted focus more directly towards the institutional protocols.One experiment in this direction was my midterm presentation. Normally, we're expected to be physically present, to show up and speak live. I wasn't. Instead, I made a video and asked a friend (thank you, Kate) to secretly swap slots with me on the schedule. People expected her to present—but instead, she played my video. It was a small surprise, a switch, a performance.

This might not sound radical, but for me it was an experiment in multiple ways. Not only did I subvert the standard presence protocol, but I also deliberately broke with the visual language we're taught to use for presenting.

I started with conventional slides—clean, beautiful, expected—and then exited the presentation mode altogether. I opened the text editor and started typing live what I was also saying partially aloud. I recorded my screen googling definitions, opening tabs, using my process documentation website instead of polished Keynote slides. This was about aesthetics, yes, but also about control. I also handed over control: Kate had the video, the MS Teams link, everything. I didn't even attend the whole session—just dropped into the feedback via call (a part where I stayed conventional, admittedly). That small act of letting go, of not overseeing every detail, was part of the experiment too.

Something I already hinted at in the introduction—is that the written part itself is an experiment. There's no strict separation between theory and practice; the two are entangled, feeding into each other throughout. Of course, this isn't radical or forbidden, but it does go against how I was trained to think about academic writing. Like hacking, my method follows a loop: theory informs experiment, experiment shifts perspective, and reflection loops back into theory. This creates a non-linear structure, where ideas emerge through doing, and documentation becomes part of the work rather than a record of it.

A key part of this is also how I write: publicly, live, and transparently. Instead of working in isolation and only revealing a final, polished result, I'm drafting in a shared Google Doc—visible in real time to anyone with the link. Writing becomes traceable, open to interruption, and stripped of its usual authority.

Even more structurally, I finished the practical part of my thesis before writing the theoretical one. Again; that's not forbidden, but it's not typical in design education. In most design institutions I know, it's the other way around — first you write, then you produce. The written part usually comes first to frame, justify, or theorize the practice that follows. But for me, that order didn't make sense, I didn't want theory to predefine or narrow down my experiments too early. Letting the work lead meant I could stay open, avoid fixing my direction too soon, and allow ideas and areas of interest to emerge *through* the act of making.

And on a more general level, I see this project as a kind of mirror, or meta-reflection, of the Master's programme itself. Many of the things one could critique about this thesis — a lack of clear focus, doing a bit of everything, not always knowing what the concrete goal is — are the same things people might say about the programme. It touches on a little coding, a bit of aesthetic exploration, some critical theory, some practice. From the outside, it might seem scattered or superficial. But to me, this thesis is the logical consequence of this study programme. It's shaped by its openness, its ambiguity, its refusal to fix what design should be. Whether you see that as a strength or a weakness is a matter of perspective, personally, I loved this programme. It confused me, stretched me, gave me space to get lost — and that's exactly what I needed. This thesis is both a product of that space and a reflection on it.

Then, another dimension of these experiments began to unfold—this time not around tools or institutions, but around the practice itself: feelings. From a certain point on, I started including a small section after each experiment where I described how I felt while doing it—what excited me, what frustrated me, what I liked, and what I didn't. It sounds simple, but for me it was a shift. Designers often have a complicated relationship with emotion. There's a lot beneath the surface—not just insecurity or self-doubt, but also pride, exhilaration, and sometimes even manic energy. These feelings are part of the process, yet we rarely speak about them. We're trained to be critical, precise, professional—but not necessarily personal. So for me, writing about my emotional state became a kind of counter-practice: a gesture of openness.

Seen together, these experiments marked a shift—from rethinking tools and habits to questioning the deeper structures that shape how we work, learn, and show up as designers. Whether through reshaping workflows, quietly bending institutional norms, or bringing emotion into the foreground, each one became a way to test what hacking might mean beyond the technical. A way of working with what's already there—protocols, expectations, materials—and nudging them into something more open, personal, or strange.

[1] Graff, V. (2004, December 14). *Meet the Yes Men who hoax the world*. The Guardian. Retrieved July 2, 2025, from https://www.theguardian.com/media/2004/dec/13/mondaymediasection5

[2] Von der Kunst, anderen auf den Sack zu gehen. (2011, September 20). Fluter, Nr. 40, Thema: Protest. Retrieved July 2, 2025, from https://www.fluter.de/sites/default/files/von_der_kunst_anderen_auf_den_sack_zu_gehen.pdf

[3] Lovink, G. (1997, May 16). *The ABC of tactical media*. Nettime. Retrieved July 2, 2025, from https://www.nettime.org/Lists-Archives/nettime-l-9705/msg00096.html

[4] Critical Art Ensemble. *Digital Resistance: Explorations in Tactical Media* (Brooklyn, NY: Autonomedia, 2001), p. 4. Retrieved from https://monoskop.org/images/3/3a/Critical_Art_Ensemble_Digital_Resistance_Explorations_in_Tactical_Media.pdf

[5] !Mediengruppe Bitnik. (n.d.). *Delivery for Mr. Assange*. Retrieved July 2, 2025, from https://wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww.bitnik.org/delivery

I am a system + Last Experiments

Seen together, these experiments marked a shift—from rethinking tools and habits to questioning the deeper structures that shape how we work, learn, and show up as designers. Whether through reshaping workflows, quietly bending institutional norms, or bringing emotion into the foreground, each one became a way to test what hacking might mean beyond the technical. A way of working with what's already there—protocols, expectations, materials—and nudging them into something more open, personal, or strange.

I am a system + Last Experiments

I started out by experimenting with outside of me – the fundamental building blocks of design. Then I moved on to systems that I am inside of, such as tools, workflows and design processes. There, I explored themes such as autonomy and technical literacy. Eventually, I arrived at institutions, questioning their protocols, templates and expectations. Throughout this process, I tried to expose or shift the norms and habits that we designers are trained into, often without noticing. In that logic, it makes sense to arrive at the self, at how we see ourselves as designers.
Still, this chapter wasn't planned. It emerged from the process. I didn't initially think of "the self" as something to hack. But the more I experimented, the more I noticed how much my own ego was shaping what I did — and how I felt about it. Expectations around originality, authorship, or recognition weren't just influencing the work — they were influencing me. Eventually, it became clear: this too is part of a system.

The shift began with a series of experiments in which I attempted to replicate hacker strategies such as copying, remixing, reusing and reverse engineering. These are typical learning methods in hacker culture — breaking something down to understand it and then putting it back together in a new way. I applied this concept to graphic design, for example by dissecting existing posters, studying their structures and recombining their elements. My goal was not simply to imitate, but to treat design as something modular — something that could be broken down and reconfigured — rather than a static artefact. I wanted to view design as a system that was free from a 'creative owner'.

In one of the first experiments, I took four posters designed by others and combined them into a new composition. There was no original content. Just recombined pieces — nothing 'mine'. Setting aside the aesthetic aspect, when I looked at the result, I felt an immediate discomfort. It seemed wrong — lazy, even shameful — and I didn't want to show it.

I had experienced that reaction before, but given my current state of mind, it surprised me. I began to ask myself why I felt this way.

Was it because I hadn't designed every part of it?
Because it didn't match my "style"?
Because it looked unrefined — and I hadn't "earned" it through effort?

This small act of remixing something instead of creating it from scratch disrupted more than just aesthetic expectations. It challenged my understanding of my own identity as a designer. I had internalised certain beliefs about what constitutes "good" design: originality, authorship, refinement and control. I even believed that suffering was part of the process.

That unease exposed an internalized system I hadn't fully acknowledged:
→ That design must be authored
→ That authorship must be visible
→ That value comes from originality and ownership

The hack was psychological: letting go of control, of authorship, of recognition. Letting go of the need to be proud of the result.

This surfaced again in a later experiment: I took a short interview text and laid it out using only default settings across several design programs. I made no custom choices: the typefaces, margins and colours were all left untouched. The idea was simple: What would happen if I removed myself from the process and let the software decide? I had assumed this would reveal something about software defaults and how they shape what we produce, and it did. But it also revealed something about me.

How deeply I tie my self-worth to the quality of the outcome.

How uncomfortable I am with producing something that feels 'empty'.

This wasn't just about defaults. It was about pride. Ego. The need to feel in control.

That kind of reflection can seem self-indulgent at first – but I've had many conversations with other designers to know this isn't just personal, it's a pattern, it's part of a system. Many of us feel it: the shame of making something "bad," the fear — but also the need — of being seen. The pressure to impress, to be original, to constantly produce.

Where does that come from?

It's not just about individual insecurity. It's part of a larger myth — the idea of the solitary genius, the auteur designer, the original creator who must suffer for their work and defend their vision. It's a legacy that runs through modernism, through design education, through studio culture. We may work in teams or under brands today, but the traces of that ideal still shape how we see ourselves and what we expect of our work.

ORIGINS

The Designer as Myth: Ego, Genius, and the Culture of Authorship

So, again: where does this myth come from — this idea of the solitary designer-genius, working in silence, producing perfect, original outcomes, untouched by criticism or collaboration? Why is it still so seductive?

Its roots go back to Romanticism and the cult of the individual. Emerging in the late 18th and early 19th centuries, Romanticism celebrated the figure of the inspired, tortured artist — someone whose creativity came not from training or context, but from innate genius. Genius, in this sense, wasn't just talent. It was almost divine. The Romantic genius was imagined as a kind of prophet, a channel for truth or beauty — *above* ordinary life, misunderstood by it. This figure became a model not only for artists and poets, but also for how we began to imagine "creative" professions more broadly.

This narrative didn't disappear. It carried over into modernism, where the designer — now framed as an author, rather than a craftsperson — was again seen as a visionary. Formal innovation, stylistic signature, and authorship were increasingly associated with *value*. The myth evolved, but it didn't dissolve. It remained tied to a powerful idea: that the designer should be original, autonomous, and, ideally, a little difficult to understand.

Monika Parrinder, in her essay *The Myth of Genius* (2000), explores how this logic plays out specifically in design. Referencing Foucault, she reminds us that the author does not "precede" the work — that ideas, references, and meanings are already present, circulating in culture. The designer's job is not to invent from nothing, but to select, filter, remix. That insight aligns closely with my own experiments, but it also undermines the very premise of the "genius": if nothing is ever fully original, then what is being authored?

Parrinder also points to how genius is constructed — and how absurd these constructions can be. She offers a "genius checklist" that includes:

1. The creator — who rises above ordinary life, once seen as a messenger for God
2. The individual — solitary, non-conformist, a rule-breaker
3. The madman — genius and madness are deeply linked in cultural imagination
4. The intuitive — whose work is natural, unlearned, and thus beyond critique
5. The pioneer — ahead of their time, misunderstood, suffering

It's almost comical when listed like this — and yet, these are exactly the roles we still subtly assign in design culture. We continue to admire the "mad genius," the self-taught outsider, the person with a "signature style," the misunderstood innovator. These figures become aspirational, even when they don't reflect how design actually happens.

And importantly: these roles are deeply gendered. The genius, historically, is almost always a *man*. As Parrinder notes, “pioneer” status is rarely assigned to a woman. Feminist critiques — from artists like the Guerrilla Girls to theorists like Fran Cottell — have long pointed out how this myth excludes and undervalues women’s labor. Cottell writes:

> “The idea perpetuated by the art market that individual geniuses arrive out of nowhere… is convenient but untrue. Artists invariably arrive at artistic solutions as a result of social influences as well as intellectual reasons.”
> The same applies to designers. But in graphic design, the genius myth is still convenient. It makes critique harder. It discourages collaboration. And it reinforces a model of authorship that isolates rather than connects.

The way graphic design history is narrated still centers around big names — often white, male “geniuses” presented as lone inventors of style or meaning. As Michael Rock points out, this myth of mysterious aesthetic inspiration is only possible if we strip design of its cultural context and flatten its complexity.[1] Jarrett Fuller similarly notes that design history is “littered with names” we’re told to memorize — shaping a culture where originality, authorship, and personal branding are rewarded, especially in education and the market.[2] Meanwhile, Rick Poynor critiques the idea that designers lack agency because of client involvement — as if collaboration were an obstacle rather than part of the work.[3] This contradiction — between the myth of the genius and the reality of anonymous studio labor — reveals how design still struggles to tell its own story in a more collective, honest way.

———

designers already try to break this up, theres a movement like hacking, one like quuer designn/typography with paul solellius that also bring up this question. i want to give an example out of the book interdisciplinary design edited by anja kaiser, ther is text by sara kamann “once upon a time” designers get more and more aware of this-example in undisciplinnary design: here she writes about design history in a very uncommon way, like she would tell a tale. the way she absurdizes names and happenings, like kalling josef-müller-brockmann “joe” and saying that he invented smth like grids. or talking abt the grid like “the grid was an arrangement of lines and measurements, drawn onto a sheet or digital sheet. the grid was a saviour and safety net. it lent security and efficiency to the things.” how she describes this in such a rational or what ist this, way, is nont only funny. it also showas how weirdle fixed history in general and design history is. plötzlich you can imagine alternativen etc. also a great example what words can do.

and thars were i also see that applying hacking strategies again could help (how hackers work, contributing to a bigger thing, still meritocratic etc) also anja groten says that in the dialogue mentioned already in a former chapter: “The paradoxes you described bring about important frictions within hacker communities, and are crucial to an understanding of the hacker way of working. These frictions don’t seem incidental. No, they’re actively made, widely publicized, and openly negotiated. Instead of idealizing a hacker archetype, designers could learn more from the dilemmas of this maker culture. This, in turn, might help them reflect on the missed opportunities and weak points of their own practices. Designers should disseminate their work in ways that force still-vulnerable processes to be exposed and possibly contested. If we stop clutching so tightly to the paradigm of making ‘convincing work’, and instead embrace the limits of our practices, designers could create our own ecology of frictions. “ (quote from:
https://hackersanddesigners.nl/hacking-designing-paradoxes-of-collaborative-practice-by-anja-groten.html) .

designers that are good at this intersection of practice given the aforementioned aspects: experimental jetset. treten immer als ganzes auf zusammen, no single “star” figure. open with their influences. they say that their design is always narrated or situated in a context of culture, its embedded.

also on another note (rubben prater): working solely makes us more weaker.. capitalism.

so what to do now. stripping of authorship or desire to be original and new was a relief, then i actually started designing.

[1] Rock, M. (2006). *The problem of provenance*. Design Observer. Retrieved July 10, 2025, from https://designobserver.com/feature/the-problem-of-provenance/5657/

[2] Fuller, J. (2020, August 18). *The end of the designer-genius*. Jarrett Fuller Blog. Retrieved July 10, 2025 from https://www.jarrettfuller.blog/2020/08/designer-hero/

[3] Poynor, R. (1998). *Design without boundaries: Visual communication in transition* (p. 115). Booth-Clibborn Editions.

so many aspects to deconstruct! but i have a maximum of 120.000 characters and its already beginning of july so i will focus on this isolation vs collaboration part, individuality.

connection collectivity

actually design is inherently a collective collaboration task. already starts with client. i hate when massimo vignelli once said”clients dont have opinions they have problems”. i like more milton glasers quote: when asked by design students if he also would do work for himself, he said every work is for himself. i like that approach, detach yourself from the output. always the idea that the client is the counterpart, limiting the designer. then other collaborators in the design taske (printers for example). and influences! every designer is influenced omg, conscious or ot. non big names, überletung experimenntal jetset.

then draw back to hacking anja groten. messy practices, undisciplinary design. and individualism makes u weak. reference in caps lock

de

___

question how design history is narrated (undisciplinary design article, welcher war das?)

graphic design history is still history of big names.

michael rock, problem of provenance: The story of mysterious aesthetic inspiration gushing forth from fertile imaginations, while romantic, is almost always miscast. Graphic design is mediated: it works because it is attached to the surrounding culture. The creative-genius story is only viable if the definition of a design object is whittled down to the point that it is no longer rich or interesting or if the designer is inflated to the cult status he or she covets.”

Jarrett Fuller: “Design history is littered with names — almost all white and male — that we’re told we should know.”

The result: a pressure to stand out, to produce original work, to be “authored.”

The market (and education) favors personal brands and clean narratives of originality.

at the same time: weird dichotomy, in agencies we are all as designers kind of anonymous. maybe thats a different topic, i will not address that right now, this is all very brief, i know.

rick poynor–design thinking or critical design: ”..enduring assumption that designers lack agency–that the designer, unlike the free artist, is fatally restrained by the presence of the client.”p.115 thats ridiculous, the designer as “enemy” from which we have to defend our great genius ideas. forgetting, ignoring even that collaboration already starts there.

—

fran cotell:
https://books.google.ch/books?hl=de&lr=&id=bM4fvz0mGtsC&oi=fnd&pg=PR8&dq=Katy+Deepwell+(ed)+New+Feminist+Art+Criticism:&ots=MrnDi316gV&sig=5-tUSvwesNWlg3auoDnLtOeaZx4#v=onepage&q=fran%20cotell&f=false

Winter 2000, The myth of genius, Monika Parrinder

funny: “ **Genius checklist**

Characteristics routinely associated with genius include the following:

1. the creator – usually artist, writer or scientist – who rises above the ordinary mortal, acquiring a semi-divine status, in past times as a messenger for “the original creator,” God

2. the individual – a pioneering, solitary non-conformist

3. the madman – links between genius and madness are legion

4. the intuitive person – whose work is “natural” and unlearnt and hence cannot be analysed

5. the pioneer – who is ahead of his or her (but rarely “her”) time and possibly a misunderstood or tortured soul (see 3 above)”

Foucault says that the author does not “precede” the work: ideas and meanings are already there and the author’s role is to “choose,” to filter and synthesise to create output. (Foucault also emphasises “limiting” and “excluding”).

Fran Cottell, in her essay “The cult of the individual” [6], says “the idea perpetuated by the art market that individual geniuses arrive out of nowhere . . . is convenient but untrue. Artists invariably arrive at artistic solutions as a result of . . . social influences as well as for intellectual reasons.”

In graphic design practice, as opposed to theory, the genius idea is still accepted, and convenient, with little sense of being a myth.

personality before content

connection to feminism/women are underrated. bsp guerilla girls (genf?)

April Greiman, who has recently become a Pentagram partner, has been a force in graphic design for three decades, but the one image of her work to which everyone returns is her poster for *Design Quarterly* [12] where the dominant image is her naked body.

rick poynor–design thinking or critical design

.."..enduring assumption that designers lack agency–that the designer, unlike the free artist, is fatally restrained by the presence of the client."p.115

work for the client is the opposite of work for himself, this is actually stupid (mitlon glaser) every work is for himself.

https://www.jarrettfuller.blog/2020/08/designer-hero/

The End of the Designer-Genius

"The design industry, too, frequently uses the narrative as the singular genius to describe our profession and the work that comes out of it. Design history is littered with names — almost all of them white and male — that we're told we should know. These people came in and, out of nowhere, invented a new style or changed the way design was made. The subliminal message here is that this is what being a designer means, this is how we are supposed to work. "

"This is not — and never was — how design operates. Design, by its very nature, is a collaborative activity. At its core, it's a collaboration between designer and client. The design process is a series of negotiations and discussions between designer and printer, designer and developer, designer and manufacturer."

for michael rock this is also a problem of how design history ist narrated:#

The Problem of Provenance:

> History belongs to the writer; so too the history of graphic design. The challenge for the design historian is to assemble bits and pieces of data — traditional primary and secondary sources as well as visual material — into coherent narrative. The state of graphic-design history is dire because the subjects themselves have a vested interest in perpetuating a closed narrative about their own ideas — designers want to build and maintain their own self-authored, and self-serving, myths — and design journalists are often unwilling to question the simple stories their subjects feed them. So both the primary and secondary are compromised. But the real problem is that the origins of a visual idea are blurry. In most cases, the attribution of a discrete idea to a single designer is simply a shorthand way to meld all the various threads of influence into a single unified point of inspiration.

> The story of mysterious aesthetic inspiration gushing forth from fertile imaginations, while romantic, is almost always miscast. Graphic design is mediated: it works because it is attached to the surrounding culture. The creative-genius story is only viable if the definition of a design object is whittled down to the point that it is no longer rich or interesting or if the designer is inflated to the cult status he or she covets.

As Thomas Carlyle (a XIXth century Scottish cultural critic, historian, philosopher) had said:

‘The history of the world is but the biography of great men.’

Paper: Social Creativity: The Challenge of Complexity, Alfonso Montuori Ronald E. Purser

California Institute of Integral Studies Loyola University Chicago

A. Montuori & R. Purser, (1997). Le dimensioni sociali della creatività. Pluriverso, 1, 2, 78-88

Slater (1991) argued that the “Individual-versus-society myth” is deeply embedded in North-American culture. This myth is closely related to the “lone genius myth” (Montuori & Purser, 1995), which also sees culture and society--other people, in other words--as an obstacle to the self-realization and self-expression of individuals. But Slater (1991, p.154), like Stewart and Bennett, pointed out that “the very wish to escape our culture is itself a product of cultural conditioning,” and therefore manifests itself in clearly preestablished roles which have taken on mythical status in North-American culture, from the Lone Ranger to James Dean to Einstein (cf. Bellah et al., 19

https://www.youtube.com/watch?v=E4ygiodaY40

---

i started experimenting with systems outside of me, addressing topics around aesthetics, then coming to systems i am inside of –tools, workflows– there i had as a topic autonomy, technical literacy. i made my way to institutions, questioning protocols etc. in general i was always questioning or making visible through hacking kind of norms & habits, what we are used to as designers, how and with what we are educated. in that way, it might seem normal to end this thesis with what we might call the “self, or the selfunderstanding of the designer. for me, This chapter wasn’t actually planned. It emerged in the process. Initially, I didn’t think of “the self” as something to question in the context of hacking. But the more I experimented, the more I noticed how much my own ego, my expectations of originality, authorship, or recognition were shaping how I worked — and how I felt about the work. Eventually, it became clear: this, too, is part of the system.

this actually started during experiments What I wanted was to borrow techniques from hacker culture — copying, remixing, reusing, reverse engineering (Hackers often learn by reverse engineering: analyzing, dissecting, breaking things down, then building them back up. I transferred that process to graphic design: I took apart an existing poster and

reassembled its elements into new images. This wasn’t just about form, but about understanding structure — seeing design not as a finished object, but as something that can be broken, studied, and remade.
) for example one time i took 4 existing, finished posters, and made a new out of them. i basically combined them, rearranged. I took four existing posters and merged them into one new image.No original content. Just recombined elements — nothing "mine."
At first, I didn’t want to show it.It felt… wrong. Lazy. Ugly. Embarrassing.
But why?
Because I didn’t design every part of it?Because it didn’t reflect my “style”?Because I didn’t suffer for it?
Suddenly I realized:This wasn’t just about design process —It was about ego.
There was arrogance in my reaction.A kind of panic: If this isn’t mine, then who am I as a designer?
I had always wanted my work to be “original,” “personal,” “refined.”But this was messy. Assembled. Uncomfortable.It challenged my self-image.
It surfaced a deeper system I’d been obeying without noticing:→ That good design must be authored→ That authorship must be visible→ That value comes from originality and control
This wasn’t imitation.It was ego exposure.
And that cracked something open.
The hack wasn’t visual — it was psychological.Letting go of control.Letting go of being seen as the creator.Letting go of needing to be proud of the result.
That’s when the experiments turned inward.
What if the most resistant system isn’t the software —But the self I’m trying to protect?
From here, the work became personal.Not just hacking systems I use —But hacking the system I am.

—

That same discomfort — the ego tension — came back in another experiment.
✹ Protocol Hack 4: Default by Design
I took a short interview excerpt and laid it out using only default settings across different design programs.Default typefaces. Default margins. Default colors. Default everything.
The idea was simple:What happens when I don’t design — when the software does?
The results were awful.Not just visually — emotionally.
I hated the process. I hated the output.I felt like a bad designer.I wanted to quit halfway through.That almost never happens.
But that was the point.
I expected the experiment to reveal something about software.Instead, it revealed something about myself:
What does it mean when I don’t design?Who am I when the work is weak?Why do I need to make something “good” in order to feel okay?
Even without control, I still made decisions —but they felt shallow, meaningless.And that alienation from the work hit deep.
This wasn’t just a process issue.It was a confrontation with authorship, pride, and control.It forced me to see how tied my self-worth is to aesthetic output.

If institutions shape what is considered legitimate, valuable, or “serious” design work, they also shape how designers see themselves: as authors, as professionals, as brands. maybe loop back to the introduction. the role of the designer, that i am looking for one. one specific role i want to look at in this is the author. this reflection somehow started with the experiments. this chapter wanst actually planned because admitettely, i didnt appear to me straight away that also i am a sysstem, that eventually design is about a person it self how important those emotions are, that we as designers are humans and humans shape the profession and so on.. and ist this even hacking still? can u hack yourself?
so how this started. it started actually with experiments where i wanted to replicate a way hackers work, transfer things like copying, remixing into graphic design. i took that literal, what i did: And I realized that after this experiment, where I combined four existing posters into a single image.No original content. Nothing “mine.”
Initially I wanted to replicate the hacking method of remixing, but it led me to something completely different

or, another remix: **In this experiment, I sought out three different sources and combined them in new ways. At first, this felt strange, but if we’re honest, it’s not that far from how things really work. We consume an overwhelming amount of content, and many designers rely on mood boards or references. Yet, there’s a certain shame attached to admitting it, as if it undermines the pride we take in claiming something as “our idea.” This experiment questions that pride — not as a critique of authorship, but as an invitation to consider the shared nature of creative work.**
or copying: **In this experiment, I straight-up copied a piece of design I liked. What interested me was how different it felt from my usual way of working — I realized how fixed my own habits have become. Copying revealed how narrow my design reflexes can be. It was also telling that the source was relatively old. Once again, it raised questions about reuse, authorship, and what we really mean when we talk about originality.**

The ting is, It felt so wrong , I didn’t even wanted to show it.
But why?
Because I didn’t design every part of it?Because it didn’t reflect my “style”?Because I didn’t suffer for it?
And that reaction exposed something in me:→ I was clinging to authorship.→ It wasn’t about design—it was about ego.
That’s when I realized:Maybe the hardest system to hack… is myself.

so those experiments started a series of reflection how tied i am to my ego! its about ego, feeling shame when i show something. and at forest, maybe especially for non designers, this sound self centred, even self indulgent. but i know its not, like i said i am also a patter, a

lot of designers have feelings like that i know it !! thats where it also connects to last experiment with the emotions, maybe thats what triggered it

Michael Rock – “The Designer as Author” (1996)
https://designopendata.wordpress.com/portfolio/the-designer-as-author-1996-michael-rock/
What does it really mean to call for a graphic designer to be an author?

“But theories of authorship also serve as legitimising strategies, and authorial aspirations may end up reinforcing certain conservative notions of design production and subjectivity – ideas that run counter to recent critical attempts to overthrow the perception of design as based on individual brilliance.”

ellen lupton, https://ellenlupton.com/Designer-as-Producer

definition: “Behind this phrase is the will to help designers to initiate content, to work in an entrepreneurial way rather than simply reacting to problems and tasks placed before them by clients. The word author suggests agency, intention, and creation, as opposed to the more passive functions of consulting, styling, and formatting. Authorship is a provocative model for rethinking the role of the graphic designer at the start of the mil-lennium; it hinges, however, on a nostalgic ideal of the writer or artist as a singular point of origin.
“

“As an alternative to “designer as author,” I propose “designer as producer.“

reference to walter benjamin text. Behind this phrase is the will to help designers to initiate content, to work in an entrepreneurial way rather than simply reacting to problems and tasks placed before them by clients. The word author suggests agency, intention, and creation, as opposed to the more passive functions of consulting, styling, and formatting. Authorship is a provocative model for rethinking the role of the graphic designer at the start of the mil-lennium; it hinges, however, on a nostalgic ideal of the writer or artist as a singular point of origin.
The challenge for design-ers today is to help become the masters, not the slaves, of technology. There exist opportunities to seize control—intellectu-ally and economically—of the means of production, and to share that control with the reading public, empowering them to become producers as well as consumers of meaning. As Benjamin phrased it in 1934, the goal is to turn “readers or specta-tors into collaborators.” His words resonate in current models of practice that view the reader as a participant in the construction of meaning.

interesting introduction but whats the point?

---

contribution over outcome.
i’ve been thinking about this very long. this feeling, that most graphic designers are so deeply entangled with their egos, that we are so attached to our outcome

The topic of authorship, originality, and ego is vast and deeply entangled with the history of design — too complex to fully unpack within a chapter of this thesis. That's why I'll focus more specifically on the collective and situated dimension of this discussion, not only because it offers a concrete counterpoint to the solitary genius myth, but also because it aligns closely with hacking as a method and mindset.

In this context, hacking isn't just about technical subversion — it's about modifying systems, working iteratively, and rethinking ownership. It invites us to see design not as the output of a singular author, but as something built through shared structures, inherited tools, and collaborative friction. This perspective helps loosen the grip of ego — not by erasing the designer, but by repositioning them within a network of influences, dialogues, and mutual dependencies.

Hackers often work collectively, iteratively, and in tension — not to erase merit, but to make contribution visible and structural. In a previously cited fictional dialogue, Anja Groten reflects on the frictions at the heart of hacker communities: "Instead of idealizing a hacker archetype, designers could learn more from the dilemmas of this maker culture… If we stop clutching so tightly to the paradigm of making 'convincing work', and instead embrace the limits of our practices, designers could create our own ecology of frictions."

Groten's call for "an ecology of frictions" points toward a design culture that embraces vulnerability, messiness, and shared authorship. While this ethos is often explored in newer or more explicitly critical scenes like hacker or queer design communities, it also resonates in more established, applied practices — such as that of Experimental Jetset. The Amsterdam-based studio, operating since 1997, is striking not only for its graphic work but for its self-presentation: always as a collective. There's no single spokesperson or star figure. Their interviews and essays often reference the band-like dynamic of their studio, suggesting a model of design that is less about individual expression and more about a shared worldview.

Crucially, they are transparent about their influences. Rather than posturing as originators of entirely new styles, they openly situate their work in relation to music, counterculture, modernist design history, and theory. For them, design is always cultural — always narrated, always embedded. In that sense, their practice quietly undermines the genius myth not through theoretical argument but through long-term, consistent collaboration grounded in dialogue, mutual authorship, and influence.

A good example of this ethos is their book *Statement and Counter-Statement* (2015). Rather than presenting a polished portfolio or a narrative of individual brilliance, the book reads more like a curated dialogue — between their own work, their many influences, and the cultural and political contexts that surround both. Even the title signals this approach: every design "statement" implicitly invites a counter — a disagreement, a recontextualization, or an extension. What's striking is how deliberately the studio avoids the language of originality or invention. Instead, they frame their work as part of a lineage — drawing from punk,

modernism, underground publishing, pop music, and theory. Their authorship is not erased but consciously situated.

This way of working — acknowledging influence, embedding oneself in a system, allowing for contradiction — shares something important with hacking: the idea that knowledge grows through modification, reuse, and dialogue rather than isolated invention.

However, the turn toward “authorship” in graphic design has itself been contested. Michael Rock, in his essay *The Designer as Author* (1996), warns that invoking authorship may not always be as radical as it seems. He writes: “But theories of authorship also serve as legitimising strategies, and authorial aspirations may end up reinforcing certain conservative notions of design production and subjectivity — ideas that run counter to recent critical attempts to overthrow the perception of design as based on individual brilliance.” In other words, simply calling the designer an “author” can risk reaffirming the very hierarchies and mythologies we might want to dismantle.

Ellen Lupton offers an alternative frame: the “designer as producer.” Drawing on Walter Benjamin’s writing, she argues for a model in which designers seize control of the means of production, not to elevate themselves, but to redistribute agency — inviting readers and audiences to become collaborators. “Authorship,” she writes, “hinges… on a nostalgic ideal of the writer or artist as a singular point of origin.” By contrast, the designer as producer is entrepreneurial, embedded, and relational. This vision doesn’t eliminate ego or authorship, but it reframes them as tools for shared meaning-making — distributed, situated, and flexible.

Together, these perspectives help refine the central claim of this chapter: that authorship in design should not be abandoned, but **reimagined**. Rather than perpetuating the figure of the designer as an isolated genius, we can understand authorship as a relational practice — rooted in context, shaped by others, and always part of a broader system. This is not a call to strip designers of agency or ambition, but to reposition those qualities within networks of mutual influence, shared production, and collaborative tension. In doing so, we begin to dissolve the ego not into absence, but into relation.

What is hacking

Ergänzung:

Finally, hacking is fundamentally a meritocracy. It rewards individuals not based on formal titles, academic credentials, or institutional affiliations, but on demonstrated skill, intellectual rigor, and creative execution. In hacker culture, respect is earned through contribution—through elegant code, problem-solving ingenuity, and the ability to make systems do things they were never originally intended to do. Hacking is undeniably hard work, but it is also work pursued as play: an engagement driven by curiosity, fascination, and joy. This merging of labor and play is central to the hacker ethos. It transforms effort into discovery and turns technical challenges into opportunities for creative intervention.

Looping back to the prologue, these values and motivations reflect what Eric S. Raymond articulates as *The Hacker Attitude*—a constellation of mindsets and ethical commitments that go beyond technical prowess. At its core, the hacker attitude is grounded in a sense of the world as something open to inquiry, deconstruction, and improvement. It resists passivity. As Raymond puts it, boredom and drudgery are not simply unpleasant—they are seen as moral failings of a system that should, and could, work better. Hackers believe in seeking elegant solutions, repurposing tools, and transforming constraints into creative possibilities. Hacking, in this deeper sense, is not just technical execution—it is a form of speculative design thinking, a way of engaging with complex systems critically and imaginatively.

But this attitude is not purely about skill or aesthetics—it also has a deeply political and ethical dimension. Hackers are naturally anti-authoritarian. They are skeptical of institutions that rely on control, censorship, secrecy, or coercion. This doesn't mean hackers reject all forms of authority—but they resist any structure that seeks unquestioning obedience, especially when it interferes with curiosity or the free flow of information. As Raymond writes, "Anyone who can give you orders can stop you from solving whatever problem you're being fascinated by," and authoritarians often do so for appallingly shortsighted reasons. This resistance to authoritarianism is not just personal—it's systemic. It involves a principled hostility to censorship, to the use of force or deception, and to closed systems that prioritize control over collaboration.

Transparency, openness, and voluntary cooperation are fundamental to hacker culture. Hackers thrive in environments where knowledge is shared freely, where code is open and remixable, and where systems are built to be understood and improved—not obscured behind proprietary walls or bureaucratic gatekeeping. These values translate directly into design practice: hacking as design becomes a method of intervention that favors open systems, user agency, and the empowerment of individuals over top-down control.

In the context of this project, *The Hacker Attitude* offers a valuable conceptual framework for thinking about hacking not just as a technical activity, but as a broader design methodology—a form of systemic engagement with the world. It encourages us to see design as an act of intervention: of reimagining existing structures, subverting dominant norms, and creating new possibilities from the remnants of the old. Hacking in this expanded sense becomes a speculative and often subversive method of working with complex

systems, whether digital, social, or material. It is a form of critical play, improvisational, curious, and often joyful.

Importantly, Raymond’s formulation of the hacker attitude also emphasizes character traits that are essential to this practice: curiosity, resilience, openness, a tolerance for ambiguity, and an intrinsic love for challenge. These are not just helpful attributes, they are foundational. The hacker's mindset thrives on the thrill of the unknown and the satisfaction of making things work in unexpected ways. It’s about being persistent in the face of difficulty, generous in the sharing of knowledge, and bold enough to question why things are the way they are, and how they might be better.

How hackers work

To further unpack hacking as a mode of working, it helps to look more closely at how hackers actually operate — the everyday practices that shape their relationship to tools, problems, and collaboration. These aren't rigid methods but rather a set of tendencies and habits that reflect shared values: openness, curiosity, playfulness, and a willingness to break and rebuild. One of the most common and defining practices is copying and remixing. In hacker cultures, this isn't seen as plagiarism or laziness, but as a productive, even generous, way of working. You take what's already there — a piece of code, a script, a tool — and adapt it to your own needs. In this sense, work is always building on someone else's work. It's a form of learning through doing, and of acknowledging that no one creates in a vacuum.

Closely related is reverse engineering — the practice of taking systems apart to figure out how they work, often without any official documentation. It's a kind of technical curiosity, but also a political gesture: a way of reclaiming access to closed or proprietary systems. It's about seeing behind the interface, opening up the black box. This ties into a broader debugging mindset, where trial and error are not just tolerated but embraced. Bugs, glitches, and errors are expected. Hackers spend hours tracing obscure problems, experimenting with fixes, trying things out just to see what happens. As Groten notes, this demands a high tolerance for frustration — but also creates space for unexpected discoveries.

None of this would work without a strong culture of transparency and sharing. Hacking is often described as a social practice, and part of that is making things visible: code, processes, mistakes, progress. Platforms like GitHub or community forums rely on this openness — others need to see your work in order to learn from it, contribute, or critique. This kind of sharing also means letting go of some control. Once your code is out there, someone else might change it, fork it, remix it — and that's the point.

Finally, many hacker projects are built through peer-to-peer, decentralized collaboration. Rather than top-down management or fixed hierarchies, work often happens in distributed teams or loosely connected communities. People contribute because they care, not because they're being paid. Recognition comes through participation and visible impact, not formal titles. Himanen calls this the "meritocratic" side of hacking — a social logic where skill and dedication are more important than credentials. But it also shows that hacking is rarely a solo pursuit. Even when working alone, hackers are in dialogue with a wider ecosystem: past code, future users, other contributors.

Taken together, these practices sketch a working culture that's messy, adaptive, and highly social. They rely on trust, experimentation, and the idea that knowledge grows through sharing, not hoarding. It's not about perfect outcomes, but about staying engaged with systems, solving problems together, and being open to where things might lead — even when they break.

Kelty, C. M. (2008). *Two bits: The cultural significance of free software* (p. 102). Duke University Press.

copying 102
bugfixing 129
reverse engineering 234

Conclusion

To conclude, maybe remind ourselves of the initial goal/idea of this thesis. feeling of personal frustration, not satisfied with the roles we as designers are offered, overwhelmed with ever evolving tools, weird prodducing of constant output... thesis was looking for a new way to engage with the practice.
so do hacking methods and mindset applied to graphic design offer solutions/or at least new ways to think abt graphic design. to rewire?
chapter 2 ethics of hacking drew an image abt what hacking is. i use a broad definition of hacking, away from hacking as technical skill. it is abt systems, understand something by deconstructingn it, everything is a system. hacking asks how does this work? what else could this be?
hackingn dont use tools passively, seeking to understand them and repurpose the creatively. hackers question authorities and gatekeeping. transparency, playfuklness, desire to learn, joy, are crucial hacker values. this chapter and the prologue and the introduction proved, that hacking indeed is transferable to graphic ddesign.
also how hackers work in chapter 3. its a different working model but it is aligned and suits creatives.hackers have working model that is very adaptive, experimentation, belief that sharing knowledge is crucial. openness also means giving up control. hacking is also highly social

Conclusion

This thesis began with a feeling — a personal frustration with the current state of graphic design practice. A sense of being boxed into predefined roles, expected to constantly adapt to new tools and workflows, always producing, never pausing. It felt rigid and unsustainable. I didn't start this project looking for a new method or grand theory, but rather with a desire to rewire my own relationship to design. I was looking for a more open, critical, and curious way to engage with the field — one that could make space for reflection, experimentation, and resistance.

Hacking became the lens through which I explored this possibility. Not hacking in the narrow, technical sense, but in a broader sense — as a mindset. As we saw in Chapter 2, the ethics of hacking are less about breaking into systems and more about understanding and rethinking them. Hacking asks: How does this work? What else could it be? It values curiosity, transparency, playfulness, and a deep desire to learn. Hackers refuse to be passive users of tools or systems — they want to take things apart, reconfigure them, and often share the results. There's an anti-authoritarian spirit here, a refusal of gatekeeping, and a strong belief in knowledge sharing and social collaboration.

Chapter 3 built on this by exploring how hackers actually work. Their modes of working differ from traditional models of labor. Rather than being driven by external obligation or economic reward, hacker work is shaped by intrinsic motivation — by joy, challenge, and experimentation. Failure and frustration aren't signs of weakness but central parts of the process. Hackers learn by doing, by breaking things, by testing limits — and by sharing their results, bugs and all. This open and iterative approach mirrors many aspects of creative work, and I found it deeply resonant with the way I (and many other designers) wish to work.

As I observed in Chapter 5, tools are not neutral. They shape our ways of working, our aesthetics, and even our sense of agency. While the tools matter — and are especially important in today's shifting technological context — what matters more is how we relate to them. Whether it's open-source software, generative AI, or design systems, the key question becomes: are we using tools passively, or are we actively interrogating and reshaping them?

Chapter 7 picked up this thread by looking at how hacking plays out in activist and artistic contexts. Here, hacking is no longer just about using tools creatively, but about questioning the very rules behind them — exposing, subverting, and reimagining systems from the inside. As one source puts it, "Hier geht es nicht mehr darum, ein Programm zu nutzen, sondern darum, die Spielregeln zu untersuchen." Given examples show that also graphic design can be critical, engaged, and systemic — without reducing itself to pure aesthetics or pure technical skill. Their work shows a broader bandwidth for what design can be: not just output, but intervention.

Finally, Chapter 9 turned to the question of authorship — and the designer's ego. This wasn't a planned chapter, but it became increasingly clear that any attempt to rethink design practice needs to engage with how we see ourselves as designers. Our field still clings to the myth of the genius designer — the singular author behind an iconic outcome. This image is not only historically constructed, but deeply limiting. Instead, I argue that authorship in design should be reimagined as relational and situated — not erased, but embedded in systems, tools, and communities. Here again, hacking offers a helpful model: hackers rarely claim individual ownership, but operate within networks of contribution, influence, and mutual dependence. As discussed through the work of Experimental Jetset or Ellen Lupton's idea of the "designer as producer," this chapter proposed a shift from solitary authorship to shared authorship — not to diminish the designer's role, but to reposition it within a broader ecology.

In all of these chapters, hacking served as a way to think differently about design — to loosen fixed roles, to question inherited tools, and to reimagine authorship. It helped reveal that design, like hacking, is never just about making things — it's about engaging with systems, asking critical questions, and shaping the conditions of production. So while this thesis doesn't offer a final answer or a universal method, it suggests that hacking — as a mindset and as a mode of working — can open new ways of thinking about design. Not as a linear process or a solo performance, but as an ongoing negotiation with tools, systems, and others.

———

EXPERIMENTS

The method of the experiment was central to this thesis. Rather than setting out with predefined research questions or objectives, I worked through action and iteration — letting the process of messing with things teach me something I hadn't known to ask. This mode of inquiry was slow, often ambiguous, and sometimes frustrating — but it allowed space for emergence. It helped me identify patterns, constraints, and tensions that might have been flattened in a purely analytical approach. The experiment became a way of thinking — not just a way of producing examples. In this way, it echoed the spirit of hacking: making sense of systems by engaging them directly, from within.

The first phase focused on defaults — typefaces, templates, shortcuts so embedded in my habits I barely noticed them. By disrupting them, they revealed their hidden assumptions: about clarity, hierarchy, authorship, and function. What appeared neutral was anything but. Still, none of these systems were entirely closed — they could be misused, bent, or reimagined.

The second phase turned to tools and environments. Switching between platforms like GIMP, Photoshop, and code editors made visible how tools shape thought. Interfaces aren't passive — they streamline or constrain, subtly guiding our decisions. Here, coding wasn't about functionality, but about curiosity and soft resistance: testing how misbehavior or awkwardness can open space for reflection.

The third phase turned outward again — toward protocols, contexts, and systems of meaning-making. These experiments weren't about breaking rules for the sake of it, but about soft subversions — finding the small points of leverage within bigger systems. The idea of the "protocol" became especially useful here: it allowed me to think about the frameworks that guide not only what designers make, but how we present, collaborate, and position our work.

Across all phases, the experiments functioned less as solutions and more as probes — tools to surface invisible conditions and internalized beliefs. They helped me confront how my own behavior as a designer had been shaped: not just by education and software, but by deeper scripts around productivity, authorship, and value. Why does default feel like failure? Why is polish equated with professionalism? These weren't rhetorical questions — they were emotional ones.

In this way, the experiments ultimately led back to the core questions of the thesis — not only how we design, but how we feel about designing. They reframed my understanding of the designer not as a genius, nor merely a technician, but as someone embedded in systems — someone who can engage those systems critically, playfully, and reflectively. The designer as hacker, in this view, is not necessarily loud or disruptive, but attentive — curious enough to trace where assumptions come from, brave enough to bend them, and open enough to let the process lead somewhere unexpected.

What the experiments revealed, more than anything, is that meaningful shifts in design practice don't always require radical tools or entirely new methodologies. Sometimes, they begin with the smallest possible tweak — a different font, an awkward script, an intentional error. In those moments, the system is made visible. And in that visibility, something new becomes thinkable.

—

Throughout the thesis it has become increasingly clear that hacking offers not a fixed solution, but a flexible, inspiring framework. A way of thinking differently about design: not just as a service, or a product, or a career — but as a form of systemic inquiry, critical engagement, and social play. Hacking invites designers to move beyond passive consumption and polished outcomes, toward more open, adaptive, and self-aware modes of working.

In that sense, hacking doesn't just offer new tools. It offers permission — to experiment, to question, to care about how things work and who they're for. It offers a way to step outside the usual loops of performance and output, and reconnect with what drew many of us to design in the first place: curiosity, joy, and a desire to make things differently.

Graphic design, as a discipline and profession, is at a crossroads. While it continues to evolve technologically and aesthetically, it is also entangled in a number of deeper structural and cultural crises—crises that urgently call for critical reflection and new modes of practice.

One of the most persistent and troubling issues is the field's widespread apathy toward its own entanglement with capitalism and its consequences. As Rupen Pater writes in *Caps Lock*, graphic design does not operate outside power structures; it actively shapes and supports them. And yet, many designers remain unwilling—or unsure how—to engage critically with this role. Instead, they continue producing visual output in service of systems they may personally disagree with. This contradiction breeds cynicism, resignation, and ultimately, a lack of direction.

At the same time, many designers feel overwhelmed. The tools we use are in constant flux, with new software, workflows, and now AI-generated content reshaping what it means to "design." The profession is declared "dead" every few years, only to be rebranded and reanimated again. For many, this instability creates insecurity and exhaustion. The pressure to be fast, to be constantly visible, to create beautiful and polished work for ever-scrolling platforms like Instagram—often while underpaid and overworked—results in a culture of burnout and self-exploitation. Behind the sleek presentation slides and carefully curated portfolios is a designer who hasn't slept in 48 hours. Why?

This condition is also fueled by ego. Design is, for many of us, personal. We pour ourselves into our work, which makes critique feel existential and success dangerously addictive. Visibility becomes a currency, often outweighing substance or process. The design scene can feel like a circus—loud, performative, and exhausting. It operates more through networks and performative presence than genuine meritocracy.

Furthermore, being a graphic designer today means navigating an increasingly complex spectrum: from service provider to author, from invisible technician to vocal commentator.

This ambiguity can be liberating—but also confusing. What do we want from design? And what do we want to teach future designers?

In my view, the skills graphic designers will need in the future go far beyond mastery of tools or aesthetics. What we must cultivate is **adaptability**, **system thinking**, **imagination**, and the ability to ask better questions. We need designers who can think critically about context, about the societal and political frameworks within which their work operates—not just people who produce clean deliverables on demand.

can applied hacking

Design must re-engage with its role in shaping culture and systems. That means rethinking not only what we do, but how and why we do it.

This is where **hacking** comes in—not as a technical skill, but as a mindset and approach. Hacking offers a way to challenge systems from within, to explore alternative functions, and to question dominant narratives. It embraces curiosity, subversion, and play—not in the name of aesthetics alone, but in pursuit of deeper understanding and transformation. In this sense, hacking becomes a **method for design**, especially when traditional frameworks no longer feel adequate. . By framing graphic design as something that operates with, and within systems, hacking becomes a mindset that invites us to question defaults, repurpose structures, and open up alternative paths.