CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

Mindful prototyping

Guidelines for applying prototyping methods focused on multi-platform user experiences

Master's thesis in Interaction Design and Technologies

OSCAR NILSSON

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg,Sweden 2018

Report Number 2018:XX

Mindful prototyping

Guidelines for applying prototyping methods focusing on multi-platform user experiences

OSCAR NILSSON

Department of Computer Science and Engineering
Division of Interaction Design
Chalmers University of Technology
Gothenburg, Sweden 2018

Mindful prototyping
Guidelines for applying prototyping methods focusing on multi-platform user experiences

OSCAR NILSSON

Master thesis at Department of Computer Science and Engineering, Chalmers University of Technology
Report No. 2018:XX

Division of Interaction Design
Department of Computer Science and Engineering
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: +46 (0)31-772 10 00

Cover: Two of the “Mindful prototyping” guideline cards

Abstract

Prototyping is an increasingly relevant activity in interaction design, for producing design work that better capture design ideas and to enable accurate evaluations. Prototypes are often used to manage complexities of digital product design for global, widely distributed apps and services. Additionally, prototypes acts as communication tools within large organizations where designers operate.

This master's thesis project aims to study how designers can improve their practises related to working with prototypes. While an increasing set of available prototyping software aim to capture processes and help designers improve their practises in these more complex design situations, little guidance exist for better understanding how to apply prototyping, and prototypes, in general. To investigate these needs of designers, this study involved *research through design*, where prototypes for the game Minecraft were built together with the design team, and analyzed.

Findings from these prototyping processes, together with researching other works on prototyping practise, resulted in identifying four core needs of designers who prototype: *Considered prototype purpose, Speed of construction and use, Considered testing or sharing* and *Considered style of representation.*

To help designers improve their practises through serving these needs, a iterative design process was used to produce a set of 25 guidelines, that provide practical suggestions for how to improve prototyping practise in design teams. These guidelines were presented as a deck of cards, called *Mindful Prototyping,* with four categories of guidelines helping designers in different phases of the design process.

Finally, the guidelines were evaluated with other design teams in workshops. Findings from these workshops indicated that the guidelines are helpful for inspiring design teams to do more considered prototyping work, better suited for their design situations. However, more research is needed to better understand how long-term application of the guidelines influences design outcomes.

Keywords: interaction design, prototyping, design process, guidelines

Preface

This report presents a master's thesis project, conducted between September 2017 and June 2018 for 60 ECTS credits. The author was a student of Interaction Design and Technologies, at Chalmers University of Technology in Gothenburg, Sweden.

—

I would like to thank my supervisor, Staffan Björk, for the continuous guidance over the course of this project. Our discussions helped, among many things, to shape my understanding of research for the purpose of interaction design practise.

I also want to thank the experience team at Mojang AB, who provided the most intriguing, friendly and engaging environment to study the work and practises of interaction designers.

Oscar Nilsson
Stockholm, June 2018

Table of contents

1. Introduction

Prototyping is a core activity in interaction design, for producing artifacts of design, and to evaluate it. Prototypes serve as a proxy for the final product being designed, to accurately and efficiently be able to acquire knowledge about how certain designs perform in more realistic tests (Hartmann 2009).

Today, a spectrum of tools and methods guide interaction designers in constructing prototypes. Prototyping with software tools is an established practise in most interaction design teams (Subtraction 2015). John Maeda also notes in his *Design in Tech Report* that a new generation of computationally native design tools are emerging (Design in Tech 2017), for constructing smarter, more dynamic artifacts of design.

However, as designers' efforts move to building these rich interactive artifacts, influenced by available tools, there is an opportunity to stop and reconsider—a survey of designers by NEA in 2017 noted that *"Design fail when we spend too much time over-thinking and designing the wrong things"* as a pattern among the respondents. At the same time, the survey results indicates that more designers used prototyping tools, than conducted qualitative user research.

With this potential shift in focus for the efforts of interaction designers, with new tools being a driving force, prototyping practice and general methodology for conducting prototyping efficiently, is of high relevance when considering how interaction designers evolve their practises.

1.1 Background and objective

I encountered these challenges related to prototyping practises when working as an interaction designer at Mojang, a Swedish game studio, which is how the idea of this project was born. When designing user interfaces for their most popular game Minecraft (Mojang 2018), the requirements on our designs, together with an organizational situation that required communication with many teams, made the design team question how we could improve our situation with new methods and practises. With some efforts, new practises could then become a defining part of our design and collaboration efforts of the design team going forward.

This project, then, will investigate the landscape of prototyping methods for interaction designers, and evaluate them in the context of the design team at Mojang. A significant part of this investigation will also be to use and develop *new* practises for prototyping in design teams, which will try to address some of the identified needs from the Mojang team. Additionally, attention will both be paid to how prototyping practises relate to collaboration with other functions of the organization, and how different phases of the design processes influence practice. To add nuance to the result, the research aspect of the project will also include studying relevant design practice of several other design teams that face related challenges.

1.2 Describing the problem area

Today, the reality for interaction designers working for software and internet companies is that their products are consumed in a wide range of contexts, that all influence what the ideal design of the product looks like. Hardware and software ecosystems and network infrastructure provide technical opportunities to have software products available anywhere and everywhere, but also add complex, sometimes abstracts constraints on the product design. Cultural and linguistic intricacies also influence how a internationally distributed software products are appreciated by end users. For these reasons, a major subfield in interaction design is field research, to try to reach a better understanding of the different contexts of use.

This situation for designers, which was exemplified to me from my experiences working at Mojang, point at the difficulties facing the design team designing for cross-platform, globally distributed software applications, with users of all ages. Currently, when the design of a specific feature is discussed at Mojang, keeping track of different end-user platforms, their interaction patterns, users of different skill levels and other intricacies makes current available methods very fragile to work with, and design discourse is either imprecise, or scattered across a multitude of design artifacts, meetings and discussions.

Today, much design work is conducted in software applications for visual expression like Photoshop, which has little native support for working with the requirements of software interfaces, such as interactivity. While tools for working with prototypes, or interactive artifacts (these definitions are discussed in chapter 1.5 of this report), have gained wider adoption, the field is changing rapidly, and designers require more and more from their tools and practises. Additionally, new tools for computational design as described by Maeda (2017), aim to further address many of these issues of scaling design work through new technical tools and platforms.

However, to serve the same purpose as these new tools, there is also an opportunity in implementing better practices, which might be more accessible and adaptable between situations that designers encounter. With few established and commonly referenced theories for such prototyping practise, the risk is that designers processes are shaped by inflexible and potentially fragile domains of technical design tools, and their capabilities.

1.2.1 Collaboration needs of designers

In the software industry today, design functions of organization rarely operate in a vacuum, but rather as an integrated part of both product and business development processes (NEA 2018).

For this reason, designers are interdependent on other functions of a company, and often need to interact with other teams that both influence the design, and make sure that the designs eventually gets built and released to customers. These relationships are discussed by Houde and Hill (1998) as different audiences for a prototypes: *Users*, *Design teams* and *Supporting organizations*. What follows from this is that designers should consider how their practises and processes enable them to serve these different groups of stakeholders.

1.2.2 Design processes for interaction designers

Additionally, prototyping work also relate to where in the design process a project is. A survey by NEA (2017) notes how *"Prototype early and often"* is a pattern that is beneficial for designers. However, this recommendation provides little detail and nuance. Several factors come in play when considering this advice, such as finding what the definition of *early* and *often* means for the specific project that the designer is working in. Additionally, how to effectively evaluate and iterate on design work, which often influence the timeline of design processes, are activities of relevance when considering when and what to prototype, and how to do it.

1.3 Research question

The thesis poses the following question:

> *Which prototyping practises are relevant to consider for interaction designers, when designing for a multi-platform audience?*

To help answer this, this thesis also aims to answer these two supporting questions:

> *Which prototyping practises help designers collaborate with other functions of an organization?*

> *Which prototyping practises are relevant to consider for different stages in a design process?*

To give answers to these questions, this research study will be conducted in the context of implementing a design process and prototyping practises at Mojang. However, the aim of this report is having the result be more generally applicable to design situations with similar properties, such a having similar collaboration needs and multi-platform design constraints as outlined by the research questions.

1.4 Desired result

- *Analysis* of the landscape of prototyping practises and previous research findings, synthesised to achieve better understanding on what value different practises bring.

- A *case study* containing Implementation of prototyping practise at Mojang. These practises will then be evaluated in how helpful it is in the design situations of the team, both with regards to quality of produced designs, ease of use for the designer, and the quality it provides in communication with different stakeholders in the organization.

- Evaluation of research and case study, synthesised into *guidelines* for how design teams should act with regards to working with prototyping methods.

1.5 Working Definitions

This report will refer to *Design artifacts*, this is defined here to be similar to *physical or informational output from design work*, similar to as described by Kellogg and Carroll (1989) and discussed by Zimmerman and Forlizzi (2008). Examples of design artifacts include sketches, mockups, prototypes, and other things that designers produce to codify design work into entities.

The method of representation in these examples will be referred to as the *Design tool*, or the *Design medium*, where *medium* focus more on presentation formats, and *tool* focus more on methods of implementation. For example: A paper prototype can is implemented with the tools of pen and paper, but the presentation medium can be a recorded video, a storyboard, or an interactive application where a facilitator manipulates the paper prototype according to user input.

When a *system* is referenced in this report, this is generally thought of as a set of rules or principles that subparts of the system adhere to. For example, an UI design *system* of UI *elements* (the subparts), means that the UI elements needs to adhere to a set of rules that make them fit into the UI design system.

Generally, when *prototypes* are discussed in this report, it concerns prototypes specific for the field of interaction design according to a definition promoted by Benyon (2010), where prototypes in interaction design carry interactive qualities encoded in the design artifact.

1.6 Delimitations

The prototyping concept is commonly applied in engineering as a way of understanding what is technologically possible with a certain implementation technique (Houde and Hill 1997). This report will be limited from discussing prototyping for technological purposes, and instead focus on how the prototyping notion is used by designers to further design work.

Additionally, this report will be limited to investigating prototyping of user interfaces for screen-based interaction design, coupled with related input methods, including but not limited to touch screens, mouse, keyboard or gamepad interaction. Prototyping of design for hardware, electronics or industrial design will not be analyzed in this report.

This investigation will also be limited to analysing design and prototyping process for design teams and products that target a global, multi-platform audience. It is hypothesised that designers working with more specific audiences might not benefit from the same level of structure from their tools and methods – and therefore they will not be actively studied in this report.

2. Background

This chapter further describes the underlying factors that together shape the problem domain analyzed in this report. The company Mojang, and the design team, is elaborated upon, as it is the main subject of study. Furthermore, the current state of design tools and practises established at Mojang is described, together with more general considerations of working within the gaming industry, and the overarching field of software interaction design.

2.1 Mojang

Mojang is a game studio based in Stockholm, Sweden (Mojang 2018). The studio was founded in 2009 when the founder Markus Persson developed and released the game Minecraft (Mojang 2018). Mojang has also published and developed other games, but Minecraft is still the most popular game from the company, which is still continuously being developed and released as of year 2018.

In 2014 Mojang was acquired by Microsoft, but it still acts as an independent game studio, and releases games on all platforms. The company employs around 100 people in Stockholm, but also collaborate with Microsoft Game Studios, and other third party studios such as 4J Studios (4Jstudios, 2018). Work is coordinated remotely between the studios, together with frequent travel by team members between the offices.

2.1.1 Minecraft

Minecraft is the second ever best selling game, with 122 million copies sold as of December 2016 (Minecraft Twitter 2016). Only one other game has reported higher number of sales, *Tetris* (Polygon 2014). Minecraft is designed and developed for a wide array of platforms and operating systems, including Microsoft Windows, Apple macOS, game console platforms (Xbox, Playstation, Nintendo), VR headsets, Google Android, Amazon FireOS, Apple iOS and Apple tvOS. The game mixes two popular game mechanics: Creation through building and arranging elements, and a first person view. Adding to this, the game is set in a fantasy world context, with algorithmically generated landscapes of different terrain.

The game is popular among all age groups, and is very popular on Youtube, where video creators play Minecraft in different ways, and fans watch. Additionally, the Minecraft Education program distribute Minecraft as a tool for teaching school children, which used at many schools, mainly in the US.

Fig 2.1: Minecraft gameplay. The player adventures and builds in a world of blocks.

The Minecraft game is currently distributed as two different codebases that target different platforms – this is something that highly influence the release schedule, platform availability and community engagement of Minecraft. The *Java* version is available on Mac, PC and Linux, and is the earliest version of Minecraft. The *Bedrock* version is newer, and has the goal of becoming the main Minecraft version. It is distributed to smartphone platforms, consoles and more.

In the context of interaction design, Minecraft is carry many gameplay design aspects, and design of interfaces supporting gameplay. However, there is also many user interfaces outside of the core game, meant to support players in accessing a wide range of functionality related to Minecraft, such as social features, and a marketplace for buying and selling content from the game, such as maps, mini-games and player avatar appearances. Generally throughout the game, the user interfaces are styled in gray, with large visible pixel elements, to capture a retro style.

Fig 2.2: The Minecraft user interface in the menu system, outside of the game.

2.1.2 Experience team

The *Experience team* at Mojang, who works with user interfaces and user experience design for Minecraft, is the context in which this project is conducted. The tools, methods and processes will be developed, used and evaluated in the context of this team, and together with other functions of the company that the team interacts with, who are based either at Mojang in Stockholm (where I am mainly based during the project), or at Microsoft Game Studios in Redmond (WA). Microsoft Game Studios is a division within Microsoft who collaborate with Mojang on the design, user research and engineering for some parts of the Minecraft game.

In this design team, the design process consists of working iteratively with prototypes of new user interface design, which can then be developed and released according to the release schedule of Minecraft, which typically gets updated 2-3 times per year. To communicate progress and specific design decision widely within the company, a common technique is to record video presentations that demonstrate prototypes of design. This is helpful to enable multiple teams across multiple continents to stay in sync without regular meetings. Between the interactions designers in Stockholm, problems that arise when designing are collaboratively discussed and designed during two meetings each week, the *Design collaboration sessions*. These meeting have aspired to be similar to what is described of design review meetings at *Pixar Animation*, with collaborative sketching and brainstorming, as described by Ed Catmull in the book Creativity Inc (Catmull 2014).

The team also plan to conduct own user research, but today most user research initiatives is conducted by teams at Microsoft Game Studios. Another core ambition is to involve more Minecraft expert users, such as players who are very active in the community and publish videos on YouTube. The team has ambition to ramp up work with A/B-testing and more rich quantitative data, but at time of writing there are still technical and organizational challenges before those practises can be implemented.

The experience team is lead by Tobias Ahlin, Experience Design Director at Mojang. Tobias, who also helped initiate this thesis project, also contribute with previous experience as a senior designer for Spotify, GitHub, and as a Industry Leader at Hyper Island, a design and technology school in Stockholm.

2.1.3 Related teams and organizational functions

To further describe the context in which this project was conducted, listed here are the other teams and functions that the Experience team typically interact with. Often design artifacts are used for communicating and collaborating with these teams, and in some cases the artifacts are specifically produced for sharing some parts of the design with one of these functions.

Mojang teams in Stockholm

- *Creative:* The Creative team produce art, sound and other resources that define and express the Minecraft style. Such resources are a major part of the new user interface initiatives, to surface the Minecraft brand and emotional qualities better, to making the interfaces more engaging.

- *Engineering:* For developing parts of the Minecraft UIs, engineering teams in Stockholm needs to be consulted.

Microsoft

- *User research:* The Microsoft user research team within the Xbox division, which focus on user research for games. Within this team several members are focused on the Minecraft game, and are available for setting up user tests and evaluations, from which reports are produced and shared, that capture the findings.

- *UX design:* Microsoft Game Studios also have a team of UX designers dedicated to Minecraft. While the Experience team in Stockholm doesn't work with them on same areas simultaneously for the most part, the teams often exchange feedback and ideas about their ongoing projects.

- *Engineering:* For specific Minecraft engineering resources that Microsoft are responsible for, engineering teams at Microsoft are consulted.

2.2 Prototyping in Mojang design practise

Prototyping can be seen as a subgenre of the overarching concept of *envisionment*, as described by Benyon in *Designing Interactive systems* (2010). Envisionment, or any kind of representation of ideas, has long been a part of design practise. Sketches and models have been tools of architects and industrial designers for idea generation, communication, specification and evaluation. As described by Hartmann (2009), this practise of product designers have been assimilated by the practitioners in the software industry.

In the video game industry, which Mojang is a part of, design has long been a part of the process of creating elements of the game. In recent years however, interaction and product design specifically has become a more serious endeavor for many gaming companies. This is a trend that correlates with many games also acting as services which are continuously developed and improved upon (Kotaku 2017). Another example of the increased efforts in interaction design is how the games publisher EA have over the recent years had a dedicated studio in Sweden for user experience design related to their games, Uprise (Uprise 2017).

Generally, major game studios now have a team responsible for the user experience of being a player, which most often translates to making sure the game has usable interfaces and sensible service design. This kind of role is typically disconnected from the design of the core gameplay – Rather it concerns the surrounding interaction surfaces that wrap the core value brought by the gameplay.

When designing for these interfaces, the typical interaction design practices are seemingly used, but being in the context of video game applications can make some aspects of the work different. For example, gaming-specific hardware devices, unconventional interaction methods, and atypical user goals might require adopting different ways of working. While these increased needs for expression might influence prototyping practice, no general conclusions are made here about how prototyping practice for interaction design in games differ from interaction design of software applications. In the experience design team at Mojang, the design practise and it's goals are intentionally not considered different from other applications of interaction design and user experience design, in the overall software industry.

2.2.1 Prototyping tools used at Mojang

While no clear patterns in prototyping practises were established in the newly-formed Experience team at Mojang, several software-based prototyping tools have been used before this project started, which influences the way the Experience team works with prototypes. The used tools and their characteristics are described here:

Origami Studio: High-fidelity prototyping tool that originated from the design team at Facebook (Origami Studio 2018). The tool uses visual programming to enable the designer to add logic, states, interactions and data manipulation through connecting nodes in a visual editor. The editor also comes with many pre-built *"patches"* for adding various functions, interactions, input and output capabilities to the program in the editor.

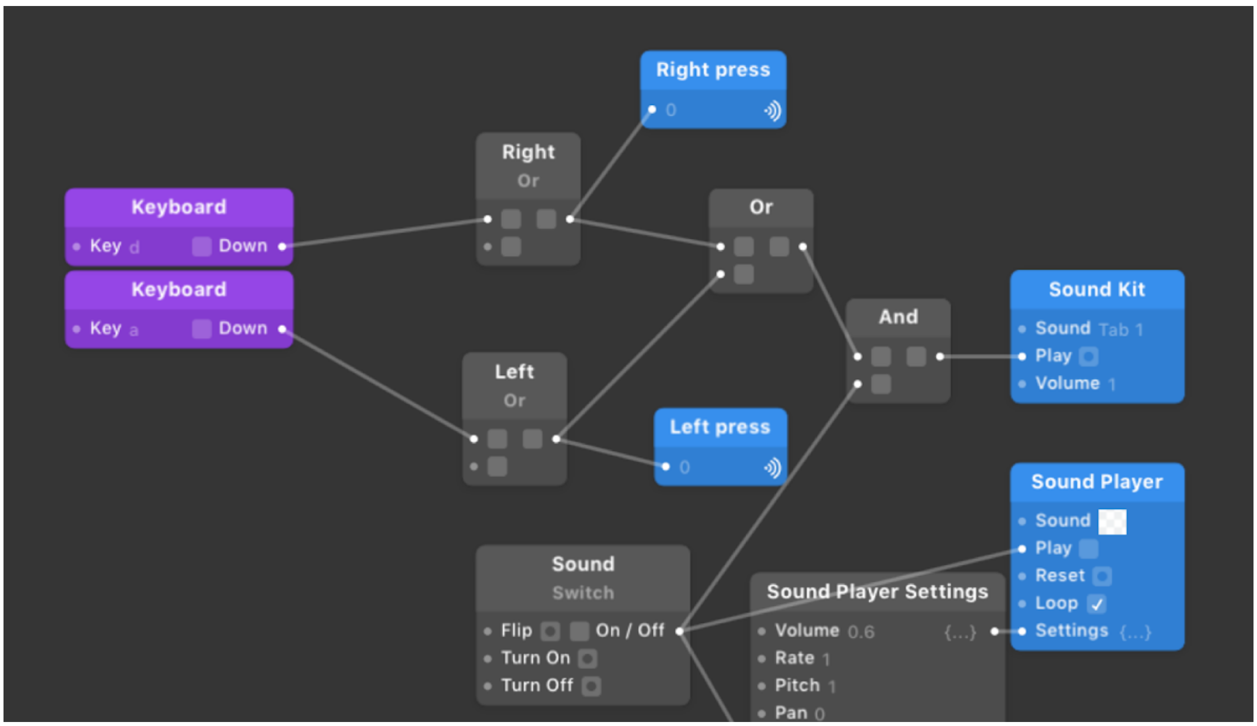


Fig 2.3: Example of node-link diagrams that adds interactive functionality in Origami Studio.

Marvel: A web-based prototyping tool for composing prototypes through adding images that represent screens of an app, and "tap areas" that takes the user to a new screen when tapped (Marvel 2018).

Unity: A game development platform (Unity 2018), which has been used at Mojang in the context of prototyping user interfaces that contain 3D-objects and many visual effects.

HTML, CSS, JavaScript: Programming and markup languages for web development, which have previously been used at Mojang for the purpose of prototyping user interfaces. While using these technologies as a prototyping tool requires web development experience, the technologies are capable for expressing a wide range of functionalities for user interfaces.

3. Theory

This chapter reviews a set of concepts and models related to interaction design practise and prototyping. As this project aims to analyze and design for *interaction design practise itself*, theoretical concepts describing the practise, and established methodologies, are collected here as part of the theoretical foundation for the project. While general design methodologies for conducting this project are collected in chapter 4, here the selection is rather based on what previous authors have connected to prototyping and envisioning practise, such as Hartmann (2009) and Ericsson (1995). Additionally, some of the concepts are discussed through the lens of prototyping or envisionment.

Concepts of design patterns, heuristics and guidelines are also discussed, to act as a foundation for this projects intended goal of providing guidelines for designers. Finally, published research specifically on the topic of prototyping is also discussed, which include both different categorizations of prototyping practise, and descriptions of aspects of prototypes themselves.

3.1 Design practise

This section describe several concepts related to work in the design field. These are hypothesised to be related to how prototyping practises and methods can be beneficial to the designer. How design practise operate with regards to these concepts will be analyzed in the further sections of this report.

3.1.1 Solving wicked problems

Problems in design are often described as *Wicked problems*, in comparison to problems that are *Tame*. (Rittel and Webber 1973). Tame problems are problems where you are able to identify that a solution has correctly solved the problem – This is the basis for the scientific method, and it’s also common in engineering.

With Wicked problems, however, it is impossible to judge whether a solution is correct. At best, one can determine whether a result is *good or bad*. Design is wicked “*because design has no special subject matter of its own apart from what a designer conceives it to be*” (Buchanan, 1992). This is also discussed from a research perspective in *What should we expect from research in design* (Gaver 2012).

Other defining characteristics of wicked problems include that it doesn’t exist a definitive, complete formulation of the problem (Rittel and Webber 1973), as an exhaustive list of all the characteristics of the problem cannot be procured from the given situation. Adding to this, it is also noted how wicked problems cannot easily be solved by trial-and-error, as each individual trial is costly and consequential to the problem situation itself—After a trial of a solution, the circumstances change from that action taken.

What often follows from this is a problematization of design practise, where it's noted that certain aspects of designing involves cognitive processes that are difficult to quantify, appreciate and optimize for.

3.1.2 Tacit knowledge

One way this concept has been reasoned about is the notion of *Tacit knowledge* (otherwise known as embedded or embodied knowledge), of which a theoretical foundation was presented by Polyani (1966), and further discussed in the context of interaction design by Moggridge (2007). Similar to Wicked problems above, Tacit knowledge is described as cognitive processes that cannot be easily explicitly expressed. Here Moggridge (2007) reasons about the nature of core design practise:

"Designers operate at a level of complexity in the synthesis of constraints where it is more effective to learn by doing, allowing the subconscious mind to inform intuitions that guide action"

Schön (1992) also describes an aspect of this notion when describing the interplay between a designer and design artifacts as a "*conversations with materials*", highlighting the surprise value in manipulation of physical manifestations such as clay, sketches or foamcore. This extends the concept of tacit knowledge out into the physical realm. The unique information that this activity produce is termed "backtalk", as if the material talks back to the designer while being manipulated. Furthermore, Schön elaborated on how this activity poses a challenge for future artificially intelligent design tools—design need representations in the physical realm to be able to progress.

3.1.3 Human-Centered Design

Another perspective on the value of more accurate, physical representations comes from a more Human-centered design mindset. Human-centered design considers how the design process should continuously involve insights from researching users and usage context, and then evaluating the design with users. Benyon (2007) notes how realistic prototypes increase the validity of testing prototypes with users. When designing, replicating the physical and sensory environment of the usage context of the end user, more representative testing data can be had, and designers can better understand how the design shapes the experience of use. Houde and Hill (1998) also notes that broader audiences of prototypes might require higher-resolution artifacts to increase accessibility.

On the other hand, Cooper (2012) recommends usability testing with low-tech prototypes primarily, and to only use specialized hardware when a very intricate interaction method is to be tested. This is also discussed by Erickson (1995), who notes how roughness in artifacts enable users to give better feedback to designers.

Designing in this more accurate medium lowers the bar for understanding a user appreciation of a design, compared to when design is made with abstraction layers in between. This concept is elaborated on in many of the design methods practised by the design agency IDEO, outlined in the IDEO Method cards (2003), where *Empathy tools* are employed both in research and development of design, to let designers and other stakeholders better understand the experiences of users.

3.2 Patterns and guidelines

Researchers and professionals within different design fields have long been concerned with the formation of patterns guiding design work. In the field of architecture, Alexander (1977) described how design patterns were of essence to the practitioner:

"At the moment when a person is faced with an act of design, he does not have time to think about it from scratch"

Together with foundational reasoning about why patterns exist, 250 patterns from architecture and urban design were published with the book *"A timeless way of building"* (Alexander 1977). Similarly, in the field of software engineering, design patterns describing engineering practises emerged, as described by works like *"Design patterns: Elements of reusable object-oriented software"* (Gamma 1994).

In interaction design, Tidwell (2010) praises Alexandres work for *"its rich interconnectedness, and its grounding in the human response to our built world"*, but also introduces patterns as a way to increase *habitability* and *familiarity* in the resulting design—if interfaces are built with certain patterns in mind, they will be easier to appreciate for users since they can reuse existing knowledge about how user interfaces work, for any new user interface they are presented with. Here, 250 design patterns for the design of user interfaces is presented, also referencing her work *Common Ground* (Tidwell 1999) as the precursor to the book.

Tidwell (2010) also introduces certain characteristics of design patterns as a preface to her work about interface design patterns, to separate the concept from other similar concepts such as heuristics or complete guides to how to solve a design problem. The six characteristics are that:

- Patterns are concrete, not general
- Patterns are valid across different platforms and systems
- Patterns are products, not processes

- Patterns are suggestions, not requirements
- Patterns are relationships among elements, not single elements
- Patterns are customized to each design context

Björk och Holopainen (2005) discusses design patterns in the context of gameplay design, and lists several characteristics for how patterns can relate to each other, and how they can form hierarchies. Furthermore, gameplay design patterns are also described as either being intentionally implemented by the game designer, or unintentional, with the patterns arising from other design decisions.

3.2.1 Comparing patterns and guidelines

Welie et al (2001) elaborates on a difference between guidelines and design patterns as tools for designers. Guidelines are noted as being lacking in specifying the context where they apply, and therefore a call for increased focus on defining design patterns is made. Examples are made of problematic definitions of guidelines, and how guidelines might lack a definition of what problem is solved, leaving the statement hard to evaluate.

Furthermore, it is noted how guidelines come in the form of “Do this” and “Do not do this”, in comparison to design patterns, which always promote design solutions currently in existence, i.e “Do this”. This position is further specified in the field of software engineering, where *Anti-patterns* is an established separate concept from design patterns, containing the “Do not do this” equivalents (Brown 1998).

Related to guidelines is how tools like IDEO method cards rather are presented as an *inspiration*, or a tool that showcase *examples* of methods (IDEO 2018). The inspirational purpose here can be seen as a function of making the advice more loosely specified—it might apply and might be good to be aware of, but the exact conditions are not given.

3.2.2 Design heuristics

Heuristics can be described as more generally applicable guiding principles. Nielsen (1995) introduces his set of *10 Usability Heuristics for User Interface Design* with that they are “*rules of thumb, not specific usability guidelines”*.

In the industrial design field, Dieter Rams (Vitsœ 2016) presents 10 *principles* for good design, noting how they could be a used as a way of questioning if a specific piece of design is good or not. The principles were assembles as a reaction to a perception of *“An impenetrable confusion of forms, colours and noises.”* in the state of the world.

As described in 3.2, Tidwell (2010) specifically describes her design patterns for interaction design as being different from heuristics, as each of them only addresses a specific situation, in comparison to heuristics which are more generally applicable.

3.3 Prototyping in Interaction design & HCI

The general notion of a prototype can be understood from a simple dictionary reference of the word. Among several definitions given, of different nuance, one is related to design:

"A first full-scale and usually functional form of a new type or design of a construction " –

(Merriam Webster 2018)

Within interaction design, however, usage of the term typically imply something more specific. Benyon (2010) notes that in interaction design, the main factor that distinguishes a prototype from screen sketches (and similar simple forms of envisioning) is that prototypes are in some form interactive—they respond to user input. Note that this interaction does not need to be digital or screen based – Benyon exemplifies the prototyping notion with an "interactive" paper prototype.

Despite this clear distinction, prototyping is still a term that can mean different things depending on the context of use in interaction design. Several authors have tried to capture these different genres of the prototyping activity in interaction design, either as frameworks and guidelines of work, or as specific design tools.

3.3.1 Categorization frameworks

A common approach among authors researching the landscape of prototyping practise includes categorizing different kinds of prototyping activities by finding notable differences in practise. These categorizations can then serve as a guideline for when to apply what methods and tools, together with examples and recommendations.

Houde and Hill (1998) analyse different kind of prototype artifacts and judge them by placing them on a triangle plane, with the corners representing different pragmatic goal categories of a prototype. A prototype, here, carries a certain value in uncovering either *Look and feel*, *Product role* or *Implementation details*, which together puts it somewhere on this triangular plane. *Look and feel* qualities are evaluated for how visual presentation is perceived by a user. *Product role* prototypes judge the value a product create for a user, and prototypes looking at *Implementation details* aim to produce deeper understanding of how technological constraints behave.

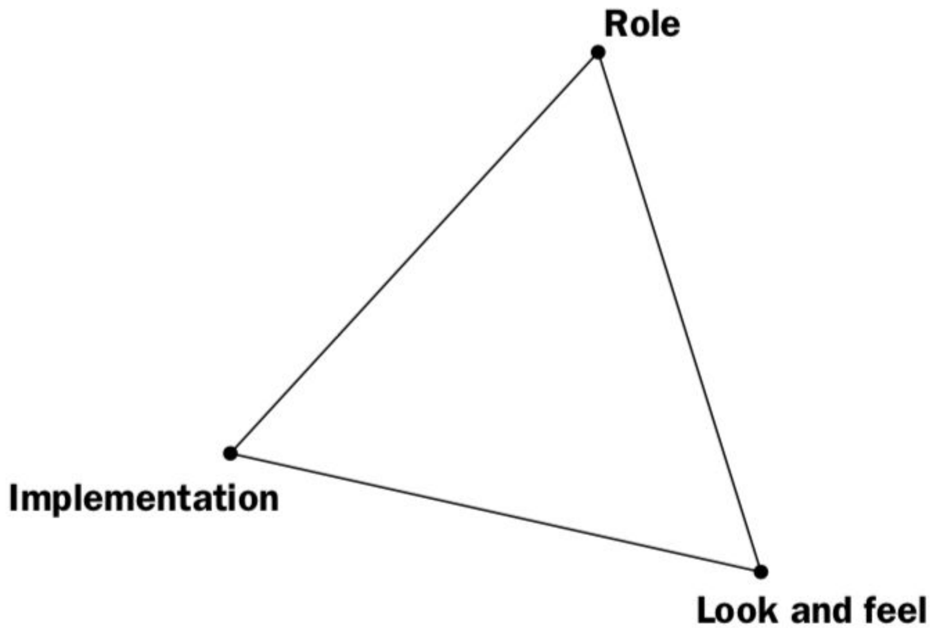


Fig 3.1: Houde and Hill's triangle model for different kinds of prototypes.

Experience Prototyping (Buchenau and Suri, 2007) further specify different areas of prototyping, by categorizing them by how well they help 1. *Understanding existing user experiences and context*, 2. *Exploring and evaluating design ideas* and 3. *Communicating ideas*.

At the design consultancy firm IDEO, a three stage process guide the usage of prototyping along the chronological timeline of a project: *Inspiration, Evolution, Validation*. Early on in a project, multiple prototypes are used to explore the solution space – Often these prototypes employ very different design solution. Following this, some prototypes (or ideas) are further evolved. Finally, in the later stages of a project, a more complete prototype of the final specification can be used to evaluate the design (Hartmann 2009).

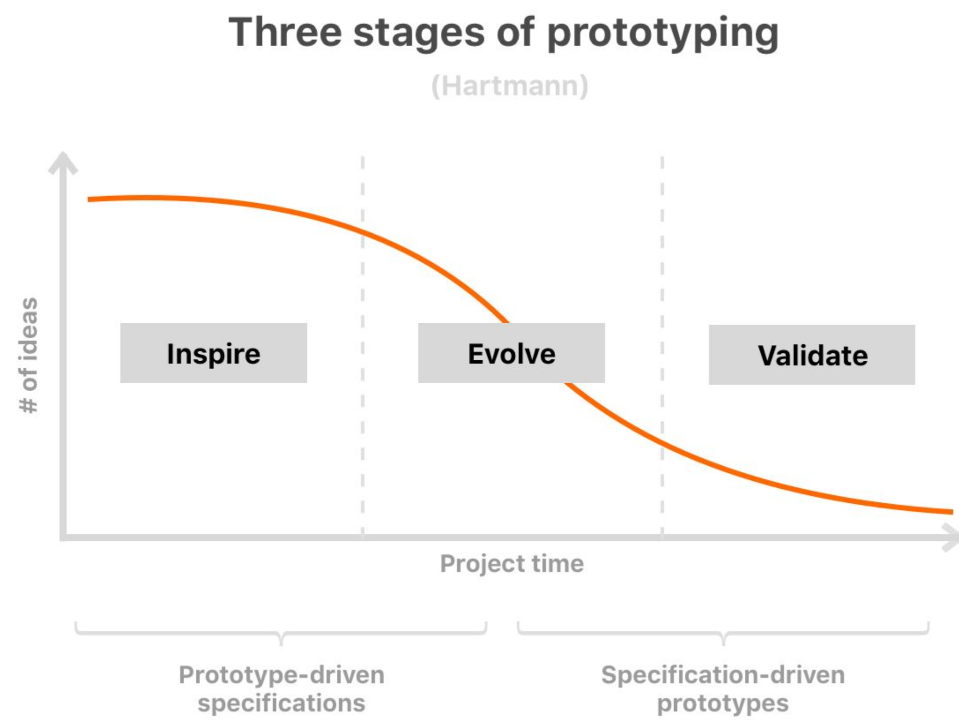


Fig 3.2: Stages of prototyping at IDEO (Hartmann 2009)

Lim *et al.* (2007) instruments going beyond focusing on the activities where prototypes play a part, and instead proposes an "anatomy" of prototypes themselves. Here prototypes are regarded as being made up of *filtering dimensions*, capturing the evaluatory purpose of the prototype, and *manifestation dimensions*, characterizing how the prototype is built and what it is supposed to represent. Manifestation dimensions are divided into three measures, here together with their definitions:

- *Material*: Medium (either visible or invisible) used to form a prototype
- *Resolution*: Level of detail or sophistication of what is manifested (similar to fidelity)
- *Scope*: Range of what is covered to be manifested

Benyon (2007) also discuss the concept of scope and resolution through the terms of *Horizontal prototypes* and *Vertical prototypes*, where horizontal prototypes represent wide range of functionalities with simpler means, and vertical ones represent a thin slice of functionality really thoroughly.

3.3.2 High and low-fidelity

A common way of characterizing prototypes is to talk about low fidelity (lo-fi) and high fidelity (hi-fi) prototypes. A typical differentiator between these categories is how the prototype can be tested:

With low fidelity prototypes, often characterized as paper prototypes, there often need to be a facilitator that “acts” as the computer, and can help explain the situation of use to the user and what actions are available. When a user decides on an action, the action is executed upon by the facilitator, who changes the appearance of the prototype, e.g by moving sheets of paper, sticky notes etc. that represent the computer interface designed for (Benyon 2010). Expanding on this, prototypes both be intentionally low-fi, to encourage more specific feedback on the design that is captured, or it can be a way of lowering the effort in creating the prototype. In the set of Human-Centered methods presented by IDEO in Method Kit (2018), *Rapid Prototyping* is presented as form of fast-paced prototyping, where less concern is given to the medium of presentation, rather to quickly capture the idea as something tangible.

High-fidelity prototypes, on the other hand, are typically software-based (Benyon 2010). Here, the prototype can be indistinguishable to a real application for the test subject, and no facilitator is therefore needed – The medium is capable enough to mimic real applications. Even if the application is simplified behind the scenes, or is missing some functionality, it can still be used as a fully implemented interface. Another area where the higher fidelity prototypes might useful is when trying to achieve aesthetic design qualities in products, since the visual experience of a hi-fi prototype can be similar to the real thing. Industrial designer Dieter Rams notes on aesthetics in relation to function – *“The aesthetic quality of a product is integral to its usefulness because products we use every day affect our person and our well-being”* (Vitsœ 2016).

In comparing the concept of fidelity with the manifestation dimensions from Lim et al (see 3.2.3), it can be regarded as being mostly tied to *Resolution*. However, looking at the examples given above, *Scope* and *Material* certainly add to a general appreciation of fidelity.

Tangential to the concept of fidelity, or whether the prototype is in software or paper, is the medium of presentation chosen for a specific prototype. Researchers such as Mackay et al (2000) have highlighted how recorded videos can be an accessible way to communicate design and engineering requirements.

3.3.3 Component-based systems

A commonality among many prototyping methods is the reliance on component based systems for composing prototypes. When arranging materials, one way of quickly reaching useful results is to arrange components who behave according to a set of rules, and that have pre-built abilities. One example of such a system is LEGO, where plastic parts are made to fit onto each other according to a set of defined rules. While a common use of LEGO is as a toy, it's ability to quickly conjure "design" makes it a popular tool in certain professional contexts. LEGO Mindstorms even extends the LEGO system so motorized, programmable components suitable for prototyping robotic systems (LEGO 2017)

Similarly, many design tools and methods rely on using a system of interface components for composing user interfaces out of a set of predefined interface components. One such example is Balsamiq mockups (Balsamiq 2017), where all interface components alight to a certain visual style. Another system is Google's Material design, of which fully featured interfaces can be composed according to Google's design standards (Google 2018). With Material Design, the interface components have predefined visuals, methods of interaction, and hierarchical relation to each other.

4. Methodology

This section describes the toolbox of interaction design methods that are available when it comes to producing research, structuring design work into design processes, and activities that can be performed as a way of designing. Collectively, these methods aim to be of use when analyzing and designing for the wicked problems that are common in the design field.

For this project, methods will both be a analyzed in their relation to how they work together with certain design tools and prototyping methods, but they will also be used as as means for the actual design of the guidelines that aim to be the end result of this project.

4.1 Design research

This sections describes different approaches to performing research in the design field.

4.1.1 Research through design

Research through design, as described by Gaver (2012) is the process of deriving insight from actual design work done, either by own practise, or synthesised from research of others'. The output is described as *annotated portfolios*, which work as a meta analysis of design work done to solve a specific problem.

A somewhat wider stance is taken by Fallman (2008) who describes three categories of design research, together visualized as a triangle of alignment for an instance of research. Here, design research is considered as a combination of *Design Practice*, *Design Studies* or *Design Exploration*. Design Practice regards research made in close conjunction with the industry, studying aspect of the field relating to the implementation context, often in commercial organizations. Design Studies, on the other hand, is research acting on a more macroscopic scale, looking at the design field without the context of a specific activity, but as a whole. On the third axis, Design Exploration describes how specific instances of new design craft influence different aspects of society.

Fallman describes any of these areas as valid design research, and that each sub-field is important for design research as a whole. Adding to this, it is described how a research project can move between these different dimensions during a project, based on differing needs and goals for different parts of the project. For instance, if design research is done on multiple instances of design practise, certain meta analysis can be made which later moves the focus closer to design studies.

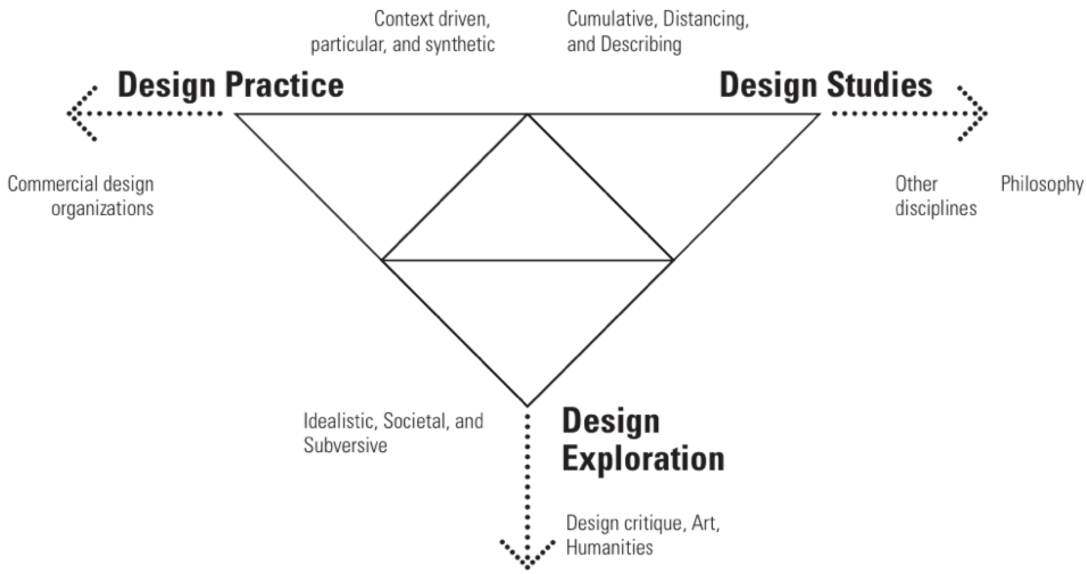


Fig 4.1: Fallman's (2008) triangle of different aspects of design research

For design practise specifically, Fallman notes the importance of the researcher acting as a designer on the team, not as an observer or researcher, to get access to the embedded knowledge appreciated from acting as, and together with, design professionals.

4.1.2 Formulation of guidelines

Similar to the annotated portfolios of Gaver (see 4.1.1), the formulation of guidelines for designers is an established methodology for synthesising empirically observed patterns in design into more terse abstractions, usable by designers in their work.

Alexandre's "The Timeless Way of Building" (1977) describes the formation and rationale behind one early attempt at such a system, in the context of architecture. Here, each pattern is described as a *"rule which establish a relationship between a context, a system of forces which arises in that context, and a configuration which allows these forces to resolve themselves in that context"*, which then is described with this shorthand:

Context ➔ System of forces ➔ Configuration

Several influencing factors in the successful formation of patterns is discussed, including presenting patterns in a way which is *operational*, and *shareable*, describing how the patterns not only concern a formulation of what is observed, but to also have a specific audience and use case in mind; here the architect using the patterns.

In the context of prototyping, several authors have identified guidelines meant to generally support prototyping work. An example of this is given by Houde and Hill (1997) in *What do prototypes prototype?*, see chapter 3.3.3. Here, many examples are given to support each given guidelines, similar to how Gaver (2012) annotates portfolios of design, but the examples tend to include more details of the design situation, to support the guidelines being more focused on design process than actual design output.

“Most fundamentally, annotated portfolios respect the 'ultimate particular' of the designed artefact, rather than abstracting across instances as pattern languages do, while allowing for the 'extensibility and verifiability' for which some of the HCI design community have called.” (Gaver 2012)

For the context of this project, the resulting guidelines are seen as a product of a design process, where practising designers are the target end users of the guidelines. Similar to how Gaver (2012) considers respecting the *ultimate particular* of having the observed patterns being closely tied to a concrete set of examples, the examples from the design practise, and the design process, should be provided when proposing a set of guidelines. Producing guidelines can be viewed as commentary to a specific design research context, which inseparable from the final result.

4.2 Design processes for interaction design

This section described different process models for planning and organizing design activities, that also more specifically have been suggested for use within the field of interaction design.

The overarching goal of many of these processes is to give designers tools for in acting within an environment of wicked problems (see 3.3.3) and where qualities of a design can be tacit. These circumstances of the design situation require operating in flexible, iterative mindsets, and through repeatedly exposing the the design to the real context being designed for.

Adding to this, defined design processes and methods also give designers a shared language, describing how to do work, which helps in collaboration and repeatability of work.

4.2.1 Iterative design

Design processes are often presented as iterative. Moggridge (2007) have one example of this, with the stated caveat of it only presenting a general pattern, and any stage of the iteration being available at any time.

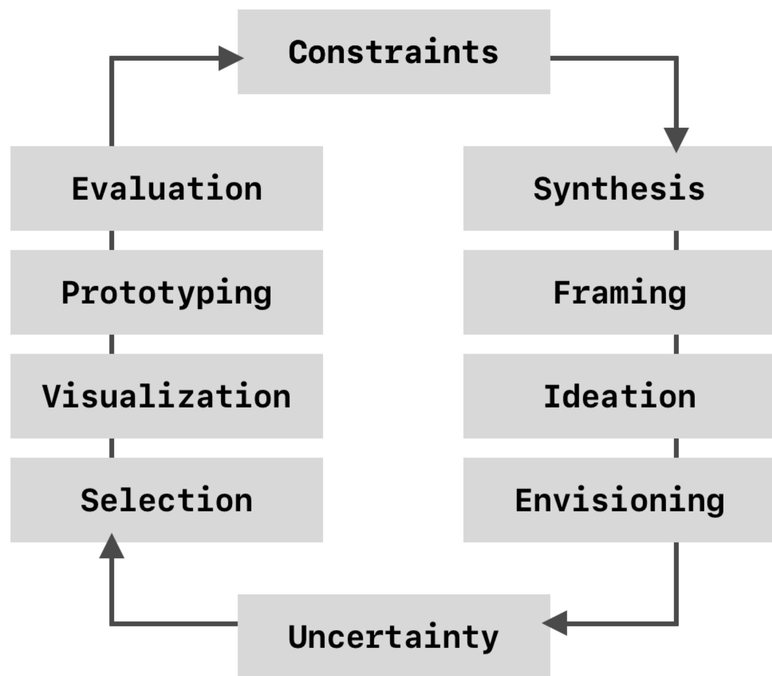


Figure 4.2: Moggridge's (2007) Iterative design process model

4.2.2 Jones' Model of the Design Process

The Jones model (Jones 1970), while not iterative, was early in capturing some of the thought in the Moggridge model into three distinct phases, modeling the general sense in which design processes progress over the course of projects – Divergence, Transformation and Convergence. Divergence represents the initial phase, which is the open-minded search for new insights and variables that can influence the design, such as research, initial ideas, observations about the context, and currently existing design. It means to widen the knowledge space, and further establish the purpose of the design activity.

In the transformation phase, research insights and and potential solutions are synthesised and elaborated upon with more concrete suggestions of design solutions. Convergence aims converge all prior work into one final design, through testing, evaluation and firmer specification.

Jones' model can be implemented in a design project with different forms of iterative process. Each iteration can either consist of the entire Jones' model, where each loop in the iteration contains the three different phases outlined above, with their corresponding activities and methods. This is similar to how the British Design Council (2017) describes a Double Diamond process, where a project follows the Jones' model in two distinct iterations.

Optionally, iterations can happen within the phases – E.g several iterations of divergence can be followed by several iterations of transformation, and then several iterations of convergence. This can be seen as using the Jones' model at different levels of scale; either in the macroscopic level of a whole project, or within the separate activities.

4.2.3 Human-centered design

Human-centered design is a design methodology whose process and methods try to make knowledge of the end user influence the end product. At its core, it's a way of making design empathize with the humans using the end products, instead of designing something for a certain business goal. The core process involve four activities:

1. Understand and specify the context of use
2. Specify the user and organizational requirements
3. Produce design solutions
4. Evaluation of design against requirements.

Human-centered design also include iterating over this core process, incorporating insights from the evaluation phase as more high-quality understanding in the research phase.

4.2.4 Design thinking

Design thinking can be seen as the most general incarnation of an iterative design process. It defines three distinct phases of an iteration: Inspiration, Ideation and Implementation. Iterating within this loop then make up a project.

Being such generally stated concept, it's become popular in areas that doesn't have design practitioners in the typical sense of the word – For example, it have be used to reimagine business models of corporations (Brown 2007), and to introduce new laws in countries such as Finland (2017)

4.2.5 Design sprint

Design sprints can be seen as a specific implementation of Design thinking, acted our over five days with a strict set of activities per day. GV (2018) describes the design sprint week as:

- **Monday:** Map out the problem
- **Tuesday:** Sketching of competing solutions
- **Wednesday:** Making of difficult decisions (based on competing solutions), turn ideas into a testable hypothesis
- **Thursday:** Build a high-fidelity prototype

- **Friday:** Test the prototype with real humans

4.3 Methods

The methods described here are categorized according to where they are hypothesised to fit into the different phases of Jones' Model for this project.

4.3.1 Divergence methods

Divergence methods (Jones 1992) share the characteristic of introducing new knowledge and ideas into the design process, or the project. The goal is to extend the boundary of a design situation.

Interviews: A research method for directly questioning a subject about certain topics (Martin 2012). Interviews can have different amounts of structure to them – It can either be more like a free-form discussion, or more structured where predefined questions are asked. One example of a more specific form of interview is *Directed Storytelling* (Martin 2012), where the questions asked explicitly involve making the subject retell their experience of something. Follow-up questions can help the researcher dive into more detail of specific parts of what the subject mentions. How you pick the subject of the interview can be of interest – As a more specialized form of user research, IDEO method cards (2003) describe *Extreme user interviews*, where more extreme, outlying users are interviewed as a way of getting unique insight into the more complete problem space, that might influence the design even for the average users.

Questionnaires: A research method for asking multiple subjects of answers to a set of questions. (Martin 2012). Questionnaires can be more efficient than interviews by being more easily distributable – This can help when more quantitative data is wanted.

Literature review: By studying previous art from within the company, or from the specific field, relevant insight about a certain design area can be synthesised. This can include internal documents, industry reports or articles published on the web. (Cooper 2014)

Fly on the wall: Research method for analyzing user and usage in the appropriate real-world context. By going out to where the situation designed for occurs naturally, and passively observing events, more representative data can be gathered. (IDEO 2003)

Brainstorming: Brainstorming is a method for a group of people to generating a large number of ideas that relate to a certain problem formulation. Generally a central guideline for a brainstorming session include to not being critical of ideas, but to instead build on what is already said – This help embrace all ideas that might arise among the participants, helping the team generate a bigger quantity of ideas or thoughts about the problem area. (Kelly and Littmann 2014)

4.3.2 Transformation methods

Transformation methods (Jones 1992) aim to synthesise and identify patterns in the generated knowledge base from the divergence phase. Methods help making the different constraining aspect of the project more actionable, and bring clarity to which potential solutions are available.

Personas: Cooper et al (2014) describes personas as a way of synthesising all previously gathered knowledge about potential users into model users, that are convenient to use when communicating needs, desires and usage context of the entire user base. Instead of having to take every single unique property of all users into account, personas can abstract the data set down into a manageable set of user characters, useful for efficiently defining requirements and to benchmark design solutions. Adlin & Pruitt uses similar terminology when describing personas as *"fictitious, specific, concrete representations of target users" which "put a face on the user—a memorable, engaging, and actionable image that serves as a design target. They convey information about users to your product team in ways that other artifacts cannot"* (Adlin & Pruitt 2010).

The use of detailed, engaging personas is not universally accepted – For example, Norman (2015) exemplifies the detailed descriptions in personas as a problematic aspect of human centered design, in his essay *Human-Centered Design Considered Harmful*, where he calls for designing of stronger conceptual models instead of attending to each and every detail about users.

Participatory design: Often described as a movement originating in Scandinavia, which aims to involve specific users in the design process (Ehn, 2008). Having users close at hand like this can help design projects where intimate knowledge of the domain is required to understand how to best solve the user's problems. Moreover, Cooper describes this as a risk in comparison to personas, since the participating users can be too specific. Personas, on the other hand, can be a more general model that better represent a broader range of intricacies in a target group, since they are modelled after aggregated sets of users. (Cooper 2014)

Scenarios: A scenario is a written story/narrative that describes an archetypical scenario that involves the product designed for. These can be used to quickly reference the contexts of use for a design. Scenarios can be based on personas, which then is denoted *persona-based scenarios* (Cooper 2014).

KJ method: A method that involves sorting and categorizing ideas or thoughts (most often written on pieces of paper) together in a group. Generally the goal is to find an intuitive sorting that makes sense to the most people in the group. The method can help team members make sense of and find patterns in a problem or solution space. It was invented and named after the Japanese scientist Jiro Kawakita (Spool 2011).

4.3.3 Convergence methods

The convergence phase (Jones 1992) aims to, based on all previous results, eliminate design solutions until the final, most refined and tested solution to the problem remains.

Usability testing: With usability testing, a user is asked to complete a certain goal with a software design being testing. The users can then be observed and recorded (sound, video, notes etc) by the evaluators, which then produce material for analysis (Cooper 2014). Optionally, the user can think aloud when completing the task, providing the evaluator with extra information about the cognitive process required when using the software that is tested.

Co-discovery: Benyon (2010) describes co-discovery as a design method for evaluating while having an open discussion with participants, and observing how the discussion goes and what is focused on.

Heuristic evaluation: A method when experts evaluate design according to a set of heuristics (Nielsen et al 1990). In interaction and design for usability, a commonly used set of heuristics is the *10 Usability Heuristics for User Interface Design* from Jakob Nielsen (Nielsen 1995), *another is the 10 Principles for good design*, by Dieter Rams (Vitsoe 2017). Given a design, an evaluator are to judge if it fulfils each of the stated principles. Since it's not guaranteed that all aspects of a design is judged appropriately, multiple evaluators can be employed to get a more complete evaluation.

Controlled experiments: Benyon (2010) describes controlled experiments as a structured way of testing when all factors are controlled to an extent that a design question is isolated enough, where the result from the test leaves no room for interpretation. While the method provided high-quality data, it is also described as being time-consuming.

4.3.4 Flexible methods

Prototyping: Prototyping is often regarded as a distinct method or activity for producing preliminary artefacts of design. However, different styles of the activity are often described as suitable for different parts of the design process, or for different purposes. Prototyping frameworks such as Experience Prototyping (See 3.3.3) or the IDEO model of working with prototypes (see 3.3.3) both describe prototyping activities mapping to all of the design process phases of the Jones model presented above.

In comparing Jones' description of the divergence phase with Schon's (3.3.3) view of manipulation of design manifestations, one can regard prototyping activities as divergent research into the capacity of the materials worked with. Alternatively, the prototype can act as a final manifestation of a design that is to be evaluated with users, placing it as a method useful within the latter phases of Human-Centered Design (see 3.3.3).

Workshop: As workshops is a general method for working on a specific topic in groups, it can be applied anywhere in the design process where exposure to users is deemed relevant, either as a form of research, participatory design, or for evaluating design solutions. A workshop can also be conducted with designers or researchers being present in the sense of *Fly on the wall*, described in 4.3.1. Spinuzzi (2005) notes that participatory design (as described in 4.3.2) typically involves workshops.

Co-creation: Method for working directly with users to involve them in design work. This can be helpful to surface potential solutions that are derived from the deep knowledge of a system that a existing user might already have (Martin 2012). Depending on the level of involvement, and open-endedness of the activities, one can regard the derived value from the activities as flexible in what context of Jones' model is contributed to.

Card sorting: Method for getting quick feedback on concepts visualized on cards, by having one or more participants arrange them in an order according to some specific parameter such as preference (IDEO 2015).

5. Planning

This chapter outlines the initial planning for how the desired result was to be reached over the course of this study, based on what was known at the start of the project.

The overall project was divided into different phases, that generally corresponded to project iterations according to how Jones' model is described to be implemented in section 5.1. For each phase, the main planned activities were described, as seen in the time plan in section 5.4.

Given the research questions of this report, this study was planned to investigate prototyping practises, with the goal of arriving at new insight into how prototypes can be built and used. However, finding these insights was determined certainly be in the domain of wicked problems – No test could help determine if the insight were correct; at best, I could say if something is good or bad in a certain tested context.

For this reason, the overarching structure of this investigation was planned to be an iterative loop that continuously refined the results of the previous iteration, based on the evaluations of the certain "designs" (I.e the synthesised prototyping methods and practises), and other research. The evaluation was to consist both of applying the ideas to be evaluated in the context of interaction design at Mojang, when designing for the game Minecraft, and through evaluating together with other design teams.

This design process makes the stages of the iterative loops akin to the the stages in Jones' model of the design process. These iterations were to produce two things—An end product was to be produced, consisting of a final set of practises relating to the three research questions. Complementing this, when analyzing the iterations completed, there were to be many examples of similar design situations that arose, and how the methods of choice at the time handled that situation. These examples, together with case studies of other companies facing similar problems, were planned to be used for meta analysis through a method similar to how Gaver (2012) describes *annotates portfolios* as research through design. This annotated library of examples were to complement the end product from the iterative design work. The two results were then planned to be used to produce a final set of answers to the research question.

The following sections of this chapter describe what was planned for these three distinct iterations, followed by the time plan containing specific activities.

5.1 Divergence iteration

In the divergence iteration, data and new knowledge were to be gathered and synthesised by studying design practices, prototyping tool use, characteristics of different prototypes a design practise outcomes. Methods here include interviews, fly-on-the-wall field research, and competitive analysis of already established recommendations from other researchers and practitioners.

In this phase the project was planned to be established at Mojang, where routines for working with prototyping within the company were to be identified, and also initiated.

5.2 Transformation iteration

In the transformation iteration, Insights from the divergence phase were to turn into hypotheses of what designers need from their prototyping methods and practises. Then new methods and practises were to be generated by brainstorming, co-creating and testing new practises together with other designers at Mojang who were interested in improving how they work with regard to prototyping.

5.3 Convergence iteration

For the convergence iteration, the hypotheses of design practises were to be evaluated based on the outcomes, both looking at produced design work, and produced value for the design team and overall organizational context. Additionally, the design practises were to be evaluated through comparing them to design processes and experiences from case studies of other design teams.

5.4 Time plan

The below sections described the time plan for the project, as considered before the project started.

- **Planning, initial research** *(Week 1-8)*
  - Writing of planning report
  - Domain research
  - Initiate project within Mojang
- **Prepare for case studies** *(Week 9-14)*
  - Develop prototyping practises or tools to meet the specific needs of the design team at Mojang.
  - Establish contact with other companies for doing case studies.
- **Design work, case studies** *(Week 15-28)*

- Use prototyping practices at Mojang, evaluate their function
  - Using practises for collaboration with other designers, UX Researchers, Artists, Engineers, Product managers and other functions of the company that the team interacts with
  - ***Halftime-presentation (Week 20)***
  - Case studies of other design teams
- **Evaluate and synthesise** *(Week 29-32)*
  - Analyse the effects and outcomes of the design and prototyping work from the team
  - Gather data from involved parties
  - Produce guidelines for interaction designers related to prototyping practices
- **Report writing** *(Week 33-40)*
  - Preparing thesis report and presentation
  - Opposition of other group

6. Process

This section describes the research process for this thesis project. The overall goal was to produce a set of guidelines, through studying, practising and evaluating with other design practitioners who do prototyping work for the purposes of interaction design. The process is generally made up of an iterative design process, with each iteration containing methods used to serve the different purposes of Jones' model of the design process. These methods and processes are described in detail in the methodology section of this report.

6.1 Pre study

For the first weeks of the project, research on the theme of prototyping for interaction design was synthesised into the background and theory sections of this report. General interaction design methods and processes were described in the methodology chapter. This research served to influence the initial set of guidelines, and initial work that proceeded at Mojang.

The relevant research on the topic was initially found through reading recommendations provided from the course *Prototyping in Interaction Design*, given to students of interaction design, which I read one year prior to starting this project. Through following references, and getting more reading recommendation from my supervisor and other teachers, more literature relevant to the project was amassed. Adding to this, internet searches for relevant keywords were made to get an overview of related work published on the web.

At the same time, a new team (described in 2.2.2) was established within Mojang, the Experience team, and introductory team building activities and project planning were completed. An initial workflow for prototyping and design system work were established to meet the immediate needs of the interaction designers. Furthermore, observations of how the the design work proceeded in relation to design tools were ongoingly collected, together with informal evaluations of the early design process.

The scope of team's project was defined, and it was presented to leadership team of Mojang and Microsoft, who responded positively.

6.1.1 Planning the project after the pre-study

This chapter outlines the planning for how the desired result will be reached over the course of this project after the pre-study, in the format of iterations to be completed, activities to be performed and approximate progress. In comparison with the planning made before the start of the project, this planning takes into account findings from the initial research into the prototyping field, and contains methods collected in the Methodology section, which here was determined to specific purposes within the iterations.

First iteration – *Week 11-17 (9 weeks)*

- Conduct additional domain research and elaborate more on methodology and theory sections for the final report.
- Implement some practises from the pre-study in Mojang design work
- Get a deeper understanding of the different functions of Mojang, and in which way the design team will collaborate with them through prototypes.
- Evaluate practises and needs of the design team
- Write first set of guidelines
- Reach out to other companies than Mojang to study design practise

Second iteration – *Week 18-24 (7 weeks)*

- Use and iterate on guidelines through design practise at Mojang.
- Evaluate guidelines with designers at other companies that face similar challenges.
- Halftime-presentation (Week 20)

Third iteration – *Week 25-30 (6 weeks)*

- Use and iterate on guidelines through design practise at Mojang.
- Evaluate guidelines with other designers
- Estimated final release of Mojang project, new design is shipped to users, further providing feedback into the success of the design process.

Evaluate and synthesise – *Week 31-32*

- Gather data from all involved parties, conduct interviews.
- Produce final set of guidelines for interaction designers and organizations

Report writing & Presentation – *Week 33-40*

- Preparing presentation
- Opposition of other group

6.2 First iteration

September ➔ November

Based on my pre-study about prototyping theory and methods, the first iteration started as I booked my first session for evaluating my insights into prototyping, which was going to be a teaching opportunity for students at the Hyper Island school in Stockholm. There I would work with the students on prototyping problems, and hold a presentation introducing the students to prototyping tools and methods. The purpose of this event for this project, was to generate new insights into the viability of my initial prototyping guidelines, which would then be used in the second iteration on the guidelines.

While the events leading up to the Hyper Island day defined the timeline of the iteration, My work together with the designers at Mojang continued—This acted as a research aspect of the iteration, while the work of synthesising these insights and the pre-study later became the outcome of the iteration in the form of my presentation, which was accompanied with guidelines. The Hyper Island presentation was followed by a presentation and discussion with a community of interaction designers in Stockholm, IxDA, further adding to the evaluative qualities of the iteration.

6.2.1 Mojang prototyping work

During the timeline of the first iteration on the guidelines, I continued to initiate the prototyping practise within the team, and deepened my understanding of the value of prototyping for the project. Me and the other interaction designer built more prototypes, while I gathered insights into the craft itself and the value that was created for the team, and the larger organization.

During this timeline, a small small library of prototypes existed form previous exploratory work, but many design decisions were still made through designing only static mockups or wireframes. Some prototypes were built, particularly to evaluation certain interaction patterns such as scrolling and navigation. Two prototypes were built that explored the navigation of the Minecraft menus through composing of existing wireframes into interactive prototypes. These are referenced as *Navigation prototype #1* and *Navigation prototype #2* in the following chapters of this report, where they are further analyzed.

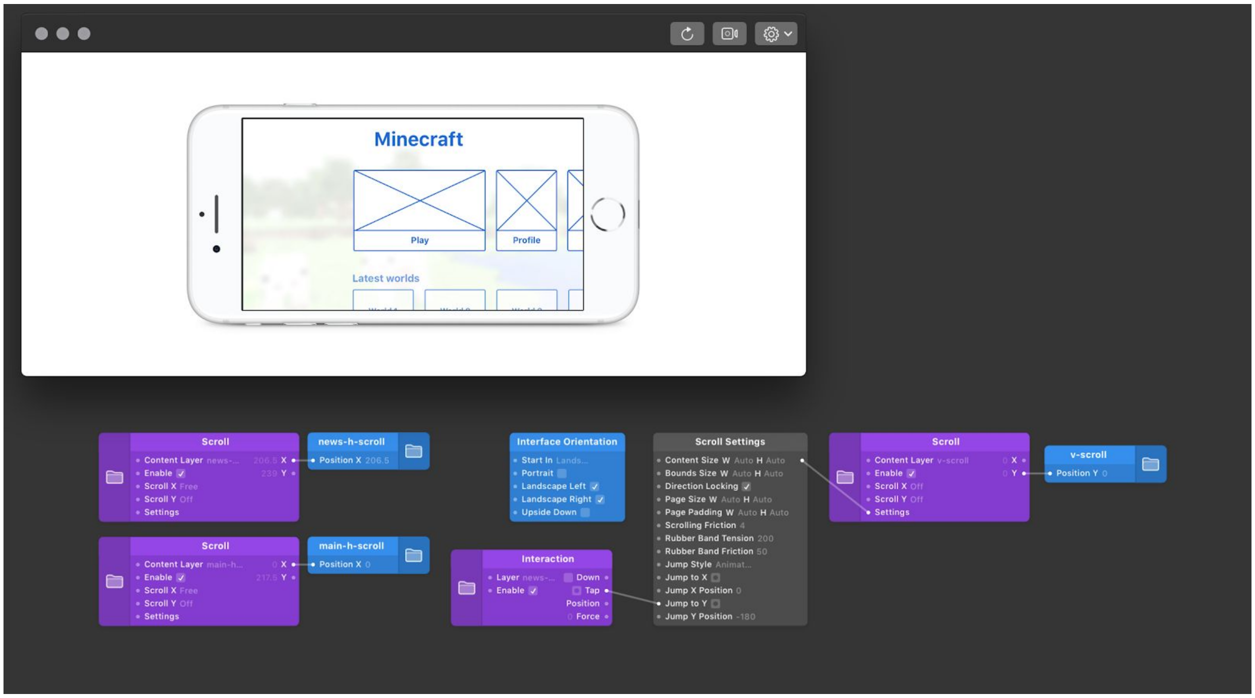


Fig 6.1: Navigation prototype #1 being built in Origami Studio

Tool wise, an early web-based prototype from an earlier exploratory stage of the project was phased out, as the team wanted to start from scratch. Tools used at this stage included Sketch, a macOS-based vector drawing app, for producing wireframes. Principle and Origami Studio, both macOS based prototyping apps, were used for adding interactivity to the designs.

6.2.2 Evaluation and research with design students at Hyper Island

During the pre-study, the design manager in our team introduced me to a friend responsible for a UX course at Hyper Island, a school in stockholm teaching design and technology. At Hyper Island, I was given the opportunity to teach and talk to the students about prototyping for one full day, through a presentation and a workshop. For this project, the idea was to have this opportunity serve as a starting point for evaluating the need for prototyping tools in a very pragmatic context—the students in the course were all working with clients on projects that required prototyping, and this served as a good opportunity to understand their needs.

A presentation titled “Purposeful prototyping” and workshop at the school Hyper Island was completed week 8, earlier than expected after a rescheduling, which delayed other activities. The presentation material, including the presented recommendations about how to work with prototyping methods, was to be used as the starting point for the initial set of guidelines, that were to be iterated upon during the following iterations of the project.

The presentation was given, but together with many discussions together with the students, answering questions and describing my experiences. Following the presentation, students started working on prototypes for their own projects, and I gave my hands-on help choosing tools, setting up projects and prototyping initial ideas.

An effort was made to produce presentation material that considered the big-picture purposes of building prototypes, but also specific examples and techniques. Figures 6.2 – 6.5 represent some of the concepts in the presentation.

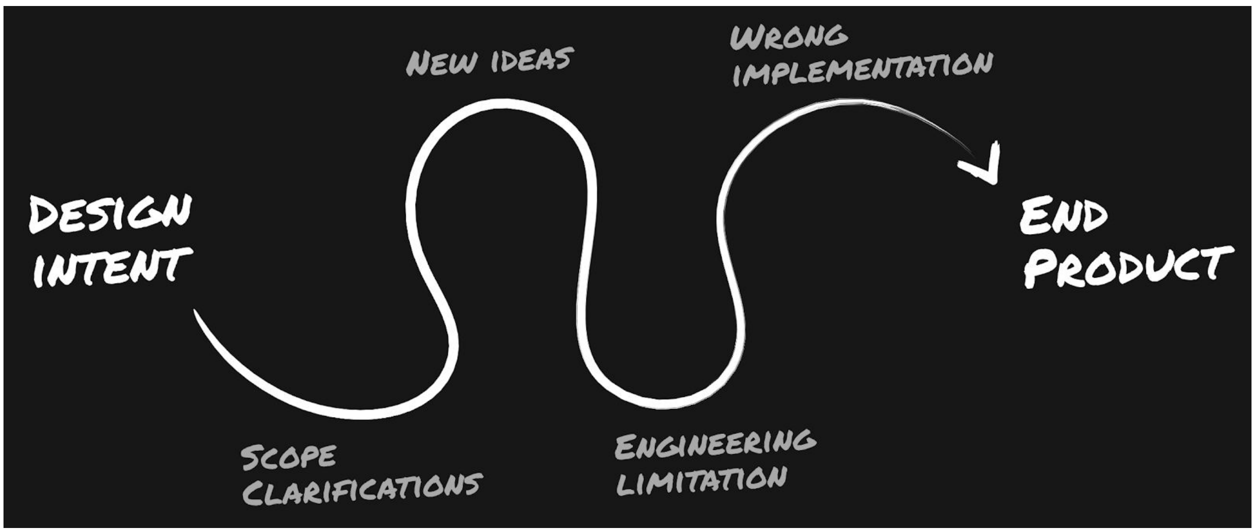


Fig 6.2. Giving general context for how prototypes are useful through a model describing the difference between design intent and the end product.

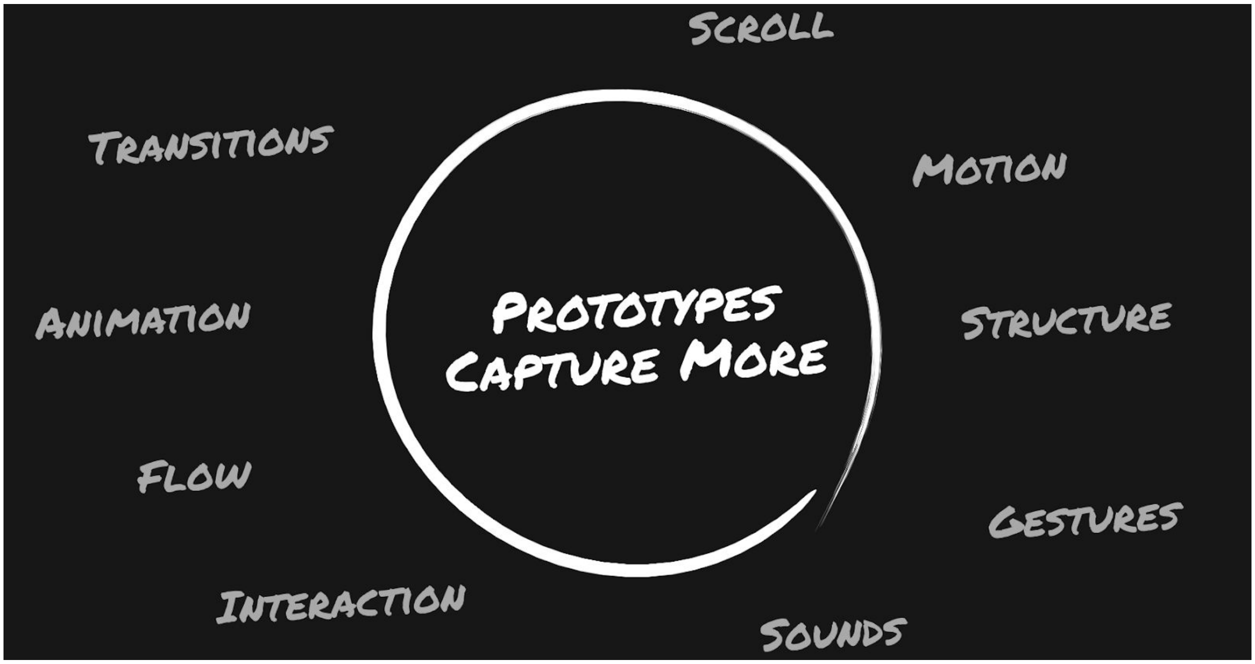


Fig 6.3. Describing what prototypes consider beyond static design artifacts.

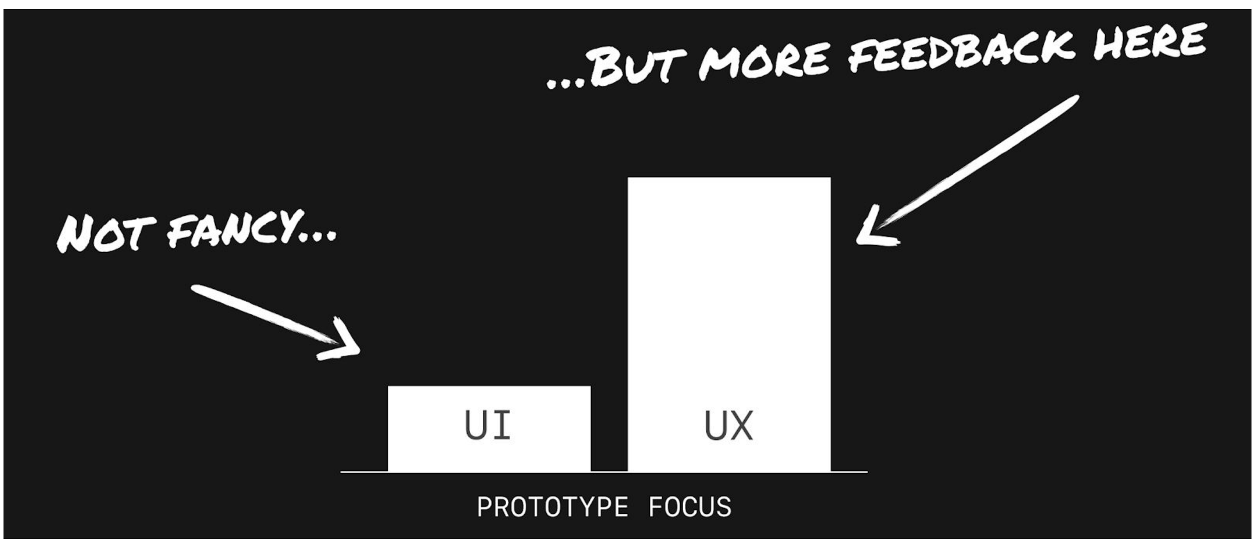


Fig 6.4. Describing prototypes through the terminology of UI (User interface) and UX (User experience)

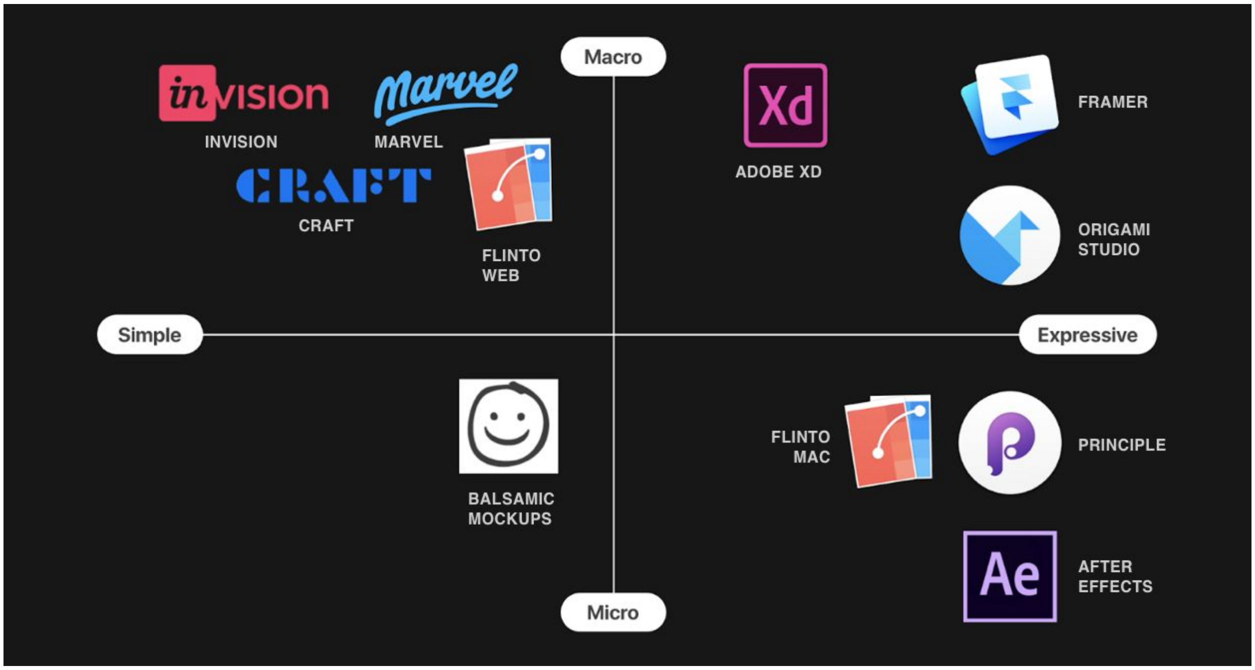


Fig 6.5. Describing my experience with different prototyping tools, according to two spectrums, Simple to Expressive design options and Macro to Micro design scope.

6.2.3 Presentation at Interaction Designers Association

Following the presentation and workshop with students, a similar but shorter presentation was held for attendants at an event for the *Interaction Designers Association* in Stockholm, where some of the above concepts were discussed with the audience.

Judging by the feedback received and discussions that followed, the concepts of discussing UX and UI qualities of a prototype, or the concept of embedded knowledge was of relevance among the designers, perhaps more so in comparison to the design students.

6.2.4 Synthesised guidelines

Based on the initial research, design practise at Mojang and discussions from the two given presentations, two guidelines were produced in this first iteration. These tried to summarise the more important aspects of my material and findings from discussions, and were arrived at through writing down the most significant insights from the research activities, and finding patterns among these. These patterns were then worded as guidelines, made to assist prototyping work.

Guideline Title	Description
Prototypes should focus on capturing “UX”	When building prototypes, give interactive characteristics of your design more emphasis, in comparison to things that can be visualised in a static way.
Consider the audience of your prototype.	Different audiences have different needs from the prototype, such as how much context is presented together with the prototype.

Table 6.1: First iteration guidelines

6.2.5 Outcomes and insights into future process

The difference in reception among the different groups of designers I met came to form the main insight from the first iteration: That the prototyping activity is practised in many different ways, and different teams and individuals have different needs from potential advice about methods or processes about prototyping.

Clearly, helpful advice can concern more fundamental aspects: Why should you prototype in the first place? How do you do it? What tool is the best one? For the junior design practitioners at Hyper Island completing their first client project, advice on which design tool to use, and the basic elements of constructing prototypes, seemed most impactful when it comes to getting more value out of the prototyping work. Alternative, the advice could carry more depth to speak to experienced professionals, such as: How do you do it in the best way? What are the underlying principles that can help you refine your craft?

6.3 Second iteration

November → January

The goal of the second iteration was to take the insights from the sessions with other designers from the first iteration, and build out a more solid foundation on which an updated set of guidelines would be based on. This second set of guidelines was planned address the problems identified in the first iteration, which required finding a slightly different character, which was more specific to the goal of this project, and less tied to the specific needs of the audiences from the first iteration. For this reason, the second iteration was planned to end with the half-time presentation of the project at Chalmers, where the results could be tested in a more neutral fashion, fully in the context of the thesis project, and less concerned with serving a specific external audience.

When it comes to collect more insight into prototyping craft and activities related to prototyping, the two main activities were continued work with prototyping within the team at Mojang, and adding to this, graphic modeling of concepts related to prototyping using sketches and drawing tools.

6.3.1 Mojang prototyping work

For the second iteration, the prototyping efforts within the design team accelerated, with most design work was being spent building prototypes to represent design ideas. Prototypes we're built to further the interaction design of controller input and touch input, but also to display the first iteration of the artistic work from the Creative team, that was planned to be displayed in the user interfaces.

One of the more significant prototypes, which was more specifically analysed for the purpose of this project, was a screen with interactions and animations for selecting an option out of different alternatives presented. Here, a close collaboration with the visual artists in the Creative team occurred, and the prototypes both influence the interaction design through implementing the visuals, and the visual assets was influenced by how they worked in the interactive context. Later in this report, the analysis of this prototype is later referenced as the *Option selection prototype*.

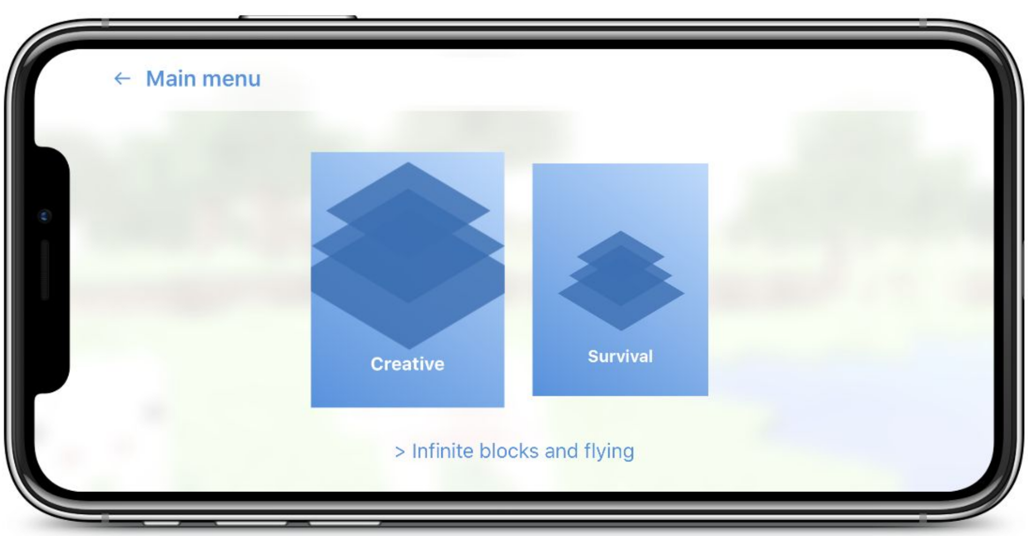


Fig 6.6: Early version of the option select prototype.

In the end of the second iteration, prototypes were documented in a spreadsheet, to keep track of the multitude of prototypes that were produced by the team. For each prototype the prototyping tool used, the target platform(s), the purpose of the prototype, and the outcome of sharing the prototype were documented.

6.3.2 Visual modeling of prototyping concepts

For my research conducted during the second iteration, I was finding that the many ideas and concepts related to prototyping were getting more difficult to reference, from the sheer multitude of project functions prototypes relate to, and the abstractness of some variables identified. For this reason, a goal of defining visual models was set, which could act as more visual, direct instances of my findings.

To begin this initiative, preliminary models that had been used in discussions and presentations previously in the project were documented and refined. More models were added, either as complements to existing ones, or as descriptions of previously un-modelled ideas from my research. Below follows some of the models that were referenced the most when analyzing prototyping work, and writing guidelines.

Fig 6.7: Dimensions of prototype characteristics.

Including – Excluding, describes how wide the intended target audience is for a specific prototype, where a including prototype is meant to be consumed by a wide audience, and excluding being for narrow audiences, who familiar with many non-explicit aspects of the prototype.

Explorative – Validative. Describes how prototypes can serve a purpose of exploring new ideas (Explorative), or validating existing ideas (Validative).

Macro – Micro. Describes how prototypes can represent many aspects of an intended design (Macro), or just a small part of it (Micro)

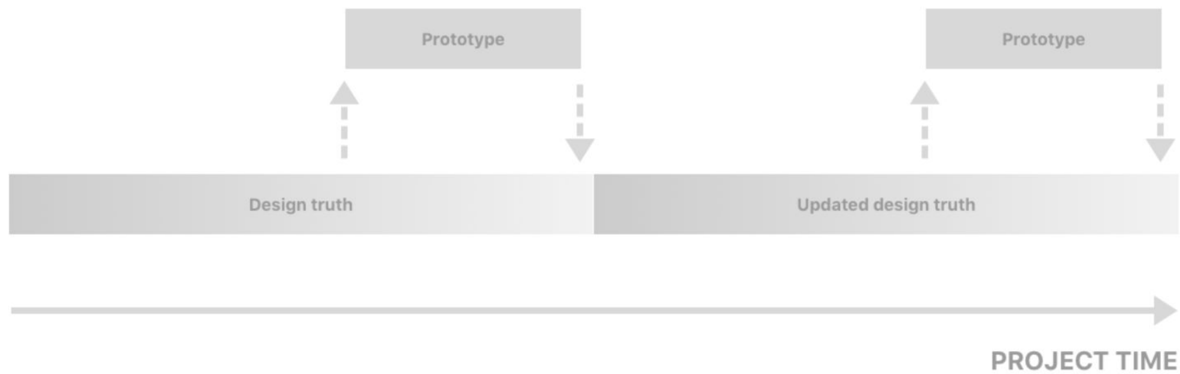


Fig 6.8: Project contribution timeline model, model 1

This model captures an idea of how value is captured in a ongoing design project, from the use of prototypes. Prototypes instantiates the current design truth, and updates it as they are tested and evaluated.

Fig 6.9: Project contribution timeline model, model 2

The model in figure 6.9 provides an alternative view to the model in figure 6.8. Here, the prototypes is the explicit representation what the design is, and improvements to the same prototype iterates on the design, and moves the project forward. For a given design project, both models be used for different phases of the project, or within the scope of each other.

These were patterns that were observed at Mojang where, for example, one prototype was iterated upon, and moved the project forward according to the model in 6.9 in one phase, but then the prototype artifact was abandoned, and instead updated the idea of the design truth within the team, according to model in figure 6.8.

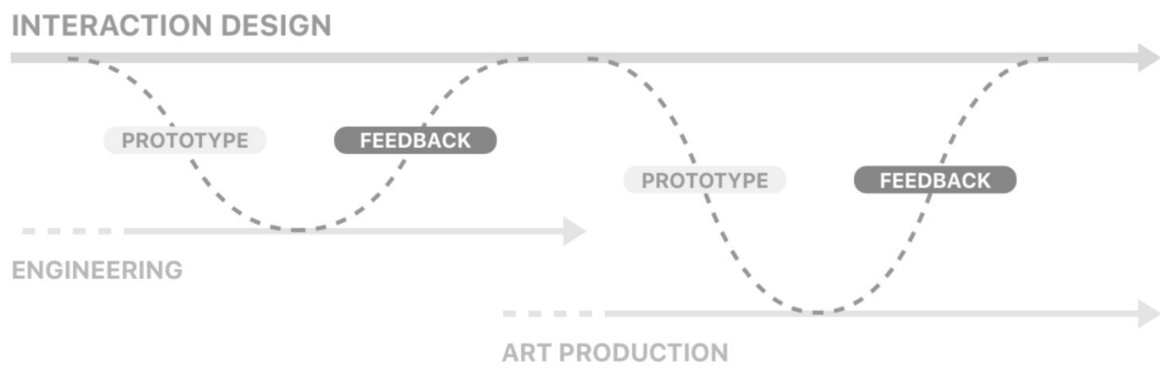


Fig 6.10: Cross-team communication model.

The model in figure 6.10 tried to capture how the prototypes built within the team at Mojang were used to communicate certain aspects of the design work, to other teams who did work related to the interaction design. For a given prototype that was shared, the other team could base their work upon that, and give us feedback on how well their work matched the interaction design. For example, for a sound designers to produce a sound effect from pressing a button, a prototype was sent over that captured the interaction of pressing a button. When the sound was added to the prototype by the sound designer, the team could evaluate if the sound and the interaction design worked well together.

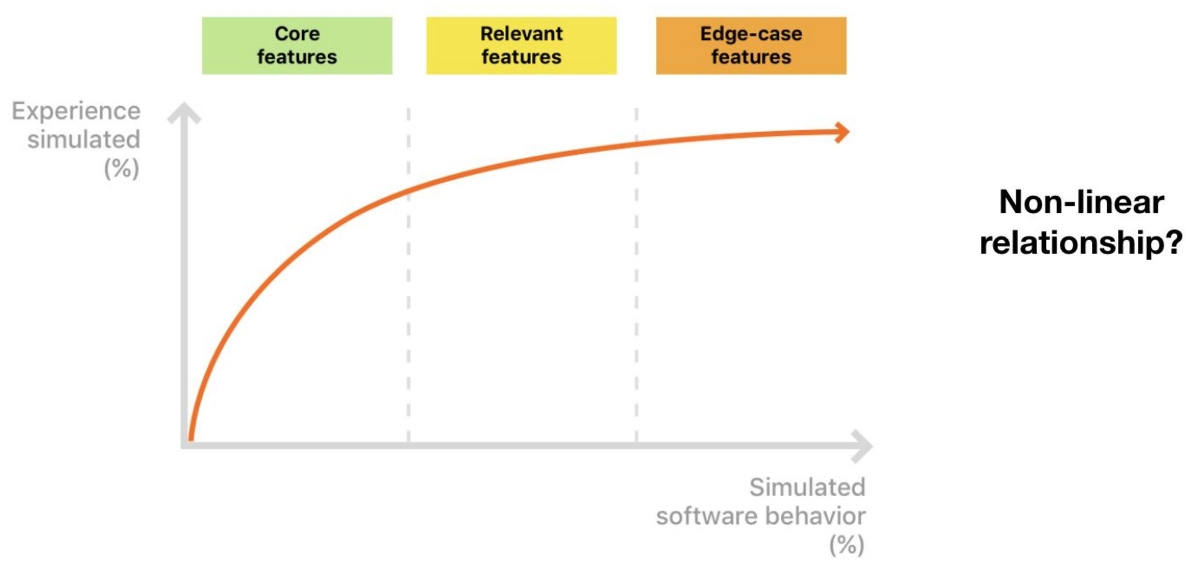


Fig 6.11: Prototyping scope impact model.

The model in figure 6.11 aims to capture how continued investment in a prototypes gives less returns as more complexity of the design is tried to be captured. This is based on observations at

Mojang, where much of the value of many prototypes was in the early work that aimed to capture a core feature, with refinements and addition only helping to a smaller extent after that, to improve the testing quality of using the prototype.

6.3.3 Writing of guidelines

When updating the guidelines for the second iteration, a clear goal was to move away from the ambiguous language of “UX” used in the initial set of guidelines. Adding to this, I wanted the guideline to capture more how to efficiently use design tools to the benefit of the designer, as this was something that emerged as crucial to the value of prototypes created at Mojang. Adding to this, grasping tools, and which tool to pick, was a major hurdle for the students at Hyper Island, from the previous iteration.

Ultimately, much of the effort in this iteration was spent on trying to visualise through the models, described in 6.3.2. However, that work was not yet influencing any of the guidelines produced here—I was not yet confident in what they captured, given my own experiences at Mojang.

Guideline Title	**Description**
Find the purpose the prototype in the context of the project.	Is it inspirational, evolutionary or validative?
The prototyping medium should support the specific usage context of your usecase.	Realistic representation should be accompanied by realistic context. Contextualization ability is often limited by the chosen tool or medium.
Consider how to communicate and share tacit qualities of prototypes	Can you record Videos? Do live demos in-person? Can you host it online to have other try it?

Table 6.2: Second iteration guidelines

6.3.4 Half-time presentation

In early February, approximately half-way through the project, a presentation was conducted at Chalmers to get feedback on my progress from teachers, students, and the to-be opponents of the final presentation of the thesis.

Before the presentation, several aspects of the project were summarised and considered, before the presentation was planned. For example, I tried to deepen my understanding of the design research aspect of the project, and doing research in the context of design practise in a design team. In the presentation, I also tried to problematize how the following iterations would have to include more concrete evaluation activities, that put my guidelines to the test.

The feedback given during the questions and answers session following the presentation was a helpful guidance for the continued planning of the project. The discussions involved questioning how well I focused my work on multi-platform aspect of prototypes, and how it might be better to start out with a large number of potential guidelines, and try to increase quality by filtering out guidelines that don't work (In contrast to what was my current approach, of trying to produce few guidelines of high quality, that carry much evidence of benefits).

The presentation also included me showing a set of prototypes built in the team at Mojang, representing some of the successful qualities that influenced guidelines.

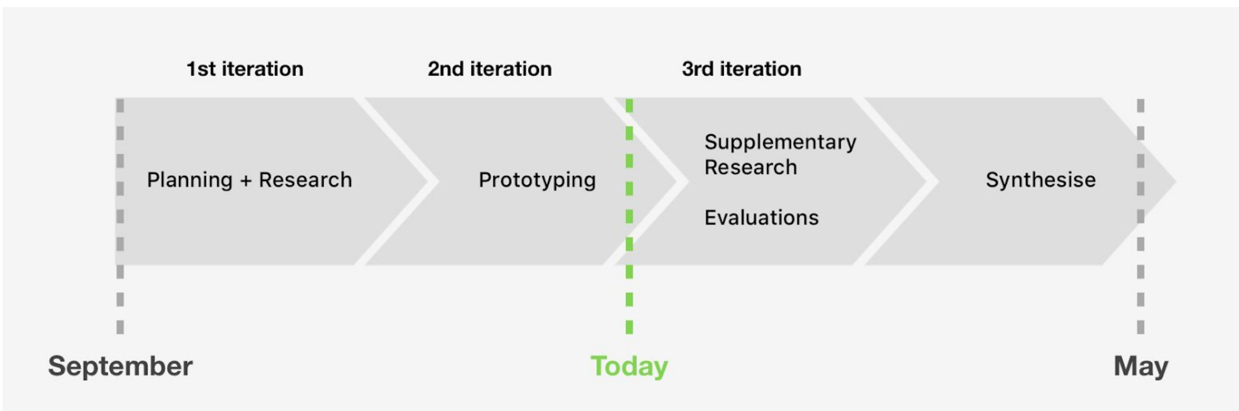


Fig 6.12. A project timeline presented during the half-time presentation.

6.3.5 Insights into future process and outcomes

Ending the second iteration, the feedback from the half-time presentation was coupled with my continued research, modeling and documentation of prototyping work within the design team. The general indication was that guidelines need to be more numerous to be able to churn through more ideas in the later iterations. Adding to this, my understanding was also that the previous goal of making neutral guidelines might have turned them into something very unspecific, that is hard to benefit from.

These two points became the starting point for the third iteration, which was also to include a more concrete form of evaluating the guidelines with designers, as problematized during the halftime presentation.

6.4 Third iteration

January → March

Based on the outcomes of the previous two iterations, the major challenge for the third iteration was to make the guidelines cover more aspects of the prototyping activity by adding more guidelines. This was initiated in this iteration mostly as a change of mindset, which included lowering the expectation for how universal a guideline should have to be. Adding to this, there was also a realization that it might be more fruitful to synthesise the end result from a large set of guidelines, than to come up with the correct small set from the beginning.

To further aid the enumeration of guidelines, I introduced scheduled brainstorming activities, where every aspect of the prototyping work at Mojang was analyzed for what had brought benefit, and what had brought more negative outcomes.

The backbone of understanding prototyping through work at Mojang continued, but with less intensity compared to previous iterations, to instead put more effort into synthesis of insights and events related to prototypes already constructed. More effort was also moved into writing the actual guidelines, designing a “card-format” to present guidelines, and planning of the activities for the workshop session.

The iteration ended with an evaluative workshop session, evaluating the guidelines with designers at Mojang, using a format of structured activities that we’re hypothesised to generate actionable insights into how the individual guidelines could be improved, together with a stronger idea of how the guidelines should be presented to be easily graspable.

6.4.1 Mojang prototyping work

While the design work together with team at Mojang progressed with more prototyping and prototypes, that was not the only thing adding to an increased understanding of the different aspects of the work. As the project progressed, more stakeholders in the organization got involved, and design learnings from previously made prototypes started to show their importance to the project, as we looked back at the progress made.

Of the prototypes build by the team, a prototype for touch interaction with a block icon was given extra attention to analyze for this project, since it involved close collaboration and iteration together with the visual artist in the creative team at Mojang. This prototype is later presented and analyzed in this report as the *Block interaction prototype*.

Adding to this, the timeline of this third iteration also included the first sessions where the prototypes were tested with actual users, by an external user research team at Microsoft, further completing the picture of the relevancy of the prototypes, together with their qualities and their shortcomings. This prototype is later in this report described (with a presented analysis) as the *Configuration flow prototype*.

6.4.2 Generating new guidelines through brainstorming

As this increased flow of information about our prototyping work was happening, the previously used tools of graphic models and prototype reference lists, continued to be of use. This library of information was used during this iteration when initiating the activity of most significance for the iteration: Brainstorming for the purpose of finding a large set of new guidelines.

By looking into the all the different aspects of the prototyping work, new guidelines could be based on where I could see a pattern of clear benefit from a specific way of working within our team, or when outspoken practises already had emerged. Ultimately, fifteen new guidelines were added this way, listed below.

Guideline Title	Description
Async communication	Prototype mean that the designer doesn't have to be in the room to explain. Use prototypes for asynchronous communication of interaction design.
Sunken-costs awareness	Prototyping when time restraint is more problematic because of sunk cost fallacy
Tool design capacity	The prototyping tool should support encoding/exploring the design qualities that needs to be analyzed.
Tool testing capacity	The prototyping tool should support simulating the specific usage context for design's use case.
Prototype sharing	Consider which methods are available to communicate and share tacit/non-verbal qualities of prototypes to stakeholders
Impact prioritization	Prioritize features that have major impact on the experience that needs simulating
Sunken-costs awareness	Invest in prototypes when you still have time to abandon the prototyping work
Side-by-side comparisons	Plan for comparing options with critical tacit qualities side-by-side
Separate investigations	Consider a separate prototype when a design quality is isolated enough to be able to be judged without other factors being implemented.

Coupled investigations	Consider making prototypes with several intersecting qualities of a design, when there is a high interdependence between the qualities.
Prioritize niche problems	Prioritize prototyping interaction design problems with non-established solutions
Balance prototype fidelity	When testing with users, consider pairing a incomplete prototype with an incomplete visual appearance, to balance expectations
Prioritize team unknowns	When prototyping and testing within a team, consider leaving out parts of the experience that can be extrapolated from the teams common knowledge.
Consider dynamic layouts	When prototyping for multiple platforms, consider your prototyping medium's ability to adapt the UI dynamically to different screen sizes and input methods
No-pressure evaluations	When sharing prototypes with tacit qualities with others, ensure that other parties can evaluate the prototypes in a "Offline" situation (e.g outside of a meeting)
Invalidate ideas	When testing a prototype, prioritize finding what would make a solution invalid, not on what is right or good about it.
Define testing goals	When building prototypes for testing purposes, make it explicit that you're aiming for qualitative data, quantitative data, or both.
Hardware cues	When sharing prototypes for a multi-platform product, use device cues, and depictions of hardware, to reinforce that a design is made specifically for a certain platform.
Multi-platform culling	When designing for multiple platforms, not all features need to be prototyped for all platforms.
Share with context	Since prototypes can be judged asynchronously, make sure that shared prototypes are accompanied with enough contextualization to be able to properly critique them.

Table 6.3: New guidelines in third iteration

After the new guidelines were generated through brainstorming and close examinations of my design practise observations, an effort was made to categorise them through tagging.

First, the core identified benefit of each activity was noted down as tags, based on the reasoning for adding the guideline to the set. The four most common of these benefits were chosen as the set of tags, to help identify how the guidelines benefitted designers.

- Testing benefits
- Multi-platform benefits
- Communication benefit

- Time management benefit

6.4.3 Developing the card format

In the preparations leading up to the first workshop that was planned to evaluate the guidelines, an effort was made to try to increase the guidelines' potential to be appreciated by other designers. This was considered an even higher priority now that there was a larger mass of guidelines, and a larger set of information to process for a potential audience.

When deciding on what format to use, no elaborate investigation was made on different available formats or mediums, from the lack of time. Instead, inspiration was drawn from IDEO method cards (IDEO 2018), which I had previous experience with using from coursework as an interaction design student. While other sets of cards exist (for example, PLEX cards (PLEX 2018) for designing playful experiences), the IDEO ones felt more similar to the type of content that would be present in a potential card set of mine, containing the guidelines. This was since the guidelines typically made recommendations on how to approach the design process, and not specific design decisions.

After sketching out a couple different layouts for the titles and descriptions, several insights were made into how the guidelines could be expanded upon to improve them in the context of cards—one example of this is the addition of a representative image for each guideline, that could help act as a visual reminder of what the card does. Additionally, an area for examples of motivations for the guideline was added. While the guidelines had been given tags in this iteration, they we're not prioritized to be added to the cards, as that work was considered to be unfinished. The final template for the guideline card is shown in figure 6.13, which was produced in Sketch (Bohemian Coding 2018), a vector drawing application.

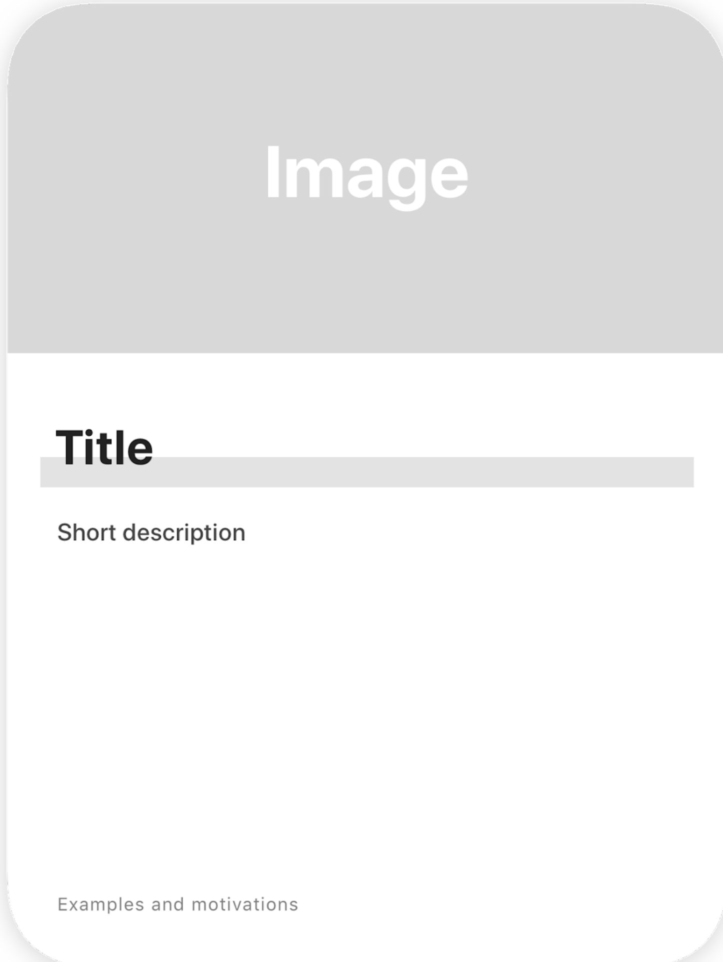


Fig 6.13: The guideline card template

6.4.4 Developing and planning workshops for evaluation

In this phase of the project, design teams at other companies were invited to participate in workshops that were to be conducted. The workshop method (see 4.3.4) was chosen for the flexibility of the format, and that it would enable working with other designers in a setting that supports having team members discussing and reflecting upon design process, and potential improvements of the design process. For this reason, the idea was to present the workshop as a learning opportunity for teams that practised prototyping. The participating team would take part of the learning material (i.e the guidelines) and I would facilitate and document how the material was appreciated.

An invitation was sent out through social media, and several designers that saw the invitation expressed that their teams would be interested. Over the course of the following weeks workshops were planned with their teams. When choosing which of these workshop opportunities to pursue, care was taken to make sure that the design teams consisted of more than three designers, to make the workshop more active, and that they considered interface prototyping to be a part of their practise. Additionally, design teams that worked with design for multiple platforms were pursued.

When developing the workshop format to be used for evaluating my guideline cards with other designers, several factors were considered when determining format and organization.

- How can I understand how the designers I talk to use prototyping in their team, to help give context for their reflections and discussions?
- How well would the workshop evaluate the guidelines?
- How can the workshop be valuable for participants?
- What types of design teams do I want to run the workshop with?

With inspiration from collections of workshop activities such as the Hyper Island toolbox (Hyper Island Toolbox 2018), a set of activities were scheduled and I planned the first workshop session with the designers in the team I worked with at Mojang.

To document the workshop discussion and activities, the plan was to collect produced written material (e.g sticky notes), take photos of the produced organization of the material, and to make sound recordings of the workshops, given that the participants would give their agreement. This material was then to be analyzed using methods for finding the distinctive themes and reflections through affinity sorting, and the KJ method (as described in chapter 4.3.2).

6.4.5 Guideline workshop 1: Mojang team

The first workshop was conducted with the design team at Mojang, as the evaluation of the guidelines that would end the third iteration. Before the workshop, all guideline cards were printed and cut to an appropriate size.

Activities during the workshop included brainstorms about why the team prototype, ranking of guideline cards according to their helpfulness, and an attempt at organizing the cards according to affinity. I also gathered feedback on which of guideline cards that the designers considered most helpful for building prototypes solving design problems related to multi-platform design work.

Fig 6.14: A photo from the affinity sorting part of the workshop.

6.4.6 Insights into future process and outcomes

During the workshop, many insights were documented into the how well the cards were understood and reasoned about by the designers at Mojang.

An acute need was an identified difficulty in appreciating the full set of guidelines in a short period of time—there was too little context given to the purpose of each card, to make set card set easily understood. For individual cards that were observed as being harder to grasp, a need for revision was also documented.

Finally, weaknesses in the workshop format were identified and documented, to help improve the workshop in the next iteration.

6.5 Fourth iteration

March → April

For the fourth iteration, the plan was to have the iteration start with the insights from the first workshop together with the interaction designers at Mojang, and iterate on the guidelines, card format, and workshop format right up until the next workshop, which would evaluate all of these changes.

Due to the limited time between the workshops, most time was spent directly implementing shortcoming and suggestions for improvement of the cards, that were discussed during the first workshop.

6.5.1 Developing the guidelines

During the fourth iteration, a few new guideline cards were added to the set, together with a set of minor improvements to the existing set of guidelines. The new cards added are are documented below, together with the explanation for why they were added.

Guideline Title	Description	Motivation for adding
Planned experimentation	Sometimes the path to the most interesting design can't be planned out. Instead, consider making room for experimental prototyping with a looser set of constraints, and be open-minded to new solutions.	Observed need for communicating the way prototypes can work for experimental purposes, and to actually recommend an activity that serves that purpose
Build Components	When building prototypes, consider building composable building blocks, that make up the final solution.	To highlight a technique that was proven useful in the design team at Mojang.
Low-hanging fruit	When building prototypes, start out with implementing design aspects are easily attainable with current prototyping tools and techniques.	To highlight a technique that was proven useful in the design team at Mojang.
Interactive wireframes	Consider keeping visual design to a minimum in an interactive prototype, or making just the wireframes interactive.	To highlight a technique that was proven useful in the design team at Mojang.
Online resources	Find and use implementation techniques or code from other designers to make prototype implementation faster.	To highlight a technique that was proven useful in the design team at Mojang.

Table 6.4: New guidelines in fourth iteration

6.5.2 Developing the card format

The most acute problem that was noted during the Mojang workshop was the designers' ability to contextualize the individual guidelines that were presented to them. While the individual cards most often contained relevant and interesting advice, it was not necessarily easy to understand what that value was when first presented with the card containing the guideline—it had to be read thoroughly, and with a new intellectual effort for each card read.

Adding to this, when the full set of cards had been presented and read, there was no organization among the cards that enabled easy reference of the different guidelines—It was merely a set of 22 discrete pieces of advice that seemingly could have been equally relevant for any situation.

To address this issue, there were several different approaches to take. While the cards clearly could have featured the tagging system that was developed previously, that design might not have been the ideal way of solving the problem, and with the added insight from the workshop, I started considering different options for contextualizing the different cards in a more effective way.

When analyzing the contents of the discussion around each card from the Mojang workshop, it was clear that most of the time the participants grasped for understanding when in their design process each strategy could have been applied. This led me to believe that this is the way the cards should have been presented at first sight—An organization that gives each card a natural chronological place of belonging, in each designers established design process.

However, when I looked at the typical stages of a design process that were reasoned about (research, ideation, prototyping, communication, evaluation), it was not obvious how the different guideline cards would be arranged in a natural way, since they all specifically related to the prototyping activity, a subset of most design processes. For this reason, I decided to arrange four separate categories of activities that all relate to prototyping, and that also typically have a chronological order.

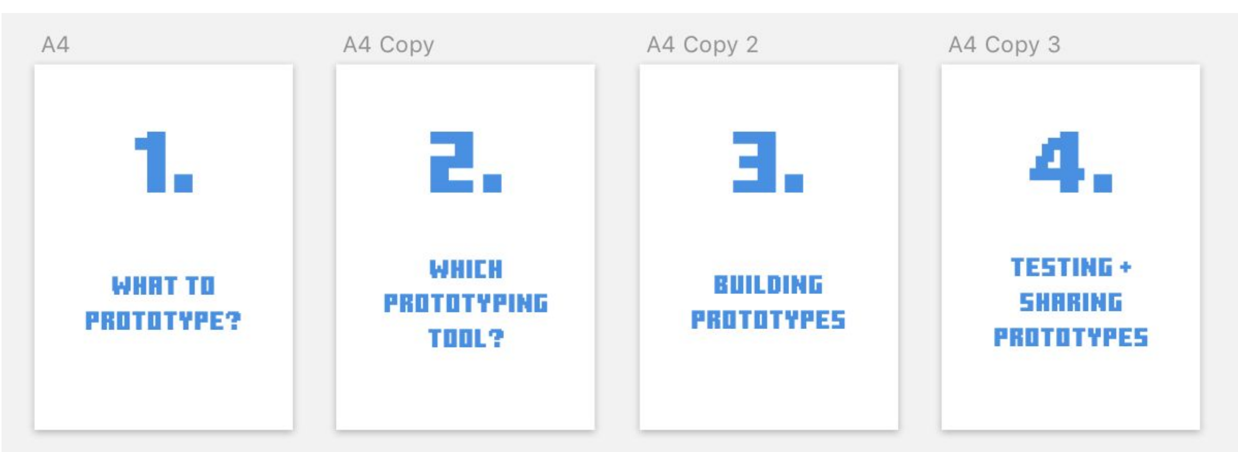


Fig 6.15: The four categories that contain the guideline cards, in the card deck

These four categories all had a place in design processes at Mojang where prototyping played a part. Adding to this, they all very much relate to the core concept of prototyping. For their ordering, that was based on the observed patterns from literature, and work at Mojang, which often followed the pattern of:

Something is to be prototyped ➜ A method of for building the prototype is chosen ➜ The prototype is built ➜ The prototype is tested

Before the craft of making the prototype starts, the scope or goal of the prototype is decided upon. After that (or in conjunction with that) a prototyping tool or medium is decided upon, to serve those goals. Then, a process of actually crafting the prototype typically takes place. Finally, the completed prototype is evaluated in some fashion, either by the designer, or through other actors which the prototype is shared with.

Together with defining these categories, the available set of guideline cards were sorted according to which activity they related the most to. The final sorting is displayed in table X

Category title	**Guidelines in category**
What to prototype?	Impact prioritization Prioritize niche problems Focus on the unknowns Prototype slicing Testing goals Validate or invalidate Sunken-costs awareness Multi-platform overlap Planned experimentation
Which prototyping tool?	Tool design ability Tool testing ability Tool sharing ability
Building prototypes	Online resources Low-hanging fruit Interactive wireframes Balance prototyping fidelity Build components Adaptive prototypes

Testing + Sharing prototypes	Match context-audience Asynchronous sharing Side-by-side comparisons Platform goals

Table 6.5: The sorting of the guideline cards according to the four categories.

6.5.3 Developing the workshop format

Based on the experiences of conducting the first workshop, several potential improvements to the format were identified. To begin with, further efforts were to be made to set the expectations for the participants, to address the identified need for better organization to support the discussion. This could be alleviated by handing out a schedule to the participants beforehand, or give a short presentation of the schedule in the beginning of the workshop. Furthermore, more clear introductions to the different activities were to be given, with more clear motivations and instructions, promoting more active participation.

Adding to this, the activities performed were to be optimised for producing even more concrete understanding of how the guideline cards were appreciated. Another goal was to adjust the activities to promote more internal reflection and discussion. One idea for helping this was introduce a silent brainstorm about examples from design work relating to different guideline cards, as silent brainstorms were proven effective to decrease the pressure of coming up with examples on the spot.

To help participants internalise the different guideline cards better, the activity of me presenting the cards was switched out to having the participants go through the material on their own for a set time period; 10 minutes.

Furthermore, the spectrum ranking activity was tweaked to focus more on the specific design situation of the company, as it was found that discussion about the guideline cards was easier in the context of the roles of the designers at the company. Two new spectrums were introduced: One ranking from:

Currently using at company — Currently not using at company
Should be considered at company — Should not be considered at company

Another spectrum was introduced to help analyze the specific benefit of the card, if they were acting on a action level or a more observational level:

Method of solving a problem — Identifier of a problem

6.5.4 Lifesum workshop

The workshop at Lifesum was conducted together with four designers from their team. The workshop was two hours long, and included me facilitating and listening to the discussions that followed from the activities related to the guideline cards.

From facilitating the workshop, it was clear that the improvements to the format helped making participating easy with fruitful reflections and discussions, both for the team and for my purpose of evaluating the cards.

To end the workshop, I also initiated a discussion about what types of cards the team was missing from the set, which was documented for the next iteration.

Fig 6.16: The workshop table after the workshop with designers at Lifesum

6.4.5 Insights into future process and outcomes

From this second workshop, I was now starting to have a wide set of evidence for how relevant the prototyping activity was for different teams, but also that design situation differ, and that a cards set can't claim to be perfect for any given recipient. While many actionable feedback points were documented from the second workshop, to some extent the data was also adding to the complexity, possibly including some conflicting points of view.

However, the updated workshop format was confirmed as being very beneficial for both the participating team and the evaluation of the guideline cards, after the improvements made.

6.5 Fifth iteration

April → May

The fifth iteration is where the final changes based on the outcomes of the Lifesum workshop was implemented into the final result. To further evaluate the guidelines cards, a third workshop was conducted with designers and a user researcher at the technology startup Doctrin.

6.5.1 Affinity sorting / KJ method

To synthesise the gathered feedback and insights from the workshop with designers at Lifesum, a KJ analysis was performed on all sticky-notes procured from the workshop. This was done as a salient and time-effective way to reference the different topics and ideas that were discussed during the workshop.

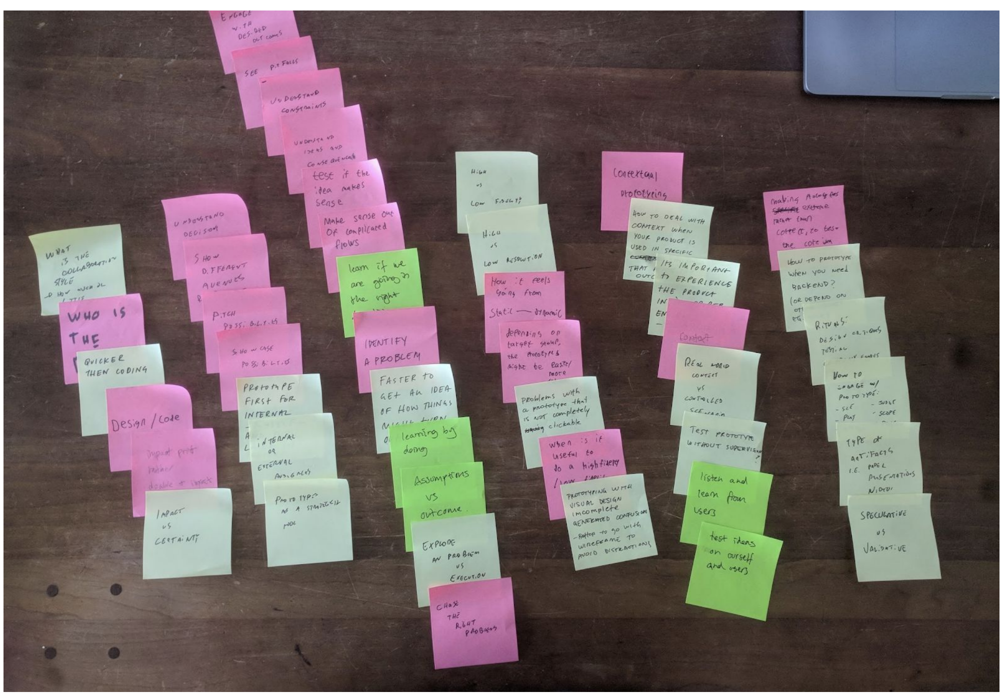


Fig 6.17: Affinity sorting of the sticky notes from the Lifesum workshop

The topics that emerged from the KJ-analysis were hypothesised to highlight topics more relevant for the prototyping work for designers at Lifesum. For each of the topics, different characteristics of the discussions could help inspire improvement of the guideline cards in different ways.

- When a certain topic was discussed a lot based reading a card, it was deemed important to designers, and therefore noted to be keep as a guideline.

- For the cards that didn't generate discussion, it was deemed either hard to understand, or less relevant for the specific design process at Lifesum. To help identify which if these were more true, the designers spectrum-sorting of cards that were relevant to Lifesum was referenced. Following this, relevant cards were either improved to help communicating the concept better, or kept as-is, while noting that it might be a less generally applicable guideline.

- When a prototyping-related topic was discussed without a corresponding card, the topic of that discussion was made into a candidates for adding new cards.

- When specific feedback was given to cards, or the format of cards, that was directly considered as improvements. Topics discussed during the "what's missing" section of the workshop were also directly considered.

- When a card was discussed with a different interpretation than intended, that was noted as a lack of clarity for the original card, while the new interpretation was noted as a potential new topic to highlight somewhere else, either as a new guideline or by modifying an existing one

The areas of interest identified from the analysis of the Lifesum workshop material were the following:

- *The need for forming a realistic testing context for the prototypes*

- *Different techniques to adjust fidelity-level and scope*

- *Prototyping as a hands-on research into a user interface domain*

- *Prototyping as communication tools and tools for forming strategy*

- *The skill-characteristics of the engineer implementing the design, and how that affects the ideal prototyping workflow*

- *Rituals and activities design teams perform*
- *Different styles of artifacts and presentations mediums*

6.5.2 Changes to guideline card format

One goal of this iteration was to consider addressing a specific feedback that was given during the Lifesum workshop: The guideline cards could feature tags which would help contextualize how the guideline could be used. For this reason, a tagging system was introduced, inspired from the initial one formulated in iteration three (see 6.3.1). However, for the card format, and with the added requirement of more cards, some changes were introduced that were hypothesised to further assist the designer using these guidelines.

Firstly, the written presentation of the tags was formalized as *boosts*. Instead of having the tag "Time management" from the previous iteration, it was here rewritten as "Speed boost". This was hypothesised to be more action oriented, making it clear that the tag communicates a positive attribute. The tags were also given unique colors, to help a user identify the abilities of the cards easily from a distance, without reading the label. Below are the set of tags introduced, together with the description of the benefit that the tag indicated would happen from implementing a guideline.

Speed boost: Indicating how implementing a guideline can lead to more efficient prototyping, with more results in less time.

Teamwork boost: Indicating how implementing a guideline can better collaboration within or across teams of an organization.

Testing boost: Indicating how implementing a guideline can higher quality evaluations of prototypes.

Creative boost: Indicating how implementing a guideline can help introduce more ideas about potential design solution, into the design process.

6.5.3 Changes and additions to guideline cards

The following tables lists the changes that were made to guideline cards in the fifth iteration.

Guideline Title	Description	Motivation for adding

Designer-engineer gap	When prototypes are tools for communication and collaboration with a technical team, make sure to understand what the team will need from your prototype, as it can affect the needed scope.	Participants in the Lifesum workshop indicated that this was a major influencing factor in their work with prototyping methods.
Reusable implementation	When possible, consider how implementation techniques in the prototype can be reused for the implementation of the end product.	Participants in the Lifesum workshop indicated that this was a major influencing factor in their work with prototyping methods.
Pick the right format	Videos, paper prototypes, shareable apps, demos, storyboards, looping gifs, voiceovers... There are many ways of sharing prototypes – Consider which one fits your purpose the best.	Lifesum workshop indicated that this was of high importance in their work, and that guideline advice can go beyond just picking a tool with a certain sharing ability, but also techniques of use.

Table 6.6: Added guideline cards

Guideline Title	**Description**	**Motivation for removing**
Prototypes with purpose	Do you experiment or validate? For each prototype, define the purpose and goal within the context of the project.	The previous workshops indicated that this guideline card was too abstract, and easy enough to take action on. Instead, cards like *Planned Experimentation* carry similar intention, but in a more actionable format.
Coupled investigations	When several qualities of a design intersect, consider investing in a complete prototype to judge them all together.	This card was removed because the card of *Separate investigations* point at the same problem. In an efforts to decrease the amount of cards, there was not strong enough reasons to keep both cards.
No-pressure evaluations	When sharing prototypes with tacit qualities, make sure that other parties can evaluate the prototypes in a “offline” situation (e.g outside of a meeting).	This card was removed because it used ambiguous language that was misinterpreted during the workshops.

Table 6.7: Removed guideline cards

The Guideline cards renamed, often with tweaked descriptions to better match the new title

Old name	**New name**	**Motivation for change**
Low-hanging fruit	Find quick wins	The initial name was not worded in an actionable way
Online resources	Build on other's work	The initial name was not worded in an actionable way
Balance prototype fidelity	Uniformity	The initial name was too complex / abstract, hard to understand by workshop participants
Async sharing	Share something testable	The wording of "Asynchronous" was misinterpreted during the workshop.
Prototype slicing	Minimize scope	The initial name was too complex / abstract, hard to understand by workshop participants
Testing goals	Set testing goals	The initial name was not worded in an actionable way
Tool design ability	Tool expressiveness	Workshops indicated that the initial name did not carry enough detail to communicate the intention.
Impact prioritization	Feature prioritization	The initial name was too complex / abstract, hard to understand by workshop participants.
Match context-audience	Prepare the audience	The initial name was too complex / abstract, hard to understand by workshop participants

Table 6.8: Renamed guideline cards

6.5.4 Doctrin workshop and final reflections

Fig 6.18: Sorting the cards on a spectrum with the Doctrin designers

To end the fifth iteration, a third workshop was conducted with designers at Doctrin, a technology startup in Stockholm making apps and software for healthcare providers. The workshop lasted two hours, using a similar set of activities as with the prior workshop with Lifesum. At Doctrin, two designers and one user researcher participated.

While this evaluation context was late in the process, some clear learnings were made following the activities and discussion, which influenced some final adjustments to the guideline cards. It was determined that time constraints made it unsuitable for considering these changes a sixth iteration.

Importantly, this evaluation opportunity came to represent the only time where the added tag-categorisations had an opportunity to be tested with other design practitioners, as they were the most significant change from the previous iteration. To get more insights into how the tags were appreciated, one new activity was introduced to the schedule, where participants were asked to try to explain, from their own experience with prototyping, why a certain card had been assigned a certain tag with benefits.

The results from this were that the tags generally made sense to the designers, and trying to connect the tag to the function of the card was a helpful way of internalizing why one should implement the practise suggested on a given card. However, this benefit was not something that made the cards easier to appreciate at first glance, rather it became a challenge to understand why the tags applied to the guidelines.

For this reason, if time, the tags were to be changed to rather describe a *method of application*, instead of describing the hypothesised benefit. With the practitioner getting hint of how to take action on the card, it can get easier to know how to apply certain cards for a given design situation.

6.5.5 How the planning and idea of end result changed

A clear set of shortcomings in the guidelines remain after the fifth iteration. As this was the final iteration that there was time for in the project, this leaves the set of guideline cards from the fifth iteration as the end product of the project, despite these unanswered questions. Since the fifth iteration contained a third workshop, but no iteration followed, naturally the projects would end with many identified opportunities for improvement. For example, the the added tags were evaluated by the last workshop at Doctrin, but not enough time were left to take action on the feedback. While no more data was gathered about the guidelines, several smaller changes were made to the guidelines as they were collected in the report, as they were reflected upon in the context of the whole project.

With two workshops completed with both teams not showing strong interest in how to multi-platform design situation affected prototyping practise, that aspect of this project had received little attention with regards to what changes are made to the guideline cards after each workshop. While the multi-platform aspect of designing was one of the most interesting challenges for the design team at Mojang, it was not one of the problems that came to define how we worked with prototypes fundamentally, and since the topic didn't get much more attention with the external workshops, less evidence have been gathered for that specific topic than was planned initially. This aspect of the project is discussed more chapter 8 and 9 of this report.

7. Final results

This chapter describes the final outcomes of the thesis project. First, analyzed situations of prototyping practice is presented as results from the design research work at Mojang. Following that, a description of the needs of interaction designers who do prototyping is proposed through analyzing and comparing research work, embedded practise at Mojang, and concept models. Then, a proposed set of guidelines are presented, together with the format of the cards, and suggestions of how the cards can be used.

7.1 Analysed prototyping practise from Mojang

This section describes seven instances of documented design practise at Mojang, where prototypes played a significant part in the design process. In sections following in chapter 7, this set of examples will be used similar to how Gaver (2012) describes annotation of design portfolios, but with the difference of this being not instances of design work, but instances of designed practise, based on the constraints of the design situation.

While the prototypes listed here does not include all prototypes that the team produced in response to the design situations (more than 30 distinct interactive prototypes were built by the team, several of which were iterated upon many times), these are the cases that were intentionally documented for the purpose of analysis. This included more close observation and taking notes about the design situation that required the prototype to be built, partaking in the process of building the prototype, managing the evaluation and response (e.g from users, teams and stakeholders), and taking notes on how the prototype affected the project over a longer period of time. The documented notes about the processes included both explicit feedback that was given from other teams on how a certain prototype was presented or used, but also indirect observations on how prototyping activities affected the project in different ways, e.g through observing the design discourse that followed in communication channels, in meetings or in person.

The selection of prototyping situations to analyze was made through trying to achieve a wide variety of analyzed practise, and to enable viewing prototyping practises from as many points of views as possible. Additionally, most of the selected situations to analyze was initiated early in the project, to enable following the entire process of building and using the prototype.

7.1.1 Navigation prototype 1

In the initial phases of the project, brainstorming (see chapter 4.3.1) within the design team generated potential ways to structure the global navigation of an application. However, these navigation methods were identified as being heavily dependant on interaction to demonstrate their potential benefits and drawbacks. For example, one design (see figure 7.1) considered revealing more actions and content through scrolling the page in both a horizontal and vertical direction. Another unknown aspect of this design, is how the left margin of the page would relate to the scrolling function—either the scroll could feature the first item in the horizontal row centered, or the page could be aligned to the left edge of the screen.

To get a better understanding of these design properties, an interactive prototype was constructed to evaluate the designs potential for working well as a global navigation pattern. This prototype was built despite very few other aspects of the design was finished. For example, both actions, buttons, content, and visual design was not considered at this stage.

To build this prototype, wireframe layouts were first designed in Sketch, as placeholders for the significant buttons and elements of the design. These were then transferred to Origami Studio as image assets, where the interactive qualities were added to make the wireframes act as a prototype. While the intention of the design was to be viable using all three major input methods that the team designed for (touch, mouse/keyboard and gamepad) , the interaction method encoded in this prototype was touch interactions, to limit the scope of the prototype. Some difficulties were had with adding scrolling interactions in both directions simultaneously, which added to the complexity of building the prototype. In the finished prototype, the user could scroll the main row horizontally, and reveal more rows by scrolling vertically, using touch interactions. Either the prototype was displayed on a computer screen, with the frame of a phone surrounding it, as seen in the figure, or the prototype could be downloaded and used normally on an iPhone.

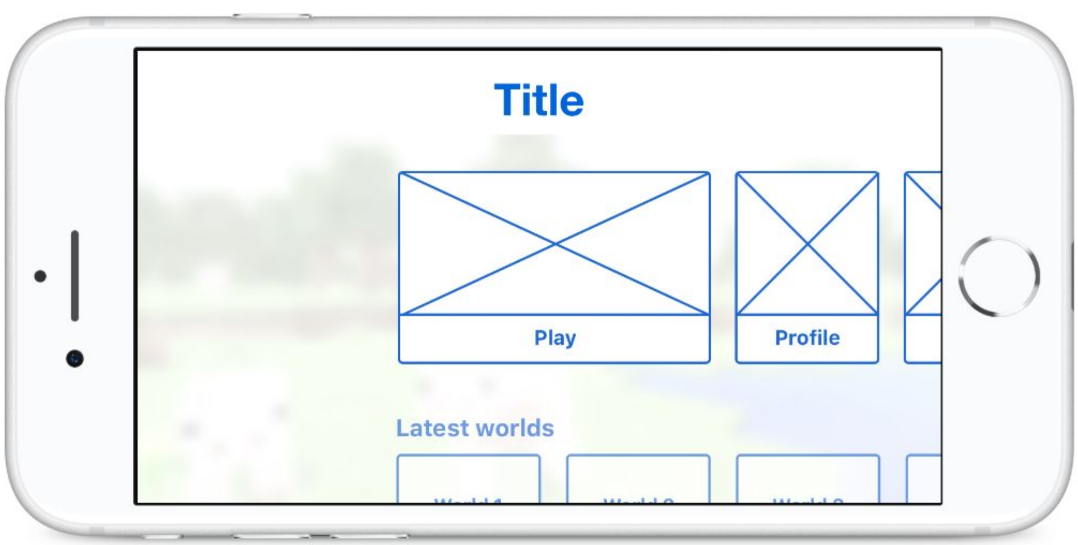


Fig 7.1: Navigation prototype 1

The prototype worked well as a quick way to to demonstrate the intention of the scrolling function, and to communicate the interactive vision of our team to other stakeholders. Adding to this, the prototype also helped communicate how the scrolling rows of content could be of any size—the interactive scrolling functionality demonstrated that with this design, we were not limited by showing all option on the screen at the same time, as scrolling could be used to reveal more content in both direction.

Furthermore, a recorded video of the prototype was used in a presentation deck that elaborated on the scope of the design project. The design team also gathered insights into how well the centering of the content worked with scrolling. While building the prototype was a very short endeavor (about 2 hours of work) it bringed more life to the wireframes that was available, and helped the team communicate more clearly for many weeks, before more rich artifacts were constructed to represent the interaction together with more elaborate functionality and visuals.

7.1.2 Navigation prototype 2

This section describes a prototype that was made to visualize a potential other candidate for organizing navigation of an application. As the prototype from chapter 7.1.2 was shown to other designers in the organization, a verbal explanation for a different design was given, that rather used tabs to organize the global navigation of the app.

To fully understand this new suggestion, and to honor the suggestion, a similarly low-fidelity prototype was constructed to represent that other idea (see fig X). Adding to this, this new prototype explored how the a technique of scrolling with inertia could “snap” the boxes in each row, to the center of the screen. This was another interaction and animation technique that was hypothesised to be of essence to any of these new design, and to early get an understanding of their usefulness was of interest.

The prototype was built using similar techniques from the other prototype, and involved approximately one hour of work for the designer building the prototype. After the prototype was constructed, it was shown to the designer who suggested it, to get confirmation that is was representing what was being described earlier.

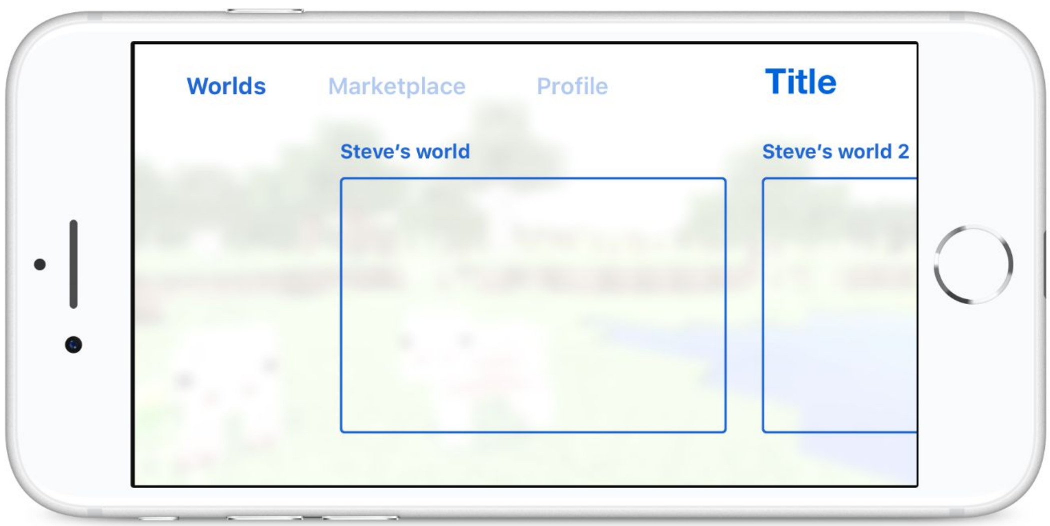


Fig 7.2: Navigation prototype 2

The result from showing this alternative prototype in a similar style to the earlier one, was that the qualities of the different design could be carefully examined side-by side, to get an understanding of what would serve the purpose of the design the best. When doing this, a clear argument against this design was uncovered, which is that the horizontal row of tabs would consume precious vertical space in the landscape view of the application, which made fitting content to the tab a challenge. Adding to this, having a scrolling view within each tab was found to be a challenge, as the scrolling content would have to overlap with the tabs as a user scrolled horizontally, for certain sets of visual design of the tab row. The prototype also raised questions of how many tabs would fit in the tab bar, and if it would be enough to serve the purpose of the global navigation.

Several benefits of the design were also made clear, such as the ability for the user to easily understand the navigational context of the application, as the tab bar always was available.

What was clear from building and sharing this prototype, was that building a prototype was a beneficial way of visualizing an idea from someone else, as that indicated that the idea was seriously considered by the team, and not just argued against on purely theoretical understandings of the other person's idea.

7.1.3 Option selection prototype

During the second iteration of this thesis project, one design situation involved a set of options that were to be displayed in a horizontal row in the user interface, with the row acting like a radio button where only one option can be selected. For this part of the interface, the team collaborated closely with the team of artists at Mojang, who produce visual assets and sounds. Here, a challenge of the design situation was to balance the expression of the visual artifacts with the interaction design, and to match certain aspects of the sounds and visuals to more interactive qualities, such as animation between different states. Additionally, there was concerns if the elements would fit horizontally on the screen, that potentially could be alleviated by scrolling or scaling of the elements on the screen.

For these reasons, an interactive prototype was constructed after the team had worked on static artifacts of the user interface. In comparison to previous examples from chapter 7.1.1-7.1.2 (*Navigation Prototypes 1* and *Navigation Prototypes 2)*, this prototype was in higher fidelity, as shown in figure X. Here, Origami Studio was still the tool used to capture the interactions and different states of the prototype, with different options selected.

One important aspect of the design that was explored when constructing the prototype, was how the elements could dynamically scale up or down depending on which was selected, to give more visual space to the selected element. When investigating to which extent this scaling could be done with the prototype, the artist team got important feedback on how much space their graphics would have, when being displayed to the user.

Fig 7.3: Option selection prototype, rightmost option selected

Fig 7.4: Option selection prototype, second option from the right selected

However, when implementing this dynamic behaviour in the prototype, an important simplification was made, in comparing the prototype to how the potential final application would work. Instead of enabling all options to be selected in the prototype, only the two rightmost options were enabled for selection. This subset of potential states was significantly easier to encode into the prototyping software, as the user interface would only have to transition between two states. If all five options would be selectable, the transition logic would have support 20 different transition (from each five options, to four others), or have a elaborate algorithm that calculates the layout needs for the transition. However, such behaviour was found not to be required to have the prototype fulfil its purpose of testing transitions, and learn about the requirements that were to be given to the artists.

After the initial interactive prototype were completed, several improvements were made to animations and interactions to better match the artistic vision with the interface. This was done in close collaboration with the artist team, through sharing video clips, or doing in-person demos of how the prototype worked. For example, a video clip was recorded of the selection interaction, that transition from one selected element to another. This video clip was used by the sound artist at Mojang, who added audio to the video to show how the interaction and the audio response to the selection worked together.

Adding to this, this prototype continued to serve as a starting point for exploring several more features over the following weeks. This included having the visual assets respond to accelerometer data from the device—Several different designs that were related to how the graphics responded to movement of the device was explored through the prototype, using the supporting functions in Origami Studio for reading data from the sensors of the device the prototype is being tested on. Another area that was explored, was to explore how the same design would behave with gamepad input. Here, logic for handling input from both keyboards and gamepads was added, to let the designer test with whatever input options was available. However, it was also found that the Origami Studio did not have complete support for working with all buttons on gamepads, which hindered certain exploration.

7.1.4 Origami device frame

Around the timeframe of the third iteration of the thesis project, many designs for TV screens and gamepad input was explored, and a custom addition was made to how videos of prototypes could be displayed using Origami Studio.

In the project, most interfaces were designed on a canvas of 667 x 375 pixels, similar to an iPhone screen. However, when configuring Origami Studio to display the prototype in a TV context, 1920 x 1080 pixels were expected from the software, to have the design fill the screen. For this reason, assets had to be adapted to work with both formats, even if the same interface was to be displayed on both TVs and phones.

At the same time, there had been instances some of misinterpreting the prototypes that were shared internally, in regards to which platforms that the design represented. For this reason, it was understood as being important to visually clarify in what context the prototype was designed for, when the prototype was shared with the team.

To help solve both of these problems, a custom device specification was added to Origami Studio by our design team, where the same assets could be used as with the 667 x 375 canvas, but displayed in the context of a TV. To further help communicate how the design was intended for gamepad interaction, a Minecraft-branded gamepad was added to the corner of the prototype display, which then accompanied every video recording of the prototype that was shared within the organization. This helped clarify that the prototype shown was meant for gamepad interaction, and switching between the touch display and TV display in the prototyping software could be done seamlessly without resizing the interface assets.

Fig 7.5: The device frame component that was added to Origami Studio

7.1.5 Shooting hearts prototype

In this design situation, occurring in the the timeframe of the second iteration, a prototype was built purely for the purpose exploring new types of interaction methods that could potentially be used for Minecraft.

The interfaces and interaction in this prototype shows a scenario where the user is watching a video stream of another person playing Minecraft. While the video stream is playing, the used can tap the heart button in the lower left hand part of the screen, to send hearts to the player that is streaming their gameplay. Adding to this, the prototype explores how to use a 3D-touch interaction in iOS (Apple 2018), to generate a repeated flow of hearts sent, with increased rate when the user increases the pressure of the finger on the screen.

While the application in this example was fictional (i.e not a part of a planned project), matching new interactive qualities to Minecraft contexts were a good learning opportunity for the team, expanding our understanding of what input methods we could use for designing Minecraft going forward. Adding to this, our ability to work with real live-streamed video better understood.

Fig 7.6: Shooting hearts prototype. Heart button in bottom left corner, with video stream in background

7.1.6 Block interaction prototype

This section describes another design situation occurring during the third iteration of the thesis project, where the design team working with prototypes was of essence when collaboration with the team of artists at Mojang, similar to what was described in section 7.1.3, *Option selection prototype*.

For this design, the visual states of a button were explored, e.g what happens to the button visually, in the different states of the button:

- *Passive state:* No user input, the button is just visible on the screen. (See figure 7.7)
- *Pressed state*: The user is currently touching the button, but without releasing the press. (See figure 7.8).
- *Tapped*: The button pressed and released.

The goal of this interaction was eventually to enable the art team to express different visual styles of the icon asset on the button, depending on the state of the button. However, initially the main need was for the design team to just give the user visual feedback when the button was pressed, giving confirmation to the user that the different touch-actions have been successfully performed. This shift in focus was inspired by how the expression of the prototyping tools showed potential for expressing different styles of the icon, depending on how it was pressed.

When building the prototype, a initial type of visual feedback was added by just modifying the icon that is seen in figure 7.7, through programmatic manipulation (e.g the image asset stayed the same, but it's width and height was manipulated through a scaling animation). When sharing this initial prototype with the team of artists, it was seen as a major opportunity to add dedicated image assets for the different states, to increase the visual effect even more when the button is interacted with. This became the foundation for a dedicated effort from the artists, with a specification for the new assets that was highly coupled to the interactive properties of the user interface.

Fig 7.7: A button with an Minecraft block icon.

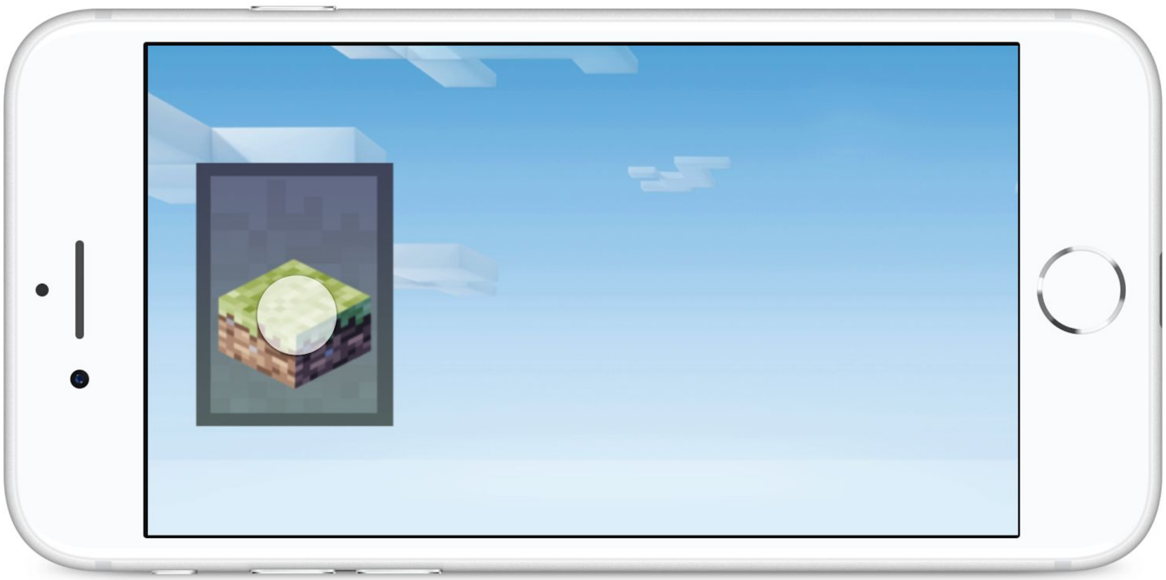


Fig 7.8: The button in it's pressed state. The white circle is how a simulated touch input is represented in Origami Studio.

7.1.7 Configuration flow prototype for user testing

As described in the section 6.4.1 of the process chapter, one design situation considered a prototype that was built for the purpose of user testing a navigation flow with users. Here, several smaller prototypes had been built (including the navigation prototypes and option selection prototypes described in 7.1.1, 7.1.2 and 7.1.3), to elaborate on parts of the navigation that would help a user configuring the application. However, the team also wanted to test this complete set of navigation and configuration actions with users, to confirm that the design was usable and easy to understand.

Since testing this flow included many different screens, and overall a large set of features to implement in the prototype, the team decided to use a more simple prototyping tool, Marvel, which only supported transitioning between screens, based on taps or clicks from a user, in comparison to Origami Studio's full set of interactions. Still, this level of interactivity would both enable the user research team to perform their intended qualitative usability testing (as described in 4.3.3), and to some extend, controlled experiments (as described in 4.3.3).

What was found however, was that as the design was implemented in Marvel, many different aspects of the experience were not fully elaborated on previously, which increased the scope of building the prototype. In a sense, the ambition of building a complete, user-navigable prototype, worked as a reality-check for the design team, to focus on the complete experience and not only on key individual parts. Simply assembling the complete prototype acted as a form of convergence method, as as described by the Jones' model in chapter 4.3.

Despite the lower level of fidelity in comparison to other prototyping tools that had been used, the prototype was able to be user tested with little loss in quality of the test, from what the team could observe. While many aspects of the design were simplified, the purpose of this evaluation was clearly linked to the more overarching aspects of the design, such as if information was ordered in a sensical way, and if the overall navigation experience felt cumbersome or not. For this clear purpose, the prototype served as a efficient artifact for getting the insights that the team needed.

7.2 Needs of interaction design prototyping

This section proposes a set of needs relevant to consider to support prototyping work for interaction designers. The set of needs is synthesised from reviewing literature and previous research work on prototyping in the field of interaction design, while also considering the needs of the design team at Mojang, students studying prototyping techniques at Hyper Island, and discussions with interaction design professionals during my workshops and presentations. Additionally, several of the visual models developed over the course of the project is used to visualize characteristics of these needs in the following chapters.

7.2.1 Considered prototype purpose

A general pattern among research include fitting the prototyping activity to a certain intentional purpose, based on the design situation. When Hartmann (2008) describes design practise of IDEO designers, prototypes at different stages of the project are categorized according to purpose: *Experimentation*, *Evolution* or *Validation*. Lim *et al.* (2008) considers the question of purpose through the lens of *Filters*, where the filter is the design variable to be analysed by constructing the prototype. Houde and Hill (1998) enumerates more specific purposes in the proposed framework, with prototypes investigating a combination of the properties of *Role*, *Implementation* or *Look and feel*. This can also seem similar to frameworks such as Design sprints (see 4.2.5), which intentionally give the prototyping activity a purpose by preceding it with a problem definition, and following it with scheduled user evaluation.

It should be noted that identifying what purpose a prototype *did fulfill*, is different from actually identifying what purpose a prototype should fill, *before* it's constructed. For many prototypes constructed in the design team at Mojang over the course of this project, the purpose of a prototype was more vague before it was constructed, and afterwards, when the prototype was either constructed, tested or simply used, the prototype helped serve more concrete purposes than was intentionally intended. *Block pressed interaction* (chapter 7.1.6) and *Option selection interaction* is two examples of this.

Often an original purpose was fulfilled, but other more unexpected purposes were also fulfilled when the prototype was tested or built. For example, several significant prototypes were built for the purpose of user testing the design. However, as the prototypes were constructed, the purpose of needing to articulate the design in more detail while building the prototype served as its own purpose to help improve the design, even before any user testing was done. This is exemplified from the prototyping situation described in 7.1.7; *Configuration flow prototype for user testing*. A more clearly defined purpose for prototypes would perhaps focus the work, and help reach intended results faster. However, it would also risk losing out on unintentional purposes that cropped up.

With this in mind, prototyping for less strict purposes can be seen as a need on it's own, which perhaps is reflected in the *"Experimentation"* terminology used by Hartmann (2008), and how Lim et al (2008) describes prototyping as an activity for exploring a design space through craft.

Fig 7.9: Prototypes as purposeful instances in the design process.

When looking at a project process models that were generated earlier in the project based on how the Mojang design team worked, (see figure 7.9 and 7.10) a comparison could be made to the concept of prototype purpose, which brings some new insight about how prototyping purpose related to these different styles of working with prototypes. In figure 7.9, the modelled situation is able to simultaneously represent multiple instances of prototypes, as the prototype artifact is disconnected from the progression of the project. Additionally, the prototypes, after being used, are discarded. With these characteristics, this modelled process show similarities to how prototyping for generative and experimental purposes is described in this chapter.

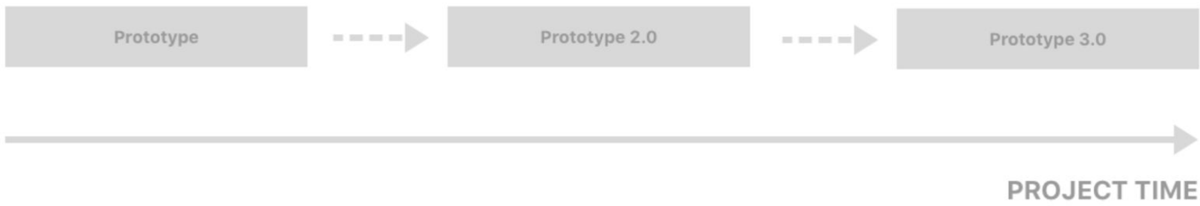


Fig 7.10: Prototypes as refinement.

On the other hand, the second process model discussed, here shown in figure 7.10, represented something more similar to prototypes being used for refinement, where the prototyping artifacts is continuously improved upon. These characteristics carry more similarities to the evolutionary and validative purposes described in this chapter.

To exemplify, when looking at the prototyping situation described in *Option selection prototype* (see 7.1.3), the overall design situation is described the best by figure 7.9, but as the prototype was built, smaller iterations and follow-up work was done on the same prototype, as described by the model in 7.10.

Further examples of reasoning about prototyping purpose was captured in the quality spectrum, see fig 7.11, developed during the second iteration of this project. Here, the scale between *Explorative* and *Validative* was an early attempt at capture how prototypes served different purposes for the team. When analysing the set of examples of prototyping situation from chapter 7.1, two prototypes serve as clear examples on the range of this spectrum—the shooting hearts prototype was more explorative, and the configuration flow prototype was more validating.

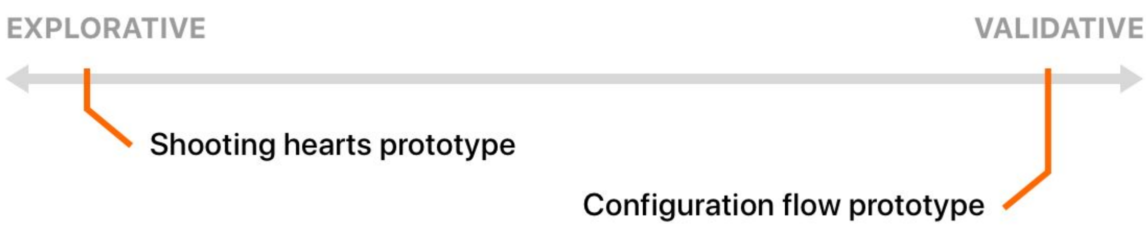


Fig 7.11: Explorative – Validative quality model

7.2.2 Speed of construction and use

Furthermore, many authors and proposed methodology frameworks concerns the ability of prototypes to be built quickly, or to quickly generate results from testing. Methods such as *Rapid*

Prototyping, as described by IDEO, (see 3.2.3), focuses the whole activity on speed. When Lim *et al.* (2008) discusses the prototyping activity, the *Economic Principle of Prototyping* is central, noting that:

The best prototype is one that, in the simplest and the most efficient way, makes the possibilities and limitations of a design idea visible and measurable.

To capture my findings from design practise at Mojang, a visual model was constructed, presenting how different features considered for a prototype had potential to capture different fractions of the modeled experience. This mirrors the economic principle: How can the prototype serve its purpose through minimal amount of work for the designer?

GV's (2018) Design sprint methodology explain the usefulness of simple prototypes in practical terminology, noting that a "*A realistic façade is all you need to test with customers*". Furthermore, the Design Sprint framework limits the prototyping activity to one just one day of work.

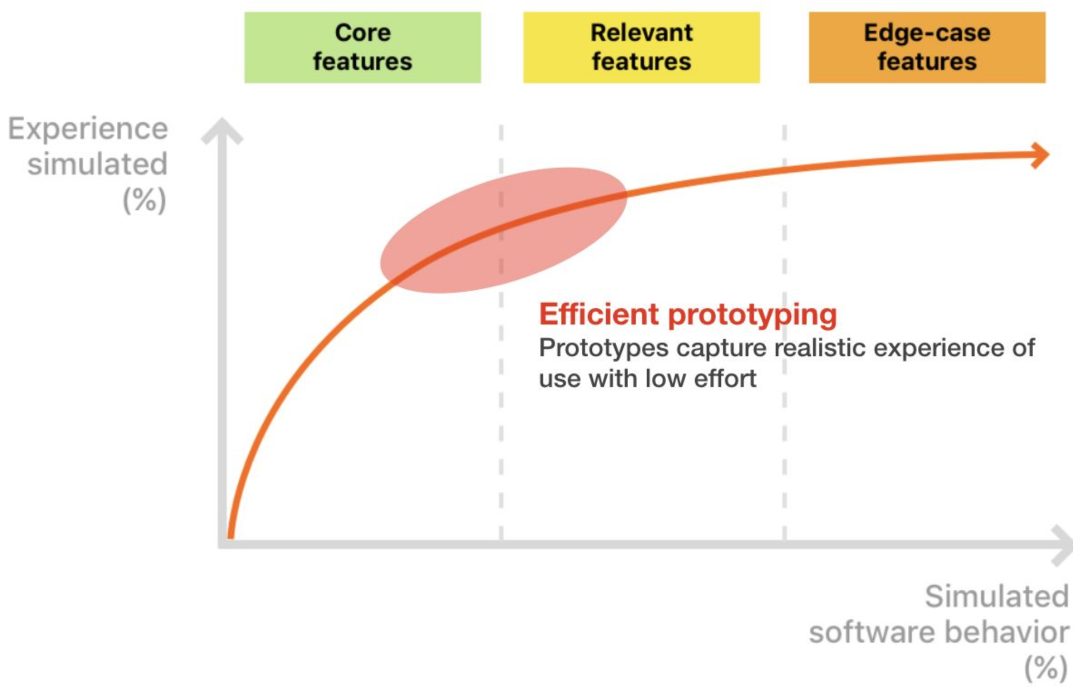


Fig 7.12: How prototyping core features matters the most to the experience

This relationship was clear from the design situations at Mojang. From the exemplified prototypes in chapter 7.1, most were conducted in design situations when speed was of essence to enable the team to either iterate more times, or move on to the next design task. Furthermore, the examples also point to how the more simple prototypes with little time investment, was of disproportionate help to the team, compared to the efforts put into them. Both the *Navigation prototype 1* and *Block interaction prototype* was referenced many times for their captured value as the project continued, despite being of really small scope and fidelity.

7.2.3 Considered testing or sharing

A need of understanding nuances of sharing a prototypes was a third theme observed during the project. Testing design on users is core to the Human-Centered Design framework (Maguire 2001), with prototypes potentially serving as representations for that design. However, with *Experience Prototyping*, Buchenau and Suri (2007) elaborates on different sets of testing purposes that a prototype can serve, noting that a prototype can help evaluate design ideas through testing, but also help communicate ideas, and act as a form of research into the design area.

Being embedded in a newly formed team at Mojang, with an ongoing project over the full year, many different nuances of how prototypes were used for communication were observed, as the project moved through different phases. Initially, prototypes we're used to present the intent of a potential project, helping defined the scope of what we wanted to do. In the later phases, prototypes specified specific design decisions meant to be implemented. Prototypes we're used as concrete deliverables to a user research team, to present team progress at company meetings, and to help contextualize art from game artists, helping display their work in the context of the user interface.

For each of these use cases, the prototypes were presented with different techniques to help serve the purpose of the prototype. While having this purpose defined was a need on it's own (see 7.2.1), the techniques and style of sharing these artifacts were an area of opportunity for better understanding. The model in figure 7.12 was used in the project to explain how the team used prototypes to help specify and contextualize the work of other functions working on the same project.

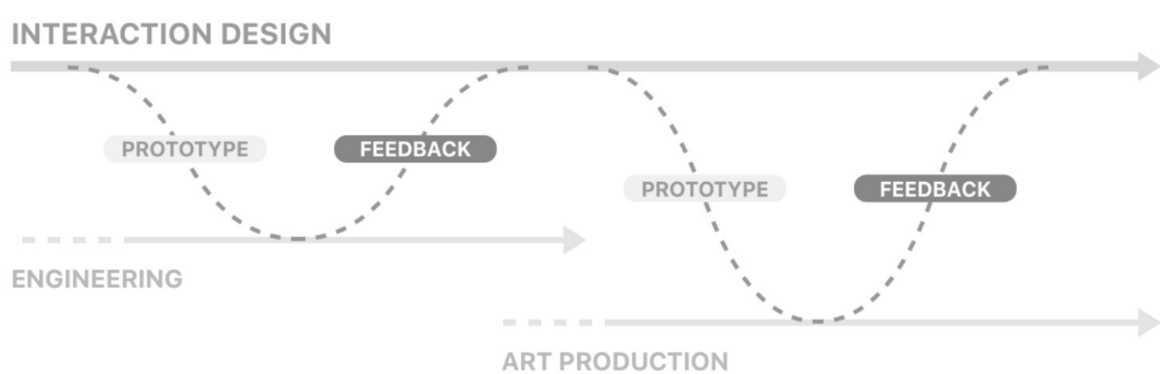


Fig 7.13: Sharing prototypes to help involve other teams

From the examples in chapter 7.1, most prototyping efforts were shared with considered techniques, and judging from the reception, this was of high importance. In the later phases the efforts put into the *Origami device frame* from chapter 7.1.3 exemplifies this, as special efforts to optimize the sharing quality became relevant to the team.

7.2.4 Considered style of representation

The fourth common theme found in literature as well as during practise, is the question of what medium is used to represent prototypes, and how scope and fidelity level influence the quality of the representation.

Paper prototypes are a recurring theme. Snyder (2001) describes the activity in detail, noting how the paper style can promote accessible collaboration and creativity. Houde and Hill (1998) however, problematizes hand-drawn graphics in a provided observation, when members of an organization *became enamored* with the hand-drawn style of the prototype, which was only meant to represent how the prototype was a rough draft. As discussed by Shrage (1996), organizations can develop their own “prototyping cultures”, where only a certain style of artifact is accepted. Examples described ranged from prototypes needing to act as technical proofs-of-concept, to prototypes requiring highly detailed representation for success within the organization.

Generally, the malleability of simple representations is central to the argument of why more simple representations should be used. Benyon (2010) describes features of Lo-fi prototypes as being design quickly, but also thrown away quickly, without hindering the process of generating prototypes for more design solutions.

Benyon also considers how the style of representation can be very different between what is used within a design team, and what is shown to people outside of the team, since the design team members are knowledgeable in the problem domain, and where the team are in the design process. For this reason, more seemingly abstract and simple representations can be used. When the designers view the artifacts through the lens of previous insights, the artifact act as a very rich representation. Similar patterns were observed at Mojang, where the design team could share prototypes very limited in scope or detail within the team, but when the same artifacts were shared outside of the team, certain areas of the design was misinterpreted. For this reason, insights into how to best share artifacts outside of the team increased over time, making the area interesting to consider for defining best practises.

Half-way into the project, the quality of representation was captured in a model that was based on the design work at Mojang done up until that point. Here, fidelity was viewed through the lens of accessibility for different people in an organization (e.g inside or outside the design team). This was captured in the spectrum model in 7.14, using the terminology of Including – Excluding.

Fig 7.14: Modelled spectrums of prototype qualities.

Here, the Micro – Macro scale also represent a similar concept to what Lim et al (2008) describes as *Scope*, which is described as how many aspects of the design idea is covered by the prototype. When looking at the prototype examples from chapter 7.1, *Configuration flow prototype* and *Block interaction prototype* can demonstrate this difference between micro and macro, as seen in figure 7.15.

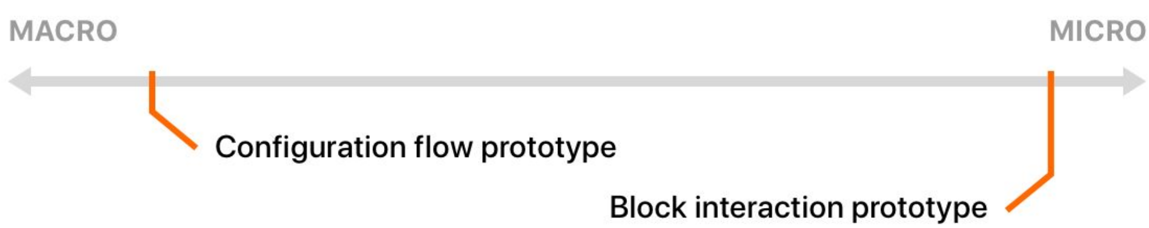


Fig 7.15: Macro – Micro quality model

When looking at the Including – Excluding scale again, from the same diagram, the *Option selection prototype* and the *Navigation prototypes* from chapters 7.1 demonstrate this difference, see figure 7.16. While the option selection prototype features visual similar to what an final product would look like, the navigation prototypes might be harder to grasp unless the viewer is more deeply involved in the design process.

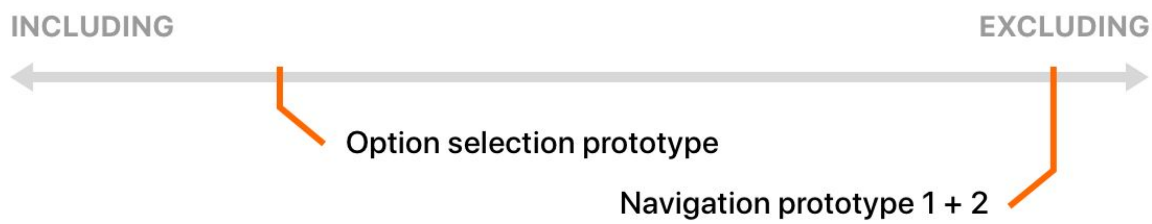


Fig 7.16: Including – Excluding quality model

The third dimension of the model in figure 7.14, was the balance of Explorative – Validative, which I later came to understand as something less related to the specific prototype representation quality, and more something related to purpose the prototype, which is elaborated on in chapter 7.2.1.

7.3 Guidelines

This section describes the final set of guidelines. The set of guidelines can be considered as an amalgamation of the presented needs of designers in section 7.2, which is based on research and practice at Mojang, and also insights into how designers can adopt new improved practises in the most effective way, derived from the workshop sessions.

The insights into ease of adoption both regard ease of internalization of the problems facing a designer who prototype (e.g *why is this guideline relevant to me*), and mapping of guidelines to concrete parts of current practise (e.g *what part of my practise can I change to help this problem*).

For the following chapters, first the guideline title is given as the chapter title, followed by the a short guideline description. After that, each guidelines is followed by longer description that gives motivation and examples, to enable the reader to fully understand the function of the guideline. Finally, the next paragraph gives a set of relations between the guideline and different research and design aspects of this project, to help document how the guideline was synthesised from this project.

In this chapter, several of the guidelines have been tweaked or expanded upon since the fifth iteration of the project—in some sense, documenting the guidelines here acted as a sixth iteration of the project, as the process of collecting and comparing the guidelines with other results of the project here acted as a form of continued evaluation, which revealed some opportunities for clarification.

7.3.1 Tool expressiveness

The prototyping tool should support encoding and exploring the design qualities that needs to be analyzed.

When building prototypes, a core ability is of the prototyping tool to enable certain expression for the designer. For example, when designing animation properties of a interactive representation of a design, a tool that can encode, show and manipulate animations is needed to enable the designer to craft a meaningful prototype. Similarly, when designing an app with many different states – a tool that supports appropriate state logic should be considered.

This guideline heavily connected to the arguments presented in chapter 7.2.4, *Considered style of representation*. However, while findings there generally suggest that simpler representations can be sufficient (much based on the need for *Speed of construction and use*, chapter 7.2.2), this guideline forges an important counterpoint—a core purpose of prototyping is to construct artifacts that carry interactive qualities encoded, and if a quality can't be captured by the chose tool, this benefit is lost. During design practise at Mojang, the design team faced situations where the expressive abilities of the design tool was not enough to fully explore the given design area. This occured mostly when relying on simpler tools such as pen and paper, but also as exemplified in 7.1.3, where Origami Studio could not fully express design related to certain gamepad inputs. When the tool did support certain qualities, benefits were very clear. This is exemplified by the the programmatic icon scaling from 7.1.6, and the support for 3D-touch input in example 7.1.4.

7.3.2 Tool testing ability

The prototyping tool should support simulating the specific usage context for design's use case.

When a prototype have been constructed, a core ability of the prototyping tool is to enable certain testing situation for this prototype. For example, when designing for touch input in a user interface, a tool that can show the prototype on a device with a touch screen is essential to enable a testing situation where the touch input is meaningfully evaluated. Similarly, when designing an app relies on testing in a specific situation of use (e.g while the user is driving a car) the output of the prototyping tool should be able to be put in such as situation, and not be constrained to a specific artificial environment.

In this project, the need of such testing abilities was exemplified by how the touch inputs in 7.1.5 and 7.1.6 could be tested on devices, as the realistic touch input was required to fully appreciate the design of interactive qualities relating to the nuances of how the user presses on the screen. The device frame from example 7.1.3 also provided the benefits of this guideline. While sharing videos of a prototype can be a lesser kind of evaluation, the ability to add a custom frame added to how well the videos could be evaluated by the viewer.

7.3.3 Tool sharing ability

Consider which methods are available to communicate and share tacit/non-verbal qualities of prototypes to stakeholders.

When a prototype have been constructed, a core ability of the prototyping tool is to enable sharing of the work in accessible formats, that can be shared using established forms of communication within the organization, that connects the relevant stakeholders, to what information about the design qualities that needs to be derived from prototype. For example, many software based prototyping tools, such as Framer (Framer 2018) enable the designers to send the a hyperlink to a prototype that is hosted on a server, that enables anyone with access to the link to download and test the prototype. Other prototyping tools such as Origami Studio (Origami 2018) give the designers elaborate functionality for recording and editing video clips of the prototype, as seen in figure 7.17. These then can be shared or used in presentations.

In the design practise at Mojang, most prototypes were shared with a wider set of stakeholders after they had been constructed. Often, significant value could be derived from the tools intentional ability to produce sharable artifacts. This was exemplified by the navigation prototypes described in section 7.1.1 and 7.1.2, which used the built in functionality of Origami Studio to output high-quality video clips.

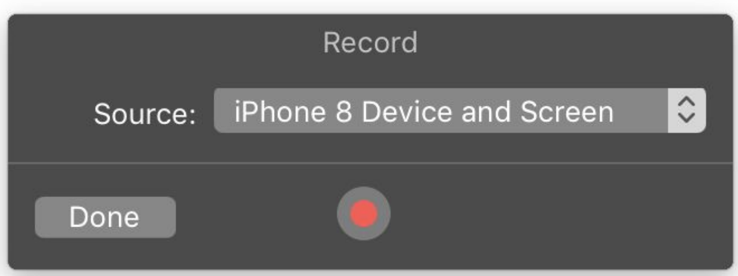


Fig 7.17: Origami Studio recording tools

7.3.4 Impact prioritization

Prioritize features that have major impact on the experience that needs simulating.

Before prototyping activities start, an important activity is to identify what parts of the design that influences the experience of using the prototype the most. Questions like: *"Which interaction is the most challenging?"* or *"Where does the user spend the most time?"* can help prioritize the impact of the different features of a design. When the most impactful features have been identified, consider focusing the prototyping efforts on those parts, as they will be the most important parts to evaluate and iterate upon, to improve the overall design.

In my synthesised needs for interaction designers, this practise is related to *Speed of construction* from chapter 7.2.2 and *Considered prototype purpose* from 7.1.2. If a prototyping project starts out with implementing high-impact features, more satient simulations of experiences can achieved faster. Furthermore, having defined a prototype purpose, helps when identifying what feature of the design that are impactful for the experience. This was exemplified in the two navigation prototypes described in chapter 7.1.1 and 7.1.2, as they successfully captured essential aspects of the design, in very little time. If efforts would have been more evenly spread across more design qualities of less relevance, the prototype would take longer to construct, with only few derived benefits from the extra efforts.

7.3.5 Sunken-costs awareness

Invest in prototypes when you still have time to abandon the prototyping work.

Prototyping is often a time-consuming task, as adding interactivity means that more states of the design needs to be present, and the interactions need to be encoded in the artifact. When designing in general, being heavily invested in a design artifact can makes designers gravitate to not switch to a different design, as that means having to start over with something less refined. However, abandoning work like that can help the team find a better foundation for a new design. For this reason, prototyping activities must continuously consider if the current prototype represents a design which is truly considered the most promising, or of you are being biased by the efforts spent refining it. Adding to this, prototyping tools often provide features that can make any demo look impressive, from the tool's ability to encode expressive qualities into the artifacts—these qualities are however not always in relation to how well the design fulfils its intended purpose. As a good starting point for decreasing the risk of bias, consider how time-restrained you are when starting work on prototypes. If there is time to abandon the work, investing in details might be an valuable effort, but if not, then the focus should perhaps be on fundamental aspect of the design.

In the design practises at Mojang, several prototypes came to explore detailed design qualities that were later not going to be present in the design, because a fundamental organizing aspect of the interface was changed. In certain situations, artifacts were improved upon with the hope that more refinement would solve certain design problems. What really was needed, however, was to let go of certain foundational design ideas encoded in the prototypes, and start over with something new. In contrast, in situations when several design options were explored to solve the same problem (such as example 7.1.1 and 7.1.2), both artifacts were equally invested in, and there was less risk of bias from the designer who made the prototypes. This guideline also relates to the need of identifying purpose, as outlined in 7.1.2—With a clearer purpose of the prototyping activity, a designer can be more intentional whether efforts is spend prototyping detailed aspects of a design, or more fundamental ideas.

7.3.6 Side-by-side comparisons

Plan for comparing options with critical interactive qualities side-by-side.

A testing-based understanding of one design can be hard to compare against an intellectual or memory-based understanding of another. For this reason, consider producing equally elaborate prototypes for competing design solutions, to minimize bias when evaluating the work. When communicating or documenting why a certain design decision was made, having the options side-by-side in the documentation helps clarifying why the design process indeed ended up with the correct solution for a problem, by showing that a fair comparison was made between alternatives.

This guideline relates to the concept of tacit knowledge described in chapter 3.1.2—actual physical experiences can carry more insights, but be hard to express or communicate. From the design situation exemplified in section 7.1.7, side-by-side comparisons were also used in user testing of prototypes. Here, when different versions were presented to the users, all differences observed between the test could serve as good indicators for how the differences between the designs would affect the experience. Without the side-by-side comparison, insights would would have been more ambiguous, with no other option to compare to. Furthermore, being able to compare the navigation patterns from 7.1.1 and 7.1.2 side-by-side, was identified as being beneficial for a quality judgement on the merits of the different designs. As another example, when the block interaction prototype was iterated upon, as described by example 7.1.6, the new version of the design could be compared to the older one side-by-side, and the benefits derived from the design iteration became more satient. For this reason, most older versions of prototypes was saved over the course of the project when iterating on a design, and this practise was often shown to be beneficial when the team had to motivate why a certain design decision was made, to new stakeholders that were introduced to the project.

7.3.7 Minimize scope

To save time, consider a separate prototype when a design quality is isolated enough to be able to be judged without other factors being implemented.

Prototype complexity can grow exponentially when many design qualities are explored at the same time. For this reason, prototypes that are more focused on evaluating very specific problem can help you arrive at actionable results faster, since making informed judgements can be hard when many different design changes influence evaluations. When breaking down a design problem into smaller parts, there is an opportunity in understanding *why* a certain part of a design work well, and not just experience a complete design that provides a good or bad experience in general. Adding to this, when a design process is organized around investigating smaller chunks of the overall design problem, the work is easier to parallelize between different designers, as there is less need for a monolith prototype that everyone needs to contribute to. When several qualities of a design do intersect, however, make a conscious decision to make a prototype with more wider scope. One reason that might be needed, for example, is to construct a realistic user test.

When looking at the needs derived from research in this project, *Speed of construction and use*, as described in 7.2.2, describes several aspects of how smaller prototype scopes enables more efficient prototyping. In the examples from the projects, most design situations showed merits of enabling healthy iteration and evaluation when investigations targeted design problems with smaller scopes, such as examples 7.1.1 and 7.1.2.

7.3.8 Prioritize niche problems

Prioritize prototyping design problems with non-established solutions, patterns or guidelines. Don't reinvent the wheel unless it's needed for your specific needs.

For unique design problems, new tacit understanding is needed the most. Established design domains often have guidelines and patterns which already describe which design will work well in a certain context or for a certain set of needs from the interface. Instead of implementing these in your prototypes, consider finding aspects of the design work that is not covered by any established design system. This could be interactive qualities that are unique to the problem domain of the specific product or feature, or a certain usage context that make currently established design systems unreliable.

From the design situations at Mojang, several of this niche problems were identified and prototyped for specifically. For example, while the designs often targeted touch devices, the requirements of featuring artistic elements related to the game, made the design context too different to fully consider the interface guidelines of the specific software platforms, such as Apple iOS Human Interface Guideline (Apple 2018) or Android Material Guideline (Google 2018). Instead, prototypes were used in these situations, to enable us to fully explore the requirements of this niche ourselves. The design situation exemplified in section 7.1.3 is an example of such a design process. Additionally, the requirements of us supporting many platforms, and with many players switching between platforms when playing Minecraft, was considered a niche interaction design constraint, which could potentially make us disregard the best practises on certain software platforms where Minecraft can be played. However, to ensure that the designs were still usable, elaborate prototypes were built and tested.

7.3.9 Uniform representation

When building prototypes, consider striving for a uniform representation of the different parts of the prototype, to make the testing experience consistent.

For example, a prototype with partial high-fidelity can make shortcomings on other areas of the design more unexpected and distracting. For user testing of prototypes, with tester who are not involved in the design process, this consideration is of critical importance. Instead of implementing partial high-fidelity, considering implementing a lower fidelity prototype that can be provide a consistent experience. With lower levels of fidelity, the design can instead rely on suspension of disbelief by presenting an artifacts that follows a consistent set of rules about how it's presented, which can be explained to the test subject beforehand. To help achieve consistency with software based prototyping tools, consider using tools that provided component libraries that are kept to a certain level of fidelity. Examples include Balsamic Mockups (Balsamic 2018) that keeps prototypes to a pen-and-paper style, or Framer which provides component libraries for different levels of fidelity (Framer 2018).

At Mojang, several times prototypes were demoed with the intention of showing of a certain aspect of the design, but a shortcoming in another place of the prototype gathered too much attention to be able to have a beneficial discourse about the detail of interest. While this inconsistency was obvious to people that just got introduced to the prototype, the designer did not have the same experience, as they had already learnt to disregard the unsightly details from the process of being exposed to the problem by building it.

7.3.10 Focus on the unknowns

Consider making smaller, more focused prototypes, and de-prioritize parts of the experience that you or you team already have a clear understanding of.

Often a prototype is used and tested within a group of people that all are very involved in the state of many aspects of the design. In this situations, consider leaving those qualities out of the prototype, and rather focus on the unknown areas, with design qualities previously not seen before. If a design quality already is explored in another design artifact, consider pointing to that artifact instead, and save the effort of adding it to the new prototype. On the contrary, also consider adding more of these established ideas when sharing more prototypes more widely, as not everyone has the same context as the team. Also consider leaving out parts of the experience that already have a clear path of refinement ahead of them, but for a more future time period or for another team. For example, if the typography is to be refined in a later stage of a project, you should leave out adding improvements to any of those qualities, since they will be more thoroughly considered at a later stage.

In the identified needs from chapter 7.2, this guideline heavily relates to *Speed of construction and use*, where a more focused prototyping effort can be built faster, and with more iterations on the identified design quality that is still unknown. In the example situations given in chapter 7.1, the navigation prototypes (see chapter 7.1.1 and 7.1.2) identified layout and navigation structure as the most important, unknown aspects of the experience, and by focusing on them, the prototypes could provide valuable insights quickly.

7.3.11 Adaptive prototypes

When prototyping for multiple platforms, consider your prototyping tools' ability to adapt the UI dynamically to different screen sizes and input methods.

When using software based prototyping tools, consider using or adding functionality that can adapt the design a design between different screen sizes or input methods. While the process can be more time-consuming when creating the artifacts, dynamic layouts that respond to their testing context can save time if the design needs to be investigated for these different contexts, and reduces the need for having several different artifacts. When sharing prototypes to others, having the design restricted to one device type or screen size can be a significant hindrance in how easy it is for others to set up the prototype on their own device, which makes it harder for others to participate in testing a shared prototype. Adaptiveness can be added either as manipulation of the visual presentation dependant on what screen size is used, but it can also consider encoding support for more input methods, to further enable the prototype to be tested by anyone in any device context. Additionally, having explored these adaptive qualities in the prototype can serve as a better starting point for when the prototype is to be implemented as a final product.

In the design situations described in chapter 7.1.3, an effort was made to have the prototype work well with different sizes of phones, which enabled more people to test the interaction on their device. Also, this prototype supported three different kinds of input methods (touch, keyboard or gamepad), which helped the prototypes ability to be tested in any situation, independant on what equipment was available. Furthermore, having the different kinds of input methods accessible at the same time, enabled benefits similar to what is described in the guideline of *Side-by-side comparisons*, with testing and comparisons being more direct and salient.

7.3.12 Invalidate ideas

When testing a prototype, consider prioritize finding what would make a solution invalid, not on what is right or good about it.

Prototypes tend to be positively appreciated because of their added richness, in comparison to static artifacts. This can make finding negative aspects more challenging, and thus it can be important to consider how the design problem is framed, according to these terms. For example, if a prototype is exploring a design quality for the purpose of finding something that invalidates it, more effort might be spent finding certain states or data that makes the solution not viable. If a button placement is tested on a prototype of a touch interface, the "validate"-framed test might just confirm that the button is clickable, but an "invalidate"-framed test could potentially go further in finding reasons for why the placement doesn't work, such as that the placement makes the button unreachable by thumbs when holding the device. This can focus the prototyping efforts on finding problems that are still unknown to the design situation, and reveal important information about a design early, before a team is heavily invested in a specific solution. Sometimes, it's of less interest whether the focus is on validation or invalidation, but rather that something in general is specified as being tested, which either of the framings of the problem can help with.

At Mojang, often when prototypes were shared the response was positive feedback about the most expressive and new aspects of the shared prototype, which left less room for feedback concerning potential critical shortcomings. However, for prototypes that captures a smaller set of design qualities, such as the navigation prototypes from chapter 7.1.1 and 7.1.2, this was less of a concern, as everything that the prototype capture was of essence, and in need of judgement.

7.3.13 Set testing goals

When building prototypes for testing purposes, make it explicit that you're aiming for qualitative data, quantitative data, or both.

Having clear testing goals can affect the scope needed for building the prototype. For example, if an app is prototyped for qualitative purposes, less structure and consistency might be needed to be able to produce useful qualitative data from the user test if a designers can help facilitate. If a quantitative test is the goal, consider how the prototype could support that, such as having clear goals that can be measured, e.g time passed or number of taps. A testing goal can also be to test a specific interaction, which can be a helpful way to focus the work. Additionally, the goal could be to not have the prototype be user testable, which could decrease the needed scope.

From the needs identified, this ties to *"Considered prototype purpose"*, as defining what testing goal the prototype has, is to give it a clear purpose. This can then help make each addition more considered. From the Mojang design situation described in chapter 7.1.7, a clear goal was set to user test the prototype, which increased the scope but also made the effort more purposeful and focused, when the design team made decision on what to put in the prototype and what to leave out. For example, a clear goal was to couple the prototype to a user goal that the user could be assigned to try to complete—Such a task required certain featured to be added for the task to make sense, but design that was believed to not affect the goal, could be left out.

7.3.14 Use platform hints

When sharing prototypes for a multi-platform product, use platform cues such as depictions of hardware, to reinforce that a design is made specifically for a certain platform.

In situations where prototypes are shared as video files, static images, storyboards or in other contexts outside of actually being presented as an application on a device, several additions could be made to give more context to the user. For example, depictions of device frames of the intended device for the design could be added to give more context on how the size of the prototype relates to the real world representation. This can be of extra importance if, for example, a prototype of a phone app is shown on a big screen for a presentation, where qualities of interface size, readability of text, reachability etc are lost from the unnatural size that the artifact is viewed in. In this scenario, if graphics are added to show how the user interface relates in size the a surrounding phone hardware, the appreciation of the design qualities become more realistic. Additionally, input devices or logos of the platforms can be shown, to further help contextualizing for viewers.

From the design situations at Mojang from chapter 7.1, this technique was used for every prototype that was built, since sharing of the prototype was almost always done through video clips, static images, or as presentations given on big screens. More specifically, example 7.1.4 shows how the specifically considered adding such platform hints, as it was seen as essential for communicating the purpose of certain prototypes. Since many of the touch prototypes were built using Origami Studio (see example 7.1.1 – 7.1.6), the team also benefited from the built in indicators of simulated touch input, which helped understand approximately how big a finger is when used to tap on the screen. This indicator (as seen in the figure in chapter 7.1.6) also clearly shows when a tap has been made by switching color, further helping the viewer of the prototype understand the how the usage context on touch devices relates to the prototype.

Fig 7.18: Surrounding the view of the prototype with a device graphic

7.3.15 Multi-platform overlap

When designing for multiple platforms, not all features need to be prototyped for all platforms.

For design situation that involve simultaneous design of interfaces for multiple platforms, consider which design qualities stay the same between platforms, and which differ. Then, to save efforts when prototyping and testing the designs, consider if there are similarities that only needs to be tested in one of the platform contexts. For example, if a new user registration interface is to be designed for phone apps and for consoles, the flow of actions might stay the same, and only needs to be tested with one of the input methods. However, the style of keyboard input can be very different between the platforms, and a new prototype could be built to just learn about that aspect. When all aspects of the designs doesn't have to be replicated across all platforms, time and effort can be saved.

This problem was key to the team design work at Mojang, as the design work generally targeted many platforms, and for each design efforts were made to make sure it's behaviour across platforms and input methods was clear. For some prototypes this was explicitly seen in what was implemented, such as the one exemplified in chapter 7.1.3. Here the method of selecting an option was clearly different between using a gamepad, using mouse input, or using touch. While the differences in selection logic were represented for each platform, the prototype also showed many qualities that stayed the same between platforms, which saved the team the effort of building three different prototypes. For the block interaction prototype exemplified in chapter 7.1.6, this is an example of a interaction that very specifically targeted touch input. For this reason, most other parts of the user interface were left out as they were represented the same across devices. The only thing was of interest was the key difference in input mechanics, which is what was prototyped and tested.

7.3.16 Prepare the audience

Make sure to provide the optimal context or information about the prototype, to get more value out of sharing it.

When sharing prototypes with others, consider providing what needs to be known be able to critique or evaluate them in the most helpful way. For example, noting what problem the design trying to solve can help the other designers give feedback for improving the design for that specific purpose, when shown the prototype. However, if this information is not given, the prototype, and the design qualities, are more open to any interpretation and judgement, which might be more helpful in certain situations. When conducting evaluations with end-users, consider instead providing what would make a realistic testing setting to a real-world usage situation, to enable similar reaction to what would happen if the user used the final product in its intended environment.

This guidelines is one of the practical implementation of the need identified in chapter 7.2.3, *Considered testing or sharing*. Since building a prototype often requires significant effort from the designer, spending time considering how to best present, share or test the artifact can be of importance to derive the most value from the now created artifact. While an initially defined purpose can guide what sharing or testing should be done after a prototype have been constructed, often the act of constructing the prototype was found to affect how it could best be evaluated, which implies that there should be a re-evaluation of purpose as the prototype is being built, to further improve how the team could learn from it. This is exemplified from the design situation in chapter 7.1.3 and 7.1.6, where the act of building the prototype changed the course of which qualities was to be represented, which then influenced what evaluations were to be made, and who was the intended audience.

7.3.17 Share something testable

Use prototypes for asynchronous communication of interaction design.

Prototypes can be complete enough to be judged without a designer presenting it, since the artifacts often carry enough qualities to let them be appreciated similar to a final product. This makes prototypes suitable for asynchronous communication, where a prototype is shared at one point in time, but it can be viewed at some other time by the recipient. When the work can speak for itself, the testing situation is also less influenced by how the designers is presenting it, which can make the usage situation more similar to how a end-user would encounter the design. Prototypes shared as testable artifacts, that can be appreciated on their own, can also help recipients make more fair judgements, with less pressure from time constraints or the social situation that could occur when a prototype is tested live, together with a designer presenting it.

This guidelines is one of the practical implementation of the need identified in chapter 7.2.3, *Considered testing or sharing*. Since prototypes have the opportunity of carrying more encapsulated interactive experiences, with tacit qualities (see 3.1.2)— it is an opportunity to consider for designers. In the design situations at Mojang, this was proven beneficial from seeing how prototypes shared as testable artifacts tended to generate more in-depth feedback, in comparison to prototypes that were shared as video-clips. Building prototypes as testable artifacts was also key for our design team collaboration with the user research team, as they could present the prototypes in the user tests similar to how the regular Minecraft application can be launched from the device.

7.3.18 Build on others' work

Find and use implementation techniques or code from other designers to make prototype implementation faster.

When building prototypes for with certain software based prototyping tools, often the software provides a platform for sharing UI components, or certain implementation techniques, between users of the tool. For this, reason, when building prototypes (and to a certain extent, when choosing which tool to use), make sure to search for if existing code or UI components can be used to quickly reach the intended design quality. Examples of where techniques are shared include Facebook groups, code sharing sites like GitHub or example projects from prototyping tool creators. Also consider which resources are provided by default in prototyping tools. For example, if an Apple iOS application is prototyped, Origami Studio (Origami 2018) provides fully interaction version of the more common UI components that make up iOS apps, which then can be used to compose the prototype.

From the design situations at Mojang where prototypes were built, this technique was found to save implementation time on a very regular basis, and for most prototypes built. In the case of our design workflow, the most valuable resource was the online message board for users of the Origami Studio app, known as Origami Community (Origami Community 2018). On there, functions and UI components were searched for by the team, and often other users had shared something similar to our needs, which was available for download and use in our prototypes. One example of this is from the design situation described in chapter 7.1.6, where at one point the team tried to implement animations through providing sequences of pictures to the application, which was to be switched between. A way of implementing this functionality was already suggested in the Origami Community, which decreased the efforts needed from us to produce the needed prototype.

7.3.19 Find quick wins

When building prototypes, start out with implementing design aspects are easily attainable with current prototyping tools and techniques.

Be mindful of the implementation effort needed to accomplish certain aspects of a prototype. By starting out with problems of higher implementation difficulty or uncertainty, more easily achievable insights could be left on the table from time constraints, or the prototype can end up with an inconsistent experience. Judging implementation difficulty can sometimes be done before the construction process of the prototype has started, to try to find the most efficient way to explore a certain design quality. However, sometimes the true technological requirements will not become clear until the construction has started, which means that upfront planning is not always enough. For this reason, consider being adaptable to how difficult certain qualities of a prototype are when constructing, as there might be more efficient ways of exploring that area that what was originally planned.

This technique of being adaptable to the implementation difficulty was often used in the design team at Mojang, which influenced the construction path of many prototypes. One example situation of this is described in section 7.1.3, where the prototype was drastically simplified based on what was found about the implementation difficulty. In the end, however, a similar value to what was originally intended could be derived from evaluating the prototype. An often observed pattern was also that technically challenging aspects of implementing certain design qualities was engaging to work with, but a more time efficient way to solve the problem would have been to search for existing solutions online, as described in guideline *Build on other's work*. This unpredictability of prototyping processes relates to a core property of *wicked problems*, as described in chapter 3.1.1—since a complete definition of the prototyping problems cannot be had, one must accept approaching a problem without complete insight into the problem domain, and then adapt to the situation.

7.3.20 Interactive wireframes

Consider keeping visual design to a minimum in an interactive prototype, or making just the wireframes interactive.

Keeping visual design simple in prototypes lets the designers focus on the qualities that prototypes uniquely have in comparison to static design artifacts, which is interactivity, navigation between states, animation and such. For examples, consider not adding visual properties that are easier to express and explore in a tool for static design, such as colors, typography and details of visual design. Alternatively, if a design project involves designing static wireframes in an early phase, consider making them interactive in a prototype to get early insight into the design of the interactions, before high-fidelity visual qualities have been explored. Interactive wireframes, or simplified visual design in general, also makes it easier to get feedback specifically on the interactive qualities that the prototype introduces, with fewer amounts of design qualities visualized simultaneously.

In the design situations exemplified from the design team at Mojang, the Navigation prototypes from chapter 7.1.1 and 7.1.2 were built by making static wireframe artifacts interactive, by adding them to a prototyping tool. As described in detail in the examples, this enabled the team to get crucial insights into foundational aspects of the design quickly. From the needs that this project synthesised, this guideline can be seen as a practical implementation of both *Speed of construction and use* and *Considered style of representation*, as this guideline can give the team both benefits of faster prototyping work, and more focused testing. The opportunity of using interactive wireframes is also considered by prototyping tools such as Framer (Framer 2018) or Balsamic Mockups (Balsamic 2018) who offers a component library for building interactive prototypes with simple visual design.

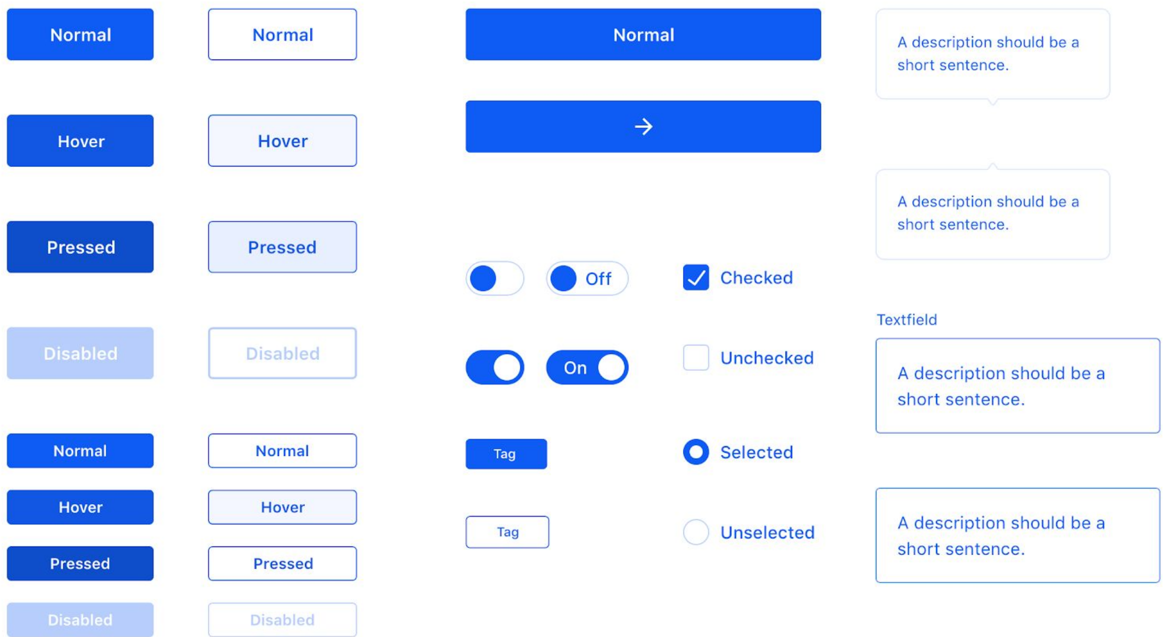


Fig 7.19: Framer wireframing components, for building interactive wireframes

7.3.21 Build components

When building prototypes, consider building composable building blocks, that make up the final solution.

When building prototypes for with certain software based prototyping tools, often the software provides a platform for sharing smaller parts of a implementation between different projects or prototypes. Similarly, if a paper prototype is constructed, the paper representation of the user interface could be made up out of smaller individual pieces of paper which can be freely rearranged to form new kinds of prototypes without having to redraw all parts of the interface. Keeping prototypes easy to arrange from larger building blocks like this can make iteration faster, and additionally, it can focus the design work on more higher-level concepts. Furthermore, when a prototype is made out of components, often work can be shared between different prototypes, and different designers who built prototypes.

In the prototyping work at Mojang, several design situations involved sharing parts of a prototypes implementation between team members and prototypes. Sometimes this was done spontaneously, simply when a problem was solved in one place and then the solution was copied to another place. However, sometimes components were also intentionally built for the purpose of being shared between designers and projects. In these situations, the built-in tools for encapsulating prototype functionality in Origami Studio application were used, which enabled visual qualities to be bundled with logic about their states and data, and then be shared as interface elements with pre-programmed interactive behavior. Additionally, components were often available to download online, or by default in the app, as described in the guideline *Build on other's work*. For example, the prototype in chapter 7.1.3 used such a component for generating *particle effects* in the interface, according to a set of input variables such as rate of particle generation, and lifetime of particles.

7.3.22 Planned experimentation

Sometimes the path to the most interesting design can't be planned out. Instead, consider making room for experimental prototyping with a looser set of constraints, and be open-minded to new solutions.

Prototyping is sometimes considered as an activity that is performed after an idea of a design is already in place, and the prototype is built to represent and validate that idea. However, prototyping can also be used as a way of generating new ideas about a design. Prototyping tools can provide the designer with creative inspiration, since available interactive functionality often is easily accessible as inspiration, which might spark new ideas of how interactive functionality can be formed or combined from the available interactive resources. For this reason, prototyping can be viewed as a kind of research into the material available to form a design. The decision to focus on experimentation (i.e generating new ideas) or validation (evaluating existing ideas) is often tied to which stage a project is in. Later stages can often include more validation, and earlier stages more experimentation, and working with prototyping tools should be considered in all stages, to get a better understanding of possibilities that the technical domain can provide.

When looking at the needs of interaction designers synthesised in this project, *Considered prototyping purpose* discusses how a deeper understanding into the different benefits prototypes can offer, together with intentional decisions about what to pursue, is important for designers to get the most value out of their work. The more generative, experimental aspect of involving prototypes early to explore design options, was often used at Mojang—chapter 7.1.5 describes one such example. While prototyping didn't replace user research and Human-centered design as inspiration for design solutions, it certainly worked as a parallel investigation into the materials that shape the problem domain.

7.3.23 Collaboration needs

When prototypes are tools for communication and collaboration with, for example, a technical team, make sure to understand what the team will need from your prototype, as it can affect the needed scope.

Prototypes can be a tool for communication, or specification, of a design to other teams. However, the relationship between these functions of an organization can strongly influence what the most efficient workflow is when using prototypes. For example, when using a prototype to communicate a design property to an engineer which is involved in the design work, much less detail might be needed from the prototype, and more design work can instead be done when refining and iterating on the final implementation. On the other hand, if teams are disconnected with little room for collaboration, highly detailed prototypes can help ensure that a design is realized according to the designers specification. Another thing to consider is potential options for transferring parts of the design work done in a prototype directly to a final implementation, for these collaborative purposes, as described in the guideline *Reusable implementation*.

In the context of this project, this guideline can be seen as a practical consideration to influence the *Considered style of representation* need, and *Considered testing or sharing*, both of which are crucial in collaboration between teams through prototype representations. In the design team at Mojang, most prototyping work was done without a technical platform being ready for the user interface engineering work to start, which put the team in a place of instead heavily relying on elaborate prototypes to get a complete understanding of how the end-product could look like. However, in later stages of the project, more incomplete prototypes were relied on, with the intention of instead iterating on the final details of a design in the code that can actually be a part of the final implementation.

7.3.24 Reusable implementation

When possible, consider how implementation techniques in the prototype can be reused for the implementation of the end product.

When building prototypes in certain software tools, often parts of the implementation can be considered for being adapted into what becomes a final implementation of a product. This can help speed up the engineering effort, but also has the potential to keep the design represented in the prototype realizable, as design qualities implemented are using techniques that more closely relate to the engineering work that will make up the end product. On a very basic level, a prototype can contain numeric values that can be copied to a final implementation, such as color values, sizes of typographic elements etc. For prototyping tools that rely on the designer writing code to represent interactive qualities, such as Framer, sometimes this code could either be copied, or be used as a starting point for an engineer building the application.

At Mojang, these opportunities became clear towards the end of the project, where the technical efforts of the project had started, and the design team started elaborating on the techniques for sharing the detailed design work with the engineering team. One example of how this was accomplished was that the design team shared the prototypes source files with the technical team, to give them access to all the input variables that made up the design in the prototype. These variables included timings values of animations, color values and sizes of visual elements.

7.3.25 Pick the right format

Videos, paper prototypes, shareable apps, demos, storyboards, looping gifs, voiceovers… There are many ways of sharing prototypes—consider which one fits your purpose the best.

The many different formats of sharing prototypes carry different benefits. To help decide on which would work best, consider what design qualities the prototype is trying to display. For example, if a prototype is representing a linear navigation flow through a set of screens in an app, a video that displays the navigation being performed chronologically might be enough to communicate how well the design works. However, if a navigation which is more unstructured and dynamic, live interaction might needed to understand if the design works well or not. To further help decide, also consider the audience and the prototype—a designer might be more fluent in understanding what is trying to be captured less refined formats, while an end-user might need an application-like representation to be able to make a judgement.

Within the design team at Mojang, often a simple static screen capture was shared to show a new design between the designers. However, as other groups were involved, formats that were more elaborate was used when sharing, such as video clips, or actual interactive version of the prototypes that were made accessible the the other party. One specific format that was found to be effective, is to share a video recording, together with a voiceover made my a designer, for explaining what is happening in the prototypes, and what design problems are considered.

7.4 Guidelines presentation format

This chapter presents the card format, the categorization system used for the set of guidelines, the tags specifying their benefit, and finally, some parts of the workshop format that were found to be helpful when socializing the guidelines within a design team.

7.4.1 Card format

The final format of the cards can be seen in figure 7.20. The card have an aspect ratio of 4:3, but can be printed at different sizes depending on context of use. The card format also feature rounded-off corners.

The top third of the card features an image, meant to give a quick visual reference point for the user, and to help spark the imagination of how the guideline is helpful. The guideline title is below the image, in bold text, with the explanatory text right below the title.

In the bottom third of the card, an extra motivation is provided for why the guideline is helpful, which is an abbreviated version of the motivations from section 7.2. To the right of the motivation, the applicable tags are featured, each tag with a specific color to help identify the ability quickly between cards.

The images are downloaded from Unsplash (Unsplash 2018), an online photography community where photographers share their work, free for anyone to use for any purpose. The full set of card are shown in appendix A.

Fig 7.20: Examples of guideline cards

7.4.2 Process categories

The final set of cards are presented in categories, meant to give the user an overview of the different contexts for designers where the card can serve helpful. The categories are not meant to be prescriptive, but rather provide a helpful starting point when considering the cards. Despite one card being put in one category, it can serve helpful in situations of the other categories. However, the given category is deemed to be most helpful situation.

The categories are presented in an intentionally ordered way, meant to describe the typical chronological progression of different activities related to prototyping. This ordering is also not meant to be applicable in all situations: For example, if a designers is working in a team with an established prototyping tool workflow, the tool category might not need to be considered.

Below are the categories listed, together with the description of what kinds of activities they encompass.

What to prototype

This first category relates to deciding on what part of the design should be in the prototype. These guidelines are meant to give actionable insight into understanding the current design situation, and how that maps onto what should go into a prototype, often according to the different prototype qualities described by Lim et al describes in *Anatomy of prototypes*. The cards consider the design situation from many different aspects, including organizational, stage in the design process, testing needs, uniqueness of the design, need for creative output, prioritization of features etc.

Prototyping tools

The second category contains guidelines relating to how to what to consider in prototyping tools when a designers chooses which prototyping tool to use for a specific design. One purpose with this category is to serve the needs of designers considering which prototyping tools to adopt, which was a commonly mentioned concern when I talked to different designers about prototyping. Specifically, the guidelines targets the abilities of digital tools, since a wide set of tools are available,

Building prototypes

The third category contains guidelines that are the most applicable when in the process of constructing prototypes. This includes techniques in implementation that can help designers reach the intended purpose faster, and ways in which the built prototype can be helpful in other areas, such as using implementation techniques reusable for engineering teams building the end product.

Testing + Sharing

The fourth category contains guidelines that are the most applicable when a prototype is constructed, and is meant to serve its purpose, either by testing, documenting, sharing or communicating design intent.

7.4.3 Tags

This section describes the final set of tags on the card. For the complete reference of which cards were assigned each tag, see appendix A for the full set of cards.

Speed boost: Indicating how implementing a guideline can lead to more efficient prototyping, with more results in less time.

Teamwork boost: Indicating how implementing a guideline can better collaboration within or across teams of an organization.

Testing boost: Indicating how implementing a guideline can higher quality evaluations of prototypes. (Ubicomp paper spectrums?)

Creative boost: Indicating how implementing a guideline can help introduce more ideas about potential design solution, into the design process.

7.3.4 Using the cards in a team

This section describes techniques from the three workshops, that were documented as being beneficial for teams when using the guidelines cards to reason about the teams design process. This result serves as a starting point for developing more techniques on how the guidelines can be presented for, and used by design teams.

As a dictionary supporting discussion

When discussing specific instances of prototyping work, having the whole set of guidelines in front of the groups proved to be a helpful framework for observing patterns in how prototyping was used within the team. Having an extensive set of aspects related to prototyping accessible seemed to provide a solid foundation for mutual understanding when the groups presented their woes of doing prototyping.

To promote such discussion, a specific technique was used to let designers reflect on their own work. The activity is described in the steps below:

1. Have the group individually read the guideline cards for 10 minutes.
2. Before reading starts, the group is instructed to think of examples from their own work, that relate to what is read on a specific card, and write them down.
3. When 10 minutes have passed, the participants take turns to presents the examples written down about instances of their prototyping work, and what guideline card that made them come to think of that.

This activity serves as a helpful way to contextualize the participants problems related to prototyping, to the design work that is done by the team. Having team members presenting to each other how they found a card helpful, was a helpful way of spreading the insight into the role of the guideline cards, between the members of the group.

Spectrums of needs

From the workshops, it was noted that arranging cards on spectrums (see fig 7.21 for example) related to the design teams' current and future ideal practise, was an helpful activity for promoting critical discussion about the teams design process, and how that relates to using prototyping techniques and tools.

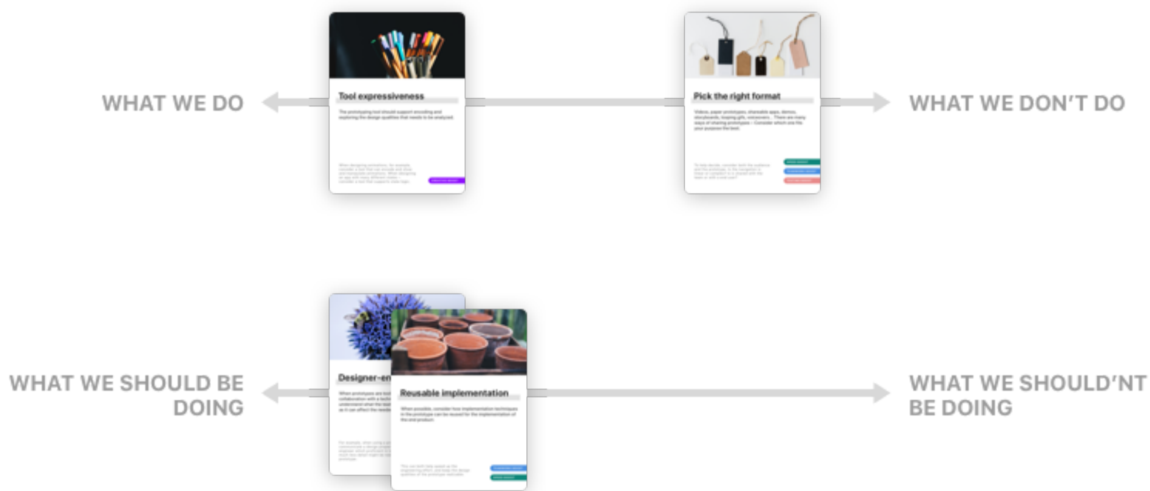


Fig 7.21: Guideline cards ranked on spectrums

Furthermore, the technique was helpful in communicating how a set of cards like this is not a set of rules to adhere to, but rather a set of guidelines that should be applied in conjunction with analyzing the own situation, and what should be considered because of that.

8. Discussion

This chapter will discuss the final results of the project, which includes reasoning about the utility of the guideline cards for designers, and how they apply to different design situations based on the research situation for this project. These results will then be compared to the research question given in the introduction of this thesis.

The research form, and the iterative design processes of the project will also be discussed, including how the idea of the end result changed over time. How this reflects on the overall validity of the final result is also elaborated upon. Finally, future work on the topic is outlined, together with presentation of potential social and ethical consequences of the work.

8.1 Discussion of final result

The end product of this project shifted slightly over the course of the project, by circumstance, but also by intention. Initially, the idea of the end product was something more similar to a technical tool or workflow for helping interaction designers with producing certain representations of design, focused on a specific set of representation challenges: The multi-platform aspect of the Minecraft product.

As this project followed the natural progression of a design project at Mojang, the idea of the end result also followed my understanding of the usefulness of what is produced from the thesis project — and what was found, was that ability to rely on technical tools to produce prototypes can be very shifting dependant in the prototyping situation. While simultaneous design for different end-user platforms was relevant in the initial stages of the project to help understand the basic constraints of that design situation, it did not scale well as a central idea for following through on the project, and specifying detailed design decisions for all specific platforms. This aligns well with my research findings of what designers need from prototyping tools: An ability to quickly reach insights into very specific design problems, which is something that is mirrored in all aspects of prototyping work synthesised from research in the *needs* chapters, 7.2.1 to 7.2.4. How this shift in focus was affected by the research situation and the chosen iterative design process, is discussed in chapter 8.2, where methodology is discussed.

What didn't change over the course of the project, however, was certainly the more general *needs* of the design teams that prototype, independant on what tool or medium was used to produce the prototypes. Serving these needs with the end product of this thesis project, guideline cards, *includes* serving our needs for multi-platform design, but as a subset of serving the overall design situation, which put many other variables at play for the guideline card to address. Having the focus of the end result shift like this, respects the *ultimate particular* of the design situation, which matches the definition of design research given by the like of Gaver (2012) and Fallman (2008). However, this also means that a wider set of variables are analysed, and the generalizability for the result suffers, which is discussed in chapter 8.3, *validity of the result*.

What also is respected is the iterative methodologies of interaction design, which framed this project, and the *wicked problem* aspect of producing design work—no idea of an end result can be fully understood before testing and iterating, which also changes the circumstance and constraints of the project. This methodology aspect is also discussed in chapter 8.2.

A motivation for this shift can also be noted by looking at the collected design situations, with corresponding prototypes, described in chapter 7.1. Prototypes occasionally considered multi-platform aspect of a design, but it was rarely the central problem to address in any given design situation, neither in the prototypes documented in this report, or others that were built by the team.

8.1.1 Guideline cards

As mentioned in the final result chapter, the guideline cards act as an amalgamation of the synthesised research on prototyping and the considered utility of the guideline cards as a learning tool, based on the continuous evaluations with designers with each iteration of the design process. As the guidelines and their card format was produced from an iterative design methodology, their relevance and qualities followed the change of focus that each iteration introduced.

The format of the cards, are very much inspired from similar works, such as the IDEO method cards, but also as a result of the iterative design process: The card format for presenting the guidelines was not introduced until the third iteration, where it was also based very much on the qualitative observations on how design process is successfully discussed with designers. Over the course of the two last iterations, the format of the cards was somewhat stable, and not observed as a critical weakness when evaluating, which increases my confidence in the cards being a relevant format of presentation and sharing. For lack of time, few other presentation formats (e.g digital formats, videos, other physical representations) were explored in this project. Potential shortcomings of the cards that other formats could potentially address include easier access to more in-depth knowledge, better organization, or better ties to workshop activities for teaching the practises.

A specific weakness of the card format also remains, however, which is the tags that tried to communicate more specific properties of the cards — their usefulness were not consistently proven, and they are a part of the format considered for future work, elaborated on in chapter 8.4.

The content of the guidelines, what aspects they consider and what they recommend, was refined for each five iterations of the project. However, they also try to codify a very diverse set of practises and design situations, which made complete evaluations more difficult. While each of the three workshops in the later iterations identified a large set of opportunities for improvement and weaknesses, no workshop discussed every guideline card at depth: They were simply too numerous to have time or energy to discuss all parts of the material. Also, since different guidelines were more or less relevant for different groups, most discussion circulated around those presented practises that felt the most relevant. This can be seen as a weakness of having a numerous set of card, but at the same time, the abundance was also a product of insights from the evaluations: The small set of guidelines from the early iterations carried many abstractions, and we perhaps to general to give advice on particular application. This mirrors Alexandre's (1977) and Welie et al's (2001) focus on providing a specific implementation context together with design patterns: Can a small set of cards be specific enough in what problem context they identify? Perhaps not, and that was also the feedback given to me during the early iteration of the work where guidelines were few, and more all-encompassing of character.

Another hypothesised problem with the content of the guidelines, based on observations from the workshops, is nuance differences between how the guidelines act as a guide for the user. Some present problems to consider, such as *"sunken-costs awareness"*. Others present opportunities, such as *"Reusable implementation"*. Some are formulated as concrete activities that can be scheduled, such as *"Planned experimentation"* and others provide advice more targeted at specific situations, such as *"Use platform hints"*. This problem of inconsistency was approached for the last iteration, but no clear organization emerged, and the task was deemed too significant to fully address in the short time period given.

Related to this, is how cards appear to be of equal value, and potentially of equal interest to designers. This might not be the case, as certain cards can serve as general recommendations, while others demand a much more specific context. One example of this is how *"Minimize scope"* considers generally recommended advice from many articles on the subject (Houde Hill 1998, Lim et al 2008), and matched my own observations at Mojang. In comparison, the others that contain much more niche advice (perhaps only based on my observations at Mojang), is in practice given equal importance as the deck of cards is presented to a designer. For these reasons, the set of guidelines can be considered fragmented, and in need of future improvement. A complete set of identified areas of improvements is outlined in chapter 8.4.

8.1.2 Ethical and social consequences of the result

When analyzing the result through the lens of ethics, one question to consider is: For who is the result ethical—end-users of produced design based on these methods, or the designers and test subjects involved in building and testing using the practises?

The result in itself, a set of guidelines made to be used by designers, does not have a direct mapping to a end products for a user—the products produced can still be ethical or not, based on what is done by the designer. However, could this work promote tendencies in what kinds of interaction design is produced, in general? Alan Cooper (Twitter, 2018) consider prototyping a move away from the user-centered aspect of interaction design, with prototypes rather serving organizations, and designers themselves. Prototypes, and therefore the promotion and refinement of prototyping work, risk distancing the interaction designers from actual needs of end users. In a very abstract sense, such a trend can move the field of interaction towards less ethical products, as organizations dependant on interaction design build their business practises on products with less consideration for the situations of end-users, and more on what turned out successful from experimentation with technologically-focused prototypes.

The other question to consider is whether *testing* with these prototyping practises is ethical, given the deceptive qualities often considered for prototypes meant to be tested with end-users. In a sense, many of the guidelines promote building prototypes that carry these deceptive qualities, as finding opportunities for successful deception is sometimes how prototypes serve their purpose when trying to evaluate design work with more economical means than building out the full implementation of a product. In practise, the ethics of this depends on how the tests are formulated for users. With our internal user testing, users we're told that they were about to test new, unfinished versions of Minecraft, which very much is a honest statement. Other potential testing practises could be more or less honest in these situations, which is mostly decoupled from the results of this project, and more dependant on how the practitioner uses these methods.

8.2 Discussion of methodology

The iterative process in this project, which applied Jones' model of a design process, framed the process in a pragmatic approach of *designing* an end result, made to be used as any product of design. This carried a sense of pragmatism to the production of the guideline cards, but also a sense of abstractness: To what extent is guidelines of prototyping practise similar to producing e.g interaction design work? To answer this question, one can note how the problem certainly is in the realm of wicked problems, as mentioned in chapter 8.1, given the nature of problems facing designers. Additionally, it's also a design context where user-centered design practises apply: Users and usage context (i.e designers who prototype) was researched, and design solutions were evaluated together with these users. To an extent, *prototypes* were also used for evaluating the current state of the design ideas, which was done as the unfinished-sets of guideline cards were evaluated with designers reflecting on their own practise.

A formative aspect of the project was the use of visual models to better understand and communicate aspects of the prototyping craft. This was not initially planned as something to heavily rely upon at the start of the project, but it emerged as a tool to facilitate both thinking about what prototypes are, what qualities they have, and how designers use them. At first, this was unintentional, when models were used in my presentation to communicate my view of prototyping to the students at Hyper Island. For example, spectrums were used to categorise prototyping tools, and abstraction such as “Generate signal” and “Generate noise” were used to communicate the characteristics of prototypes used for evaluative aspects, and idea generation aspects respectively. The effectiveness of these models as vehicles for ideas, made me later *intentionally* try to define visual models, which helped further my understanding the underlying mechanisms of the craft from observing and participating in prototyping practise at Mojang. It served as a good in-between step for formulating guidelines—to try to identify the systems and patterns in the work that the guidelines were meant to address.

In comparison with other sets of cards meant to be used for designers, IDEO method cards (see chapter 3.2) are close at hand, since they also aren’t considering specific qualities of design, but rather processes for doing design work, which indirectly produces design. However, is a design agency such as IDEO perhaps a better breeding ground for developing such methodologies? The nature of work as consultants can involve repeated application of similar methodologies in different contexts (e.g for different clients), and by nature patterns that produce beneficial results across different contexts, could perhaps emerge. For a very specific design situation such as for the design team at Mojang, identifying such general patterns should be more challenging, as only one project is studied. One way this need for context diversity was introduced in this project was through workshops and presentations with other groups of designers, outside of the Mojang team. However, each of these situations are by nature more artificial, since no actual design work is produced in these contexts. Rather, they serve as a proxy, hinting at how a wider set of context relates to the produced set of guidelines from each iteration.

This situation can be considered a weakness of this project, since it is very dependent on to which extent these situations carry different evaluationary qualities from the reality of design projects. At the same time, arranging these events was not a forced process, rather it was met with enthusiasm from the designers I talked to. Discussing the topics of prototyping process was highly appreciated by the audiences, hinting at that these topics have a high relevance for the actual design work produced, and that participation was done with very pragmatic intentions, closely related to actual design practise.

More challenging aspects here include how the different workshops were conducted in different environments. For example, the first workshop was performed at Mojang together with members of the same team I worked with from before. This made the participants have more insights into the research project and its goals, which could possibly influence the discussion. Additionally, the teams the workshop was conducted with had different needs from their practise, and used prototypes to different extents, which influenced which insights would be collected. Having more similarities in for example the needs of designing for multiple platforms, might have produced insights more specific for answering the research question. While a goal was set during the search for workshop participants to have multi-platform experience in the teams, this requirement could have been stricter.

When comparing the project plan procured from after the pre-study, to the final process, several changes stand out. Firstly, a research method of interviewing designers at other companies was replaced with the more evaluative format of workshops, combined with the multiple events where I got the opportunity to talk to designers when giving lectures on the topic of prototyping at Hyper Island and the Interaction Designers Association. While this did give me qualitative insights into what was of most relevance to designers, and a starting point for defining guidelines in the initial iteration, it lacked the rigor that possibly would have followed from structured interviews, which perhaps could have given a more strong foundation for shaping my guidelines in the initial iterations. The workshops, however, acted as a healthy addition to the process as specific evaluation methods was lacking from the initial planning. Furthermore, the workshop format potentially carried some similarities to how the guideline cards could actually be implemented by teams, not by just reading the material, but to actually have collaborative sessions focused on the design process instead of design work. For this reason, the workshops evaluated not only the content of the guidelines, but also the format of implementing the knowledge into practical situations.

In comparison with the initial planning, two more iterations were added to the overall process. This followed partly from the generalization of the scope—with a wider scope being studied, more evaluation contexts and iterations were needed to approach a result that was more exhaustive. Even so, the ambition of the final scope of the project is not completely fulfilled, and future work is much needed to validate and add more detail to the result.

8.3 Validity of the result

The validity of the guideline cards is naturally shaped by the research context of design practise at Mojang, which framed this project and introduced me to many practical problems of using prototypes. Furthermore, the practises suggested by the cards are very much now ingrained in the prototyping work of the team, either through trial-and-error learning in early work, explicit learning of certain concepts through the workshop, or from sharing my own prototyping contributions that followed from writing the guidelines, which then acted as inspiration.

Many things make the design situation at Mojang very *specific*, both from a organizational point of view, and from what is being designed—both of which affects what is required from the prototyping practises of the designers. The team was formed right as I joined, which means that I was a part of shaping how we wanted to work with prototypes, together with the other designer, and the design manager. There was no established practise to consider, despite our heavy reliance on prototypes in the project. Additionally, the project situation also involved communication and collaboration with many external teams, including art, engineering, sound production, user research, leadership teams and communication teams. For each of these contexts certain types of prototypes and prototype mediums were used based on what we tried to communicate.

When it comes to what was being designed, many other factors contributed to the project having niche needs. For one, the project was very much specified by us, the members of the team, which means that ideas were equally important as execution in certain phases. Moreover, the many platforms that we were targeting for all new design, carry an almost industry-unique challenge: To simultaneously design for gamepad, touch input and desktop interactions at the same time, and for screens of almost any size. While other software products have the same wide distribution across platforms, the interaction design work can for instance be distributed across many teams, or having each platform designed one-by-one sequentially—having all interfaces are tackled at the same time, in one project, seems to be a rare situation.

Adding to this, since the user interfaces were to be used in a game, almost no established design systems were used, such as Google's Material design or Apple's iOS Human Interface Guidelines. With games, designers typically give themselves more freedom of expression, and with more need for unique expressing through interaction design. This aspect gives this project unique characteristics compared to interaction design outside of a gaming context.

Taken together, it's reasonable to consider how guidelines specifically made for designers in the project at Mojang would not equally apply to other design situation, and other design teams. For this reason, evaluations of the guidelines were conducted with other designers, as discussed in chapter 8.2. So where does that leave the validity of the result? I think a comparison to user tests help here: Typically when you evaluate usability through user testing, it can be observed if a task was completed or not by the user. What would the equivalent test be for a set of guidelines for design processes? What *task* does the produced design help the user complete? For prototyping activities, I can identify different levels of such completion. On a more basic level, simply producing a prototype that helps a team make a design decision, can be regarded as successful prototyping—and if that same prototyping would not have fulfilled its purpose without the given guideline of how to prototype, the guideline makes the important difference that makes one able to deem it beneficial. Clearly, such a test was not performed, either at Mojang, or with the designers i met in the workshops. Instead, I can only see indications of how the guidelines can be beneficial in the different contexts analyzed, based on the feedback of designers during the workshop, and the overall success of our prototyping practise at Mojang based on the state of the project at time of writing.

8.4 Future work

Based on the questions that remain on validity outlined in chapter 8.3, there is a clear need for the further evaluation and research, both to further evaluate the current state of the guideline cards, but also to do further research on the prototyping activity, to approach a set of cards that is more exhaustive and tested. Does a set of guidelines need to be exhaustive to be considered beneficial and good? I think this depends on how the guidelines are presented. If the guidelines are presented as "everything you need to know" about prototyping, certainly additions would be needed to make it more general. Perhaps such a set of cards would be more useful for more wide distribution within the field of interaction design. However, this was not in the scope of this project, but rather to just identify a set guidelines, not the definitive set.

The current set of guidelines can also be improved, just based on findings from the evaluations in the project, for example by giving the copywriting more attention. Putting more effort into a succinct and cohesive writing style for the titles, descriptions and examples would make set set of guidelines easier to digest for users. Additionally, as outlined in chapter 8.1.1, the type of advice that the different guidelines provide could be more consistent.

The most specific problem that was identified from the workshops, was how the addition of the tags between the fourth and the fifth iteration did not seem to accomplish their purpose of making the cards more accessible, which was the reason given for that “feature request” from the Lifesum workshop. This set of tags would have to be considered for either more easily digestible information, or being removed entirely. A hypothesis here would be to rather have the tags represent what type of activity or recommendation the card proposes, as problematized in 8.2.1.

Furthermore, more research and design work can be done to improve the formats for presenting these cards to designers. While the workshop format from the three workshops contain indications of how facilitated workshops might be effective, no alternative formats were evaluated, only the activities within the workshop were iterated upon for the three workshops. Perhaps the presentation format could carry more gamified aspects, like Planning Poker (Grenning 2002), or a interactive digital version could be explored.

9. Conclusion

In this chapter, the conclusion of the project is presented. This project had the purpose of answering the question: *Which prototyping practises are relevant to consider for interaction designers, when designing for a multi-platform audience?* To help answer this question, two supporting questions related to design practise were raised, regarding which prototyping practises relate to different stages in the design process, and which practises that relate to collaborating with other functions of an organization.

To help answer this question, this project employed *research through design practise*, where I worked with a design team at the game studio Mojang, on their game Minecraft. To exemplify my findings from design practise, several prototypes used in the design process at Mojang were documented and analyzed, together with the practises used in conjunction with the prototypes. These examples included prototyping app navigation, touch and gamepad interactions, user interface elements and more. This work resulted in identifying four needs of interaction designers who use prototyping methods, through comparing findings from practise with established research and theory about prototyping methods. The four needs for interaction designers that were identified included *Considered prototype purpose*, *Speed of construction and use*, *Considered testing or sharing* and finally *Considered style of representation.*

To capture these needs as practises, and to evaluate my findings with other interaction design practitioners, guidelines related to specific prototyping practises were produced through an iterative design process. These guidelines, presented as a set of cards, were used in three workshops, where they were evaluated through analyzing the guidelines' ability to help the other designers learn about how to improve their prototyping practise. After the last workshop, and the last iteration phase, 25 guidelines became the final set of guideline cards, ordered into four groups based on where in the design process the guideline was found to apply. The cards cover a range of prototyping-related advice, such as how to better decide what to prototype, and what to consider to share prototypes more successfully.

These guidelines serve as this project's answer to the research question, by suggesting practises relevant to consider. The results from the workshops indicated that these guidelines can help teams identify weaknesses in their prototyping practise, and get inspiration for how to improve in different areas. These results included designers considering *where in the design process* prototyping efforts are made, *what is needed for effective collaboration* using prototypes and *how to use prototyping tools* for different benefits. These results adds indications that the guidelines have mandate to answer the main research question, and the supporting questions. However, more research need to be done to understand the long-term effectiveness of applying the guidelines to prototyping practice, by analysing outcomes of projects that apply the guidelines. Another weakness of the evaluations performed was how the workshops could give false indications of needs and problems that were discussed with the designers, as the setting is artificial and disconnected from their design practise.

With regards to the multi-platform aspect of the research question, most guidelines were found to generally support problems of designers, such as managing a wider scope of what is being designed, which is a consequence of designing for multiple platforms simultaneously. A few guidelines were specifically argued for based on the multi-platform design challenges at Mojang. However, little opportunity was given to evaluate these, as the constraints of working with multi-platform design were not found to be of high priority when running the workshops with the external teams, and less data could be gathered about that aspect of design work. For these reasons, the full set of guidelines is not strictly tied to multi-platform design situations, but are still to be considered for designers facing these challenges.

10. References

Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I. and Angel, S.: A Pattern Language. Oxford University Press, New York (1977)

Apple 2018. *Human Interface Guidelines, 3D Touch*, URL: https://developer.apple.com/ios/human-interface-guidelines/user-interaction/3d-touch/

Balsamiq 2016, *Balsamiq Mockups*. URL: https://balsamiq.com/products/mockups/

Benyon, D. (2010). *Envisionment*, Chapter 8 in Designing Interactive Systems, Addison-Wesley, pp 177-197.

Bjork S, Holopainen J (2005), *Patterns In Game design*. Charles River Media

Bohemian Coding (2018) *Sketch*. URL: https://www.sketchapp.com/

British Design Council (2017). *The Design Process: What is the Double Diamond?*. URL: http://www.designcouncil.org.uk/news-opinion/design-process-what-double-diamond

Brown, T. (2008). *Design thinking*. Harvard business review, 86(6), 84.

Brown, W.J., Malveau, R.C., McCormick, H.W. and Mowbray, T.J.: Anti Patterns, Refactoring Software, Architectures and Projects in Crisis. John Wiley, New York (1998)

Buchanan, R. (1992) Wicked Problems in Design Thinking. Design Issues, Vol 8, Issue 2, pp 5-31. MIT Press.

Buchenau M, J Suri (2000). *Experience prototyping*. DIS '00 Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques
Pages 424-433 New York City, New York, USA — August 17 - 19, 2000 ACM New York, NY, USA ©2000 ISBN:1-58113-219-0 URL: http://doi.acm.org/10.1145/347642.347802

Catmull E (2014) *Creativity, Inc.: Overcoming the Unseen Forces That Stand in the Way of True Inspiration*. 1st edition. Random House.

Cooper, Alan et al. (2014). *About Face: The Essentials of Interaction Design*, 4th Edition. John Wiley & Sons, Inc.

Cooper, Alan (2018), Twitter. URL: https://twitter.com/mralancooper/status/900728419087900672

Ehn, P. (2008), *Participation in design things*, in ‘Proceedings of the Tenth Anniversary Conference on Participatory Design 2008’, PDC ’08, Indiana University, Indianapolis, IN, USA, pp. 92–101.

Erickson, T (1995) *Notes on Design Practice: Stories and Prototypes as Catalysts for Communication*. URL: http://www.pliant.org/personal/Tom_Erickson/Stories.html

D Fallman (2008), *The Interaction Design Research Triangle of Design Practice, Design Studies, and Design Exploration*. Massachusetts Institute of Technology Design Issues: Volume 24, Number 3

E Gamma, R Helm, R Johnson, J Vlissides (1995), *Design Patterns: Elements of reusable Object-Oriented software*. Addison-Wesley.

Gaver, William (2012). *“What Should We Expect from Research Through Design?”* In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI ’12. Austin, Texas, USA: ACM, pp. 937–946. isbn: 978-1-4503-1015- 4. doi: 10.1145/2207676.2208538. URL: http://doi.acm.org/10.1145/ 2207676.2208538.

Google (2017), *Material Design.* URL: https://material.io/guidelines/

Grenning (2002) *Planning Poker.* URL: https://wingman-sw.com/articles/planning-poker

Hartmann B (2009), *Gaining design insight through interaction prototyping tools,* Department of Computer Science and the Committee on Graduate Studies, Stanford University.

Houde S, Hill C (1997) *What do Prototypes Prototype?* Apple Computer, Inc. URL: *https://pdfs.semanticscholar.org/30bc/6125fab9d9b2d5854223aeea7900a218f149.pdf*

Hyper Island toolbox (2018) URL: http://toolbox.hyperisland.com

IDEO (2003) *Method cards*. URL: https://www.ideo.com/post/method-cards

Jones C, 1992, *Design Methods: The seeds of human futures*. 2nd edition. Wiley.

Kelly, T and J Littman (2000). *The art of innovation*. Chapter 4.

Kotaku (2017). *Top Video Game Companies Won't Stop Talking About 'Games As A Service'*. URL: https://kotaku.com/top-video-game-companies-wont-stop-talking-about-games-1795663927

Lego Group (2017), *Mindstorms* URL: http://mindstorms.lego.com/

Lim, Y.-K., Stolterman, E. and Tenenberg, J (2007). *The Anatomy of Prototypes: Prototypes as Filters, Prototypes as Manifestations of Design Ideas*. ACM TOCHI 15, 2, Article 7 (July 2008), 7:1-7:27.

Mackay W, Ratzer A & Janecek P (2000), *Video Artifacts for Design, Bridging the Gap Between Abstraction and Detail*. URL: https://www.lri.fr/~mackay/pdffiles/DIS2000.VideoArtifacts.pdf

Martin Bella; Hanington, Bruce M (2012). *Universal methods of design: 100 ways to research complex problems, develop innovative ideas, and design effective solutions.* Rockport Publishers.

Maeda, J (2017) *Design In Tech Report, 2017 edition*, URL: https://designintech.report/2017/03/11/design-in-tech-report-2017/

Maguire, M. (2001). *Methods to support human-centred design*. International journal of human-computer studies, 55(4), 587-634.

Method Kit (2018) *Rapid prototyping*. URL: http://www.designkit.org/methods/26

Merriam-Webster 2018. *Definition of Prototype*. URL: https://www.merriam-webster.com/dictionary/prototype

Minecraft Twitter Account (2017), URL: https://twitter.com/Minecraft/status/836214948380409858

Moggridge B (2007), *Designing Interactions*. The MIT press

Mojang (2018), *Games*, URL: https://mojang.com/games/

Nielsen, Jakob (1995). *10 Usability Heuristics for User Interface Design*. URL: https://www.nngroup.com/articles/ten-usability-heuristics/.

NEA (2017). *Future of Design in Startups*. URL: http://www.futureof.design/

NEA (2016). *Future of Design in Startups*. URL: http://2016.futureof.design/

Nielsen, Jakob and Rolf Molich (1990). *"Heuristic Evaluation of User Interfaces"*. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI '90. Seattle, Washington, USA: ACM, pp. 249–256. isbn: 0-201-50932-6. doi: 10.1145/97243.97281. URL: http://doi.acm.org/10.1145/97243.97281.

NPR (2017): *Planet Money Episode 796: The Basic Income Experiment*. URL: http://www.npr.org/sections/money/2017/09/22/552850245/episode-796-the-basic-income-experiment

Norman D (2005) *Human centered design considered harmful.* URL: http://www.jnd.org/dn.mss/human-centered_design_considered_harmful.html

Polygon (2014) *This is how Tetris wants you to celebrate for its 30th anniversary*. URL: https://www.polygon.com/2014/5/21/5737488/tetris-turns-30-alexey-pajitnov

Lucero A (2018), *PLEX Cards · Playful Experiences Cards,* URL: http://www.funkydesignspaces.com/plex/

Rittel H, Webber M (1973), *Dilemmas in a General Theory of Planning*. Policy Sciences 4 155-169, Elsevier Scientific Publishing Company, Amsterdam

Schöen D A, (1992), *Designing as reflective conversation with the materials of a design situation*, Butterworth Heinemann Ltd.

Spool, Jared M. (2004). The KJ-Technique: A Group Process for Establishing Prior-
ities. URL: https://articles.uie.com/kj_technique/

Subtraction (2015), *The Tools Designers Are Using Today*, URL: http://tools.subtraction.com/index.html, accessed 2017-07-31

Spinuzzi, C. (2005) *The methodology of participatory design*. Technical Communication, 52(2), 163-174.

Tidwell J, (2010) *Designing Interfaces: Patterns For Effective Interaction Design.* O'Reilly Media

Tidwell J (1999) *Common ground: A Pattern Language for Human-Computer Interface Design.* URL: http://www.mit.edu/~jtidwell/common_ground.html

Vitsoe (2016), *Good Design*. URL: https://www.vitsoe.com/eu/about/good-design

Unsplash (2018), URL: https://www.unsplash.com

Wikipedia (2017) *List of best-selling video games.* URL: https://en.wikipedia.org/wiki/List_of_best-selling_video_games

M Welie, G van der Veer, Eliëns A (2001) *Patterns as Tools for User Interface Design*. URL: https://pdfs.semanticscholar.org/4ea5/9ce773f3f44646f34f7d76eba05438e35452.pdf

Zimmerman J and Forlizzi J. (2008) The Role of Design Artifacts in Design Theory Construction. Human-Computer Interaction Institute. Paper 37.

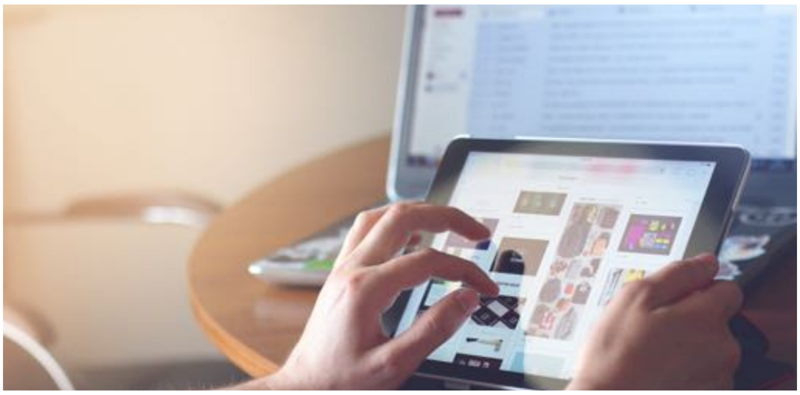
Appendix A: Guideline cards

Tool expressiveness

The prototyping tool should support encoding and exploring the design qualities that needs to be analyzed.

When designing animations, for example, consider a tool that can encode and show and manipulate animations. When designing an app with many different states – consider a tool that supports state logic.

CREATIVE BOOST



Tool testing ability

The prototyping tool should support simulating the specific usage context for design's use case.

Does the tool allow testing with the intended input method? Real devices? Real data?

TESTING BOOST

Tool sharing ability

Consider which methods are available to communicate and share tacit/non-verbal qualities of prototypes to stakeholders

Can you share the prototype as a video? Downloadable apps? In-person demos? Hosted online?

TESTING BOOST

SPEED BOOST

Impact prioritization

Prioritize features that have major impact on the experience that needs simulating

Questions like: "Which interaction is the most challenging?" or "Where does the user spend the most time?" can help prioritize what to prototype.

SPEED BOOST

Sunken-costs awareness

Invest in prototypes when you still have time to abandon the prototyping work

Prototyping is often time-consuming, and the sunken-costs can leave you with a non-ideal design simply because so much was invested into a prototype with a certain design.

CREATIVE BOOST

TESTING BOOST

Side-by-side comparisons

Plan for comparing options with critical tacit qualities side-by-side

Tacit/non-verbal understanding of one design can be hard to compare against an intellectual or memory-based understanding of another.

When communicating or documenting why a certain design decision was made, having the options side-by-side helps clarifying the message.

TESTING BOOST

Minimize scope

To save time, consider a separate prototype when a design quality is isolated enough to be able to be judged without other factors being implemented.

Prototype complexity often grows exponentially with many dimensions. Also, making informed judgements can be hard when many different design changes influence a test.

Separate prototypes can also be easier to parallelize between designers.

When several qualities of a design intersect, consider investing in a complete prototype to judge them all together.

TESTING BOOST

SPEED BOOST

Prioritize niche problems

Prioritize prototyping design problems with non-established solutions, patterns or guidelines. Don't reinvent the wheel unless it's needed for your specific needs.

For unique design problems, new tacit understanding is needed the most. Established problem domains often have guidelines and patterns which already describe which design will work well.

SPEED BOOST

Uniform representation

When building prototypes, consider striving for a uniform representation of the different parts of the prototype, to make the testing experience consistent.

For example, a prototype with partial high-fidelity can make shortcomings on other areas of the design more unexpected and distracting.

TESTING BOOST

Focus on the unknowns

Consider making smaller, more focused prototypes, and de-prioritize parts of the experience that you or you team already have a clear understanding of.

Adding all details and complexity might only be needed when sharing prototypes outside of the team, or with testing subjects

SPEED BOOST

Adaptive prototypes

When prototyping for multiple platforms, consider your prototyping tools' ability to adapt the UI dynamically to different screen sizes and input methods

Dynamic layouts can save time when building the prototype, and reduces the need for several different artifacts.

When sharing prototypes to others, having the design restricted to one device or screen size can be a hindrance in how easy it is for others to test.

SPEED BOOST

Validate or invalidate?

When testing a prototype, consider prioritize finding what would make a solution invalid, not on what is right or good about it.

Prototypes tend to be positively appreciated because of their richness. This can make finding negatives more challenging, and thus more important.

TESTING BOOST

Set testing goals

When building prototypes for testing purposes, make it explicit that you're aiming for qualitative data, quantitative data, or both.

TESTING BOOST

SPEED BOOST

Testing goals can affect the scope needed for the prototype.

Use platform hints

When sharing prototypes for a multi-platform product, use platform cues such as depictions of hardware, to reinforce that a design is made specifically for a certain platform.

Platform cues can include input devices, screen frames, platform logotypes and more.

SPEED BOOST

TEAMWORK BOOST

TESTING BOOST

Multi-platform overlap

When designing for multiple platforms, not all features need to be prototyped for all platforms.

Different platforms can have unique challenging aspects that need to be investigated separately, but other design qualities stay the same between platforms (e.g flow between screens). When these aspects of the designs doesn't have to be replicated, time and effort can be saved.

SPEED BOOST

Prepare the audience

Make sure to provide the optimal context or information about the prototype, to get more value out of sharing it

When sharing with team members or other designers, consider providing what needs to be known be able to critique them in the most helpful way – E.g what problem is the design trying to solve?

When sharing with end-users, instead provide what would make a realistic testing setting, to enable realistic candid reactions.

TEAMWORK BOOST

TESTING BOOST

Share something testable

Use prototypes for asynchronous communication of interaction design.

Prototypes can be complete enough to be judged without a designer presenting it. This makes prototypes suitable for asynchronous communication – the work can speak for itself.

Prototypes shared asynchronously can also help make more fair judgements, with less pressure from time constraints, or the social situation when a prototype is demoed live, in-front of the designer

SPEED BOOST

TEAMWORK BOOST

TESTING BOOST

Build on other's work

Find and use implementation techniques or code from other designers to make prototype implementation faster.

Examples of where techniques are shared include Facebook groups, code sharing sites like GitHub or example projects from prototyping tool creators.

CREATIVE BOOST

TEAMWORK BOOST

SPEED BOOST

Find quick wins

When building prototypes, start out with implementing design aspects are easily attainable with current prototyping tools and techniques.

Be mindful that of implementation effort needed to accomplish certain aspects of a prototype. By starting out with problems of higher implemementation difficulty or uncertainty, more easily achivable insights could be left on the table.

SPEED BOOST

Interactive wireframes

Consider keeping visual design to a minimum in an interactive prototype, or making just the wireframes interactive.

Simpler visual design enable faster iteration on the interactions and flows. Interactive wireframes also makes it easier to get feedback specifically on the interactive qualities.

TESTING BOOST

SPEED BOOST

Build components

When building prototypes, consider building composable building blocks, that make up the final solution.

Keeping prototypes easy to arrange from larger building blocks can make iteration faster, and can focus the work on more higher-level concepts.

CREATIVE BOOST

TEAMWORK BOOST

SPEED BOOST

Planned experimentation

Sometimes the path to the most interesting design can't be planned out. Instead, consider making room for experimental prototyping with a looser set of constraints, and be open-minded to new solutions.

The decision to focus on experimentation or validation is often tied to which stage in a project the prototyping is done. Later stages can often include more validation, and earlier stages more experimentation.

CREATIVE BOOST

Collaboration needs

When prototypes are tools for communication and collaboration with a technical team, make sure to understand what the team will need from your prototype, as it can affect the needed scope.

For example, when using a prototype to communicate a design property to an engineer which proficient in UI design, much less detail might be needed from the prototype.

TEAMWORK BOOST

SPEED BOOST

Reusable implementation

When possible, consider how implementation techniques in the prototype can be reused for the implementation of the end product.

This can both help speed up the engineering effort, and keep the design qualities of the prototype realizable.

TEAMWORK BOOST

SPEED BOOST

Pick the right format

Videos, paper prototypes, shareable apps, demos, storyboards, looping gifs, voiceovers... There are many ways of sharing prototypes – Consider which one fits your purpose the best.

To help decide, consider both the audience and the prototype. Is the navigation is linear or complex? Is is shared with the team or with a end user?

SPEED BOOST

TEAMWORK BOOST

TESTING BOOST