Doctoral Thesis in Computer Science
Specialty Human-Machine Interaction

Investigating the Influence of Visual Signifiers to Foster the Discovery of Touch-Based Interactions

by Eva Mackamul

Jury Composition

Jury President	Jean-Philippe Vandeborre	Professor, IMT Nord Europe
Reviewers	Anastazia Bezerianos Céline Coutrix	Professor, Université Paris Saclay Chargée de recherche HDR, CNRS
Examiners	Andrea Bunt Nathalie Henry Riche	Professor, University of Manitoba Principal Researcher, Microsoft Research
Thesis Director	Sylvain Malacria	Chargé de recherche HDR, Inria
Co-Supervisor	Géry Casiez	Professor, Université de Lille

Thesis submitted on October 25, 2023 and defended on December 21, 2023 in fulfilment of the requirements for the Degree of Doctor of Philosophy of the Univeristé de Lille in the specialty Computer Science.

Thèse de Doctorat en Informatique
Spécialité Interaction Humain - Machine

Étudier l'influence des signifiants visuels pour favoriser la découverte d'interactions tactiles

Eva Mackamul

Composition du jury

Président du jury	Jean-Philippe Vandeborre	Professeur, IMT Nord Europe
Rapporteures	Anastasia Bezerianos Céline Coutrix	Professeure, Université Paris Saclay Chargée de recherche HDR, CNRS
Examinatrices	Andrea Bunt Nathalie Henry Riche	Professeure, University of Manitoba Chercheuse Principale, Microsoft Research
Directeur de thèse	Sylvain Malacria	Chargé de recherche HDR, Inria
Co-encadrant	Géry Casiez	Professeur, Université de Lille

Thèse soumise le 25 octobre 2023 et soutenue le 21 decembre 2023 pour obtenir le grade de Docteur de l'Université de Lille.

Abstract

Discoverability of interactions is a core component of successful user interaction yet it is almost systematically dismissed by interface designers and researchers. This is not surprising given the variety of associated concepts that are inconsistently defined, and the consequent lack of clarity on how to address these problems. In this thesis we emphasise the importance of interaction discoverability and investigate whether visually signifying the availability of inputs improves their discovery using commercial touch-based devices as the test case.

We examine usage and definitions of discoverability as well as related concepts and offer clarifications where definitions are ambiguous while highlighting focus areas and limitations. To examine the circumstance of initial interactions and discovery more structurally, we then propose a conceptual framework of relevant impacting factors. We consider how individual components interrelate and reflect on how they can be used to promote discoverability. Concentrating on the factor 'visibility' on the system side of the framework, we examine the potential of added visual signifiers in mobile touch-based interfaces through two projects.

First we examine how the availability in-place single finger inputs such as 'Tap', 'Dwell', 'Double Tap' and 'Force Press' can be communicated through added visual signifiers in the context of buttons. We propose a design space of visual signifier characteristics. Combining these characteristics, we generated 36 designs and investigated their perception in an online survey and an interactive experiment. The results suggest that visual signifiers increase the perception of input possibilities beyond 'Tap', and reduce perceived mental effort for participants, who also prefer added visual signifiers over a baseline. This work informs how future touch-based interfaces could be designed to better communicate in-place one finger input possibilities. Second, we explore the viability of transient interaction hints during animated transitions between UI screens to improve the discovery of swipe-revealed hidden widgets (swhidgets). To do so, we adapt three established transition patterns to include visual depictions of the swhidgets. We then study how noticeable the transitions are at different durations (750ms, 1000ms). Results suggest substantial variation in noticeability between participants. Subsequently, we explore the impact on the discoverability of interactions with two types of transition (Container Transform, Panels) and a baseline and find that transitions do not significantly impact the use of swhidgets. This may suggest that temporary interaction hints are not comprehended in a task context if they do not coincide with a need for the interaction in question. These projects highlight the potential of improving discoverability of interaction through visual signifiers in the interface while demonstrating that there are limitations of when visual signifiers impact the perceived affordance of interaction possibilities.

Résumé

La découvrabilité des interactions est un élément essentiel d'une expérience utilisateur réussie, mais elle est presque systématiquement ignorée par les concepteurs d'interfaces et les chercheurs. Cela n'est pas surprenant compte tenu de la diversité des concepts associés qui sont définis de manière inconsistante et du manque de clarté qui en résulte sur la manière d'appréhender ces problèmes. Dans ma thèse, je souligne l'importance de la découvrabilité des interactions et j'étudie si le fait de signifier visuellement les entrées utilisateur disponibles améliore leur découverte, utilisant les appareils tactiles commerciaux de type smartphone comme plateformes d'étude.

J'examine l'utilisation et les définitions de la découvrabilité dans la littérature ainsi que les concepts associés, et propose des clarifications là où les définitions sont ambiguës tout en soulignant les domaines d'intérêt et les limites. Pour examiner les circonstances menant à ces éventuelles découvertes d'interaction de manière plus structurelle, je propose ensuite un cadre conceptuel de facteurs susceptibles de jouer un rôle déterminant. J'examine les interdépendances éventuelles entre ces facteurs et réfléchis à la manière dont ils peuvent être utilisés pour promouvoir la découvrabilité des interactions. En me concentrant sur le facteur « visibilité » du côté système du framework, j'examine le potentiel de signifiants visuels supplémentaires dans les interfaces tactiles mobiles à travers deux projets.

Tout d'abord, j'examine comment la disponibilité d'entrées utilisateur tactiles à un doigt telles que « Tap », « Dwell », « Double Tap » et « Force Press » peut être communiquée via des signifiants visuels ajoutés au représentations graphiques de boutons. Je propose un espace de conception de caractéristiques pour signifiants visuels, et génère 36 modèles combinant ces caractéristiques que je compare dans une enquête en ligne et une expérience interactive en laboratoire. Les résultats suggèrent que les signifiants visuels augmentent la perception des possibilités d'entrées utilisateur au-delà du « Tap » et réduisent l'effort mental perçu par les participants, qui préfèrent également des signifiants visuels supplémentaires à un représentation graphique basique de bouton. Ce travail explique comment les futures interfaces tactiles pourraient être conçues pour mieux communiquer les entrées utilisateur tactiles avec un seul doigt. Deuxièmement, j'explore la viabilité d'utiliser les transitions animées entre différentes vues de l'interface utilisateur pour suggérer l'existence et faciliter la découverte de "Swhidgets", des composants cachés révélés par balayage. Pour ce faire, j'adapte trois modèles de transition établis pour qu'elles affichent subtilement ces swhidgets. J'étudie ensuite à quel point ces transitions sont perceptibles à différentes durées (750 ms, 1000 ms). Les résultats suggèrent une variation substantielle de la visibilité entre les participants. Par la suite, j'explore l'impact sur la découvrabilité des interactions avec deux types de transition (Container Transform, Panels) et une animation de référence, et constate que les transitions n'ont pas d'impact significatif sur la découvrabilité des swhidgets. Cela peut suggérer que l'affichage temporaire des swhidgets lors de l'animation de transition n'est pas compris dans le contexte d'une tâche s'il ne coïncide pas avec un besoin pour l'interaction en question. Somme toute, ces projets mettent en évidence le potentiel d'amélioration de la découvrabilité de l'interaction grâce aux signifiants visuels dans l'interface tout en démontrant qu'il existe des limites quant au moment auquel les signifiants visuels peuvent avoir un impact sur la perception des possibilités d'interaction.

Research Ethics

The studies performed as part of this thesis involved human participants. In accordance with guidelines, the studies were only performed once approval was given by the responsible Ethics Board.

Studies ran in France were approved by the Inria Comité opérationnel d'évaluation des risques légaux et éthiques (Coerle). Studies ran in Canada were approved by the University of Toronto's Research Ethics Board.

All participants were informed of their right to withdraw and signed consent forms before any data, demographic information, or survey responses were collected.

Acknowledgements

This thesis would not have been possible if not for all the support I received.

First and foremost I want to thank my supervisors Sylvain Malacria and Géry Casiez for their guidance and support. I am incredibly grateful for all the opportunities you have provided me with and have truly enjoyed our discussions and learning from you. It has been a pleasure to be supervised by you.
Thank you also to the LOKI team and especially the 'starter', your company and help throughout the last three years has been invaluable. Thanks to Lucas Plé for his work on the touch affordance project.

I would like to thank Fanny Chevalier for temporarily co-supervising and hosting me at dgp at the University of Toronto, it was truly a pleasure to work with and learn from you. Thanks to the people at dgp for welcoming me so warmly, Book, Mian, Kinjal and everyone who became a part of Thursday lunches.

Thanks to everyone who donated their time to me by participating in my user studies - I am incredibly grateful. A big thanks also to Emilie and Mengfei for their moral support and help with recruiting participants (Mengfei - also thanks for the pictures of Euler, they brighten up my day).

Thanks to Joshua Leung, for so graciously sharing his thesis Latex template and saving me a lot of time.

Finally thanks to all my friends and family for their unwavering love and support. Danke Mama, Papa und Anna, ich bin unendlich dankbar euch zu haben. Thanks to Aïa for letting me rant and providing some much needed distractions from work. And to Ash and Aqib for keeping me sane while we were all locked up in our rooms during the pandemic. Finally, thanks to Rowen for always believing in me, I know you would be so proud.

Contents

Abstract **i**

Research Ethics **iii**

Acknowledgements **iii**

Table of Contents **iv**

1 Introduction **1**
1.1 Overview 3

I Foundation of Theoretical Research of Discoverability **5**

2 Terminology **6**
2.1 The progression of initial interactions 7
2.2 Discovery in HCI 7
2.2.1 Surveying Discoverability 8
2.2.2 Defining Discoverability 11
2.2.3 Designing for Discoverability 11
2.2.4 Research Addressing Discoverability 14
2.2.5 Studying Discoverability 15
2.3 Related Interaction Concepts 16
2.3.1 Learnability 17
2.3.2 Communicability 19
2.3.3 Navigability 20
2.3.4 Findability 20
2.3.5 Guessability 21
2.3.6 Noticeability 22
2.3.7 Elucidating Differences - How Do These Concepts Compare 22
2.4 Conclusion 24
2.4.1 Importance of Clear Terminology for Discoverability 24

3 Framework of Impacting Factors **25**
3.1 Framework of Impacting Factors 25
3.1.1 System 26
3.1.2 User 30
3.1.3 Environment 36
3.2 Discussion 39
3.2.1 Interplay of actors 39
3.2.2 Boundaries of the framework 40
3.3 Conclusion 40

II Using Visual Signifiers to Improve the Discoverability of Interactions 41

4 Investigating Visual Signifier Characteristics Communicating Input Availability 43
4.1 Introduction 43
4.2 Related Work 44
4.2.1 Communicating Input Possibilities 45
4.2.2 Signifying interaction in HCI Research 46
4.3 What makes a signifier successful? 47
4.4 Design Space for Visual signifiers 47
4.4.1 Current signifiers in the design space 49
4.4.2 Filled Design Space 50
4.5 User studies 52
4.5.1 Study 1 - Online Survey 52
4.5.1.1 Methodology 52
4.5.1.2 Presenting DESIGNS 53
4.5.1.3 Procedure 53
4.5.1.4 Participants and apparatus 54
4.5.1.5 Experimental Design 54
4.5.2 Study 2 - Interactive Experiment 54
4.5.2.1 Methodology 55
4.5.2.2 Procedure 55
4.5.2.3 Participants and apparatus 56
4.5.2.4 Experimental Design 56
4.6 Results 56
4.6.1 RQ1 - Do added visual signifiers increase the perception of available inputs? 57
4.6.2 RQ2 - Which inputs are communicated by which design, and therefore what are the perceived affordances? 57
4.6.2.1 Study 1 - Online Survey 57
4.6.2.2 Study 2 - Interactive Experiment 58
4.6.3 RQ3 - Do the designs convey the intended inputs? 59
4.6.3.1 Study 1 - Online Survey 59
4.6.3.2 Study 2 - Interactive Experiment 59
4.6.3.2.1 Comparison to baseline 60
4.6.4 RQ4 - Does the success of signifiers depend on previous familiarity with the input in question? 60
4.6.5 RQ5 - Do signifiers reduce the perceived mental effort required? . . . 61
4.6.6 RQ6 - Do users prefer the presence of signifiers over standard widgets? 62
4.6.7 RQ7 - Does Explicitness, Timing and Proximity impact the success of a design? 62
4.6.7.1 Study 1 - Online Survey 62
4.6.7.2 Study 2 - Interactive Experiment 64
4.7 Discussion 65
4.7.1 Lessons learnt from our studies 65
4.7.2 Clarity of signifiers for similar interactions 67
4.7.3 Impact of physical interaction and personal differences 68
4.7.4 The impact of familiarity 68

4.7.5 Future Work 68
4.8 Conclusion 69

5 Incorporating Visual Signifiers in Existing User Interface Animations 70
5.1 Introduction 70
5.2 Related Work 72
5.2.1 Swipe-Revealed Hidden Widgets 72
5.2.2 Communicating 'swipe-ability' 74
5.2.3 Animated Transitions 75
5.3 Animated Transitions displaying transient interaction hints 76
5.3.1 Transition Type 77
5.3.2 Transition Duration 78
5.4 Study 1 - Noticeability 78
5.4.1 Experimental Design and Procedure 78
5.4.2 Participants 79
5.4.3 Data Coding 79
5.4.4 Results 80
5.4.5 Discussion 80
5.5 Study 2 - Discoverability 81
5.5.1 Experimental Design and Procedure 81
5.5.1.1 Swhidgets 82
5.5.1.2 Transitions 83
5.5.2 Participants 84
5.5.3 Results 84
5.5.3.1 Usage/Discovery Percentage 84
5.5.3.2 Timing 86
5.5.3.3 Transition Perception 87
5.6 Discussion 88
5.6.1 Are Transient Interaction Hints Noticeable? 88
5.6.2 Why doesn't Noticeability Translate to Improved Discoverability? . . 89
5.7 Conclusion 90

---

6 General Discussion and Perspectives 92
6.1 Summary of Contributions 92
6.1.1 Theoretical - Clarification and Exploration of Discoverability 92
6.1.2 Methodological 93
6.1.3 Practical 93
6.2 The Importance of Discoverability 94
6.3 Measuring Initial Interactions and Discoverability 95
6.3.1 Reflecting on Methods in this Thesis 96
6.4 The Potential of Visual Signifiers in the Interface 98
6.4.1 The Clutter Argument 98
6.4.2 Examining Factors Influencing the Success of Visual Signifiers 99
6.4.2.1 On the system side. 100
6.4.2.2 On the user side. 100
6.4.3 Limitations 101

6.5 Directions for Future Work 102
6.5.1 Studying Discoverability 102
6.5.2 Signifiers and the Perception of UI Elements 102
6.5.3 Noticeability and Discoverability 103
6.5.4 Visual Signifiers Providing Feedforward 104
6.6 Conclusion 104

Bibliography **123**

Appendices **125**

A II In-Place Single Finger Inputs **125**
A.1 Finite State Machine 125
A.2 Touch Interaction Categorisation 126
A.3 Table visualising study 1 responses 127
A.4 Table visualising study 2 responses 128
A.5 Figure visualising the input attempts in the interactive study 129

1 Introduction

In an increasingly complex technological world, we are perpetually confronted with new or updated systems. To successfully operate, this requires users to be able to continually detect and interpret what they are confronted with. This need is reflected in usability guidelines highlighting how important it is for users to be able to *"figure out what actions are possible [with a system] and where and how to perform them"* [167] or stating that *"I should be able to "get it" — what it is and how to use it — without expending any effort thinking about it"* [114]. According to these guidelines, systems should be designed to support the users initial recognition and comprehension of their different features and interaction methods.

Nonetheless, modern computing systems often fail to support their own discovery in favour of minimalist design aesthetics. The subsequent lack in discoverability is further exacerbated by the rapid introduction of novel devices and frequent updates to the functionality and interfaces of existing ones. While devices like smartphones, on the surface, may appear "uncluttered" and simple, they are simultaneously overloaded with controls to facilitate vast functionalities within a small interface. This creates a challenge for users, who are left alone to determine which inputs are available to them. This is exemplified by a New York Times article stating the majority of users were not using the iPhones 3D touch capacity, simply because they were unaware of it, highlighting *"it's not something you can really see [...] people don't understand it"* [76]. Subsequently, the 3D touch functionality was phased out and, in 2019, replaced with "haptic touch". This illustrates how novel interaction methods, that do not facilitate their own discovery, may fail to be adopted in commercial products.

However, even established inputs present a challenge, since touch screens technically *afford* all types of touch interaction on the entire screen [163]. This means users have to rely on previous knowledge and supplementary materials or on the visualisation on screen to communicate which type of input provides a meaningful interaction at a given location and time. In the absence of visual signifiers [166, 167] (depictions signalling what can be done) aiding the perception of input opportunities, it is entirely left to the users memory or guesswork to identify input availability. Given the large variety of inputs registered on touch screens, this creates an immense burden on users to continuously discover available interactions as interfaces change while also adding to the mental workload of recalling *which* input is supported *when* and *where*. Despite this, both research and commercial products almost systematically disregard discoverability when introducing novel systems and interactions or update existing ones.

The fundamental thesis of this dissertation is that this disregard should be rectified. In an effort to do so, we identify causes and explore potential resources to support discoverability. One possible reason for the lack of consideration for discoverability lies in the wide variety of technical terms used in this context such as affordance [106], learnability [82] and discoverability [167], which have frequently been defined in fuzzy ways. Although variances in terminology may appear trivial, as [87] highlights *"the terms we use for concepts are not inherently important, but the semantics behind the terminology commands our attention"*. By often being considered solely as a means to achieve learnability [66, 71], discoverability does not receive the attention nor the holistic investigation its pervasive effects should command. In an attempt to clarify this situation, we reviewed discoverability and its usage in the context of HCI. We then expanded our survey to incorporate related concepts and clarify and contrast the surveyed terms in order to elucidate and highlight their different focal points, which address specific interaction problems, as well as their limitations and relation to one another. In doing so, we provide an extensive literature review of discoverability, illustrating its many different application areas in HCI and, through highlighting the differentiation of *what* needs to be discovered, emphasize its importance.

Concurrent to the inconsistent terminology, the literature, depending on context, considered different human factors and system characteristics. We collected and reviewed these factors and considered them in the context of initial interactions, and discoverability in particular, by separating them into three main components of a conceptual framework, user, system and environment, that we then examined more in depth. Subsequently, we reflected on how individual factors and components interrelate and can be leveraged to foster successful discovery of interactions.

From this framework we isolate visual signifiers, whose absence we previously identified as a factor in the lack of interaction discoverability [76, 177], and focus on their possible implementation as they allow information to be conveyed in the place of, and concurrent to, the interaction opportunity meant to be discovered. We investigate the possible implementation of visual signifiers through two projects.

First, we examine the perception of *in-place touch inputs*, single finger contacts that do not move across the surface. Specifically, we focus on variance in duration (*Dwell*), frequency (*Double Tap*) and Force (*Force Press*). We propose a design space to investigate visual signifier characteristics and how they may influence the success of signifying different inputs. This design space encapsulates high-level principles that materialise in the dimensions *Explicitness*, *Timing* and *Proximity*. Through this design space we then generate 36 designs and investigate their perception in an online survey (N=32) and an interactive experiment (N=24). The results suggest that visual signifiers increase the perception of input possibilities beyond 'Tap', and reduce perceived mental effort for participants, who also prefer added visual signifiers over a baseline.

Given the common concern of clutter through added visualisation, our follow up project expands on temporary interaction signifiers by integrating them into animated transitions as transient interaction hints. To do so, we adapt existing interface transitions to temporarily reveal swipe revealed hidden widgets (swhidgets) that disappear as the final UI state of an animation is reached. We conduct two experiments to assess the viability of this approach. In a first study (N=60), we verify the noticeability of visual signifiers in animated transitions, for different transition durations and styles. In a second study (N=33), we assess the discoverability of swhidgets when they are exposed to users using our animated transition approach. Contrary to the results of the previous project as well as the results of the noticeability study, the results of this study suggest that the visual signifiers embedded into the transitions did not improve discoverability of the interaction. This suggests that, while visual signifiers are a valuable tool in conveying the availability of inputs, they are not universally applicable.

Finally, we discuss the results of our different approaches in conjunction and examine them in the context of the established theoretical foundations. We reflect on the importance of discoverability and the difficulties inherent to it's examination before highlighting the potential of visual signifiers. Further, we examine this potential by considering our different signifier implementations through the timeline of initial interactions and framework of impacting factors we previously introduced. We then provide a perspective of future research directions building on this thesis.

1.1 Overview

The core thesis of this dissertation is that interaction discoverability can and *should* be supported in interactive systems, for which visual signifiers provide a valuable resource which is critically underutilised.

The projects pursued in this dissertation contribute to this end goal in various ways, as summarised in Figure 1.1.

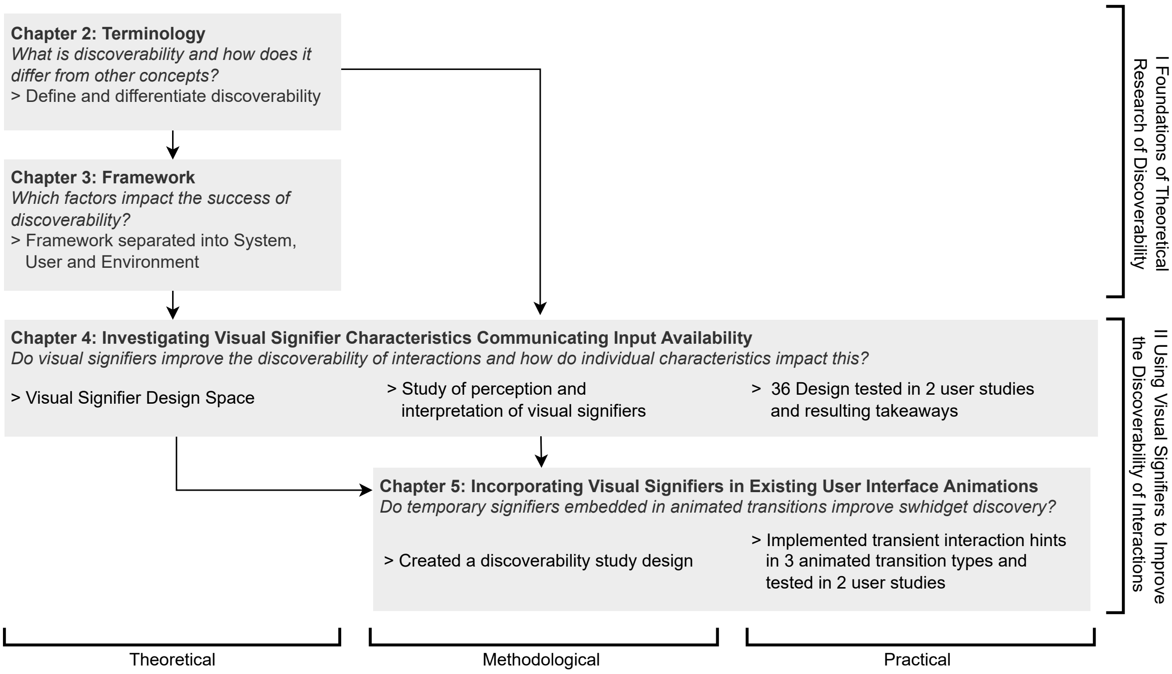


Figure 1.1 *– Overview of chapters summarising their theoretical, methodological and practical contributions to this thesis.*

This thesis is divided into 2 parts: I) Foundation of Theoretical Research of Discoverability; and II) Using Visual Signifiers to Improve the Discoverability of Interactions.

Foundation of Theoretical Research of Discoverability provides an overview and analysis of prior and related work. In chapter 2 we analyse the terminology and clarify "Discoverability" before highlighting how it relates to and differs from similar concepts such as learnability and findability. We then briefly touch on recommendations for its design and how it has been studied. In chapter 3 we offer a framework of related factors impacting the likelihood of discovery separated into System, User and Environment before considering the interplay between these actors.

Using Visual Signifiers to Improve the Discoverability of Interactions covers two approaches investigating the viability of improving interaction discoverability through increased visual signifiers in the interface. In chapter 4 we first propose a design space of visual signifier characteristics. We then combine those characteristics to create signifier designs for single finger in-place input. In chapter 5 we then include temporary visual signifiers in animated transitions between screens to communicate the availability of swipe-revealed hidden widgets. Both approaches are examined through 2 user studies respectively.

Finally, *General Discussion and Perspectives* provides a discussion of the two approaches and their contrasting findings as well as examines potential causes and avenues for future research.

Part I

Foundation of Theoretical Research of Discoverability

2 Terminology

While both commercial products and HCI research perpetually propose novel interaction devices, techniques, applications and functionalities, this rarely includes consideration of how their future users would know how to utilize them. Instead, research focuses primarily on proposing novel interactions methods or the feasibility of reliable interaction recognition on the system side [85, 123, 190] rather than the user's awareness and comprehension of the technique itself. While such considerations may not be the priority of research publications, the introduction of novel interaction methods without any regard of how the user would become aware of them, disregards an important factor of how this interaction would work outside of a laboratory setting. Although this consideration may be most pertinent in products being introduced to a large audience, it should be made at all stages of development. This does not necessarily imply an all encompassing need for design changes, but a need for deliberate examinations of the design and how its functionality is conveyed to intended users.

However, this need is difficult to address, as there is very little knowledge of how initial interaction [1] with these novel systems, input techniques and functionalities should be facilitated. This is exacerbated by the wide variety of technical terms used in this context which have been used and defined in inconsistent and overlapping ways, such as affordance [106], learnability [82] and discoverability [167]. While learnability and affordances have been clarified through extensive surveys [82, 145] their inconsistent usage persists. Additionally, the variety of terminology complicates the collection of relevant previous research. This illustrates the importance of semantics and terminology, as Hartson [87] highlights *"the terms we use for concepts are not inherently important, but the semantics behind the terminology commands our attention"*.

The term learnability has been investigated in depth [82], however its considerations often focus on specific aspects related to performance while also covering a timeframe beyond initial interactions. Similarly, discoverability has been primarily investigated in very narrow contexts such as public displays or voice and gesture user interfaces as well as in conjunction with learnability. By often being considered solely as a means to achieve learnability [66,71], discoverability does not receive the attention nor the holistic investigation its pervasive effects should command.

Given this lack of clarity, this chapter reviews discoverability and its usage in the context of HCI in an attempt to elucidate its differing definitions and highlight its application areas. We then expand our survey to incorporate other concepts concerned with the initial stages of first-time interactions and find various terms are used to describe related interaction phenomena. To then clarify these terms, which often have ambiguous usage or definitions, we analyse how they have been utilised. We then contrast the surveyed terms in order to elucidate and highlight their different focal points, which address specific interaction problems, as well as their limitations and relation to one another.

In doing so, this chapter provides an extensive literature review of discoverability, illustrat-

1. In the remaining of the document, we may use the term *"initial interaction"* alone to refer to the first time a user successfully interacts with a novel system, input technique or functionality.

ing its many different application areas in HCI and, through highlighting the differentiation of *what* needs to be discovered, emphasises its importance. We offer a clear and comprehensive definition. Further, we highlight the variance of semantically related terms that have been used interchangeably, thus advocating for their explicitly separate consideration. Consequently, we shed light on how the importance of supporting discovery is a well known problem statement which may suffer a lack of recognition due to a lack of clarity which we aim to provide.

2.1 The progression of initial interactions

First time interactions with something that is novel to a user follows a given path. We present a basic timeline in figure 2.1 in order to clearly communicate which part of an interaction we refer to in later sections.

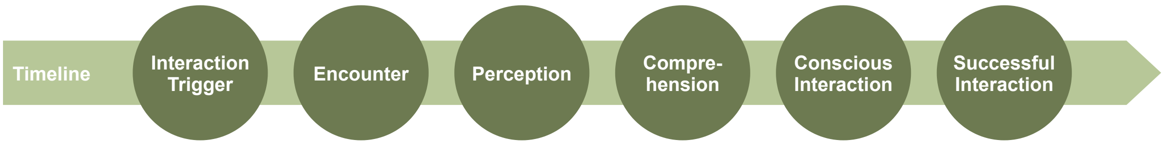


Figure 2.1 *– Conceptual timeline of the different stages of initial interaction, moving from before the interaction is initiated to successful interaction.*

First, an interaction needs to be triggered. This may happen through various factors, for example internally through a goal to interact, curiosity, happenstance or errors, as well as externally through examples, word of mouth or instructions. For example, seeing someone else use a feature might trigger a user to seek out said feature for themselves, or a user might accidentally encounter new interaction methods by failing to perform another one correctly (i.e. trying to tap a button but lingering to long and performing a long-press instead). As the final example suggests, for some interaction triggers the encounter with an object happens simultaneously while for others, like the first example, an encounter follows the interaction trigger. This is then followed by perception which depends on the sensory affordance [87] of the object in question as well as considerations of attention and awareness. If the user perceives the object they then process their impression [32]. Given they arrive at an accurate conclusion of what they are perceiving , they *comprehend* what they are interacting with. This then allows them to consciously interact with something and, depending on the difficulty of the interaction, immediately or eventually successfully interact.

2.2 Discovery in HCI

The initial discovery of interaction possibilities is the basis for allowing users to engage with and utilise interactive technology. Such discoveries rely, before all else, on the potential user's ability to perceive relevant parts of the system, comprehend them as such and interpret their functionality correctly.

However, initial perception and comprehension of systems and interactivity is inhibited by progressively more complex technology, which allows for a wide variety of functions and different input methods. This is exacerbated as visual guidance is reduced to avoid cluttering interfaces [177, 178]. Consequently, initial, intuitive perception and understanding of interaction opportunities and features is progressively harder to achieve. The resulting lack of comprehension of, or complete obliviousness to, the tools available to a potential user can lead to either complete disregard of a system or missed opportunities to use a system to its full potential [9, 98]. This creates a significant issue, not only for designers wanting their system to be used, but also for potential users that would benefit from, or are required to interact with, a system they fail to recognize or comprehend.
In an effort to investigate the persistence of discovery problems and possible solutions, we survey the concept of *discoverability*, its application areas and proposed definitions.

2.2.1 Surveying Discoverability

Definitions and application of the term *discoverability* vary, but generally describe users encountering something new that they were not previously aware of, therefore *discovering* it. While the concept is also widely used in the context of content discovery within search engines, databases and library systems [133, 146] our focus is on discoverability that facilitates and improves interactions. In the context of usability and human-computer interaction the term discoverability was popularized by Don Norman [167], who describes it through the following question:

> *"Is it possible to even figure out what actions are possible and where and how to perform them?"* [167, p.3]

However, it is ambiguous what this definition refers to exactly. Indeed, in the context of HCI, discoverability can be split into three different focus areas: System, Interaction and Feature. System discoverability is focused on whether potential users notice the overall system and recognize it as something they can interact with, for instance, that a public display is interactive in a mall. Interaction discoverability is concerned with whether users can identify and comprehend interaction and input methods when they encounter them without instruction or guidance, for example that a force press can be performed on an iOS icon to reveal a menu. Finally, feature discoverability covers the user's ability to discover features or functionality they were previously unaware of while using a system, for example a new tool in a graphics editor like Adobe Photoshop. While this separation between different discoverabilities has been implicit with research specifying a focus on interaction discoverability [36] or feature discoverability [198], these focus areas have not been collated and considered in relation to one another. This is unfortunate given that, even though they are separate considerations, these different discoverabilities are interconnected. System discoverability is a precondition for users to discover different aspects of said system. Similarly, features can only be accessed and discovered if the user is able to successfully interact with the system by being able to discover the interaction methods available to them. While features may be accessible through a variety of input methods, often times novel input methods also reveal features linked to them exclusively, for instance the shake-to-undo on iOS where shaking the device used to be the only way to activate system-level undo implying that shake gestures cannot be used for anything else. The clarification of *what* needs to be discovered is essential when examining the connection of different discoveries since they may not always be facilitated through the same measures. The following sections consider the different focal points of discoverability in more detail.

System Discoverability

In the context of complete systems, the term discoverability is often used to describe the user's overall awareness of a system's existence and interactivity. As such it is most commonly considered in the context of public displays aiming to attract users. Cheung [36] describes the discoverability process for large public displays as *"the process through which a user transitions from being unaware of the system to interacting with it to, eventually, effectively using the system"*, with the particular aim to focus on the earlier stages of display interaction. Their work focuses on capturing attention and consequently guiding interactions [38], appropriate methodology to study attraction power of interactive public displays [39] and ultimately how to increase casual engagement with large, public interactive displays [37]. Other examples of system discoverability investigations include public polling interfaces [91], multiplayer games [46] and the transition between implicit and explicit interaction [217].
Similar research has been conducted without explicitly referencing discoverability, including investigations of *"how passers-by notice the interactivity of public displays"* [152] by evaluating how visual feedback to incidental movements by potential users in front of the display effectively communicates interactivity and invites interaction.
Once investigations of discoverability go past initially attracting users, getting their attention and making them aware of the system, research can move into the territory of interaction discoverability as designers want potential users to understand how to interact with their system.

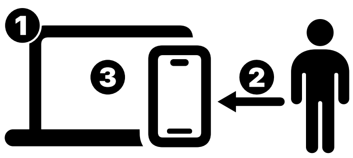
Interaction Discoverability

Interaction discoverability describes the ability of users to notice and identify interaction possibilities they were previously unaware of. As such, the focus of interaction discoverability considerations is often on overall novel interfaces such as Voice User Interfaces (VUIs) [71,109,132,181] or Embodied Interactions [29,30,75]. Other examples are interactions with, among others, tabletops [86], mid-air gestures [219], interactive touch surfaces [77–79], VR objects [154] or smart textiles [171]. Notably, both VUIs and embodied interactions commonly lack visual signifiers suggesting the possible interactions to new users. These examples indicate that the discoverability of input methods is primarily studied if researchers or designers, proposing a new method, already presume discoverability to be an issue impeding successful interaction. However, interactions in traditional interfaces that users are not aware of, such as keyboard shortcuts on desktop, force-press on mobile and gestures on both, similarly present discoverability problems that designers commonly overlook. Further, the majority of interaction methods examined are yet to be established with a wide user base and therefore lack the advantage of similar interactions with other devices aiding the discoverability of the input methods in question. Meanwhile the discoverability of interactions which may be established but novel to individual users in a given context is generally ignored. As public displays also commonly employ less established interaction techniques than devices designed for personal use, research about their design often includes considerations of how to communicate their interaction possibilities and functionalities. In this context, a lack of interaction discoverability has also been termed *interaction blindness* [170], while systems with well supported system and interaction discoverability have been termed walk-up-and-use systems [49, 141], implying the recognition and comprehension of interaction to be inherent to the system.

Feature Discoverability

Feature discoverability describes the ability of users to discover individual parts and features of a system they were previously unaware of.
Reports [83,207] suggest that only around 20% of features within applications are commonly used. While this can partially be attributed to some features not being of any interest or use to some of the users, other features are simply never discovered.
This discovery is often times hindered by the fact that the system is functional and the user is able to perform a majority of their intended actions without being aware of all features at their disposal. This can especially be the case if users are not encouraged to explore further options because, what a new feature offers, can be achieved in another, potentially more strenuous, way that the user is already aware of. On the other hand, if users are not at all aware of the existence of an available function in the system they are interacting with, their lack of expectation and familiarity may lead to them overlooking potential signifiers. Other examples of features suffering discoverability problems are items hidden deep in the hierarchical menus [3] which are not explored by users or features that are simply overlooked because they are not essential to the user.
A significant number of research in this category considers issues in software development [52,66,139,179,211], while others consider discoverability of applications [95] or menus [197] in public displays. Additionally, the discoverability of existing and new functionality for operating systems [176] and applications [147] is explored.
Notably, some literature explores "tool" discovery [155, 156, 158] which, following our separation fits into *feature* discoverability, as it does not consider the discovery of an interaction per se. However, while tools do not change the method through which users interact with the system, they largely present the discovery of new actions available within a system, which in some facets resembles interaction discovery.
Furthermore, feature discoverability is considered without explicit reference to the concept, for example, by Cockburn et al. [43]. They consider *vocabulary extension* as one domain of performance improvement and describe it as a consideration of ways to help users broaden their knowledge of the range of available functions in an interface, which would subsequently ease feature discoverability when successfully implemented.

To summarize, discoverability is always concerned with the user becoming aware of and comprehending something. *What* a user can or should discover can be separated into 3 categories:



1 System The discovery of interactive systems overall.
2 Interaction The discovery of actions a user can perform.
3 Feature The discovery of actions a system can perform.

2.2.2 Defining Discoverability

While considering the same notion and process, all of the "discoverabilities" identified above are focused on different aspects of an interaction. This illustrates how varied the understanding, of what discoverability describes, is. Despite these discrepancies, discoverability is rarely specified or defined.

The definitions that are offered vary widely in scope and specificity. While some offer loose explanations of discoverability such as *"the effort required to locate or "discover" an application amongst others"* [95] or *"the ability for users to find and execute features through a user interface"* [109], others define discoverability specifically for their context like noting how *"the "invisible" nature of VUIs can challenge users' ability to discover its capabilities and limitations"* [71] or how gestural interfaces *"are not discoverable in the sense that it is hard to know what gesture to do and we often cannot look it up in the way we would look at a menu to see what options are on it"* [75].

Two comprehensive definitions that inform our understanding of discoverability come from Cardello [33] and Srinivasan et al [204]. Cardello described discoverability as users encountering *"new content or functionality that they were not aware of previously"*. Srinivasan et al. consider discoverability in the context of speech input commands and make the important distinction of what discoverability entails in their interpretation: *"Discoverability, in this context, entails: (1)* awareness *— making users aware of the operations that can be performed using speech; and (2)* understanding *— educating users on how requests should be phrased so the system can interpret them correctly"* [204].

Combining these approaches, we consider discoverability as:

> The ability for users to perceive and comprehend a system, function or input method as such when encountering it for the first time despite a lack of previous awareness or knowledge. This may be through intentional effort or serendipitously.

As such, discoverability covers the time period from interaction being triggered, the user encountering and perceiving the system, function or input method to the user comprehending what they have encountered.

2.2.3 Designing for Discoverability

The varying definitions of the term have led to a lack of clarity on how to design for discoverability and even a discussion on whether designing for discoverability is possible or desirable.
Furqan et al. [71] look at improving design for the discoverability of VUIs by surveying previous projects with the same aim. Notably all surveyed projects resort to converging purely voice-based interfaces with visual assistance to improve discoverability, exemplifying the challenge of discovering interactivity that is not visually communicated. They identify the characteristics *"Visualizing what can be done"*, *"Visualizing what can be said"*, *"Visualizing New Discoveries"* and *"Contextual Relevance"*. Notably the differentiation of *"what can be done"* and *"what can be said"* mirrors the discovery of features and interactions while *"visualizing new discoveries"* and *"contextual relevance"* can be seen as general principles when designing for discoverability in VUIs.

The *VoiceNavigator* [45] included in this review, revealed an established discoverability problem that accompanies the benefits of visualization - clutter. When the display showed too many supposedly helpful prompts, these discovery prompts fought for the users' attention and caused confusion and frustration rather than easing the interaction. This problem is also discussed by Berkun [21] who opines that discoverability overall is a myth by arguing that if everything is discoverable - nothing is. Maybe speaking hyperbolically when making this initial argument he then goes on to discuss the problem statement of clutter and essentially makes the argument to prioritize discoverability of important and essential features over others.

This reasoning suggests not the absence or non-viability of discoverability but more so the importance of a *hierarchy of discovery* suggesting a prioritization according to how commonly things are used by how many people. He notes that there are exceptions to such hierarchies like essential tasks not often needed but enabling overall use or safety and emergency features that, while hopefully not required, should be immediately discoverable and accessible should the need arise. Although prioritization is an important guideline, it should be noted that the one offered here is oversimplified as many tasks can be achieved in multiple ways. Similarly, if a newly introduced feature suffers from bad discoverability, it may be used by very few people simply because potential users are not aware of the option. While Berkun's considerations are certainly relevant and inform *what* should be prioritized for discovery, they only graze the question of *how* to make it discoverable by providing a list of resources that can be utilized to "emphasize" things.

Seto et al. [197] note the scarcity of research focusing directly on designing for discoverability in public displays. They reviewed existing literature and found that written instructions near the display aided successful discovery (of the system as well as available interactions and features) while also considering that ambiguity in design can entice users to engage by appealing to their natural curiosity. Gradual discovery, through scaffolding of continuously revealed elements, was identified as a way not to overwhelm users and therefore ease discoverability [101,226]. This finding again supports the notion that clutter or too much visual representation at once does not benefit discovery. Additionally, they found that subtle animation can be helpful to suggest to users that objects can be manipulated in a certain way, noting that touch gestures particularly suffer from bad discoverability due to a lack of visible controls affording manipulation [94]. Notably, such subtle animation has since been implemented for Instagram Reels (Version 285.0.0.25.62), where the possibility to swipe up to see the next video is occasionally suggested by a small animation of the next video moving in from the bottom. Seto et al. confirm the statement that animation can be helpful to suggest interactions in their own experiment and add the design recommendations, that interface elements be discernible by utilizing universal GUI elements, and giving immediate feedback to exploratory interactions.

Their consideration of written instructions touches on the use of external sources supporting discovery. Especially complex systems designed for expert users are difficult to make independently discoverable. Therefore, designers circumvent having everything discoverable by itself through the use of instructive material like manuals. This creates different problems due to the non-use of manuals [22]. There are attempts to rectify this issue, while utilizing the benefits of formal instructions, by incorporating instructive material in the system. One way to do so while avoiding permanent changes is the temporary inclusion of UI elements focused on aiding understanding. An example of this is the discontinued Microsoft Office Assistant attempt "Clippy" that offered context-related tips. Joyce [104] describes this as onboarding, defining it as *"the process of getting users familiar with a new interface, using dedicated flows and UI elements that are not part of the regular app interface"*. They do, however, highlight, that *"instructional onboarding should not be used to supplement poor design"*. While these methods of supplementary material can be useful and are a necessary consideration in novel user interfaces, inherent discoverability should not be disregarded in favour of defaulting to discovery support through explicit instructions of how to use a system. Rather designers should aim to make the system *communicate* its interaction possibilities independently wherever feasible.

Investigating this, Mayer et al. [144] examined how novel input techniques for smartphones can best be communicated to users and identified *depiction* (visualizing input techniques through icons), *pop-up* (modal dialog shown when the technique is available) and *tutorial* (centralized explanation of all available input techniques). Their study revealed that depiction was preferred, however they comment that which communication method would be most appropriate depends on the context of the interaction. Nonetheless this further supports the notion that instructive material is not a suitable, or popular, replacement of visual signifiers supporting discovery.

For their part, Lambert et al. [118] investigated the understandability of different visual presentation of single-handed *microgestures* and extracted design guidelines on which features of such gestures should be represented. Notably, they observed which microgesture users interpreted from a visual presentation without any prior knowledge on the available gestures. Their conclusion is that in order to optimize understandability, the interface should not present only the trajectory of the microgesture, but also the *actuator* (the finger initiating the microgesture).

To assess how similar the above discussed academic recommendations are to practical applications, we reviewed tips and guidelines provided for interaction, usability and UX design to support discoverability. We limited our search of non-academic guidelines to immediate search engine results for the terms "designing for discoverability" and simply "discoverability" to identify practical advice that was commonly available and most likely to be accessed, read and utilized widely. Through this we found and compared 4 articles advising readers, two of which were quite detailed with 15 [227] and 12 [17] tips respectively while the other two took more direct, feature focused, approaches and in doing so offered 5 [188] and 3 [210] recommendations. The 2 more extensive articles had 5 overlapping recommendations, one of which was mirrored by another of the shorter articles. They are as follows:

- **Design familiar interfaces:** Follow universally accepted standards and design conventions to ease recognition for the user.
- **Group elements logically:** Items with strong relationships or similar functions should be grouped together.
- **Simplify:** Reduce the number of available options to prioritize important content over superfluous extras.
- **Consider size:** Size interface elements appropriately to direct attention towards key information. Both [227] and [188] note that bigger elements are easier to discover.
- **Provide visual cues:** Guide users towards what to do and see next.

These, as well as other parts of the full articles, have significant overlap with the previously considered findings. Examples of recommendations or challenges identified are clutter [17,21,45], use of hints and tips [104,227], animation [17,197] and prioritization of what to emphasize [17,21]. Notably, some of these guidelines, such as a hierarchy of prioritization or grouping and simplifying options, mainly apply to feature discoverability.

2.2.4 Research Addressing Discoverability

The HCI research community frequently proposes novel features and ways of interacting without considering how users would discover this functionality (e.g. [11,16,80,85,123,189, 190, 231]). However, rare exceptions exist of authors who consider how to alleviate this problem, although not always with explicit reference to the concept of discoverability. These considerations often take the form of either attempting to make proposed novel interaction methods discoverable from the beginning [12,13,69,78,219] or improving the discoverability of existing features and input methods [78,128,142,143,197]. Additionally, existing designs are compared and evaluated [71,105,197], modes of discovery are explore [157], adapted design procedures are suggested [29] and study methodologies proposed [37,39] in an effort to improve discoverability.
Notably, previous work has examined the importance of social surroundings on the general discoverability of features and tools [107,156,157]. While some focus on interviews with experts in specific domains (designers [107], programmers [156]), other combine interviews with diary studies [157] to explore how users discover tools. Murphy-Hill and Murphy [156] specifically put aside technical solutions to explore social components of discovery and found that peer interaction is an effective, yet infrequent source of discovery. In contrast, Kiani et al. [107] explored how users learn feature-rich software and found that interpersonal interactions still constituted a preferred and commonly used method of discovering, and learning about, features. Finally, Murphy-Hill et al.'s [157] focus is exclusively on serendipitous discovery, for which they found tool encounters by happenstance to be the most common cause of discovery. Additionally, they report discovery through peer observation and recommendations, tutorials, written descriptions and social media. Nonetheless, their investigation and categorisation of discovery causes is limited due to their primary focus on serendipitous discoveries through peer interaction. Subsequently, the inferences from their work are based on improved tool discovery through peer interaction and recommender systems rather than discovery supported by the system itself.
However, the majority of considerations of discoverability are focused on the discoverability of specific inputs, features or systems. For example, Appert et al. [12,13] introduced the novel undo mechanism *Dwell-and-Spring* and considered whether users would discover the technique. They found a lower discovery rate than expected and concluded that there may be a trade off between the stronger feedforward required to improve discovery and the design aim of making the mechanism easy to discard. Fennedy et al. [69] attempted to adapt hotkey use for touch-based devices by testing different designs and found their designs were discoverable and yielded efficient performance. Walter et al. [219] explored how initial mid-air gestures on public displays may best be revealed to potential users, fostering their discovery. They found that *temporal division*, in which the running application is interrupted to show the gesture on the full screen was most successful and noted that the discovery of gestures was aided by imitation of other users and competition within a group. While these explorations focus on the discoverability of interactions they themselves intro-

duce, other research focuses on improving existing designs. Goguey et al. [78] aim to improve the discoverability of force sensitive text selection and introduce techniques that provide improved feedback through visualization. Although they compare the discoverability of their designs to the normal iOS design, they informed participants beforehand that the touchscreen had force sensing capabilities and found that all overall techniques were discovered by participants while parts of them remained unclear to some. Matejka et al. [142] created *Patina*, a system visualizing usage data in order to replicate usage clues through wear and tear in physical objects. One of the benefits of the displayed usage heatmaps they consider is *"exposing functionality relevant to a specific document"*, which in turn eases the discovery of these functionalities. While a short-term internal deployment had users reporting increased exploration out of curiosity what others were using, the limitations of the study make it unclear whether these effects would persist. Nonetheless the study results, similarly to Walter et al. [219], point towards a social component of discoverability. Dong et al. [63] built a puzzle game in an effort to improve learning through discovery-based games. The design aimed to incentivize the users independent discovery of different features within Photoshop through gamification. Furthermore, command recommendation systems have been explored [128, 143] which suggest unused or novel commands relevant to the user or task in an attempt to improve and foster the discoverability of these commands. Finally, Cafaro et al. [29] consider how design methodology can be optimized to elicit discoverable full body interaction movements.

Additionally, some studies include partial considerations of discoverability. Bonnet et al. [23] consider whether different feedback successfully communicates the success of the interaction but do not expand on or study whether this feedback and feedforward aids in the discovery of the methods. Goguey et al. [77] claim the design of their technique improved discoverability, but do not include a study confirming that assertion. Agrawala et al. [2] list discoverability as one of their goals and gave participants time for independent discovery within their study before being introduced to the remaining functionality that was not discovered. They report *"encouraging"* results that participants were able to discover functionality on their own but do not share any concrete data regarding discovery.

While these considerations confirm the continued consideration and relevance of discoverability, they are extremely rare in comparison to the number of novel interaction techniques and systems proposed every year, not to mention persisting discoverability problems in existing systems.

2.2.5 Studying Discoverability

Among the papers that do aim to examine discoverability, methodology varies significantly, and, often-times the study design itself impacts the discoverability it aims to measure. Discoverability studies often focus around comparing the discoverability of different conditions [78, 197, 219]. They are conducted as observational studies of publicly deployed prototypes [197, 219], within tasks performed in laboratory settings [78] or on online crowdworking platforms [69] as well as self-reported discovery and perceived discoverability in surveys [177]. These studies, at times, include results that show both the difficulty of achieving as well as measuring discoverability, with [78] highlighting that no participants discovered a particular feature within an interaction method but *"all participants suggested during the interview, however, that they would have stumbled upon these features and eventually figure* *them out"*. This, as well as the variance in approaches highlights the difficulty of studying discoverability, with some studies including only vague observational data [2, 78], priming the participants towards discovery by giving them tasks that reveal visual signifiers [12] or telling participants about the presence of the targeted interaction method [78]. These concessions, which are likely made in an effort to gather relevant data, however do mean that some studies claiming to investigate discoverability, fail to do so.

2.3 Related Interaction Concepts

Given the prevalence of difficulties with initial interactions, but also the lack of clear characterisation, significant research has been conducted to examine, ease and improve the beginning phases of interaction under many terms. Like discoverability, these terms have been used with varying definitions and interpretations, resulting in differing interaction concepts being used to describe overlapping problem statements. Their close semantic relation and them being applied to similar interaction contexts, have further led to confusion, with these terms being used in conjunction or even interchangeably [71, 109, 149]. Consequently, which term is being used to describe a specific problem may vary between researchers, making a comprehensive look at the problem statement, and research addressing it, difficult. As Allen et al. [6] point out when considering *"intuitive"*, which is yet another word often used with inconsistent definitions when describing initial interaction goals, *"a rich and evocative word [...] is wasted as long as it sits in a fog of uncertain association"*. The following section will examine how various concepts, which are important to initial interactions and related to discovery, have been used and defined. Further it will highlight how their different focus points differentiate them from discoverability and one another and why their separate consideration matters.

The lack of consistency in how discoverability is defined and used as well as the overlap with concepts considered here led us to approach the problem through a scoping review in order to explore which concepts are used, why and how they differ, rather than a systematic review that would require a pre-existing clear terminology to gain meaningful insights [117, 180]. Therefore, in order to identify relevant concepts, we followed an exploratory approach, but limited this investigation to interaction research concepts that describe the users *ability* to interact with and comprehend a system initially. When examining semantically related words to the verb discover, *find*, *learn* and *notice* appear alongside multiple other synonyms. We highlight these terms as they are the ones commonly referred to as -abilities in HCI literature (i.e. findability, learnability, noticeability). Furthermore, both *learnability* and *findability* are often related to discoverability on UX websites [33, 188, 227]. *Navigability* appears in standard work [183] with a definition closely resembling that of discoverability. Additionally theses terms, as well as *guessability* and *communicability*, appear repeatedly in relevant literature in conjunction with discovery and the initial stages of interaction.

While this selection of concepts is broad and highlights how many approaches focus on different aspects of initial interactions, we make no claim to cover *all* relevant interaction concepts. Nonetheless, the following section covers the concepts most frequently overlapping with, or considered alongside, discoverability.

2.3.1 Learnability

In HCI the beginning phases of interaction have been primarily examined in the context of *learning* how to interact with the system. Consequently, learnability is the most commonly used term to describe the users' first understanding and subsequent improvement when using an interactive system.
The Grossman et al. [82] survey of learnability research found that, while there was an agreement among researchers that learnability is an important aspect of usability, there was a lack of consensus in the definition, evaluation and improvement of learnability in user interfaces. They grouped definitions into two main categories: *initial learnability* and *extended learnability*. The definitions falling under *initial learnability* concerning themselves with new users gaining proficiency with a system and *extended learnability* definitions expanding this, to incorporate long-term learning and mastering a system. A brief survey of literature related to learnability since then shows that definitions widely appear to still align with the categorisation established by Grossman et al., with some publications focusing on initial learnability [5,71,149] and others on extended learnability [4,68,126,140]. Some also directly cite this survey [68,71,126,140,149], suggesting that the compilation and clarification of learnability definitions has been embraced and utilized. The scope of these definitions has been discussed, with Marrella et al. [140] as well as Santos and Badre [192] arguing that the assessment of initial learnability, while crucial for the study of the system, at best represents a measure of the system's *intuitiveness*. Notably, Santos and Badre's considerations predate Grossman et al.'s survey. They use the term "initial learnability" to describe learnability for first-time users. Neither they nor Marrella and Catarci explicitly define intuitiveness. Nonetheless, the separation of learnability into *initial* and *extended* illustrates the broad use of the term in both time frame and scope.
When specifically considering initial interactions, initial learnability is the pertinent consideration and a part of the success of those interactions. Initial learnability is commonly used as a performance measure for interaction occurring for the first time. Literature examining it often concerns itself with users performing well in a task [5,71] and how initial usage may be easily facilitated and improved.
Notably, initial learnability often does not address the first identification of a system, function or input method as such and largely fails to address interactions happening without specific intent or aim.
Nonetheless, learnability is often mentioned in conjunction or synonymously with discoverability [71,109,149]. Many of the offered explanations of discoverability either allude to or directly reference *learnability* by referring to discoverability as *"a means to achieve learnability"* [71] or describing discoverability as a challenge one is confronted with then trying to *learn* a system [66]. Furthermore, definitions of discoverability as, for example, *"the ability with which a user can find features of the system to increase proficiency over time"* [179] resemble those offered for learnability. It is therefore no surprise that the differentiation of the terms can be ambiguous.

Differentiation between Discoverability and Learnability

Since the concepts of discoverability and learnability have often been used in conjunction or defined interchangeably, it is pertinent to highlight their differences to emphasize the importance of separate considerations of discoverability. To elucidate these differences, figure

2.2 depicts a timeline of initial interactions and where each concept applies.
As an example, a user we name Alice, who is taking pictures on her phone, tries to swat away a fly that keeps landing on it. In the process she twists the phone twice, triggering the "Switch Camera" function, changing from the main camera to the front facing-camera. The unintentional action *triggers* the interaction and leads to her first *encounter* with the function. Notified by the vibration of the phone, she realizes something happened and inspects her phone, noticing the camera view has switched - her first *perception* of the function. If she is able to deduce that the movement of her hand has triggered the function (*comprehension*) she has successfully discovered it. If she does not understand this connection, she finishes this interaction without discovering the feature. The actual interaction with or usage of the feature is not part of discoverability. Her ability to utilize this feature is covered by learnability. There is a separation between conscious and successful interaction since, depending on the complexity of the interaction, successful interaction may require skills beyond the simple understanding of the action. Initial learning covers how to first consciously and intentionally use the feature successfully, while extended learning covers perfecting its usage by incorporating it seamlessly into her usual interaction.

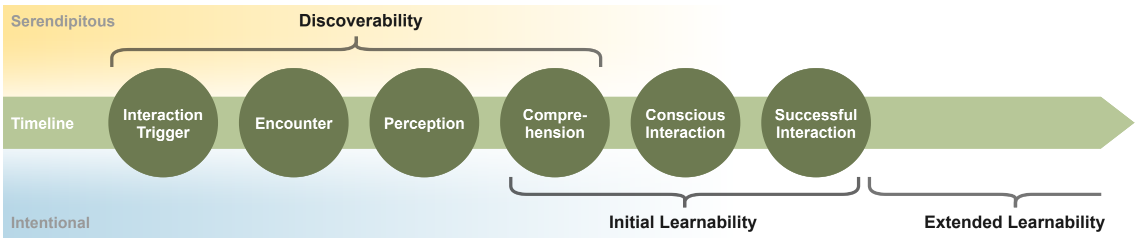


***Figure 2.2** – Timeline of progressing initial interaction. Both discoverability and learnability apply during this timeframe and influence the interaction's occurrence and success.*

When considering the abstract sequence in figure 2.2, it should be noted, that while it depicts one isolated interaction, discoveries and learning can happen continuously and concurrently as users encounter new interactivity within a system and during other interactions. Furthermore, although the diagram suggests a strict separation between serendipitous and intentional interaction, these commonly overlap. Discoverability and learnability are each more applicable to the circumstance they are placed in, but they can both occur serendipitously or intentionally.
To elucidate how this timeline applies in an intentional context, we again use Alice as an example user. After she discovered the "Switch Camera" function, she is now curious if there is other functionality she doesn't know about. The interaction overall is triggered by her curiosity rather than the previous accident. Through this the entire discovery phase falls into the category of *Conscious Interaction*. She encounters a "More settings" button and discovers the option to use gestures on the volume keys and double taps on the screen to optimize her camera usage. As she is consciously exploring and seeking out functionality, encounter and perception are likely to coincide and immediately followed by comprehension. The functionality is displayed (*encounter*) and as she is consciously exploring she immediately sees (*perceives*) it. In the case of functionality in menus (such as this example) this is most likely followed by immediate comprehension. While both examples here involve direct interaction, in other cases external interaction triggers such as examples may lead to discovery without any action from the user as they encounter functionality passively through someone else's action. If they still perceive the action and reaction and comprehend their connection

they may discover functionality without ever interacting.
Overall, discoverability precedes and triggers learning since, if a potential user encounters functionality but does not notice or comprehend it as such, they are unlikely to interact with it consciously or retain information that would allow improvement of the interaction. However, learnability exceeds discoverability in both scope and time frame as it continues long beyond the initial interaction. Even initial learnability, which is generally focused on the interactions of first-time users, succeeds the discovery phase by encompassing the user's ability to perform the interaction *successfully*. As Berkun notes, *"discoverability does not guarantee success"* [21]. He argues that simply being able to discover something, does not necessarily mean people can, or know how to, use it. For example, users being confronted with a keyboard and being able to understand that they can press the WASD keys to interact with a desktop game does not guarantee they will deduce how to use said controls successfully to play a game. Additionally, if the interaction through those keys in the game is challenging, the sole discovery of the interaction method does not facilitate successful usage. Thus, discoverability is not a measure of efficiency or improvement. In contrast, learnability is often considered in the context of performance. Accordingly, considerations of it generally assume the users know how to perform certain interactions even when encountering the system for the first time.
Consequently discoverability is a precondition to the learnability of interactions and features the user is unfamiliar with. In order to learn how to use something, one has to first become aware of and comprehend its existence.
When learnability is examined as a whole, this often incorporates discoverability. However the prevalence of interaction failures due to discovery problems shows that the two concepts should be considered independently before any conjoined examinations. Nonetheless both phenomena occur within a similar timeframe of initial interaction and occur successively. They may therefore be influenced by similar factors.

2.3.2 Communicability

Communicability as a concept within Human Computer Interaction originated in the context of Semiotic Engineering by Clarisse de Souza. De Souza et al. [51] describe communicability as *"the distinctive quality of interactive computer-based systems that communicate efficiently and effectively to users their underlying design intent and interactive principles"*. In an earlier publication Prates et al. [182] describe a communicability evaluation method aimed at *"measuring whether the software successfully conveys the users the designers' intentions and interactive principles"*. How such system communication is presented was investigated by [144] and is discussed in section 2.2.3.
In summary, communicability describes the system's ability to communicate its purpose and interactivity to the user. It presents a system perspective of the discovery and comprehension process but without actually focusing on whether or not users will perceive these principles.

Differentiation from Discoverability

Communicability, in contrast to discoverability presents a solely system based approach. As such it considers how a system communicates with the user but is not focused on a specific phase of interaction. Furthermore, discoverability includes interaction triggers in the environment, user characteristics as well as the context of interactions occurring which communicability, due to its system focus, does not.

2.3.3 Navigability

Due to the spatial connotation of the word "navigate", navigability has been used substantially in literature concerned with navigable media such as 3D environments and how to create them [47,218,220]. However, it has also been defined in broader terms. Preece et al. [183] describe navigability with the question *"Is it obvious what to do and where to go in an interface?"* while Takagi et al. [206] define navigability simply as *"Ease of navigation"* (of webpages). This discrepancy and vagueness present an issue when considering that a substantial amount of literature mentioning or focused on navigability does not offer a definition [88,120,121,200]. Given the differences in definition it is necessary to clarify that we consider navigability as the ability of a system to communicate to the user how to (optimally) move through it considering location and timing in the context of this work.
Whether a system successfully guides a user through which options are available to them at a given time and aids them in moving towards intended interactions, influences the success of initial interactions. This is especially due to the user's initial lack of independent knowledge of available content and what to do when.
Generally, navigation assumes a destination, leading to this term being primarily used when describing users with concrete intentions and purpose. Navigating without a concrete goal e.g. surveying in a leisurely and casual manner has been described as *browsing* (Oxford Languages) and relies on similar characteristics.

Differentiation from Discoverability

While there is no evidence of these concepts having been used interchangeably, the proximity and relation of their focus points make their conjoined consideration helpful to clarify how they differ. While discoverability includes considerations of awareness and how users notice interactions, navigability is focused on how to move through a system considering timing and location. Consequently good navigability can aid discoverability by easing access to different parts of a system and emphasizing those relevant to the interaction context. Notably navigability is especially relevant when considering the discovery of features.

2.3.4 Findability

In HCI findability was introduced in the context of finding individual websites on the internet by Morville [151] and has since been adapted to a much broader scope of describing *"the degree to which a particular object is easy to discover or locate"* [183] or *"how easy is it to navigate and locate objects in the system"* [169]. Cardello [33] defines findability as *"users can easily find content or functionality that they assume is present in a website"*.
For the understanding of findability in this context we adapt Cardello, to include other functionalities and usage contexts than websites, and consider findability as the ability to locate or come upon a system, feature or input method the user knows or assumes to exist within their current usage context through conscious effort. As such it requires the previous knowledge or assumption of the user that something they want to *find* is present somewhere in the system. It therefore presumes the user knows *what* they are looking for, and is concerned with how they know *where* to find it. As such, navigability enables findability by allowing

potential users to move through a system to locate what they are searching for.
Overall, the ability to locate a system, feature or input method greatly influences the user's ability to begin interacting with it since if they fail to do so, they never encounter the interaction in question.

Differentiation from Discoverability

Cardello [33] differentiates discoverability and findability through the users previous knowledge and awareness, defining findability as *"users can easily find content or functionality that they assume is present in a website"*, and contrasting it with discoverability defined as *"users encounter new content or functionality that they were not aware of previously"*. Findability, through the implication of searching to find, is intentional while discoverability can be both intentional and serendipitous. Additionally findability assumes the awareness of the user and their conscious pursuit of a goal.

2.3.5 Guessability

Guessability has been defined as a quality *"of symbols which allows a user to access intended referents via those symbols despite a lack of knowledge of those symbols"* [228] as well as a measure, considering it as *"the effectiveness, efficiency and satisfaction with which specified users can complete specified tasks with a particular product for the first time"* [103]. However, guessability is most commonly used in the context of *guessability studies* who often focus on gesture-based interactions [124, 159, 184, 216, 229]. Vieira et al. [216] describe *"Guessability or user-elicitation studies"* as useful as their results *"allow building general taxonomies or specific gestures when a high agreement rate is observed among participants"* who are *"prompted to create gestures for a particular action, with no given training or feedback"*.
Consequently, guessability is frequently being considered as a methodology to evaluate other usability principles or inform the design process rather than a principle to be defined separately. However, when considered as an independent concept, as the definitions provided show, guessability has been used to describe both, a quality of a system, and a measure of how well this quality facilitates interaction.
We follow Wobbrock et al.'s definition of guessability as a quality, but extend it past the original definition focused on symbols, to incorporate all parts of a system. Consequently, we consider guessability as the quality of the system allowing first time users to estimate the function of systems, features or input methods they encounter despite a lack of prior knowledge of them. This definition makes the importance of guessability to initial interactions obvious as it is explicitly focused on initial, or first, interaction. The ability of the user to guess the functionality of what they are presented with alleviates the difficulties unique to initial interaction and increases the likelihood of a successful interaction.
Guessability can aid discoverability, learnability, navigability and findability by making it easier for users to approximate how to interact with and navigate the system. It also contributes to the overall communicability of the system.

Differentiation from Discoverability

In contrast to discoverability, guessability does not incorporate considerations of awareness or interaction context. It solely considers whether in the moment of perception, the user can estimate the meaning of what they are presented with without extended consideration, previous knowledge or outside help. While this moment of perception necessarily includes context, as users do not encounter anything in a vacuum, guessability is focused on the immediate impression of the user rather than a careful consideration of context. As such it aids discoverability but does not include considerations of how the encounter in question came to be or what its outcome will be.

2.3.6 Noticeability

Noticeability describes the likelihood of something being *noticed* i.e. capturing someone's attention and making them conscious of something's existence. In HCI, noticeability has been explored primarily in the context of notifications and interactivity. Considerations around notifications commonly explore the impact of visual features like size, colour and motion of notifications as well as location in the context of differing factors such as background [153], resemblance with a primary task [137], AR HMDs [115] and gaze contingent positioning [110]. Explorations of noticeability of interactivity often occur in the context of public displays, with both the system itself and its interactivity needing to be *noticed* which is then followed by thoughts on enticing engagement and facilitating understanding [39,153,197]. However these considerations do not always explicitly refer to the concept of noticeability. Walter et al. [219] measure a "noticeability rate" which they define as the percentage of users that notice a provided hint about interactivity on a public display. They explicitly separate this noticeability from comprehensibility which they define as the percentage of users that understand the presented interaction technique.

Differentiation from Discoverability

Noticeability, while possibly a part of discoverability by focusing on the potential users perception and attracting their attention, does not include considerations of comprehension. As such, noticeability is often considered in connection with system or interaction discoverability, especially in the context of public displays. These examinations often explicitly separate noticeability from comprehensibility [219] or understanding [152].

2.3.7 Elucidating Differences - How Do These Concepts Compare

As is shown in the previous sections, various interaction concepts have been used with inconsistent and overlapping interpretations due to their semantic closeness and being applicable to a similar timeframe. This section aims to elucidate the different focus areas these concepts represent and how they relate to one another.

Figure 2.3 shows an expanded version of the sequence of actions considered in figure 2.2 in an aim to visualize the different focal points and overlaps. While this is not a definitive classification, it provides guidance for the general parts of initial interactions, different concepts focus on. However, specific parts of the timeline may vary depending on context. Additionally, while this selection of terms is extensive, it is, by no means, exhaustive. Not only are different terms introduced and redefined constantly, other concepts are often used to describe similar problem statements from different perspectives.

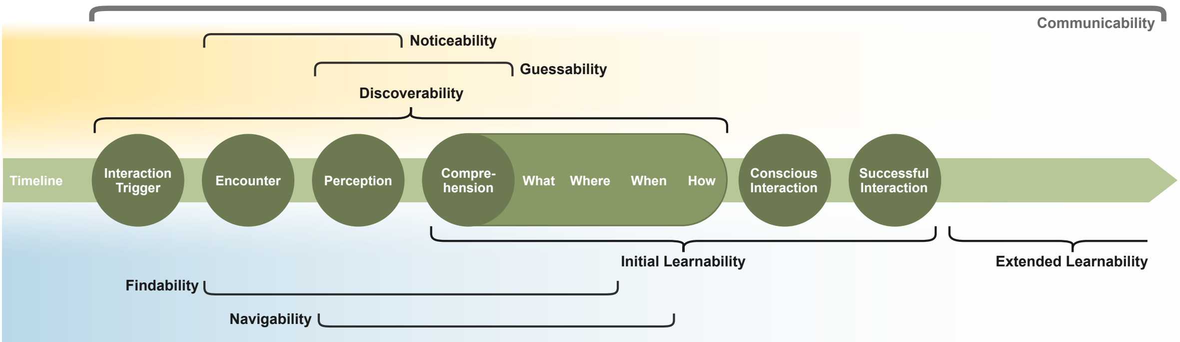


***Figure 2.3** – Timeline of interactions showing at which points different interaction concepts apply. While this generic timeline presents a lot of overlap with all concepts including some form of comprehension, they may only apply to very specific interactions which is not visually represented here.*

In contrast to all other concepts considered, communicability applies to the entire timeframe presented in the sequence of actions, however it presents a focus on the system capabilities and how they are communicated rather than the user interaction itself. As such, it presents an overarching concept that highlights the importance of system design aimed at communicating interaction possibilities and outcomes to users, but does not focus explicitly on initial interactions and the context surrounding and enabling the interaction.

Navigability is focused on the comprehension of *where* and *when* in order to facilitate movement through a system considering location and timing. Consequently, a sufficient level of navigability is necessary to enable findability. Without a comprehension of how to move through a system, it would be impossible to locate specific parts of it which are not immediately presented in the user interface.

While discoverability and findability cover a very similar timeframe, findability focuses on specific circumstances where the interaction trigger is invariably the users' previous knowledge of the interaction possibility and conscious desire to locate and utilize it. As such all concerned interactions happen intentionally and focus on the comprehension of *where* the desired feature is located. The comprehension of *what* predates and motivates the interaction, while questions of *how* exceed the focus of findability.

Noticeability tends to focus on facilitating or promoting perception of specific systems or system components. It does not include considerations of comprehension or interaction and is generally considered separate from understanding or enticing engagement. Nonetheless, noticeability is crucial in enabling all other interaction concepts as perception is a precondition for further consideration and engagement.

Guessability reflects the aim to have comprehension occur concurrently with perception. As such it is not concerned with how a user encounters or eventually uses an interaction possibility, but is solely concerned with whether the user can estimate the meaning of what they are presented with. Consequently, good guessability increases the likelihood of discovery and generally successful interaction by focusing on easing the comprehension of *what* one is confronted with.

Sorting these concepts along the presented timeline elucidates their different focus areas while also highlighting why their usage varies and differentiation is difficult. As they are all focused on aspects of initial interactions, they apply concurrently or in close proximity and the successful implementation of focused concepts, such as guessability, positively impacts all broader concepts covering that interaction timeframe.

2.4 Conclusion

In this chapter, we surveyed and clarified the different usages of discoverability, how it has been defined and which contexts it is applied to. In doing so, we introduced the separation of system, interaction and feature discoverability. Subsequently, we surveyed associated interaction concepts like learnability and contrasted them with discoverability to highlight the significant overlap in the usage of the terms while elucidating how they differ.
Through this we provide a broad overview of the many concepts concerned with the ability to engage with technology without previous knowledge or supplementary material and highlight the different focal points previous research has considered.
While the selection of terms is extensive, it is, by no means, exhaustive. Not only are different terms introduced and redefined constantly, but other concepts are often used to describe similar problem statements from different perspectives, making an exhaustive list not only not feasible but also incomprehensible. We may fail to include niche or yet-to-be-established concepts. However, the breakdown of discoverability and initial interactions in more elementary concepts, as well as the clarification of their temporal articulation, paves the way for future explorations that can easily incorporate additional terms, should they become established.

2.4.1 Importance of Clear Terminology for Discoverability

Given the constant increase of technology in everyday life, as well as the constant introduction of progressively complex and disparate interaction methods, understanding and easing the discovery of features and interactions is imperative. Despite extensive research around "learning" and "affordances" this problem has not been resolved, suggesting it is not sufficiently addressed through these concepts. We argue that the clarification of concepts and terms used in this context can contribute to both theoretical discourse and practical design implications. As an example, affordances in HCI were clarified by McGrenere et al. [145] breaking down the difference between Gibsonian affordances and Norman's perceived affordances. This was expanded upon by Hartson [87] proposing a separation into cognitive, physical, sensory and functional affordances, suggesting both the usefulness of overarching concepts as well as the need to clarify and break them down.
While the clear separation of different terms may appear trivial on the surface, clear and shared terminology is crucial when defining and exploring a research area. As Allen and Buie [6] state, *"There is no such thing as* just *semantics. Semantics is all about the very meanings of the words we use, the intention with which we use them, and the understanding they create in our audience. In communication, nothing is more important than semantics"*. When the aim is to address a widely established yet fuzzily defined problem such as discoverability, we share the view of Hartson [87] that *"shared meanings and representations (through common language) are an absolute must"*.
The differentiation and clarification of discoverability and related terms helps to consolidate relevant research that uses varying terms to describe similar problem statements while also exploring the different focuses and approaches these terms originate from and how they relate. As such, while not introducing novel concepts, this survey provides clear definitions and separations. These can aid future research to clearly define the specific focal points of their explorations and provide a convenient overview of concepts worth looking into when exploring related work.

3 Framework of Impacting Factors

Having clarified the concept of discoverability, in the previous chapter, it is pertinent to discuss *how* discovery occurs and which *factors* may influence its likelihood and success to further support its increased consideration.
We do so by identifying different human factors and system characteristics which are considered concurrent to the interaction concepts examined in chapter 2. We collect and review these factors and consider them specifically in the context of initial interactions. In doing so, we identified that they can be classified to be a part of one of the three main actors impacting initial interactions: The user, system and environment. Subsequently, we present the considered factors within the individual actors and reflect on how they interrelate before expanding our consideration to examine how the factors of all three actors interrelate and how they can and should be considered to foster successful discovery of system, features and interactions. Through this, we provide a holistic view of how initial interactions develop and identify individual factors that may be leveraged to improve the discoverability of interactions.
While our particular focus is on graphical user interface and in particular touch screens, the following framework can largely be applied to initial interaction in general.

3.1 Framework of Impacting Factors

Due to the lack of work considering discoverability, and what may influence it, in a holistic approach, as well as the previously discussed lack of consistency and focus, we again followed an exploratory scoping literature review approach following references and comparing the relevant literature. This approach was found to be most suitable, as the aim of this review was not an overview of research on a focused question but to elucidate connections and reflect on how factors interconnect to make discovery work [117,180].
Starting from the academic work reviewed in the previous chapters we examined which (human) factors and human computer interaction concepts are considered frequently in conjunction with the concept of discoverability (*e.g.* affordances, awareness), designing for discoverability (*e.g.* familiarity, visibility, feedback) or the success of initial interactions (*e.g.* examples, social context). We then extracted these *factors* and investigated their usage, definitions and associated concepts (*e.g.* affordances led to perceived affordances, signifiers, feedforward, etc.), to elucidate their connection and how they impact initial interactions. We categorized these factors into three main *actors*: system, user and environment. Subsequently, we organized them accordingly and analysed how factors of each individual actor interrelate. Through this we constructed a conceptual framework of factors impacting the discovery of novel interactions.
The three actors influence the interaction in different ways. The system and user present two entities between which the interaction occurs, while the environment establishes the context of said interaction. As such, the environment may influence the interaction directly and indirectly through its impact on the system and user. The *actors* each have various in-

terconnected *factors* that influence one another. Due to the complexity of this interplay, the following sections break each of these actors down independently to examine the factors associated with them and review corresponding literature. Subsequently, we will discuss how they influence one another, other actors and ultimately the interaction.
The interaction leading to a discovery can be triggered both *endogenously*, within the user, or *exogenously*, through environmental or system factors. Where triggers of initial interaction are particular to one actor, they are mentioned in the corresponding section.

3.1.1 System

The system generally presents the more passive counterpart of an interaction between user and technology. Outside some exceptions, such as expecting users to respond to a notification, it rarely instigates an interaction and must incentivize and guide potential users to do so. It needs to convey its purpose, functionality and interactivity to users well enough to maintain interest and facilitate successful use and exploration of available functions. As designers cannot select or change the user significantly and only occasionally have influence over the environment in which technology is used, the system is the main actor through which interactions can be guided and influenced.

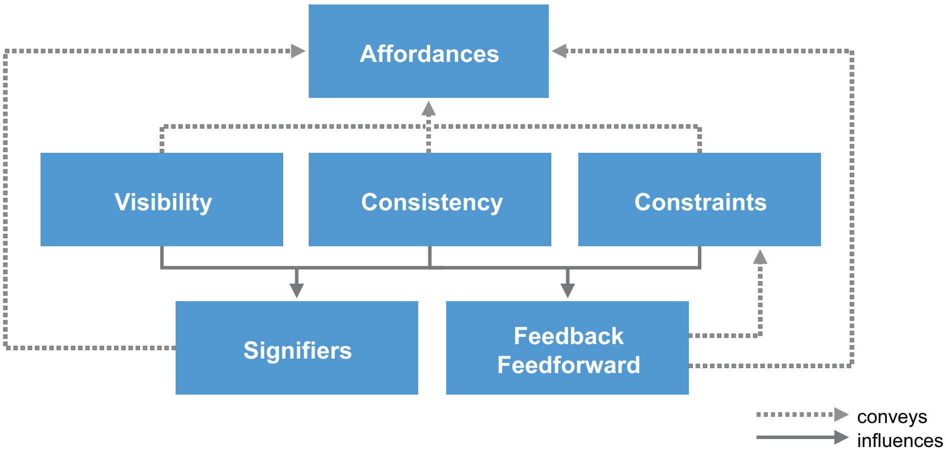


Figure 3.1 *– System factors influencing initial interaction and their relations to one another are shown. They either influence the success of other factors or convey their presence to the user.*

Figure 3.1 shows a conceptual model of the system factors. In this model we differentiate between two relationships: influence and convey. One factor influencing the other means they are not essential to its existence but aid its success and can have a significant impact on its perception. Factors that convey another, communicate the existence of said characteristic to the user.

The following sections will look at the definition and meaning of these factors, as well as their influence on initial interaction and other system factors.

Affordances

Affordances in the context of HCI have long been studied and debated [18,106,165,208]. We do not detail the theory of affordances but use them as they were defined by Gibson [73] and clarified by McGrenere et al. [145] in the HCI context, as:

— Offerings or action possibilities in the environment in relation to the action capabilities of an actor
— Independent of the actor's experience, knowledge, culture, or ability to perceive
— Binary – an affordance exists or it does not exist

As such, they are what a specific user can do with a system. Whether these possibilities are perceived as possible, registered, understood or utilised does not change the existence of the affordances. Due to this explicit focus on the *system* offerings and not the user perception or understanding, they are distinctly a part of the system. As the consideration of which offerings and action possibilities are factually available to the user, they play an important role in initial interactions as they are what a user needs to perceive and utilise. For example, mobile keyboard keys afford being pressed individually for users whose finger size fits individual keys, without pressing other keys in the process, and who possess the steadiness and mobility in their fingers to press the key. Note that here we consider affordances as defined by Gibson, while we consider Norman's definition as perceived affordances on the user side in section 3.1.2.

Visibility

Visibility refers to the state in which something can be seen. It has been widely regarded as an essential design principle facilitating ease of use [61,160,183]. In these considerations, the term visibility is used to simply describe whether something is visually represented in a way the potential user can see it. However, the term has previously also been used to incorporate user comprehension and understanding beyond that. Norman [164], for example, defined visibility as one of the most important principles in design stating *"the correct parts need to be visible, and they must convey the correct message"*, thus expanding the meaning of visibility beyond solely being visible, to incorporate provoking specific user interpretations. Notably the later revised and expanded edition of his book [167] does not include this focus or definition and instead concentrates on discoverability. The expanded definition is included here to acknowledge that the term visibility has previously been used to incorporate more than visual representation and to clearly separate this from visibility as a system characteristic. We, leaning on the most common usage, consider visibility as strictly the characteristic of visually representing something in a way potential users can see it. As such, visibility of a system, feature or input method can facilitate initial interaction and therefore is a desirable system quality. However, visibility does not guarantee the user's comprehension and consequent ability to utilise what they see.
The importance of visibility in easing initial interaction can be illustrated when considering invisible features or input methods that are consequently overlooked. For example force press on iOS, which, as discussed in chapter 1, was never communicated to users and lacked visual representation, leading to it being underutilised and eventually discontinued. If the existence of features and the interaction possibility of revealing them are not communicated in the interface, they are unlikely to be discovered.
Due to interaction still primarily happening on *graphical* user interfaces (GUIs), visibility remains the predominant way of conveying information to the user leading to signifiers and feedback often occurring as visual representations. Similarly, the majority of affordances are

communicated to the user through graphic depictions. However, the limited screen estate and ever-expanding functionality in modern smartphones, in combination with a aesthetic preference for minimalism, has lead to a lack of visual cues on screen. That being said, the exposure of features can increasingly be facilitated by non-visual cues as a rising number of systems and input methods rely less on GUIs, but, for example, haptic feedback or VUIs, prompting interactions through audio cues.

Consistency

Consistency has been widely identified and acknowledged as an important principle when designing user interfaces [20, 44, 183, 199]. It *"refers to designing interfaces to have similar operations and use similar elements for achieving similar tasks"* [183]. Consistency eases user understanding by reducing the mental effort required to identify unique interaction. It has been noted that consistency is relevant both within the system (internal consistency) as well as externally with similar systems (external consistency) and as the system relates to outside factors [20]. An example for internal consistency is the "next" button looking the same everywhere within an application (visual consistency) while also being consistently in the same place (spatial consistency). An external example would be the Hamburger icon which is frequently placed in the top corner of mobile user interfaces across multiple applications and is consistently used as a toggle to show and hide a menu or navigation bar.
Through the cohesion within itself as well as with external systems, consistency helps signify interaction possibilities and conveys affordances of the system. It utilizes the user's familiarity with similar systems, applications or other functions within the same system to ease understanding and learning. Smith and Mosier [201] in their guidelines for designing user interface software add that if *"no strong user experiences exist with respect to a particular design feature, then designers can help establish valid user expectations by careful consistency in interface design"* . When considering the discoverability of novel features or interaction methods it is therefore pertinent to consider established design patterns to either utilize them in guiding the user or avoid them when implementing novel functionality to avoid the assimilation paradox [34]. Additional, if novel functionality is implemented in multiple parts of the system, it should be presented consistently to allow discovery through transfer (more see section 3.1.2) once an initial discovery has occurred.

Constraints

Preece et al. [183] describe constraints as *"restricting the kinds of user interaction that can take place at a given moment"* and Norman includes constraints as one of his seven fundamental principles of design, stating *"providing physical, logical, semantic, and cultural constraints guides actions and eases interpretation"* [167]. As such, designed system constraints restrict the kinds of interactions that are available to users at any given moment to prevent them from doing things that are inappropriate. Therefore, when aware of these constraints, users can leverage them to infer possible interactions. Benyon [20] highlights, that constraints should be particularly utilised to prevent users from making serious errors by *"constraining allowable actions and seeking confirmation of dangerous operations"*. In addition to explicitly designed system constraints, e.g. greyed out options in a menu, there are logical constraints that inform proper use of a system. They are based on a *"logical relationship between the spatial of functional layout of components and the things that they affect or are affected by"* [164]. Typically, Norman

gives the example of a view with an element half visible near its edge, from which one can guess that the view can probably be scrolled [166].
In summary, constraints ease user understanding by limiting the options available to them. As such they can provide feedback to the user, signify interactivity and convey affordances. This is achieved by limiting the number of actions to consider as alternatives to those "suggested" by affordances [178].
Further constraints, which are not directly related to system functionality, are considered in the subsection 3.1.3.

Signifiers

A signifier according to Norman [167] refers to *"any mark or sound, any perceivable indicator that communicates appropriate behaviour to a person"*. He contrasts them with affordances, stating that affordances determine *what* is possible while signifiers are employed to communicate *where* said action should take place. He emphasizes that both are needed for successful interaction. Additionally, Norman distinguishes between deliberate, intentional signifiers like door handles or push signs and accidental, unintentional ones such as flags indicating wind direction or a beaten path showing established walking routes. However, he notes that the intentionality or lack thereof behind a signifier does not matter to the user recognizing and interpreting it. Whether something was placed specifically to suggest meaning to a user or whether its aid is coincidental does not influence its effectiveness. The users' ability to perceive the signifier, however, does: if the signifier is not perceivable, it fails to function as such successfully.
Considering this, signifiers are crucial to convey affordances to the user, and as such, possible interactions. They are facilitated through often visual representation and more easily recognizable through consistency and constraints. Signifiers also play a critical role in the concept of *self-revelation*, which has been defined as a system providing *"information about what commands are available and how to invoke those commands"* [116].

Feedforward and Feedback

Feedback is a clearly established concept and has been described as the system continuously informing the user about the system state, what it is doing, when user input is registered and how that input is being interpreted [20, 162, 183]. Feedforward, while less commonly mentioned, is equally important. It was introduced by Djajadiningrat et al. [62] as informing *"the user about what the result of his action will be"* prior to the interaction taking place. Wensveen et al. [224] later differentiated between three types of feedback and feedforward: Inherent, Augmented and Functional. Inherent feedforward describes the information that communicates what action is possible and how. Augmented feedforward refers to information from an additional source such as words in an on-screen message. Functional feedforward informs the user of the general purpose of a system and its features. Vermeulen et al. [215] examined this and other definitions in an effort to re-frame feedforward and distinguish it from related design principles in more detail. As such they highlighted the difference between feedback and feedforward through when they occur while distinguishing them from perceived affordances (discussed in section 3.1.2). Feedforward occurs before the user action while feedback is provided during or after an interaction. They note how feedback *"can later turn into feedforward again [...] for another action that logically follows the previous one"*. Norman

distinguishes between the two concepts by describing feedforward as *"the information that helps answer questions of execution"* and thereby doing and feedback as *"information that aids in understanding what has happened"* [167]. As such feedforward is more important in the context of initial interactions and their inception.
By clarifying to the user what result a potential action will have and, once they start the interaction, that their input is registered and what effects it has, Feedforward and Feedback help the user evaluate the consequences of their actions and subsequently how to interact with the system. They further aid users by conveying system constraints (e.g. greyed out options once something is selected) and communicating successive affordances to the user.

3.1.2 User

The user, as the independent counterpart in the interaction, is influenced by not only the system's behaviour but also their own experiences, expectations and intentions. As shown in Figure 3.2 there are various characteristics that influence each other and the users' mental state and consequently their actions. While those characteristics cannot be altered directly to facilitate the discovery of novel interactions, features and systems, they greatly influence its success and therefore need to be considered when aiming to facilitate successful initial interaction.

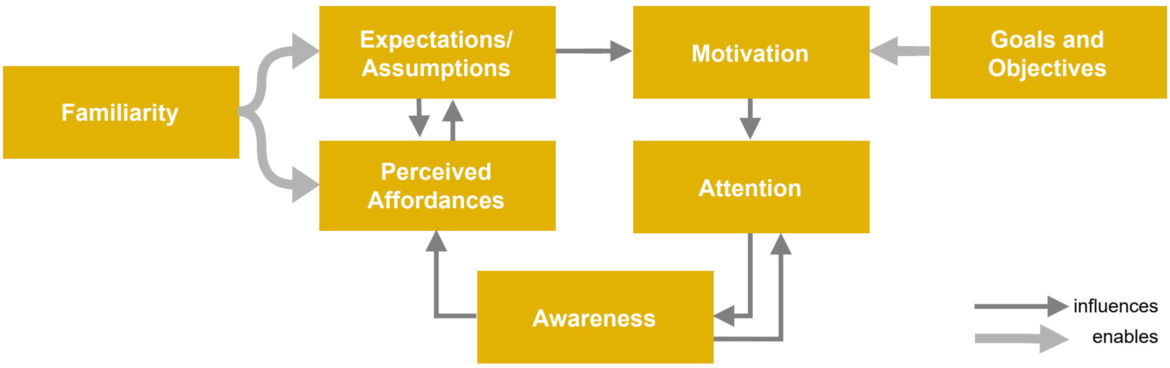


***Figure 3.2** – Depicted are different user characteristics and how they interconnect. While some characteristics influence one another, others are preconditions that are essential for the formation and development of others.*

Expectations and Assumptions

The ISO standard describing principles for interaction between a user and a system notes the importance of expectations by making *"conformity with user expectations"* one of their principles and recommendations for interaction design [70]. Similarly, in an early guideline for user interface design, expectations are considered by advising designers to *"ensure that the results of any control entry are compatible with user expectations, so that a change in the state or value of a controlled element is displayed in an expected or natural form"* [201]. Despite these guidelines, Noyes [168] argues that the consideration of expectations in the HCI context has been neglected. He notes that expectations are an important aspect of human behaviour and performance and as such are pertinent to consider when designing interactions, as mismatches of expectations and actual system performance lead to frustration, irritation and unnecessary mistakes. This is supported by work on spatial consistency analysing how users adapt their expectations of the location of known targets when interfaces are transformed (e.g. ro-

tation, stretch) [195].
While the importance of expectations has been noted and discussed, they are rarely defined since they are not a novel concept or term developed for or in the context of HCI. In an effort to be clear on what we consider when using the terms expectations and assumptions, we define them as follows. An expectation is something the user considers to be either probable or necessary. One is based on the user's conviction that they have interpreted the system and interaction possibilities correctly and therefore know what will happen. The other is based on the presumption that some features or functionalities *must* be part of the system they are confronted with due to its intended use or circumstance, therefore predicting the presence of said feature or functionality. Similarly, assumptions are things the user has accepted to be true, present or certain to happen without having any proof. The user *assumes* to know an interaction possibility and due to this assumption and their interpretation of the situation, *expects* a certain outcome. These expectations and assumptions greatly influence how the user interacts, or tries to interact, with a system.

The importance of this to the discovery of novel interactions or features is supported by Alvina et al.'s [10] findings from a study conducted in the context of cross-device learnability. They found the disconnect between an application's capabilities and the user's expectation of it, to be a major contributing factor to, what they identify as the most salient cross-device learnability issues, related to awareness of features and locating features. Similarly, Pearson et al. [174] found that users' expectations about a system influenced their behaviour when interacting. They found that users adapted their language to match the system language more if they believed the system to be unsophisticated in comparison to systems they believe to be more sophisticated. They further highlight that this tendency is solely based on the user's assumptions about the system and is unaffected by the actual system behaviour and capabilities.

When considering this factor in relation to others, the development of expectations and assumptions is heavily based on which aspects of a system the user is aware of as well as what they even perceive to be an interaction possibility. In addition to stemming from what users perceive as affordances of the system, they are based on prior experiences of the user and their familiarity with similar interactions. This relationship has been noted by Noyes [168] who highlights the important role of past experience and knowledge in developing expectations. The projection of these familiar interactions on to a new one creates expectations for the new interaction to be similar to the ones the user has already experienced as well as the assumption that they will be.

Motivation

Motivation is what drives the user to initiate, continue or end actions or behaviours at any given time [53]. As such motivation and its associated goals can be the source of initiation of interaction. A motivated user can intentionally discover functionality or interactivity. This can either be through conscious exploration of a system or the seeking out of features a user might hope to find. The user seeking out something they already *expect* to be present is commonly referred to as *findability* [33] (see subsection 2.3.4). In the absence of concrete goals and objectives, a user may casually interact with a system and serendipitously come across unfamiliar interactions or features and decide to explore them.
The source of motivation can be intrinsic (arising from internal preferences, aims, values or

wishes), or extrinsic (stemming from outside factors such as the environment and a means to an end rather than for the behaviours own sake) [53, 212]. There is extensive literature regarding different facets and sources of motivation that exceed the scope of this thesis (see [53] for a comprehensive review). In HCI in particular, motivation has been studied extensively, especially in the context of education and learning [1, 41, 221], which are common topics in motivation psychology as well. Additionally, *curiosity* has been explored as a new type of intrinsic motivation for exploration, learning and creativity [122, 230].

The degree to which a user is motivated can be influenced by how well their expectations and assumptions of system functionality match their goals and objectives for the interaction. Additionally, goals and objectives of great importance to the user may motivate them more, regardless of accordance with expectations. Furthermore, the type of motivation may influence its degree, especially in the absence of concrete goals and objectives. Since extrinsic motivation is considered as *engaging in something as a means to an end*, it may be significantly lower in the absence of goals, while intrinsic motivation as *"doing an activity for itself"* [212] remains stable. Intrinsic motivation might additionally be influenced by a person's mood. Finally, the degree of motivation of users can influence their commitment to the interaction and through that, the amount of attention they are willing to allocate to it.

Attention

Attention describes the sustained focus of the cognitive resources of users on something that they pay "attention" to. As such it has been extensively studied in psychology but also Human Factors and HCI. There are multiple theories and models aiming to examine and clarify the concept. Among the most widely adopted is Christopher Wickens' Multiple Resource Theory of Attention [225]. According to this model, tasks can be carried out simultaneously if they do not have the same type of resource demand. This recognizes the limit of attention resources available. Consequently, there is a difference to be considered between for example auditory and visual attention and whether stimuli can be attended to, given the presence of other stimuli in a situation.
Another widely accepted model of attention based on the clinical model of Sohlberg and Mateer [202] differentiates between 4 types of attention, not in terms of modalities but mental effort:

- **Focused attention** refers to directing focus to specific stimuli and responding to only it, excluding other sources
- **Divided attention** refers to simultaneous focus on two or more tasks simultaneously
- **Selective attention** refers to attending to a specific stimulus in the presence of another which should be disregarded
- **Sustained attention** refers to maintaining a consistent response to stimuli over prolonged periods of time of continuous or repetitive activity

What kind of attention users pay to the interaction, and how much it requires of them, greatly influences whether they are able to perceive interaction cues given by the system. For example, focused attention on a task can lead to inattention blindness, where the visual focus on a specific task leads to a failure to detect similar stimuli, even when they appear in the area of attention as long as they are not disruptive [138]. This is further illustrated by Davies and Beeharee [50] who found that participants focused on a primary driving game task (control of direction and speed) failed to notice a third of the given speed limit notifications. Similarly, Leung [125] notes how inattention blindness may lead to users not only missing highlighted sections when they appear but not registering that they occurred at all.

Depending on the type of attention required, the system needs to ensure it sufficiently captures or directs the user's focus and supports the user re-entering the interaction when, for example, sustained attention was required but the user got distracted. Through this, the user's motivation impacts attention as it influences the resources they are willing to allocate to a given interaction. Designers need to be aware of which kind of attention and resource is required to successfully interact with their system and change stimuli accordingly when aiming to redirect attention or introduce changes.
In HCI literature concerned with discoverability, attention has been especially studied in the context of system discoverability for public displays [8, 39] where the focus is on initially getting or attracting the users' attention.

In general, sufficient attention on the interaction at hand leads to the user being able to observe the system and environment, which allows them to be aware of their surroundings. Conversely, it has been found that increased awareness can support stable attention and increase automatic suppression of task-irrelevant stimulus [222].

Awareness

Awareness describes the quality or state of knowing and understanding that something is happening or exists [60]. In HCI some of the main considerations are situational and social awareness. Situational awareness has been defined as *"the perception of environmental elements and events with respect to time or space, the comprehension of their meaning, and the projection of their future status"* [67]. Social awareness is generally considered as the ability to empathize with others, understand social and ethical norms for behaviour and consequently react appropriately in social and interpersonal interactions and has been defined as *"states of mind in which the person is consciously aware of a specific range of social experiences from a specific point of view"* [223].
Considering awareness is especially relevant in the discovery of interaction where the user might not be aware of inputs or controls at their disposal, as well as the overall system and functionality. Notably simple awareness of something does not guarantee the right understanding or interpretation. It is, however, a prerequisite to be aware of, for example, the existence of a button before the user can then interpret how, or for what, it might be used. Nonetheless awareness significantly impacts initial interaction because potential users, who are accurately aware that a feature is available or an interaction method is supported, are more likely to begin interacting with it.

The relation between awareness and attention has been widely examined, with some arguing attention is necessary for awareness [54, 134] while others question the necessity of attention for awareness [111, 119], arguing that enhanced attention to specific stimuli can come at the expense of overall awareness. Furthermore, Baijal and Srinivasan [19] argue that the type of attention, in their case considering focused and distributed, influences awareness.

Perceived Affordances

Perceived affordances are the actions a user perceives to be possible, based on what they are confronted with [163]. They are the user's interpretation of what the system properties seem to be and how the system appears to suggest that these properties can be utilised. As such they are influenced by the user's awareness of said system and its properties. Since perceived affordances are personal interpretations, they are influenced by the previous experience and knowledge as well as the culture of the potential user. If a user is familiar with certain inputs and how they are generally signified, the users will assume these inputs are present if they perceive the system to have these signifiers regardless of whether the affordance actually exist. Thereby the sole perception of the presence of a property does not guarantee it actually exists [145, 167]. This perception does, however, inform the users expectations of a system capabilities, and assumptions about functionality and how to successfully interact with the system. Notably, in HCI the term affordances is commonly used incorrectly where perceived affordances should be used instead [145]. Similarly, signifiers are often wrongly referred to as affordances [166].

To summarise, *affordances* are what a specific user can do, *perceived affordances* are what a specific user *thinks* they can do and *signifiers* are indicators communicating affordances to users.

Familiarity

Familiarity, outside its previously described impact on expectations and assumptions as well as perceived affordances, can in itself be a cause for discovery of new interaction methods through transfer. We assume something to be possible within a system (interaction methods, functions, etc.) because we already know it works in another. This can lead to discovery of novel interactivity but predominantly allows the discovery of unknown features. If the user is already familiar with similar interactivity, they are more likely to correctly estimate its functionality and location within the system [10]. Through the knowledge of how they can interact with one application, the users have a concept of which interactions or features may be possible in another. This knowledge then allows them to test these concepts in a new environment which can facilitate the discovery of new aspects of said environment. For example, if a user is aware that they can swipe from the left side of their mobile phone screen to the middle in one application to open a menu, the same interactivity, which is not explicitly suggested anywhere, can lead to a different function in another application [177]. This assimilation bias [34] can lead to discovery but also inhibit it as the perceived understanding of a system limits further exploration and consideration. Furthermore, a user may prescribe limitations of previous experience that do not exist in the new environment to their interaction with it. The users, while adhering to the concepts they are familiar with, never realize the new possibilities available to them.

Endogenous Interaction Triggers

The user characteristics considered previously directly influence discovery and the circumstances in which it occurs (see figure 3.3). As such these *endogenous triggers* focus on the users by themselves, and without outside influence, discovering unfamiliar functionality or interactivity through a set of different circumstances. As this discovery does not rely on outside impulses, the potential users own motivation greatly influences the likelihood of these to occur.

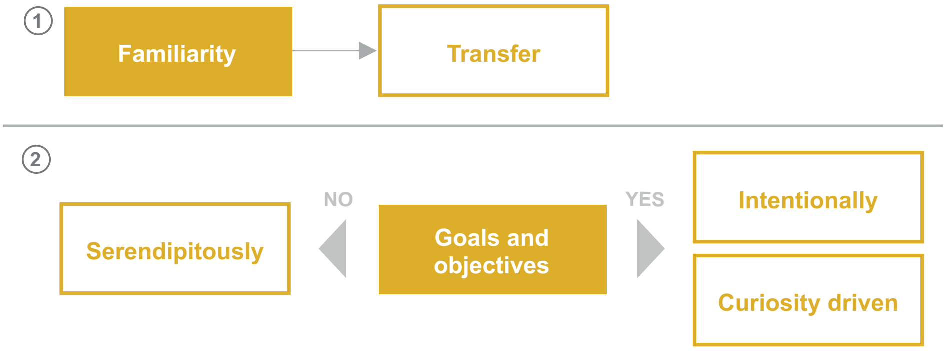


Figure 3.3 *– Shown are endogenous triggers for initial interaction and their source. 1) Familiarity allows users to transfer previous knowledge on to the current scenario and deduce interaction possibilities 2) Goals and objectives incentivize the user to intentionally seek out functionality or consciously explore the system. In their absence exploration or random encounters of new interactivity happen serendipitously.*

Transfer describes discovery through familiarity with similar system, function, input methods or signifiers. Although the user may not be aware or familiar with the novel functionality in question, if it resembles interactivity the user is familiar with they may deduce or assume similar functionality and subsequently discover it. Additionally, if the user is familiar with signifiers used in the system (through resemblance to established digital or real-world affordances) they similarly may transfer this knowledge to infer functionality.

Serendipitous discovery describes users encountering and utilizing things by chance without previous intention to do so [157]. It can lead to improved interaction with the system, but may not have a long-lasting impact as the user may now be aware of the existence of a feature or input method but not able to replicate their interaction since it happened unintentionally. Furthermore, the user may discover the existence of a feature, but be oblivious to the novel interaction method they utilized, barring them from repeating executing the function purposefully.

Intentional discovery is based on concrete goals and objectives without specific knowledge of the functionality in question. This may be the intention to learn a system, a directed effort to solve a specific problem or an aim improve ones general performance with the system.

One subsection of intentional discovery is *curiosity driven* discovery which includes intention to discover a system or its functionality for the purpose of the discovery of itself.

Finally, while not displayed in figure 3.3, *errors* in an intended interaction can lead to discovery by unintentionally triggering unfamiliar functionality. It is not visually represented as errors can occur due to multiple factors (individually or in combination). A lack of attention, awareness or motivation may all lead to faulty interaction, as can the wrong perception of affordances. Furthermore external factors can cause errors. It is nonetheless important to consider errors as a discovery mechanism since the unintentional triggering of functionality while aiming to perform another interaction makes awareness and attention especially pertinent. If mistakes trigger novel interactivity, in the absence of awareness and attention, the user may never perceive the outcome or comprehend that it is connected to the interaction in question.

While not novel, this distinction between triggers may help designers in adjusting systems according to the intended future interactions. If designers are aware that parts of a system may lack initial discoverability it may be beneficial to consider how they expect future users to encounter this functionality. Once identified, they may then be able to create targeted supplementary materials or adjust designs accordingly to support the type of discovery they wish to achieve.

3.1.3 Environment

The environment in which an interaction happens or the system is placed can also influence the user, the system and the interaction between them. In this context, the environment does not only include the physical surroundings, but other people present in the space as well. Figure 3.4 displays factors within the environment that influence initial interaction, which will be discussed in the following section, as well as their relationships to one another.

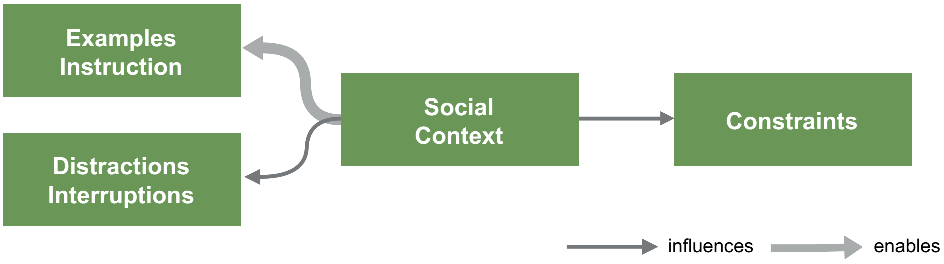


Figure 3.4 *– The environmental factors influencing potential users as well as the system and interaction itself and their relation*

Social Context

Social context refers to the immediate physical and social setting in which an interaction is taking place. As such the social context of an interaction includes other users and bystanders as well as media consumption and common practice in the environment. Through these factors, it can play an important role in the inception and success of initial interactions. This can be both by increasing the likelihood and success of interactions through examples and instructions or by decreasing their probability through interruptions and constraints.

Nonetheless, the beneficial role of social context to initial interaction has been confirmed in various ways. One example is the extensive literature concerned with social learning [92, 127, 186, 187, 234] which often incorporates discovery. Considerations include learning through imitation, increased understanding through conversation and observation as well as increased awareness of available interaction possibilities highlighted through others interacting. However, Murphy et al. [157] found that social learning is infrequent, with discovery of new tools predominantly occurring due to other factors.
While not focused on discovery, a lot of technology adoption relies on discovery through social networks. Vannoy and Palvia [213] make the argument that social computing influences technology adoption through both, the technology being embraced by individuals and it being embedded in society overall. This is demonstrated through users being more likely to use hotkeys if their co-workers already do so [175]. Several studies show that social networks play a critical role in the adoption and adaptation of customizable software [72, 136].

Moreover, Draxler et al. [65] note that technology appropriation is an inherently social activity. Furthermore, Brignull and Rogers [26] found what they called a honeypot effect, in which the number of people around a public display progressively increased, attracting further engagement. This suggests the behaviour of other users in the environment increases attraction power and therefore can increase discoverability.
Ultimately, discovery of functionalities or tools can be increased by the social context through help from colleagues, fortuitous observation of examples and casual or explicit sharing of information about available tools [156].

Examples and Instructions

Examples can allow users to discover systems, functionalities or input methods they may not have recognised or successfully utilised independently. Users may witness others interacting and recognise the interaction as something desirable or helpful for themselves which they were previously unaware of. This then allows them to either replicate the interaction or explicitly seek out how to achieve a similar interaction. Depending on the complexity of interaction or feature, purely mimicking other users' behaviour may not permit successful engagement with it.
Such examples stem from the social context of the interaction and can be either witnessed directly or consumed through media. One case of a directly witnessed example leading to the discovery of a new functionality was observed by Murphy-Hill and Murphy [156] who recorded that one programmer witnessed another one using a keyboard shortcut in a remote screen-sharing session. Realising they were not familiar with the performed interaction they then questioned their colleague about what had transpired and discovered the option. This example highlights two facets of examples in the context of initial interactions. First, in some cases witnessing the effect of an interaction may be enough to incentivise users into investigating unfamiliar interaction methods. And second, examples can support discovery in both co-located and remote work settings as long as technology allows the observation of others interacting.
Apple staged presentations of new products clearly illustrate that examples through media depiction can be effective in introducing new interactivity. New input techniques like pinch-to-zoom on the first iPhone were never communicated on the device itself but through demonstration in the keynote introducing it as well as subsequent ads [144].
In addition to visual examples prompting imitation, anecdotes from other users in the environment may increase awareness of interactivity. For example, hidden widgets may be discovered thanks to a user's social network sharing knowledge of their existence with them [178].
While some of these examples and anecdotes may be shared unwittingly, the social context can also aid initial interaction through explicit instructions. Other actors in the space may directly instruct the user, therefore revealing interactivity or a sequence of actions to them, which they themselves would not have perceived as such or comprehended well enough to perform. While this is not a source of independent discovery, it increases the likelihood of initial interactions occurring since the users are more likely to perform the interaction that has been demonstrated to them.
Instructions can additionally come directly with the product or system, blurring the line between environmental, external, guidance and internal system guidance and instructions. However, the difficulty of including explicit instruction directly with the product is illus-

trated by a general reluctance to read enclosed instruction manuals [22]. To circumvent this reluctance, attempts have been made to integrate instructions more closely in the interface. One example of this is the augmentation of traditional tool-tips with video tutorials and documentation [81] within the system interface.
Given these considerations, the social context and resulting examples and instructions generate exogenous triggers for initial interactions and discovery. These can be implicit in the form of examples or word of mouth or explicit through instructions as illustrated in figure 3.5.

Figure 3.5 – *Exogenous interaction triggers stem from the social context of an interaction. It can induce interaction through implicit interaction guidance like examples or word of mouth as well as explicit instructions.*

Distractions/Interruptions

Factors like noise, lights and other people, as well as other technologies, can distract users from focusing on the interaction possibilities they are confronted with. Such interruptions are prevalent and have been proven disruptive as it takes time to resume previous activity and error-rates increase following an interruption [24]. Various studies have found that the disruptiveness of interruptions is influenced by their duration, complexity and similarity to the original interaction [27, 28, 74, 150]. Interruptions can, furthermore, lead to a change of work strategy where the execution of the primary interaction is modified to avoid deterioration of performance [235].
In addition to simply taking the user or their focus away from an interaction, distractions may also lead to a lack of attention and consequently awareness, that may be required to explore more sophisticated interactions and features. Other actors in the environment may interfere to either interact on behalf of the original user, preventing them from familiarizing themselves with the interaction, or comment on the interaction. While these comments may be aimed at instructing the user to successfully embrace an interaction possibility, they can also lead to overlooking features the original user might have paid more attention to, if they were not explicitly directed towards another method or feature.
While not unique to initial interactions, these considerations are especially relevant since they may prevent discovery. Although internal factors may also be the cause of distraction, they are not considered explicitly but through their impact on motivation and attention in section 3.1.2.

Constraints

There are various constraints limiting interaction and influencing the user and system in the environment [167]. While there are semantic and physical constraints, the constraints most pertinent to initial interactions and their inception in the environment are social and cultural. Social constraints include restrictions through laws but extend to the adherence to social norms and traditions. Even if a system clearly communicates its purpose and the user assumes interactivity they would like to utilise, they may not test it if the interaction suggested is inappropriate to the social context or their environment. Examples of this are voice interactions in quiet environments like libraries or expansive gestures in formal settings. The perception of these constraints and therefore their effect on user behaviour can be amplified through the presence of other people in the environment.

3.2 Discussion

We compiled a conceptual framework of factors that may impact the inception and progression of discovery and initial interactions. In doing so we aim to advocate for the distinct consideration of what facilitates and enables discovery and how to support it through design. In this context we additionally explored various triggers of interaction, both exogenous and endogenous, as well as whether the interaction happens intentionally or serendipitously.

3.2.1 Interplay of actors

The factors identified in the interaction framework in the previous sections are separated into actors for clarity and to highlight their immediate relations within that actor as well as which part of the interaction they impact. This additionally aids in identifying which factors the system designers may be able to influence or modify.

However, the factors' impact is certainly not contained within the actor in which they are placed. Factors placed in one actor influence other actors and the factors considered in them. In fact, each factor can be considered to have an impact on all other factors. As a full consideration exceeds the scope of this manuscript and does not add substantially new insights, we highlight prominent examples below to emphasize the interplay of actors.

Consistency between systems becomes especially relevant if it can leverage the user's familiarity with other systems, to inform their expectations of interactivity and features [20]. Through leveraging familiarity, this consistency additionally impacts the users' perceived affordances [205]. Similarly, constraints in the environment influence the user side through their impact on expectations and goals, while influencing the system by potentially limiting affordances [167]. This is shown when users, in a quiet environment like a library, expect technology to be similarly quiet and consequently disregard considerations of interaction through voice. Another example of this would be rain that may make public interactive touchscreens harder or impossible to use.

Similarly, motivation is often influenced by external factors. In HCI, gamification has been used in an attempt to increase motivation [58] through system design. Additionally, social factors influence motivation through shared goals and social motivation, social interactions increasing motivation as well as social pressure. Notably gamification has also been used in order to support discovery in an effort to improve overall learning [63].

These examples illustrate the importance of considering initial interactions through a holistic approach including how it is supported or hindered through specific user characteristics, environments and system designs and how they interrelate. For most systems it is not feasible to be discoverable to every potential kind of user in every potential circumstance, but a careful consideration of the presented factors can aid designers to create systems that increase the chance of discovery for the intended user groups and environments.

3.2.2 Boundaries of the framework

While the collection and consideration of impacting factors in the presented framework is extensive, it is by no means exhaustive. Due to the complexity of human interaction and the mental processes involved, as well as the intentional lack of focus on specific systems or actions, an exhaustive list is not feasible.
Consequently, we limited the selection of factors to those we consider to be pertinent to the initial discovery of features, interactions and systems. Furthermore, as this thesis is focused on touch interactions on smartphones, the framework is predominantly centred around graphical user interfaces including consideration of visual stimuli and how they are processed. The considered factors were chosen due to their close relation and common consideration in the literature associated with discoverability, learnability and the initiation of interaction.
Additionally, many of the factors considered are themselves complex and widely discussed topics within HCI research. This includes broad concepts aimed generally on the user's overall understanding (e.g. perceived affordances [163], cognitive affordances [87]) as well as how the system may facilitate such an understanding (e.g. signifiers [167], suggested interactivity [25], feedforward [215]). However, these considerations do not provide a holistic overview of factors that may impact initial interactions and discoverability beyond their specific focal points. While we aim to provide an overview to elucidate their relevance as well as connection to the overall framework, extensive investigations exceed the scope of this thesis.
We aim to explore these factors in conjunction to elucidate their connection and importance to the discovery of interactions. Consequently, we limited our work to the main factors in order for the framework to remain comprehensible while offering an extensive overview. Despite the manifold factors that could be included, we argue that this framework is broad enough to elucidate the complexity of discovery and initial interactions and highlight especially relevant considerations. Further factors like cognitive biases and accessibility exist and, while important, were not considered critical to discuss at this point.
The framework can be utilised to identify which aspects of initial interactions designs fail to address and whether that results in lower discoverability. Given this, the model can aid in the identification of discoverability problems and determining their origin as well as be beneficial when analysing results from other methods employed to test discoverability.

3.3 Conclusion

In this chapter, we examine core concepts and factors impacting initial interactions and the discovery of interactions. We propose a framework of impacting factors to aid a more systematic consideration of initial interactions and discovery and, in doing so, differentiate between the actors and factors involved. The relevant actors were identified as the system, user and environment, each with distinct factors influencing the likelihood, success and progression of an initial interaction. In doing so, we not only provide a collection of factors to consider and leverage when aiming to improve discoverability but also reflect on how they interrelate.

Part II

Using Visual Signifiers to Improve the Discoverability of Interactions

Given the aim of this thesis to improve the discoverability of interactions in touch screens, we examine the framework introduced in chapter 3 to identify which factors to leverage to increase discoverability. As we aim to improve interaction discoverability in general rather than in specific social contexts or usage environments, we focus on factors within the *system* (subsection 3.1.1). Examining these factors reveals a failure in current commercial systems in conveying their affordances. While touch interactions may be constraint, they can be inconsistent and are almost never visually signified which is reflected in the motivation for this thesis discussed in chapter 1. This is further exemplified by the rarity of interaction signifiers in commercial touch-screens. Nonetheless there are exception, like 'Tap', which as the most common input is often communicated through buttons presented as outlines, icons, text, or geometric shapes featuring any of the previous [99, 100]. However, the majority of inputs are never visually communicated and are subsequently hard to discover. For example, while 'Dwelling' on touch screen elements is widely available (e.g. dwell to select text or open menus, dwell and drag app icons or mobile keyboard spacebar) this is seldom conveyed to the user. Similarly, the interaction possibility through 'Swipe' gestures is rarely directly communicated to users. While it is occasionally implied through context like partially displayed items at the edge of the display indicating they can be brought into view [178], it is more often not conveyed at all. For example, in the Gmail app users are able to 'Tap' the profile icon to open a popup and switch between accounts. However, they are also able to swipe up or down on the icon to switch to the next account, which is never signified and subsequently largely unknown.

Given this lack of visualisation, we isolate visibility as a key contributing factor to the current lack of interaction discoverability and subsequently as a large potential for improvement.

Increased visibility may be achieved through various means, such as onboarding, increased instructions or signifiers in the interface. While onboarding [104] can assist users in perceiving an affordance, it is usually presented at an inappropriate *timing*, when the interaction is not needed, and requires users to remember it for later use. Supplementary materials, such as instruction manuals on websites, require an explicit search from users and are *away* from the GUI which might explain why they are often not read [22] and similarly require memorisation. Other material like online tutorials or demonstrations in keynotes [144], which only the most tech savvy users would watch, suffer from the same problems of physical and temporal distance.

In an effort to reduce these distances from the point of interaction, we focus on how visual signifiers *in the interface* can improve the discovery of input methods on touchscreens. Given the theoretical affordance of all touch-gestures on a touchscreen we in particular examine how to visually signify *which* input is available *where* and *when*.

The following two chapters explore the potential of visual signifiers to improve the discoverability of interaction through practical implementations of added visual signifiers which were then examined through different user studies. We first investigate visual signifier characteristics through introducing a design space, creating designs, and examining their ability to communicate single finger in-place inputs. We then investigate the viability of incorporating transient interaction hints into animated transitions to improve the discoverability of swipe-revealed hidden widgets.

4 Investigating Visual Signifier Characteristics Communicating Input Availability

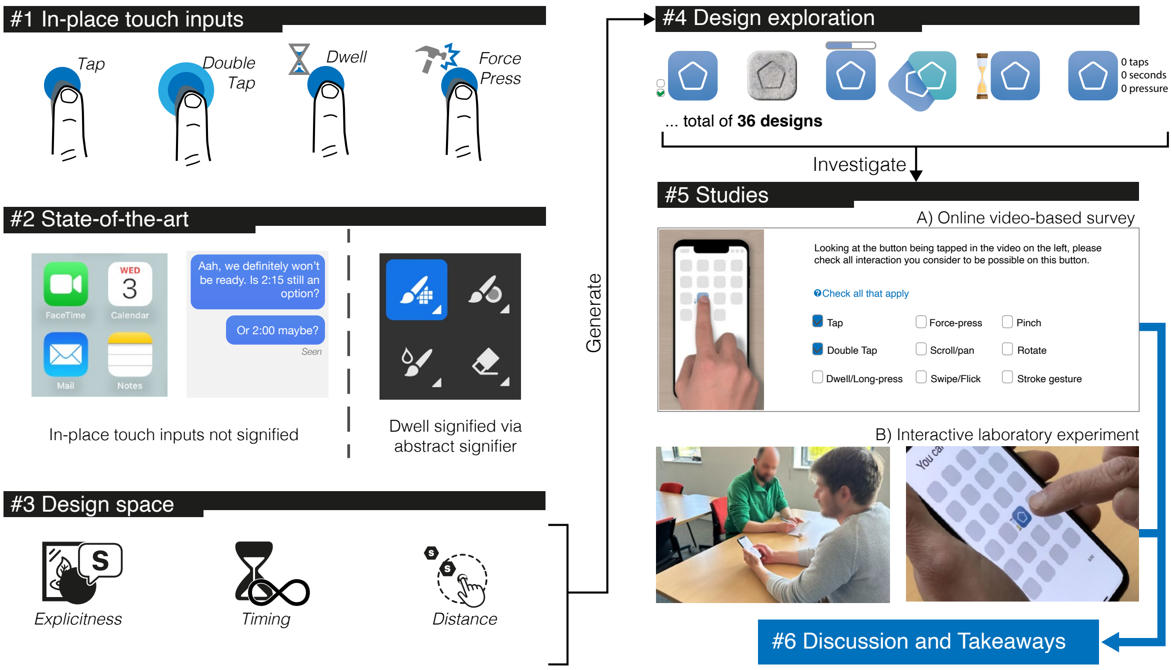


Figure 4.1 *– 1) In-place touch inputs (tap, double-tap, long-press and force-press) are common in modern touch-based devices yet 2) these inputs are almost never signified by the interface. 3) I elaborate a design space of visual signifier characteristics that may impact perception of these inputs, 4) generate 36 designs of buttons that I contrast in 5) an online survey (N=32) and an interactive laboratory experiment (N=24), resulting in 6) takeaways on how touch-based interfaces could be designed to better communicate the availability of in-place touch inputs.*

4.1 Introduction

Some core input gestures like tap, dwell or double tap are widely known and established and therefore likely to be familiar to users [42]. Nonetheless, it is often unclear where in the interface these inputs are available. For example, although we can expect users to be familiar with the input of dwelling on a widget, they may fail to deduce in which context this interaction is meaningful. Typically, users can dwell on application buttons on home-screens or messages in messaging applications to access shortcut menus. But they can also dwell on their camera feed, gallery button, quoted tweets and Slack reactions to name a few. Knowing one of these does not automatically allow the user to extrapolate the others, especially since these UI elements also allow interaction via a simple 'Tap'. This creates uncertainty, leaving a heavy burden on the user to accurately anticipate when and where a given input is available. Moreover, interfaces frequently changing and new input techniques being intro-

duced [11, 16, 85, 189, 190] make these challenges more and more persistent.
In this chapter, we investigate the possibility of reducing the burden on users of discovering and remembering inputs through visual signifiers in the interface. In particular, we investigate which signifier *characteristics* may be suitable in conveying different inputs. We focus specifically on expanding the perception of *in-place touch inputs* (Figure 4.1), single finger contacts that do not move across the surface. Specifically, we focus on variance in duration (*Dwell*), frequency (*Double Tap*) and Force (*Force Press*).
We propose a design space to investigate visual signifier characteristics and how they may influence the success of signifying different inputs. This design space encapsulates high-level principles that materialise in the dimensions *Explicitness*, *Timing* and *Proximity*.
Exploring the design space, we create 36 DESIGNS and examine how they influence the perception of available inputs in a two-step study. First, we conduct an online survey in which participants are presented with a video of a finger tapping the button and asked which inputs they consider possible. Second, we conduct a laboratory experiment where participants interact with a smartphone displaying the different DESIGNS. Participants are instructed to interact with the widget featuring the DESIGNS in whatever way they consider possible, while making the least errors possible.
Altogether, the results in this chapter inform that different signifier designs suggest different inputs to users and that most added visual signifiers increase the perception of additional input possibilities. Additionally, participants indicated a preference for visual signifiers over a signifier-less baseline and reported DESIGNS with added visual signifiers to require less mental effort. Further, we found that real world metaphors are particularly successful when signifying less known or ambiguous *inputs*, and that timing is especially impacting when presenting unusual *inputs*. Finally, we found that where designs are ambiguous but strongly suggest an input, initial perceived affordances such as 'Tap' are discarded in favour of what is suggested, warning to not inadvertently remove perceived affordances when trying to add others.
In summary, in this chapter we introduce a design space for visual signifiers, investigate the impact of its dimensions on user perception of possible in-place touch input methods and examine which signifier characteristics are particularly suitable, or unsuitable, to communicate additional input possibilities. In doing so, we provide takeaway considerations for future implementations of visual signifiers.

4.2 Related Work

While touch screen widgets often support inputs beyond tap, these alternative input methods are largely not signified. For examples, iOS home screen buttons support the use of 'Dwell' and 'Force' (on iPhone models with force sensing capabilities) to open a context menu (Figure 4.2-a). Similarly, the availability of 'Double Tap', while available to 'Like' content on several social media apps, select text, edit messages in Slack, or zoom in on maps, images and video feeds, is never signified. An exception is in the mobile application Adobe Fresco Version 4.4.2 (2023), which permits users to dwell on menu buttons to access tool specific menus. This is signified by a small triangle pointing outward the right corner of the buttons in question (Figure 4.2-b). This mirrors the same functionality and signifier in their desktop applications.

In contrast, desktop interfaces alongside many of the same signifiers as mobile interfaces (such as buttons, coloured text and progress bars) feature a cursor as an important visual signifier. Since hovering over widgets is possible, the cursor can signify which input is appropriate (e.g. a caret for text input, hand for click) while widgets can indicate interactivity through animation and text pop-ups. Despite this additional visual guidance, the appropriateness of inputs such as double click is still commonly misinterpreted. As there are no visual signifiers indicating which items on screen can be double clicked, it is up to the user to determine when the input is appropriate, leading to interaction mishaps [59, 185] named "Double Dysclicksia" [185]. Once established, a misunderstanding of when 'Double Click' is useful and necessary persists, with users wrongly assuming the need to 'Double Click' to open links [93].

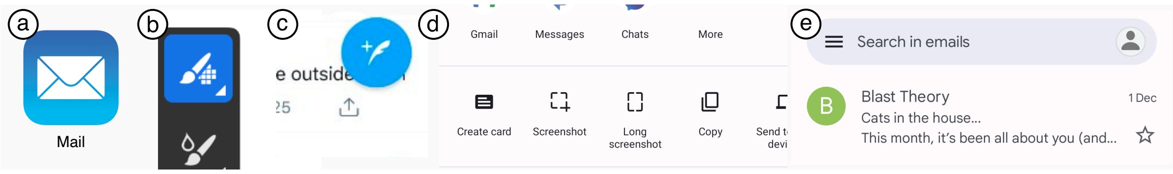


Figure 4.2 – Examples of visual signifiers in current mobile interfaces. a) iOS home-screen application button b) Adobe Fresco menu c) Twitter floating 'Add Tweet' button d) Share panel on Android OS e) Gmail mobile interface

4.2.1 Communicating Input Possibilities

Since core gestures such as 'Dwell' or 'Swipe' are available across devices and platforms, the assumption is that users will discover and understand them independently [42]. However, as input availability is inconsistent between contexts, there is a vast burden on the user to discover and recall interactions.

Previous research suggests that if novel input methods are communicated, this may take the form of explicit instructions through tutorials and pop-ups, or the form of visual depictions, with users indicating a preference for depiction [144] and a dislike of supplementary instructive material [22]. Nonetheless, input opportunities on touchscreens are often not signified at all, and bet on their commercial establishment to reach the point at which they are so conventionally known that their understanding is widely implicit. Examples are often found in keynotes or advertisements, that introduce an interaction (e.g. Steve Jobs introducing *touch-based scrolling* to great excitement of the crowd [102]).

For depictions, animation has been found efficient [7, 197] while potentially also being a distraction [144]. Additionally, layout and element sizing on touchscreens have been found to impact perceived affordances of touch gestures [108].

When examining visual depictions, different design styles such as *skeuomorphism* and *flat design* are often debated. It generally revolves around the assumptions that skeuomorphic design, that mimics the appearance of similar functionalities in the physical world, conveys affordances more successfully, while flat design, more minimalist and depthless, is more aesthetically pleasing [203]. However, in recent years, studies found no significant difference in performance between the two design styles [108, 173, 203]. They suggest that differences may not be attributed to design styles at all, but rather to the general quality of individual designs [173] or the level of abstraction or novelty of an object [108]. It should be noted that a design classified as skeuomorphic does not necessarily reference physical objects successfully and potentially invokes false affordances through *faulty metaphors*, one of the reason why metaphors remain debated in the context of interaction design [84, 163]. Similarly, flat designs are not necessarily detached from references to real world affordances and may employ symbolisms that aim to invoke physical qualities of real-world objects.

4.2.2 Signifying interaction in HCI Research

While much research aims to "expand" [89, 190] or "extend" [23, 90] the interaction vocabulary for touch interfaces, very little is said about how it is established or communicated to users. This omission is common not only in research but also commercial products.

Nevertheless, some research expanding interaction, includes considerations of visual signifiers, either reflecting about how to signify interaction and propose a design [23] or examining how to communicate proposed interactivity in more detail. For example, visual signifiers have been explored in the context of force-based text selection [78], undo mechanisms [12] and digital tabletops [197]. While they studied the success of the interaction method they support, they however do not include separate considerations on the visualisation and its potential impact. Nonetheless, some found the method including visualisation to be more successful than the standard it was compared to [78] while others revealed user difficulties with comprehending novel interaction solely on the basis of visualisation [197]. Interestingly, some visualisation attempted to mirror real world metaphors [12] while others employed established visual concepts such as gauges [78]. In general, users appreciated visual feedback for how their interaction was registered by the system [78, 197]. Visual characteristics of signifiers have been explored in the context of directional selection gestures [232], however, while different colours and shapes were considered through qualitative feedback from participants, they only collected quantitative measures regarding the minimum size required to perceive how to interact with their UI element.

Similarly, studies have been conducted to investigate signifiers for existing touch gestures. Arleth et al. explored the effect of temporal placement of added visual signifiers for 'Dwell' and 'Double Tap' being shown before, during or after a gesture was completed and found neither temporal placement, nor the addition of the signifier in general to be significant [14]. The signifiers employed in their study, were a pulse of two rings expanding and then disappearing after one another for 'Double Tap' and a gradually filling ring for 'Dwell'. Since these were the only designs investigated, and were not validated, it is unclear whether the lack of impact of temporal placement may be partially attributed to misunderstanding of the visualisations rather than the time of their appearance. In contrast, Damkjær et al. focus on different visual depictions when examining signifiers for 'Drag' and 'Double Tap' [48]. They included two designs for each input method, using a 'drag handle' and drop shadow for 'Drag' and two borderlines and a pulse animation for 'Double Tap'. Notably, both their 'Drag' signifiers mirror existing digital interface designs and are static with no feedback animation, while the 'Double Tap' signifiers are temporary and feature animations, making comparisons between inputs difficult. Similarly to Arleth et al., they found no significant difference between the signifier designs and a control. They note that 'Drag' signifiers were more efficient than 'Double Tap' overall and highlight that this may be attributed to higher familiarity with 'Drag' in general, which they do not control for.

In conclusion, some work has been done to explore individual facets of signifiers and their impact on communicating the possibility of in-place touch inputs. In this chapter, we propose a more detailed look at signifier characteristics for diverse in-place touch inputs.

4.3 What makes a signifier successful?

While signifiers have been identified as a requirement of *"good design"* [167] there is a lack of clarity of what constitutes good designs for signifiers themselves. Equally, their success is seen as a binary - something is successfully signified, or not. To allow for a more granular investigation, we define three performance levels for interaction signifiers:

1. Does not convey the intended interaction(s).
2. Conveys the intended interaction(s) but may convey other, non-intended ones.
3. Conveys only the intended interaction(s).

These can then be further separated by differentiating between users who are previously familiar with the signified interaction method and those who are not. The most successful interaction signifier would, e.g., convey specifically this interaction, even to people who were previously not aware of the existence of the interactions, *i.e.* conveys only the intended interaction to (a) individuals unfamiliar with the interaction or (b) individuals already familiar with the interaction. Furthermore it should be considered whether the added signifiers reduce the perception of established inputs therefore interfering with conventional tasks.
When testing and comparing signifiers, these can then be quantified, making the most successful signifiers those that communicate the intended interaction to the largest number of people regardless of their previous familiarity with it, while not interfering with established interactions.

4.4 Design Space for Visual signifiers

To explore how visual signifiers convey specific interaction possibilities, we propose a design space focused on abstract characteristics that can then be realised regardless of aesthetic considerations (unlike colour or style [108, 173, 203]). In doing so, we compile signifier characteristics that have previously been considered in isolation [14, 48].
As visibility is an essential part of signifiers in graphical user interfaces, we consider when (*timing*) and where (*proximity*) a signifier is displayed in relation to the interaction. In addition to these concrete concepts, we examine the level of interpretation expected from users to understand the intended meaning once they have noticed the signifier. Therefore, we include the more conceptual characteristic *explicitness* in the design space. In doing so, we do not consider visuals beyond what certain metaphors may dictate in an effort to allow abstraction and application of the design space to any desired context regardless of design requirements. Similarly, we decided against including *context* as a design space category despite its relevance. This decision was made as (1) context itself is a signifier and often times a major contributor to convey meaning to users, (2) it is not part of the widget itself and (3) contexts are way too diverse to be sufficiently explored in a first study. As such, it would broaden the design space excessively and would complicate its exploration. Nonetheless, it would be remiss not to acknowledge the importance of context for the interpretation of visual signifiers. Subsequently, we provide the same context through a grey grid layout in all signifier designs.

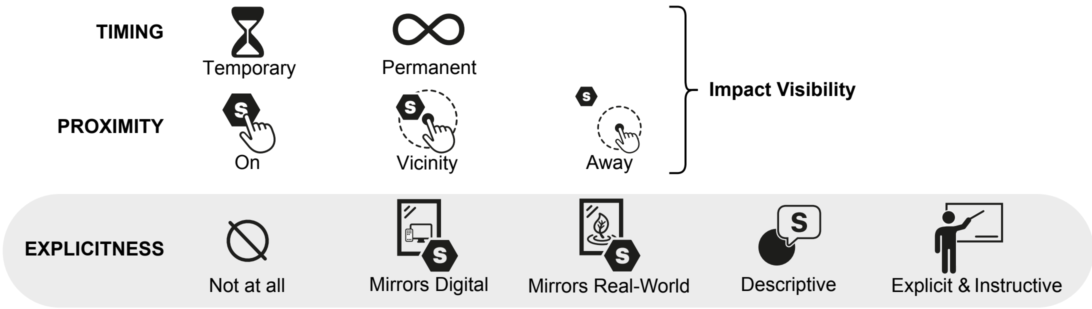


Figure 4.3 – Visual representation of the signifier design space featuring the characteristics Explicitness, Timing *and* Proximity. *Each characteristic then includes the levels considered for it which are represented through text as well as small graphical representations. Where characteristics are considered to be influencing the visibility of a signifier is indicated through the curly bracket.*

Timing. Describes when and for how long a signifier is *displayed* by the interface. The main separation considered in this respect is between signifiers that are permanently displayed and those temporarily displayed (concurrently to interaction). Notably, concurrent interaction feedback within permanently displayed signifiers may still be temporary, while the overall signifier is displayed at all times (e.g. a permanent displayed scale that fills upon interaction). As such, the permanently displayed parts of the design may be considered feedforward informing the potential interaction before it occurs. Our consideration of when the signifier is displayed is made relative to the interaction and is therefore separated into before (feedforward), during (concurrent feedback) and after (terminal feedback) interaction [55].
We consider signifiers to be *permanent* if displayed before, during, and after the interaction; and *temporary* if displayed during and immediately after the interaction. A finer consideration between different levels of temporary visibility exceeds the scope of this study but may be worth considering in future work.

Proximity. We consider proximity to the point of interaction, which can influence the perception of a signifier. It can highlight the connection of the signifier to the widget in question by being close as well as increase the likelihood of the signifier being noticed by being displayed within the area of the users focus. On the contrary, close proximity can also lead to the signifier being occluded by the touch interaction. We propose three levels of proximity. *On the widget*: the visual signifier is superimposed on the widget in question, in our case the button. *Vicinity of the widget*: the signifier is displayed in the area immediately surrounding the widget, but not the widget itself. *Away from the widget*: the signifier is visible elsewhere on the display but not connected to the widget (e.g. the edges of the screen).

Explicitness. Describes how clearly the interaction is communicated to the user and, consequently, how much interpretation, and possibly mental effort, is expected. We propose 5 levels of explicitness: *Not at all* explicit, *Mirror established digital signifiers*, *Mirror physical affordances*, *Descriptive* and *Explicit and Instructive*.

Not at all: signifiers have no link to the interaction intended to be signified. Examples are the complete absence of visual signifiers in Swhidgets [177] or force on the space-bar of iOS keyboards [78].

Mirror Established Digital Signifiers: apply commonly understood digital signifiers to communicate interactivity in a different context. Examples are progress bars, widely used to signify loading content or level of completion.

Mirror Real-World Affordances: aims at leveraging users' understanding of physical objects and processes in order to communicate appropriate interactions. Examples are controls designed to resemble physical buttons or switches, as well as slightly more abstract examples like "stacks" of images or documents that can be moved through by swiping them as if leaving through a physical collection of paper. We do not attempt to create skeuomorphic or photorealistic designs but attempt to take advantage of the users' familiarity with physical objects to improve guessability [103, 228].

Descriptive: show how the system is considering the input but do not instruct appropriate interaction. A typical example are countdowns that may show that the system is recording the progression of time but do not elaborate on how the system further utilises this information, its outcome, or which interaction it triggers.

Explicit and Instructive: unambiguously communicate interaction possibilities to the user and instruct them on how to use them, typically by using text. An example of this is mobile-app onboarding [104] which includes concrete instructions and information on how to interact with and use an application.

Note that these levels may vary in effectiveness depending on the user and their previous experience and therefore should be considered as categorical and not ordinal.

4.4.1 Current signifiers in the design space

As the most common interaction, 'Tap' is signified most often. Especially popular for signifying 'Tap' are buttons (Figure 4.2) which are signifier designs *mirroring established digital signifiers* that are *permanent* and *on the interaction widget* (since they largely consist of simply the visual depiction of the interactive widget). Example Figure 4.2 a), b) and c) signify being tappable through shapes mirroring digital conventions while conveying the effect of the button through mirroring real-world objects. However, a) also supports 'Dwell' and 'Force' that are not visually communicated. Nonetheless, dragging is signified once a user dwells through the icons, starting to 'wriggle' in place, indicating they can be moved. The moving animation mirrors the real world, is concurrent with the interaction and on the widget.

The profile icon in example e) is a button signifying 'Tap' by mirroring established digital signifiers, however the button also supports swiping vertically to switch between accounts, which is not at all signified. Similarly, both example d) and e) allow the user to swipe, while only d) communicates this visually by showing parts of the controls accessible through a horizontal swipe at the corner of the screen. Subsequently swiping on e) represents a lack of signifiers, or a 'Not at all' explicit signifier. The categorisation of d) can be argued, as the partially shown menu option somewhat resembles sliding puzzles mirroring real world affordances but also functions to visually describe its function.

The button in example b) is a rare example of a button communicating not only 'Tap' but also 'Dwell' visually. The arrow in the bottom right corner pointing outwards is well established in desktop applications for changing tools via a right click or a long press, particularly within the Adobe Creative Suite. While this signifier is not adapted in most mobile adaptations of these applications (e.g. Photoshop, Lightroom), Adobe Fresco 4.2.1 on iOS 16.2 utilises it similarly to desktop applications where a short tap simply selects the tool while a longer press opens the tool selection. This signifier can therefore be classified as mirroring established digital signifiers, permanent and on the widget.

4.4.2 Filled Design Space

We use an approach inspired by the *research through design* methodology [236] in which we produce various designs to explore and reflect on the characteristics of the design space. More specifically, we created signifier designs for the three selected interaction methods (Double Tap, Dwell, Force Press) combining different levels of explicitness, timing and proximity. The created DESIGNS are instances of a signifier, featuring characteristics explored in the design space. To limit confounding factors, we used buttons, which are widely used and can be expected to be familiar to most users. To account for the 'Not at all' explicitness level, we included a condition of a simple button with no added signifier.

Furthermore, to avoid redundancy in designs, we combined some descriptive and all instructive designs to address all the interactions explored at the same time. While explicit instruction can also take the form of audio or video content, we limited our designs to text and explored the other dimensions (Timing and Proximity) in more detail with this text.

Outside these constraints, the design space was filled semi systematically, with all combinations being considered but those presenting particular difficulties not being implemented. Nonetheless, we created a broad group of designs (36) which are presented in figure 4.4, covering the design space broadly and fulfilling the majority of possibilities. The only designs which have been previously implemented or tested, are 01 and 24 which are based on commercial interfaces, and 15 and 25 which are adapted from existing signifier research [14, 48].

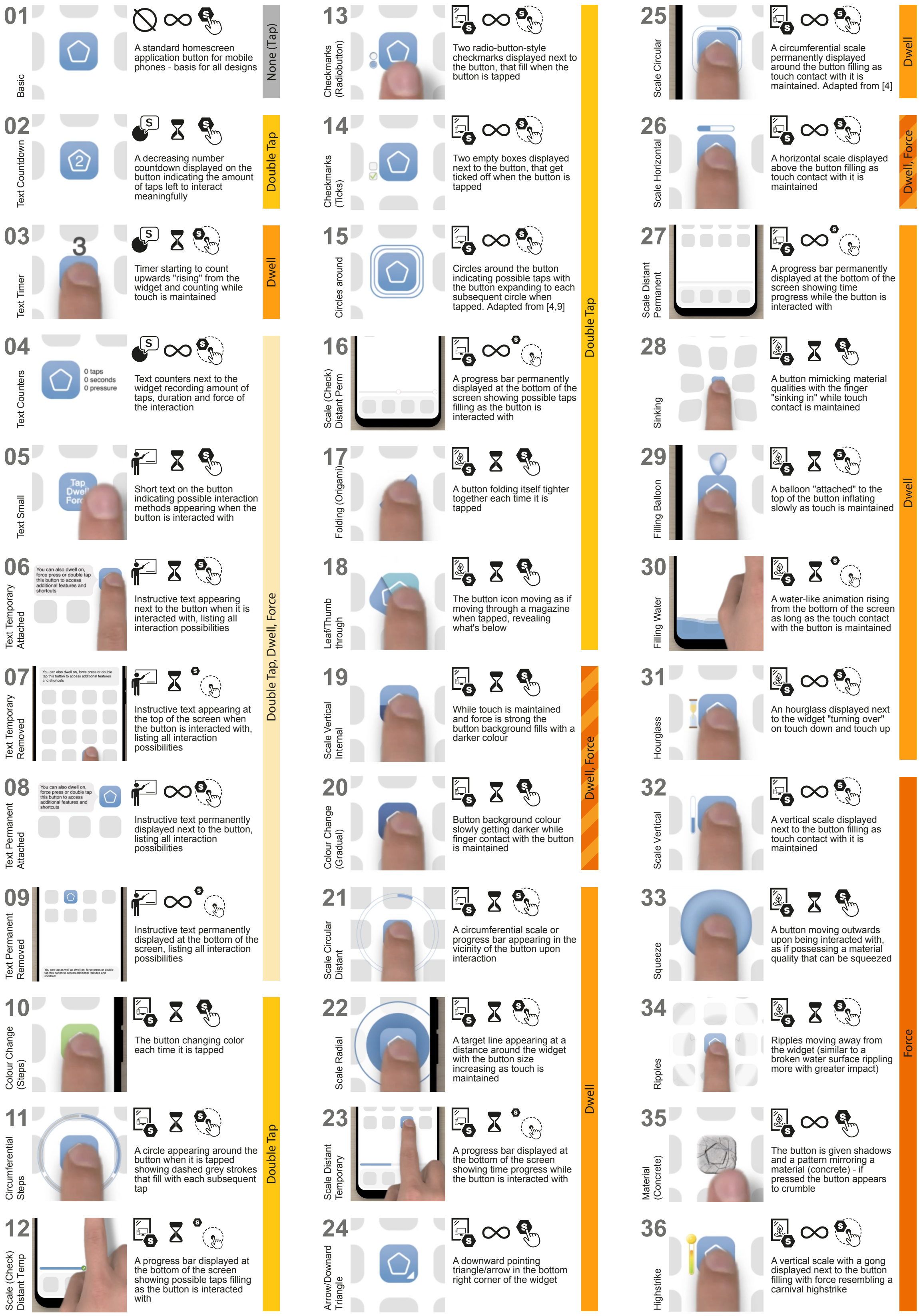


Figure 4.4 – Table of designs filling the design space. Each design is displayed alongside the design space icons that apply to it and a text description. The vertical strip on the right displays the assumed signified interaction.

4.5 User studies

We investigate the following questions regarding which *inputs* each DESIGN communicate.

- **RQ1 -** Do added visual signifiers increase the perception of available *inputs*?
- **RQ2 -** Which *inputs* are communicated by which DESIGNS, and therefore what are the perceived affordances?
- **RQ3 -** Do the DESIGNS convey the intended *inputs*?
- **RQ4 -** Does the success of signifiers depend on previous familiarity with the *input* in question?
- **RQ5 -** Do signifiers reduce the perceived mental effort required?
- **RQ6 -** Do users prefer the presence of signifiers over standard widgets?
- **RQ7 -** Does Explicitness, Timing and Proximity impact the success of a DESIGN?

To answer these questions, we conducted 1 - an online survey followed by 2 - an in-person user study. Beginning with an online survey presents two main benefits. First, it provides relatively rapid insights regarding which *inputs* each DESIGN communicates, without biasing users' perception of *input* viability nor suffering from demand characteristics [172] resulting from interpersonal contact between researchers and participants. Second, it informs which *input* to associate as *correct* for each DESIGN for the second, in-person, user study. Then, conducting an in-person study investigates not only which *inputs* users perceive as possible, but which ones they actually perform when exposed to each DESIGN. Altogether, these two studies provide significant elements of answer the above research questions.

In the following, we describe the methodology and design of each study.

4.5.1 Study 1 - Online Survey

We first conducted a remote online survey to establish which *input* participants perceive as possible with a given DESIGN, and confirm whether the DESIGNS communicate the *input* we assumed they would when implementing them (*assumed input*), thus justifying a sufficient coverage of all in-place touch inputs.

4.5.1.1 Methodology

The survey relies on videos that present each DESIGN implemented in a button, placed on a grid similar to a smartphone home screen, and being 'tapped' by a finger (Figure 4.5). For each DESIGN, participants are invited to select from a list all *inputs* they consider to be possible with the button.

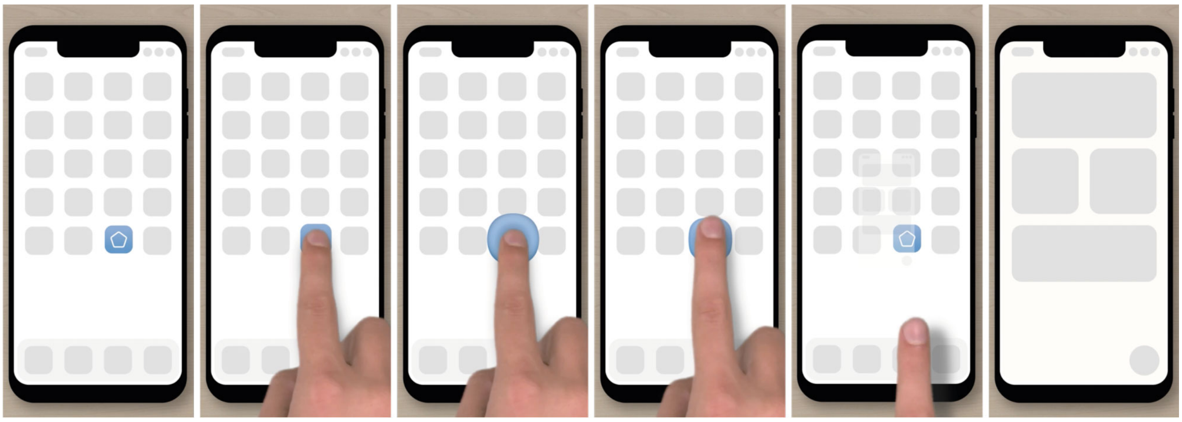


Figure 4.5 – *Setup for presenting* DESIGNS *in the online study. Images show how the button and its context are displayed throughout the videos depicting them. Each design is shown by itself, before a hand moves into frame and taps the button. In response to the tap, an application opens as the hand moves out of frame.*

4.5.1.2 Presenting DESIGNS

Each video shows a finger tapping the button, which triggers a view to open. All videos are 6 seconds long and feature the same tap animation, while the position in the grid and mockup application opening varied. This variation was implemented to clearly distinguish designs from one another, after initial testing of the survey revealed that participants were confused by the changing visual characteristics of what they perceived to be the same button. To minimise variance, interaction feedback is timed to be the same length for all the design visualisations.

DESIGNS are all presented as a button in the same mobile homescreen-esque grid to suggest interaction taking place on a touchscreen. Also, while concurrent and terminal feedback of each DESIGN may vary, we provide the same 'expected' outcome to the tap interaction by revealing another screen (Figure 4.5). Finally, DESIGNS are implemented in a basic button whose shape and icon remain the same, mirroring a mobile phone homescreen application button.

4.5.1.3 Procedure

Participants accessed the online survey using their own computer. After being informed that the survey examined different interface designs for touch-based interactions, participants were asked to provide informed consent and to answer demographic questions. They then indicated their familiarity with Android and iOS on a 7-point Likert item ranging from *1 - None* to *7 - Extremely High*. Then, they were presented with a list of touch-based inputs (Tap, Double Tap, Dwell, Force-Press, Scroll/Pan, Swipe/Flick, Pinch, Rotate, Stroke Gesture) each including a text description and video illustrating the interaction, for which they indicated their familiarity with and usage of on a similar 7-point Likert item. Subsequently, the main body of the survey began that displayed, for each DESIGN, two main parts (Figure 4.1, #5-A). On the left, a video-loop where a finger taps a button implementing the corresponding DESIGN, thus showcasing possible concurrent and terminal feedback one would witness if tapping it. On the right, one checkbox for each common touch screen input method (Tap, Double Tap, Dwell / Long-Press, Force-Press, Scroll / Pan, Swipe / Flick, Pinch, Rotate, Stroke Gesture). Participants were instructed to check all *inputs* they considered to be possible with the DESIGN shown in the video. When done, participants proceeded with the next button, displaying a similar page for the following DESIGN. The study ended after participants answered for all DESIGNS.

4.5.1.4 Participants and apparatus

The survey was implemented in Limesurvey 5.3.12. 32 participants recruited via direct message completed the survey. They were aged 22 to 58 years (M=31.7, SD=9.6), with 30 of them indicating that they use a smartphone at least once a week. 21 of them reported a higher familiarity with Android, 7 with iOS and 2 similar familiarities. Overall, they indicated a high familiarity level with their favourite mobile OS (Mdn=6, SD=1.09).

4.5.1.5 Experimental Design

The survey was set up to test our main independent variable DESIGN, with the 36 instances illustrated in figure 4.4. To control potentially confounding variables, presentation order was randomised for each participant. Similarly, to avoid biasing participants, response options were shown either in a "logical" or "random" order [1], but stayed consistent throughout the survey for each participant. This variation was added in order to evaluate the potential influence of presentation order on which answers were selected [113]. Additional independent variables emerge by grouping DESIGNS according to the three main dimensions of design space: EXPLICITNESS, TIMING, PROXIMITY.

In summary, the experiment followed a within-subjects design: 32 participants × 36 DESIGN = 1,152 forms filled. The survey took an average of 30 minutes per respondent to complete.

Our main dependent variable is ***Selection%***: for each *input* × DESIGN combination, it corresponds to the proportion of participants who considered this *input* to be possible with this DESIGN, regardless of whether this *input* was intended or if others *inputs* were considered to be possible. It is computed as the number of participants who selected this *input* for this DESIGN, divided by the total number of participants (compiled in a table in Appendix A.3).

For the additional independent variables EXPLICITNESS, TIMING, PROXIMITY, the measures for the dependent variable are first aggregated by computing the average *Selection%* for all assumed inputs for designs corresponding to a level of the design space categories per participant.

4.5.2 Study 2 - Interactive Experiment

We then conducted an in laboratory interactive user study.

1. The "logical" order starts with the most common interaction "Tap" and then groups interactions according to the fingers used as well as the similarity and complexity of the interactions. Logical interaction order: 'Tap', 'Double Tap', 'Dwell', 'Force-Press', 'Scroll/Pan', 'Swipe/Flick', 'Pinch', 'Rotate', 'Stroke Gesture'; Random interaction order: 'Scroll/Pan', 'Rotate', 'Stroke Gesture', 'Swipe/Flick', 'Force-Press', 'Double Tap', 'Tap', 'Pinch', 'Dwell/Long-Press'

4.5.2.1 Methodology

Setup

The experiment setup (Figure 4.1-5B) consists of a mobile application featuring a homescreen grid populated with grey squares with one DESIGN featured in the middle, mirroring the online study. In the bottom right corner a progress counter is displayed. At the top, a text is displayed which states "You can start" until the participant starts interacting with the button. If the supported *input* is performed, the text states "GOOD!" and a green button stating "Move to next trial" appears in the middle of the screen. When any other input is performed on the button, the text displays a counter of errors. After 6 errors, a red button stating "Move to next trial" appears. This repeats for all DESIGNS with the counter at the bottom indicating progress throughout.

Presenting Designs

All DESIGNS are presented interactively, i.e. if they feature animated elements they react when interacted with. We split the DESIGNS featuring all *inputs* in the design space in Figure 4.4 (04, 05, 06, 07, 08, 09) into individual DESIGNS for 'Double Tap', 'Dwell' and 'Force Press' (e.g. 04A 'Double Tap', 04B 'Dwell', 04C 'Force'). Similarly, we include the '01 Basic' three times, once for each intended *input*. This leads to 21 DESIGNS to be included (7 times 3) in addition to the other 29 DESIGNS resulting in a total of 50 DESIGNS included in the experiment. The interactive nature of the experiment requires *inputs* to trigger an event that is either correct or incorrect. We determined a 'correct' *input* for each of the other 29 DESIGNS by following a 'draft pick' system to have all *inputs* represented equally. In the order of 'Double Tap', 'Dwell', 'Force' the *inputs* were assigned the DESIGN with the highest *Selection%* for the *input* from the online survey, removing the DESIGN from the pool to be picked from, until all DESIGNS were assigned to an *input*. This resulted in 10 additional DESIGNS for 'Double Tap' (02, 10, 11, 12, 13, 14, 15, 16, 17, 18) and 'Dwell' (03, 19, 22, 23, 24, 25, 27, 29, 30, 31) and 9 DESIGNS assigned to 'Force' (20, 21, 26, 28, 32, 33, 34, 35, 36).

4.5.2.2 Procedure

Mirroring the procedure in section 4.5.1.3, participants were asked demographic questions as well as about their familiarity with mobile operating systems and interactions, however 'Scroll', 'Pinch' and 'Rotate' were removed from the rating, as they were not likely to be considered according to the results of study 1. Following this, participants were given a smartphone running the application described in section 4.5.2.1. Participants were instructed to progress through all DESIGNS while making the least errors possible, and encouraged to comment on their thought process throughout. They were told that one of the *inputs* they just indicated familiarity with would achieve this and that it may vary between DESIGNS, but were given no further guidance on how to achieve this beyond the visual signifiers in the DESIGNS themselves. At the end of the experiment, participants were asked to rate the mental effort demanded to reach the correct *input* for all DESIGNS, as well as their personal preference on a 7-point Likert item.

4.5.2.3 Participants and apparatus

The survey part of the experiment was implemented in Limesurvey 5.6.1, and the experiment in typescript, using PixiJS 7.2.2 for the rendering. It was conducted on an iPhone XS running iOS 16.4 which supports force sensing. Each design was created using Adobe After Effects and exported as an image sequence to create animations based on the input from the user. Input was classified using the the state machine presented in Appendix A.1.

Participants were recruited through direct messages. The experiment was completed by 24 participants (different from study 1 except for 2) aged 20 to 59 (M=27.8, SD=9.8) with all of them indicating that they use a smartphone at least once a week. 4 had a higher familiarity with Android, 12 with iOS and 8 an equal familiarity. Participants rated their overall smartphone expertise as Mdn=5, M=5.33 on the 7-point Likert scale, with the lowest being 4 - Average.

4.5.2.4 Experimental Design

Our main independent variable remains DESIGN. However, in cases where multiple *inputs* were combined in one single DESIGN, they were split into separate DESIGNS, leading to 50 instances, as described in section 4.5.2.1. We maintain the additional independent variables EXPLICITNESS, TIMING, PROXIMITY. The experiment followed a within-subject design with 24 participants × 50 DESIGNS = 1200 DESIGNS interacted with. It took an average of 21 minutes to complete. We used a pseudo random order of presentation with the condition that the 'correct' input could not be the same more than twice in a row. The presentation order was inverted for half of the participants.

Our dependent variables are *Attempt%*, *AttemptNumber*, mental effort rating and personal preference rating. For each *input* × DESIGN combination, *Attempt%* corresponds to the proportion of participants who attempted to interact through this *input* with the given DESIGN, regardless of whether this *input* was correct. This mirrors *Selection%* in study 1. *AttemptNumber* for each DESIGN corresponds to the interaction attempt at which participants performed the *correct input* for the given DESIGN. If participants did not perform the *correct input* within the 6 trials, *AttemptNumber* was saved as 7, representing 'Failed'. Mental effort and personal preference ratings were measured on 7-point Likert items.

4.6 Results

First, we examined the data from both studies for outliers. For study 1 we did so by examining the total number of selected checkboxes per participant (out of the maximum 36 DESIGNS × 9 interaction methods = 324). For study 2 we examined the total trials performed per participant (out of the maximum 50 DESIGNS × 6 permitted trials = 300). We found no outliers in either study and subsequently proceeded using all participants.

Then, we examined a potential effect of order for both studies. For study 1, we examined whether the order of response options impacted the results and compared the selection of 'Tap' between the two response order groups. Tap selection was used for this comparison as it is the most basic and established interaction method for mobile touch interface buttons. 'Tap' was presented first in the logical order and 7th in the random order, but no significant difference in selection amount was found between the logical and random order (Mann-Whitney U = 109.5, p ¿ 0.05). For study 2, we examined the trials required to perform the correct input for each DESIGN between the two order groups, and found a significant difference for only one DESIGN (Design 15 - U=39, p=0.04). Since the DESIGN in question is in 23rd place in the regular order and 27th in the reverse order, while the other 49 DESIGNS with larger variance in placement show no significant difference, we decide to ignore this minor effect. Therefore, we analyze the results of both studies by considering all participants as a single group.

4.6.1 RQ1 - Do added visual signifiers increase the perception of available inputs?

Visual signifiers, in general, do increase the perception of available *inputs*.

When examining which *inputs* were assumed to be available in study 1, 12/16 DESIGNS created for 'Double Tap', 19/20 DESIGNS created for 'Dwell' and 14/14 DESIGNS created for 'Force Press' increase the *Selection%* for the *input* in comparison to the baseline '01 Basic' ('Double Tap' 47%, 'Dwell' 56%, 'Force Press' 31%).

Similarly, in study 2, 13/16 DESIGNS for 'Double Tap', 12/16 DESIGNS for 'Dwell' and 13/15 DESIGNS for 'Force' increase the *Attempt%* for the *input* in comparison to 01A, 01B and 01C. However, note that the results of study 2 are less indicative, given that the participants were only given a chance to interact until they performed the correct *input* for a given DESIGN.

When considering which *inputs* participants attempted first in study 2, out of the 1200 total DESIGNS interacted with, the first *input* performed was 'Dwell' (400 times), 'Tap' (393 times), 'Double Tap' (173 times), 'Force Press' (155 times) and 'Swipe' (79 times), showing that participants frequently attempted different *inputs* than 'Tap'.

4.6.2 RQ2 - Which inputs are communicated by which design, and therefore what are the perceived affordances?

Examining the *Selection%* across *inputs* for each DESIGN in study 1, we found clusters of DESIGNS communicating either discrete, continuous or ambiguous *inputs*. The *Attempt%* for the first two attempts only in study 2 showed how perceived affordances change beyond the initial perception of temporary signifiers but revealed no meaningful cluster.

4.6.2.1 Study 1 - Online Survey

To examine the similarity between DESIGNS in terms of *inputs* selected, we clustered the DESIGNS using K-mean clustering approach. Each design was considered as a vector of 9 values comprised of the *Selection%* for each *input* method for the given DESIGN. To determine the ideal number of clusters (k) we calculated silhouette scores. This resulted in 5 clusters that can be described as 'Tap' + DESIGNS that signify:

All 'Tap', 'Double Tap', 'Dwell' and 'Force Press' (DESIGNS 04, 06, 07, 08, 09)

Continuous 'Tap', 'Dwell' and 'Force Press'
(DESIGNS 03, 05, 19, 21, 22, 23, 25, 26, 27, 29, 30, 32, 33, 36)

Discrete	'Tap' and 'Double Tap' (DESIGNS 02, 13, 14, 15)
Ambiguous	'Tap' with ambiguous results for 'Double Tap', 'Dwell' and 'Force Press' (DESIGNS 01, 10, 11, 12, 16, 20, 28, 31, 34, 35)
Ambiguous+	'Tap' with ambiguous results for 'Double Tap', 'Dwell', 'Force Press' and 'Swipe' (DESIGNS 17, 18, 24)

The **All** cluster encapsulates DESIGNS where participants selected all *inputs* that correspond to in-place touch interaction. It contains all *Explicit and Instructive* DESIGNS as well as the *Descriptive* 04 Text Counters DESIGN. This cluster is explained by the fact that text-based DESIGNS were created to include all *inputs* to avoid excessive repetition.

The **Continuous** cluster contains DESIGNS for which participants selected 'Dwell' and 'Force' alongside 'Tap'. This includes DESIGNS created to convey one or both of the continuous *input* with most of the "scale" DESIGNS falling within this cluster. This grouping shows that the perceived affordance for continuous *input* overlaps, with most DESIGNS aimed to signify 'Dwell' or 'Force Press' signifying the other as well.

The **Discrete** cluster contains DESIGNS for which participants selected 'Double Tap' as well as 'Tap'. Examples of DESIGNS in this cluster, are the checkmarks (13 (Radio) and 14 (Tick)). This clustering indicates that there is a clear difference in perception of affordance of discrete *inputs* versus continuous *inputs*.

Meanwhile, both clusters **Ambiguous** and **Ambiguous+** include DESIGNS with high 'Tap' selection and middling selection percentages for 'Double Tap', 'Dwell' and 'Force'. Cluster **Ambiguous+** contains DESIGNS for which some participants additionally selected 'Swipe'. These DESIGNS were unclear in what they conveyed and, in the case of **Ambiguous+**, conveyed a non in-place touch input. Notably, *17 Folding* had a high selection rate for 'Dwell' (84%) but a low score for 'Force Press' as well as a relatively high selection rate for 'Swipe'

This clustering highlights the difficulty of achieving signifiers that successfully convey only the intended interaction(s) (see section 4.3). Since the *inputs* 'Dwell' and 'Force Press' are both perceived as continuous, the *Selection%* for them tends to rise in tandem. The highest difference between 'Dwell' and 'Force Press' *Selection%* can be seen in 3 DESIGNS. *03 Text Timer* which is a top performing DESIGN for 'Dwell' has a *Selection%* that is 40 percentage points lower for 'Force Press'. Interestingly, the other two DESIGNS featuring the highest difference between the two continuous *inputs* (*17 Folding* - 53pp difference, *31 Hourglass* - 44pp difference) are both mirroring real world affordances.

4.6.2.2 Study 2 - Interactive Experiment

Study 2 influences the perception of available *inputs* by expecting a 'correct' *input*. Therefore, we examine only the first attempt for each DESIGN and participants rather than the full *Attempt%*.

When examining which *input* has the highest *Attempt%* in the first interaction, for each of the 50 DESIGNS, 22 are 'Tap' (DESIGNS 01A, 01C, 03, 05A, 05B, 05C, 6B, 6C, 12, 17, 18, 19, 20, 22, 23, 27, 28, 29, 30, 33, 34, 35), 7 'Double Tap' (02, 04A, 08A, 09A, 13, 14, 25), 17 'Dwell' (DESIGNS 01B, 04B, 06A, 07A, 07B, 07C, 08B, 09B, 10, 11, 15, 16, 21, 26, 31, 32, 36), 3 'Force' (DESIGNS 04C, 08C, 09C) and 1 'Swipe'(DESIGN 24).

We then examined the *Attempt%* when looking at both the first and second interaction attempt rather than just the first, to incorporate participants perception following the first exposure to temporarily displayed DESIGNS. For 11 DESIGNS, 'Tap' achieves the highest *Attempt%* after 2 attempts (01A, 01C, 05C, 10, 12, 17, 18, 19, 20, 28, 29). For another 13 DESIGNS, it is 'Double Tap' (02, 04A, 05A, 06A, 07A, 08A, 09A, 13, 14, 15, 16, 25, 35), for 21 DESIGNS it is 'Dwell' (01B, 03, 04B, 05B, 06B, 07B, 08B, 09B, 11, 15, 19, 21, 22, 23, 25, 27, 29, 30, 31, 33, 34), for 8 it is 'Force' (04C, 06C, 07C, 08C, 09C, 26, 32, 36) and for 1 it is 'Swipe' (24). Note, that 4 DESIGNS have an equally high *Attempt%* after 2 attempts for 2 *inputs* ('Double Tap' & 'Dwell' for 15, 25; 'Tap' & 'Dwell' for 19, 29). Silhouette scores, revealed that there were no meaningful clusters, which may be due to no further interactions being performed once the *correct input* has been performed.

4.6.3 RQ3 - Do the designs convey the intended inputs?

Overall, the DESIGNS convey the intended *inputs*. Study 2 reveals, that, when compared to the baseline, multiple DESIGNS improve the perception of the intended *inputs*.

4.6.3.1 Study 1 - Online Survey

Using an arbitrary threshold of 75% to consider that a DESIGN communicates the assumed *input*, which means that above 75% of participants considered this *input* as possible to perform with the corresponding DESIGN, we found 22 DESIGNS that communicate at least one of the assumed *inputs*. Looking in detail for each *input*, we found 9/16 DESIGNS for 'Double Tap' (Mdn=75%, M=65%, SD=22%), 16/20 for 'Dwell' (Mdn=91%, M=85%, SD=13%), 9/14 for 'Force Press' (Mdn=77%, M=76%, SD=9%).

Note that looking at assumed *inputs* is not the ultimate aim of our study as, in the end, what matters is what *inputs* participants did report suggesting which affordances were perceived. Interestingly, there were 4 DESIGNS, which were not assumed to communicate 'Dwell', that had a *AssumedSelection%* over the 75% threshold (*11 Circumferential Steps*: 81%, *17 Folding*: 84%, *32 Scale Vertical*: 94%, *33 Squeeze*: 84%).

Further, it should be noted, that 5 designs (*17 Folding, 23 Scale Distant Temporary, 25 Scale Circular, 32 Scale Vertical, 33 Squeeze*) fall under the 75% threshold for 'Tap' which should be signified by the button itself and its context as well as the example interaction in the videos. Notably, all of these designs do score high selection percentages for other interaction methods, suggesting that strong signifiers for one input may reduce the perception of otherwise established inputs.

4.6.3.2 Study 2 - Interactive Experiment

Considering the same 75% threshold and only the *Attempt%* for the correct intended *input* for each DESIGN, only 2 DESIGNS ('Double Tap' 17 - Folding, 'Dwell' 19 - Scale Vertical Internal) fall under it for the interactive study. However, it should be noted, that the confirmation through interaction and the potential to arrive at the correct input through trial and error are confounding factors for these numbers.

When examining the *Attempt%* for only the first interaction performed by each user only 8 DESIGNS pass the 75% threshold ('Double Tap' 08A, 09A, 13, 14; 'Dwell' 08B, 09B, 31; 'Force' 04C, 08C). When extending this to the *Attempt%* for the first two interactions by each user, this grows to 21 DESIGNS ('Double Tap' 02, 04A, 05A, 06A, 07A, 08A, 09A, 13, 14; 'Dwell' 05B, 06B, 07B, 08B, 09B, 31; 'Force' 04C, 07C, 08C, 09C, 26, 32).

4.6.3.2.1 Comparison to baseline

When comparing the *AttemptNumber* of the DESIGNS to the respective baseline for their *input* (Double Tap' DESIGNS to 01A, 'Dwell' DESIGNS to 01B and 'Force' DESIGNS to 01C), a Wilcoxon revealed, that there was a significant difference (p¡0.05) between the design and the baseline for 22/47 DESIGNS (11/16 'Double Tap' DESIGNS, 3/16 'Dwell' DESIGNS and 8/15 'Force' DESIGNS). Out of those, only 2 DESIGNS ('Double Tap' DESIGN 17 (p=0.010, M=4.96), 'Dwell' DESIGN 24(p=0.013, M=3.92)) performed significantly worse than the baseline ('Double Tap' M=3.63, 'Dwell' M=2.71. 'Force' M=3.71), while all others significantly improved the perception of their respective *input*.

The measure of *AttemptNumber* illustrates (see figure 4.6), how the effectiveness of DESIGNS can vary widely, despite achieving similar *Attempt%* depending on how quickly participants perform a communicated *input*.

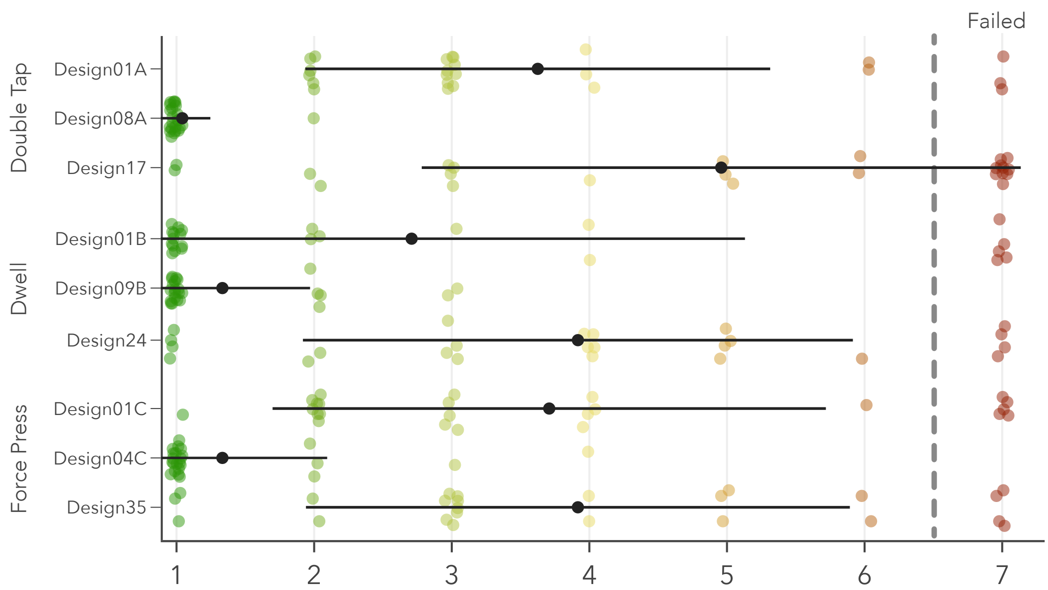


***Figure 4.6** – Trials required to achieve the correct interaction subset showing the baseline, as well as best and worst performing* DESIGN *for each* input. *The full figure is shown in appendix A.5.*

4.6.4 RQ4 - Does the success of signifiers depend on previous familiarity with the input in question?

We did not find evidence that previous familiarity with an *input* impacts *Selection%* (Study 1) or *Attempt%* (Study 2).

For study 1, we ran a Pearson's test between both reported 'familiarity' and 'usage' and *Selection%*. We examined both the correlations overall, and the correlations for DESIGNS where the *input* was assumed to be signified (vertical text in Figure 4.4). For *Familiarity*, we found a moderate linear correlation for 'Tap' (r=0.41) and 'Dwell' (Assumed r=0.57, Overall r=0.46), but none for 'Double Tap' (Assumed r=-0.03, Overall r=0.14) and 'Force Press' (Assumed r=0, Overall=0.02). Regarding reported *usage* of an *input*, we found a low linear correlation for 'Tap' (r=0.27) and 'Dwell'(Signified r=0.35, Overall r=0.26), but none for 'Double Tap' (Signified r=-0.15, Overall r= 0.14) and 'Force Press' (Signified r=-0.07, Overall=0.06).

For study 2, we ran a Pearson's test between 'familiarity' and *Attempt%* for all three *inputs* and found no correlation ('Double Tap' r(22)=-.07, p¿0.05; 'Dwell' r(22)=.06, p¿0.05; 'Force' r(22)=.07, p¿0.05).

4.6.5 RQ5 - Do signifiers reduce the perceived mental effort required?

Overall, signifiers reduce the perceived mental effort required to find an *input*.

Subjective responses in study 2 (figure 4.7-left) reveal that the majority of DESIGNS are perceived to require a lower mental effort. Wilcoxon tests comparing the mental effort of the individual DESIGNS to the '01 - Basic' baseline, show a significant difference (p¡0.05) for all DESIGNS except DESIGNS 17 (M=5.54, Mdn=6.0) and 24 (M=5.63, Mdn=6.0). All other DESIGNS were perceived as requiring significantly less mental effort. Overall, DESIGNS 08 (M=1.29), 09 (M=1.54) and 14 (M=1.67) were perceived as the least mentally demanding (Mdn=1.0), followed by DESIGNS 06 (M=1.58, Mdn=1.5) and 31 (M=2.13, Mdn=1.5) and finally with a median of 2.0, DESIGNS 04 (M=2.33), 05 (M=2.67), 07 (M=1.92), 13 (M=1.83), 16 (M=2.79), 22 (M=2.46), 23 (M=2.58), 25 (M=2.29), 26 (M=2.71), 27 (M=2.50), 35 (M=2.67) and 36 (M=2.13).

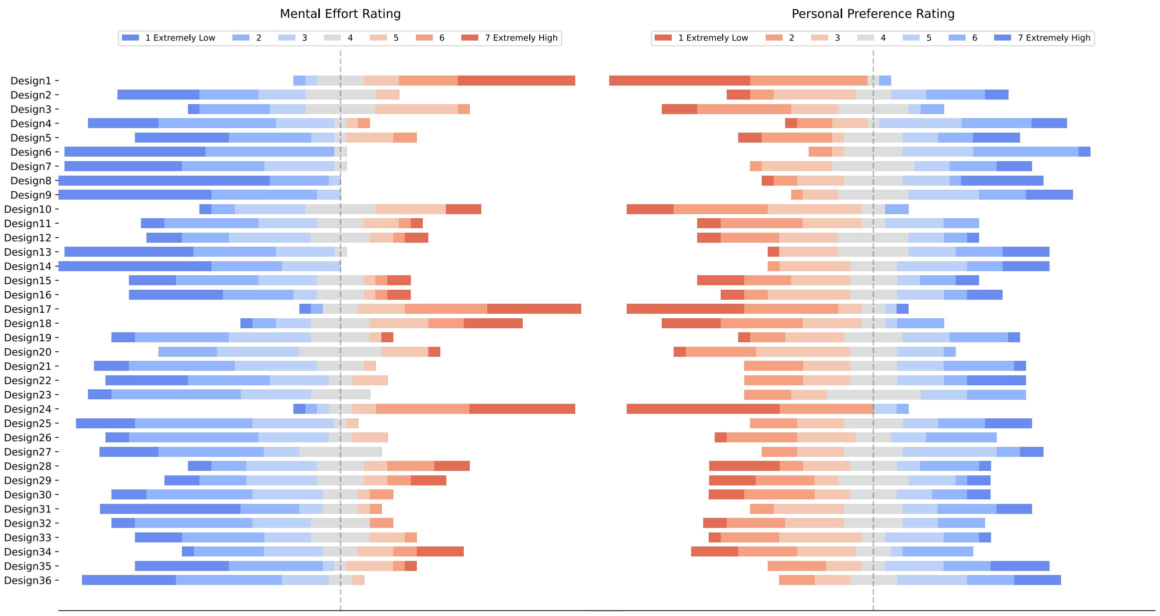


Figure 4.7 *– Stacked bar graphs showing likert scale responses for mental effort required (left) and personal preference (right).*

We only measured perceived mental effort in study 2. Since the aim of study 1 was to establish which *inputs* participants considered to be available, we did not have a set of 'correct' *inputs*. This made it impossible for participants to rate the effort required to deduce the availability of a specific *input*, because they were never given any confirmation of an *input* being available or unavailable.

4.6.6 RQ6 - Do users prefer the presence of signifiers over standard widgets?

In general, users prefer the inclusion of signifiers over a baseline without them.

While most DESIGNS received varying preference ratings (see figure 4.7-right), they were widely preferred over a complete absence of signifiers. Wilcoxon tests comparing the preference ratings of the individual DESIGNS to the baseline showed a significant difference (p¡0.05) for all except DESIGNS 17 (M=2.17, Mdn=2.0) and 24 (M=1.86, Mdn=1.0). All other DESIGNS were preferred over the baseline (M=1.75, Mdn=1.5), the favourite ones being DESIGNS 04 (M=4.67, Mdn=5.0), 06 (M=4.92, Mdn=5.0), 09 (M=4.88, Mdn=5.0), 14 (M=4.67, Mdn=5.0), 35 (M=4.63, Mdn=5.0) and 36 (M=4.71, Mdn=5.0).

Note that pearson's tests revealed significant correlations between personal preference and mental effort as well as interaction attempts for all three *inputs* ('Double Tap', 'Dwell', 'Force'). Notably, there is a high negative correlation between personal preference and mental effort rating ('Double Tap' (r(22)=-.71, p¡0.001), 'Dwell' (r(22)=-.69, p¡0.001), 'Force' (r(22)=-.66, p¡0.001)). Meanwhile, there is only a low to moderate negative correlation between personal preference and interaction attempts ('Double Tap' (r(22)=-.43, p¡0.001), 'Dwell' (r(22)=-.20, p¡0.001), 'Force' (r(22)=-.19, p¡0.001)).

4.6.7 RQ7 - Does Explicitness, Timing and Proximity impact the success of a design?

There are significant differences in the success of DESIGNS depending on design space categories. *Explicit* and *permanent* DESIGNS perform better while DESIGNS with signifiers *on* them generally perform worse.

4.6.7.1 Study 1 - Online Survey

EXPLICITNESS. All levels work well. *Descriptive* works best, followed by *Explicit*, *Mirrors Digital* and *Mirrors Real World*.

As each level of EXPLICITNESS is associated to a different number of designs, we first aggregated the corresponding designs for each level of EXPLICITNESS and each participant using the average *Selection%* for assumed inputs. Note that *Descriptive* is represented by only three DESIGNS (02, 03, 04). A Friedman test showed a significant effect of EXPLICITNESS on *Selection%* ($\chi^2(3) = 22.6$, $p < 0.001$). Post-hoc comparisons using Wilcoxon signed-rank tests revealed significant difference (p=0.01) between the *Mirrors Real World* (Mdn=80%) and *Descriptive* (Mdn=100%) levels. Further there were significant differences (p¡0.05) between *Mirrors Digital* (Mdn=84%) and *Mirrors Real World* levels (see Figure 4.8).

The different levels of EXPLICITNESS can also be analysed for each individual assumed input. The different levels of EXPLICITNESS affect *Selection%* differently depending on the assumed input with 'Double Tap' particularly affected, 'Dwell' affected and 'Force Press' not affected.

For 'Double Tap', a Friedman test showed a significant effect ($\chi^2(3) = 35.8$, $p < 0.001$) of EXPLICITNESS on *Selection%* with post-hoc revealing significant differences (p¡0.001) between the *Mirrors Real World* (Mdn=0%) and all other levels (*Descriptive*: Mdn=100%, *Explicit*: Mdn=80%, *Mirrors Digital*: Mdn=71%).

For 'Dwell', a Friedman test showed a significant effect ($\chi^2(3) = 23.5$, $p < 0.001$) of EXPLICITNESS on *Selection%* with post-hoc revealing significant differences (p¡0.01) between the *Descriptive* (Mdn=100%) and *Mirrors Digital* (Mdn=89%) as well as *Mirrors Real World* (Mdn=75%). It further showed a significant difference (p¡0.05) between the *Explicit* (Mdn=100%) level and *Mirrors Real World*.

Finally, for 'Force Press' only, no effect of EXPLICITNESS was found on *Selection%*.

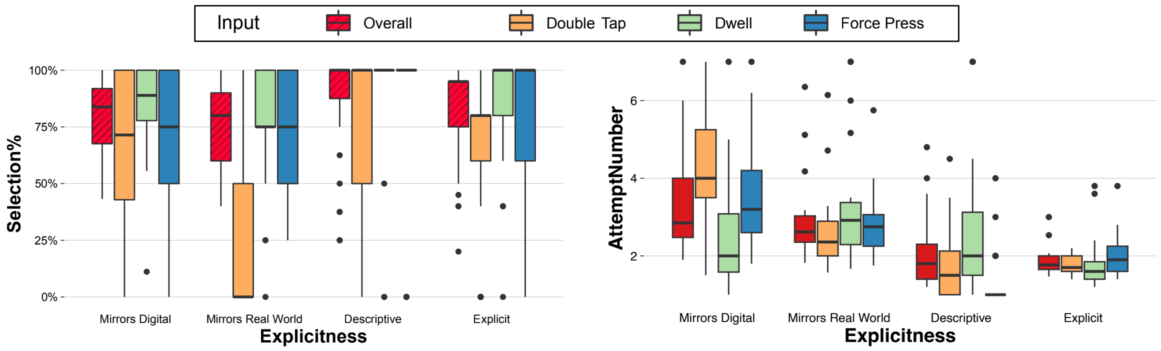


Figure 4.8 *– Left: Average Selection% by* EXPLICITNESS *level for each* input *study 1. Right: Average AttemptNumber by* EXPLICITNESS *level for each* input *study 2.*

TIMING. *Permanent* is more efficient than *Temporary* for 'Double Tap'. A Wilcoxon signed rank test found a significant effect ($V = 403.5$, $p < 0.001$) of TIMING on *Selection%*, showing that *Permanent* (Mdn=89%) is more efficient than *Temporary* (Mdn=82%). An investigation for each assumed input shows a significant effect ($V = 375$, $p < 0.001$) of TIMING only for 'Double Tap' (*Permanent*: Mdn=93%, *Temporary*: Mdn=56%).

PROXIMITY. *On* has lower performance compared to *Vicinity* and *Away*, especially for 'Double Tap' and 'Dwell'. A Friedman test revealed a significant effect ($\chi^2(2) = 21.7$, $p < 0.001$) of PROXIMITY on *Selection%*. Post-hoc analysis revealed that *On* (Mdn=77%) is significantly (p¡0.001) less efficient than *Vicinity* (Mdn=88%) and *Away* (Mdn=92%). An investigation for each assumed input revealed a significant effect ($\chi^2(2) = 28.9$, $p < 0.001$) of PROXIMITY on *Selection%* for 'Double Tap' with post-hoc analysis showing significant differences (p¡0.001) between the *On* (Mdn=40%) and both other levels (*Away* (Mdn=75%), *Vicinity* (Mdn=86%)). Similarly, there is a significant effect ($\chi^2(2) = 28.1$, $p < 0.001$) of PROXIMITY for 'Dwell' with post-hoc analysis showing significant differences (p¡0.002) between *On* (Mdn=80%) and both other levels (*Away* (Mdn=100%), *Vicinity* (Mdn=95%)). No significant effect was found for 'Tap' and 'Force Press'.

4.6.7.2 Study 2 - Interactive Experiment

The nature of study 2 offers confirmation for correct *inputs* and allows participants multiple trials. Therefore the average *Attempt%* is much higher than the average *Selection%* from study 1. However, since we collect *AttemptNumber*, we can compare not only whether an *input* was considered possible, but also when.

EXPLICITNESS. *Explicit* works the best, followed by *Descriptive*, *Mirrors Real World* and *Mirrors Digital*.

We aggregated the corresponding DESIGNS corresponding to each EXPLICITNESS level and each participant using *AttemptNumber*. A Friedman test showed a significant effect of EXPLICITNESS on *AttemptNumber* ($\chi^2(3) = 56.6$, $p < 0.001$). Post-hoc comparisons using Wilcoxon signed-rank tests revealed a significant difference (p¡0.001) between the *Mirrors Digital* (Mdn=2.85) and *Descriptive* (Mdn=1.8) as well as *Explicit* (Mdn=1.77) levels. There were also significant differences (p¡0.001) between *Mirrors Real World* (Mdn=2.62) and *Descriptive* (Mdn=1.8) as well as *Explicit* (Mdn=1.77) levels. (Figure 4.8).

The different levels of EXPLICITNESS can also be analysed for each individual assumed input. In contrast to study 1, the different levels of EXPLICITNESS affect *AttemptNumber* for all *inputs*.

For 'Double Tap', a Friedman test showed a significant effect ($\chi^2(3) = 48.93$, $p < 0.001$) of EXPLICITNESS on *AttemptNumber* with post-hoc revealing significant differences (p¡0.001) between all levels except for *Explicit* and *Descriptive*.

For 'Dwell' a Friedman test showed a significant effect ($\chi^2(3) = 23.94$, $p < 0.001$) of EXPLICITNESS on *AttemptNumber* with post-hoc revealing a difference (p¡0.001) only between *Explicit* (Mdn=1.6) and *Mirrors Real World* (Mdn=2.92).

For 'Force Press' a Friedman test showed a significant effect ($\chi^2(3) = 49.34$, $p < 0.001$) of EXPLICITNESS on *AttemptNumber* with post-hoc revealing differences (p¡0.005) between all levels except between *Mirrors Digital* and *Mirrors Real World*.

Given that the design space category 'Explicitness' aims to categorise how much interpretation and therefore mental effort is expected from the user, we also compared the mental effort ratings for the DESIGNS grouped according to explicitness level (Figure 4.9). There is a significant difference between the mental effort for 'Explicit' DESIGNS and 'Descriptive' (U = 2396, p ¡ 0.001), 'Mirrors Real World' (U = 6892, p ¡ 0.001), 'Mirrors Digital' (U = 13030, p ¡ 0.001) and 'Not at all' (U = 191, p ¡ 0.001). Additionally, there is a significant difference between 'Descriptive' and 'Mirrors Real World' (U=7322.5, p ¡ 0.05) as well as 'Not at all' (U=178.5, p ¡ 0.001). There is also a significant difference between 'Mirrors Real World' and 'Mirrors Digital' (U=55321.0, p ¡0.005), 'Mirrors Real World' and 'Not at all' (U=1130.0, p ¡ 0.001) as well as 'Mirrors Digital' and 'Not at all' (U=1220.0, p ¡ 0.001).

TIMING. *Permanent* is generally more efficient than *Temporary*. A Wilcoxon signed rank test found a significant effect ($V = 13$, $p < 0.001$) of TIMING on *AttemptNumber*, showing that *Permanent* (Mdn=1.98) is more efficient than *Temporary* (Mdn=2.56). An investigation for each assumed input shows a significant effect of TIMING for 'Double Tap' ($V = 8$, $p < 0.001$,

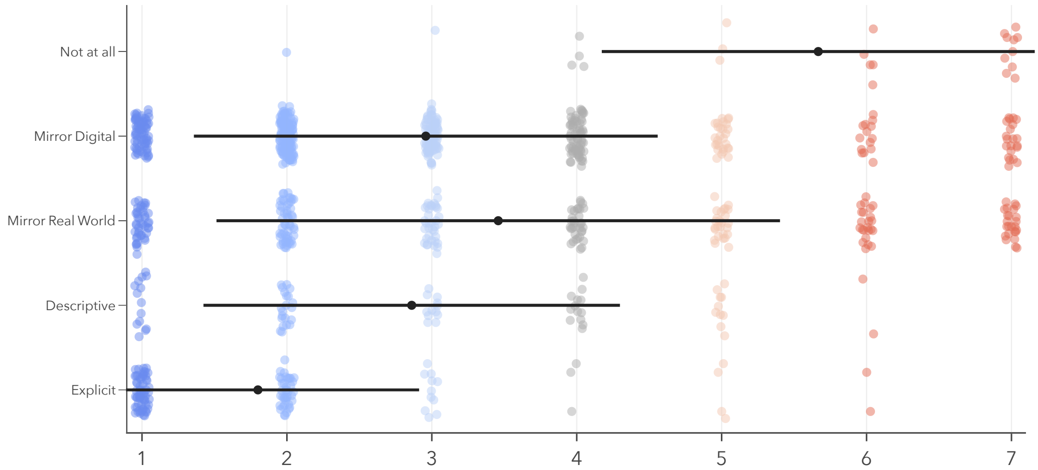


***Figure 4.9** – Plot showing the mental effort ratings grouped by* EXPLICITNESS *of the corresponding* DESIGNS.

Permanent: Mdn=1.71, *Temporary*: Mdn=2.89) as well as 'Force' ($V = 15$, $p < 0.001$, *Permanent*: Mdn=1.93, *Temporary*: Mdn=2.94).

PROXIMITY. *On* performs worse compared to *Vicinity* and *Away*. A Friedman test revealed a significant effect ($\chi^2(2) = 33.25$, $p < 0.001$) of PROXIMITY on *AttemptNumber*. Post-hoc analysis revealed that *On* (Mdn=2.92) is significantly (p¡0.001) less efficient than *Vicinity* (Mdn=2.09) and *Away* (Mdn=2.21). An investigation for each assumed input revealed a significant effect ($\chi^2(2) = 16.695$, $p < 0.001$) of PROXIMITY on *AttemptNumber* for 'Double Tap' with post-hoc analysis showing significant differences (p¡0.001) between the *On* (Mdn=3) and both other levels (*Away* (Mdn=2), *Vicinity* (Mdn=1.86)). There is a significant effect ($\chi^2(2) = 11.77, p < 0.005$) of PROXIMITY for 'Dwell' with post-hoc analysis showing significant differences (p¡0.05) between *On* (Mdn=2.67) and both other levels (*Away* (Mdn=2), *Vicinity* (Mdn=2.17)). Similarly, there is a significant effect ($\chi^2(2) = 27.85$, $p < 0.001$) of PROXIMITY for 'Force' with post-hoc analysis showing significant differences (p¡0.05) between *On* (Mdn=3.1) and both other levels (*Away* (Mdn=2), *Vicinity* (Mdn=2.25)).

4.7 Discussion

In this section, we discuss the implications for design and limitations of the two user studies described above.

4.7.1 Lessons learnt from our studies

Takeaway 1: *Users prefer added visual signifiers.* Despite the lack of significant difference in performance for some DESIGNS and *inputs* in study 2, all but 2 DESIGNS (17, 24) were perceived as less mentally demanding. This suggests that these DESIGNS can still improve the subjective experience when these *inputs* should be communicated. This is further supported by participants indicating a significant preference for the DESIGNS with added visual signifiers over the baseline except the aforementioned 2 DESIGNS. Overall, this still suggests the usefulness of visual signifiers for improving user experience with a widget.

Takeaway 2: *Which signifier is best depends on context and target audience.* The preference ratings for the different DESIGNS did vary between participants. For instance, P21 considered the permanently displayed instructions to be *"a waste of space to a certain degree"*. In contrast, P13 would prefer redundancy in signifiers, including explicit ones, noting that *"the best would be to first have text to tell me what to do and then a scale that tells me how I'm doing"*. In addition, several participants noted that they may prefer certain DESIGNS in some contexts. For example, P12 noted for DESIGN 35, that *"this is fun [...] if it's in a game I'd like it, but if it was my alarm clock I'd hate it"*.

Takeaway 3: *When inputs are established, adding signifiers may be unnecessary.* In general, the results of both studies suggest that signifiers increase the perception of available inputs and reduce mental effort. However, study 2 revealed that for 'Dwell' only 2 DESIGNS performed significantly better than the baseline 01B, those DESIGNS both being permanently displayed explicit text (DESIGNS 08 & 09). This can partially be attributed to the fact that 'Dwell' is a well established *input* and was frequently performed as the first *input* attempt regardless of DESIGN. An example of this, is P10 stating that *"[their] default [interaction] is just kind of hold it there and see if there is anything that pops up and act accordingly"*. As a result, the 'Dwell' baseline DESIGN 01B performed significantly better than the baseline for the less common *inputs* 'Double Tap' and 'Force Press'. This suggests that if an *input* is established, added signifiers do not increase the perception of the *input*'s availability overall, questioning their necessity.

Takeaway 4: *When signifying 'added' input methods, proceed with caution to not inadvertently remove the perception of availability of established ones.* Analysis of the *Selection%* in study 1 according to the design space revealed a significant difference in 'Tap' responses between different explicitness levels, in spite of 'Tap' always being shown in the animation, suggesting that the explicitness of a design may impact previously established perceived affordances. Furthermore, designs for which the response to 'Tap' does not surpass our 75% effectiveness threshold achieved high selection percentages for other interaction methods. This may indicate that for ambiguous design that still strongly suggest an interaction, initial perceived affordances are discarded in favour of the new signified one. Given this observation, designers need to be careful not to inadvertently remove perceived affordances when attempting to add additional ones.

Takeaway 5: *The perception of signifiers varies between users.* If signifiers are not explicit, designers need to account for users who may interpret them in unexpected ways. For example, for the DESIGNS 04A, 04B, 04C which displayed explicit counters above the icon, the majority of participants interpreted the text as intended. However, P12 interpreted the text as a warning telling them not to tap, not linger and apply zero pressure to the button. Similarly, P10, when presented with a DESIGN including a scale interpreted it *"kind of a slider"* and tried to swipe on the scale, while the majority of other participants considered scales as an indication of continuous *input*, leading them to either dwell or force press. These differences suggest that even widely understood visual guidance can be interpreted in various ways.

Takeaway 6: *Users previous experiences with contexts and devices impact their perception of metaphors.* When interacting with DESIGN 18, P4, once they had noticed the animation, attempted to dwell on the button and expressed their surprise when the *input* was considered incorrect, stating *"on the kindle page flip happens on dwell so I expected that"*. Similarly, P11 attempted to 'Double Tap' both DESIGNS 29 and 35 multiple times, commenting *"from the [Nintendo]* *switch, I'm sort of programmed to tap multiple times in quick succession - that's just my go to instinct"*.

These quotes highlight, that users extrapolate meaning from previous interactions. If DESIGNS mirror other usages of similar visual metaphors, they should either align with what is conveyed in those other use cases, or very clearly diverge visually.

Takeaway 7: *Real world metaphors are particularly successful when signifying less known or ambiguous interactions.* Explicit instructions very efficiently communicated *inputs*. However, in study 1, for both 'Double Tap' and 'Dwell' the most successful DESIGN was one mirroring established digital signifiers. In contrast, 'Force Press' is the only interaction method whose top results include DESIGNS that mirror real world affordances. Furthermore, in study 2 multiple participants commented on the difficulty of differentiating between continuous *inputs* but pointed to DESIGN 31 (Hourglass) as an exception, e.g. P7 stated *"for this one it was clear it's time related [in comparison to the scales]"*. This may suggest that for interactions that are less established, leaning on real world metaphors is particularly helpful.

Takeaway 8: *Timing matters for less established* inputs. The comparison of *permanent* and *temporary* DESIGNS for both studies shows that *permanent* DESIGNS perform better. Notably, when considering the DESIGNS separated by *input*, there is no significant difference for 'Dwell' suggesting temporarily displayed signifiers may sufficiently communicate more established *inputs* while less common *inputs* may benefit from extended visibility.

Takeaway 9: *Existing visual guidance may only work in context - consider this when adapting it to a new context.* The *24 Arrow* DESIGN was not successful in conveying 'Dwell' (53% *Selection%*, 83% *Attempt%*, M *AttemptNumber* = 3.92) despite being the only commercial visual signifier we found for that *input* (figure 4.2b). Even worse, half of the participants in study 1 did not consider the signified interaction to be possible at all, and it was one of only two DESIGNS that (1) participants did not prefer over the baseline and (2) did not reduce the perceived mental effort. While this may be due to our participants being unfamiliar with Adobe Fresco, or Adobe products in general, it is certainly noteworthy that one of the few existing visual signifiers for mobile interfaces appears to not convey the intended meaning universally. A potential explanation may be that the context of a tool menu within an application is as much of a signifier as the visualisation itself and the visual arrow depiction only works within this context. Further work could investigate how visual depictions work in tandem with the context in which they are placed to convey meaning to users.

4.7.2 Clarity of signifiers for similar interactions

Clustering the response patterns for different designs in study 1 revealed that, while designs tend to be most effective in communicating a specific input, similarity between the interactions increases the likelihood that multiple inputs are considered possible. Continuous inputs such as 'Force Press' and 'Dwell' could be clearly separated from the discrete input of 'Double Tap'. This was echoed in study 2, with the overall most frequent comments being about the difficulty to differentiate between 'Dwell' and 'Force' (e.g. *"the problem between force and dwell with scales is it is hard to tell which one you are supposed to do"*-P6). Similarly, multiple participants commented on 'Double Tap' due to its discreet sequential interactions (e.g. P9 *"whenever there's two of anything I immediately double tap, that's a no brainer for me"*). Given that, in study 1 where no 'correct' *input* was provided, 'Dwell' largely achieved a higher

Selection% than 'Force', these responses suggest that, where less common inputs may resemble more established ones, special attention is required to ensure that the intended *input* is signified first and foremost.

4.7.3 Impact of physical interaction and personal differences

The laboratory study revealed that correct perception of available *inputs* is not always sufficient to achieve successful interactions since how participants interacted with the phone also impacted the interaction. All participants were handed the phone, but while some kept it in their hands, others put it down on the table in front of them. For the participants who held it, some interacted with the thumb of the hand holding it while others interacted with the index finger of the other hand. Participant 20 noted *"Hold on, my hand was covering that"* and then put the phone down on the table, while participant 17 commented, that *"it depends on how you hold your phone, since I use my thumb I block a lot of the phone, but if I would hold the phone in one hand and tap with a finger on the other, maybe I would block less of the screen"*. In addition, participant 24 had long acrylic nails which changed their way of interacting. They were the sole participant that interacted with the phone on the table moving their hand horizontally with their nails pointing towards the left edge of the phone. Subsequently, there were multiple DESIGNS that were occluded by their nails. These observations are in line with, and a possible explanation for, the finding in study 2, that signifiers *on* the DESIGN generally perform worse than those with some distance (*vicinity, away*). If the user physically covers the signifier, it will be harder to perceive and therefore may not improve the perception of available *inputs*.

4.7.4 The impact of familiarity

Although we found no impact of previous familiarity with *inputs* on their subsequent *Selection%* in study 1 or *Attempt%* in study 2, this may be due to the way the studies are setup. In asking participants about their familiarity with *inputs* ahead of their encounter with the DESIGNS, we remove the possibility of their complete lack of familiarity. In study 1 participants are additionally reminded of the existence of different *inputs* through their representation in the available checkboxes. The decision to preface the DESIGNS with questions regarding familiarity with different *inputs*, was made to increase the likelihood of meaningful engagement with the DESIGNS. Nonetheless, it should be noted that this order may impact the observable correlation between familiarity and *input* selection. Therefore, it can not be ruled out, that familiarity with *inputs* may have an impact on their perception outside of a test environment.

4.7.5 Future Work

In an effort to explore the root of signifier performance, these studies were limited to in-place touch interactions. However, many other inputs are available on touchscreens and rarely signified (e.g. 'Swipe' [177]). Future work should explore how these inputs may best be communicated to users.

Furthermore, while the studies investigated which DESIGNS conveyed which *inputs* to the participants, they did so within a context that introduced the *inputs* to them and increased the participants consideration of them. Therefore, to confirm whether participants would *discover* the interaction method in question thanks to the signifier in real-world implementations, a discoverability study [69,78] needs to be performed which does not include guidance on which interaction methods may be available. Such an investigation could further examine whether the lack of correlation between familiarity and interaction selection may be attributed to the fact that participants were presented with all interaction methods beforehand or had predefined answer options that did not require them to recall any interactions. Furthermore, since early pilot studies revealed that context is a major signifier of buttons, we presented the same context for all buttons. It can be assumed that most signifier designs could be adapted to different contexts, or possibly even widgets, but this is not verified within the studies performed. Future research should examine context as a signifier and how it is linked and interacts with visualisation.

4.8 Conclusion

In this chapter, we investigate visual signifier characteristics and their impact on perceived availability of in-place single finger inputs on UI buttons.
We propose a design space that not only acts as a framework to create and compare designs but also provides an opportunity to reflect on the qualities considered within it. In the design space, we examine explicitness, temporality and proximity. Through these characteristics, the design space allows consideration of which metaphors and design patterns can and should be replicated.
Evaluating the perception of signifiers in graphical user interfaces, and consequently the affordances user might perceive, remains uncommon and no clear protocols exist in that respect. Motivated by the success of online studies deployed to evaluate the perception of affordances in product design [96], we decided to first deploy an online survey. Subsequently, we conducted an interactive in-person user study, to further examine the perception of input availability and the mental effort required to deduce available inputs. Overall both studies confirmed that added visual signifiers increase the perception of *input* availability. This suggests that, where interactivity on touchscreens extends past simple tapping on clearly displayed targets, added visual signifiers can be a helpful tool to communicate additional input availability to users. Nonetheless, this chapter focuses on very specific inputs and widgets. Thus, signifiers designs for other inputs and interaction contexts need to be further explored.

5 Incorporating Visual Signifiers in Existing User Interface Animations

The previous chapter shows that visual signifiers have the capacity to communicate inputs to users where they would otherwise not perceive them. As such, they are a valuable resource in conveying the diverse functionality of modern devices that overload controls in small interfaces. Accordingly, the viability of increased visual signifiers for improved interaction discoverability on touch screens should be examined further.
Therefore, we explore visual signifiers for a different type of input, specifically swhidgets (swipe-revealed-hidden-widgets) [177], that by their very nature are not visually signified and thus suffer from poor discoverability. Given the by-default-hidden nature of swhidgets, we further explore where signifiers can be displayed and test the viability of incorporating them into existing temporary user interface components. In doing so, we additionally address the previously discussed need to run a discoverability study to truly examine the effect of visual signifiers.

5.1 Introduction

Swiping on touch screens has been increasingly relied upon to give users access to additional UI functionality by revealing controls that are otherwise hidden. Swipe-revealed hidden widgets, or *Swhidgets* [177], are interactive elements that are hidden by default and introduce a new component when revealed through swiping away from the screen edge at which the component is revealed, but do not permanently change the main view of the interface (Figure 5.1, left). For instance, swhidgets allow users to reply to messages in applications like WhatsApp and Messenger, as well as archive or delete things like chats, emails, memos, reminders and alarms. Additionally, they have also been implemented to add songs to play queues, display precise timings and reveal actions within list views. In doing so, they enable feature-rich applications despite the limited screen real estate in smartphones. As the examples above show, this is becoming increasingly popular given the abundance of functionality available in applications. By hiding controls such as buttons outside of the screen, swhidgets

***Figure 5.1** – Left: Example of a swipe-revealed widget, or Swhidget: a user swipes an email away from the edge of a touch screen to reveal hidden buttons. Right: We explore the potential of adding interaction hints in animated transitions to temporarily expose users to Swhidgets as a means to increase their discoverability.*

keep the main interface uncluttered, while enriching the set of widgets (i.e. buttons/actions) readily available to users [196]. The rapid swipe movement is easy to perform, making the controls embedded in the swhidgets easy and quick to access. While they are hidden, they are quick to reveal and use, in comparison to sequences of taps opening menus and submenus, and allow controls to be presented in the context they are relevant to.

The reliance on gestures and hidden widgets, however, comes with discoverability challenges as the large variety of devices, platforms, and usage contexts make it difficult for users to be aware *when* and *where* touch gestures provide a meaningful function. This is especially problematic when it comes to swhidgets, since they are implemented inconsistently, with swipe availability, direction and thresholds varying between applications and platforms. One example is the general view of different chat applications. While all of the following resemble each other in layout, Telegram (Version 10.0.5) allows users to swipe *left* to archive chats, Signal (Version 6.31.2) offers the same functionality when swiping *right*, Android Messages (Android 13) allows both *left and right* swipes to archive, and Whatsapp (Version 2.23.17.80) implements no swhidgets on the chat overview at all in favour of lateral navigation between tabs. In addition to making knowledge transfer difficult, this inconsistency in combination with a lack of visual signifiers communicating the availability of swhidgets, makes them hard to discover [177, 178], depriving some users of important functionality.

Subsequently, visual signifiers could prove to be a valuable resource increasing the discoverability of swhidgets. We specifically explore the use of animated transitions to temporarily expose users to these hidden elements. Animated transitions are well established in commercial mobile interfaces [131] and have been shown to increase the perception of continuity in interfaces [97], establish clear causal connections between states [209], guide user attention and highlight changes [148], as well as support accurate mental models of applications [112]. We explore the possibility of extending their function to include transient interaction hints as a means to reveal the affordance of swhidgets (Figure 5.1, right). To do so, we adapt existing interface transitions to temporarily reveal the swhidgets that disappear as the final UI state of an animation is reached. We implement three types of animated transitions featuring transient interaction hints based on existing transition styles and conventions [131]: 'Container Transform', 'Panel Exit' and 'Side Slide'.

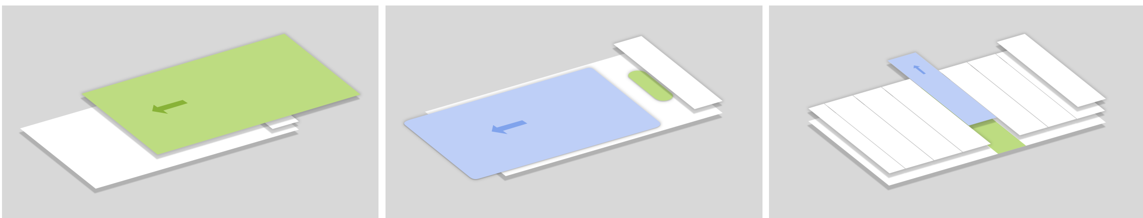
We conduct two experiments to assess the viability of this approach. In a first study (N=60), we verify the noticeability of the visual cues provided by interaction hints contained in animated transitions, for different transition durations and styles. Results of this study suggest that the amount of detail that people recall varies widely between participants. We also find that making the animated transition duration last as long as the maximum recommended value (i.e. 1000ms) [160] does not seem to carry much benefit. Not only do we not observe a notable increase in noticeability compared to a shorter animation (i.e. 750ms), but some participants also reported that the animation was too long and caused confusion. In a second study (N=33), we assess the discoverability of swhidgets when they are exposed to users using our animated transition approach. We evaluate the impact of two types of transitions: 'Panel Exit', the most noticeable transition per our first study, and 'Container Transform', which builds on the most widespread transition used in mobile interfaces by comparing these transitions incorporating transient interaction hints to a baseline without them. While interaction hints were noticeable in the first study, the results of this second study surprisingly reveal that this was not sufficient to impact discoverability. Neither the overall swhidget discovery percentage, nor the time until the first usage of swhidgets im-

proved in comparison to the baseline. Nonetheless, the study allowed insights in how people discover swhidgets, with participants discovering them through conscious search for tools, accidentally, or through familiarity with similar features. The study also revealed that users were able to discover a swhidget when it was necessary to complete a task. Further, the study results suggest that noticeability does not necessarily translate to comprehension and discoverability as multiple participants commented on features of the transition, but did not deduce that they may indicate the availability of interactions.
In this chapter we make the following contributions. We offer an extended examination and classification of swhidgets and introduce the addition of transient interaction hints to animated transitions, which we then implement in three types of animated transitions. We report on a study examining the noticeability of animated transitions featuring transient interaction hints and find noticeability varies substantially between users. Finally we contribute empirical results of a study investigating the impact of animated transitions featuring transient interaction hints on the discoverability of swhidgets, revealing a disconnect between noticeability and discovery, and offer possible causes.

5.2 Related Work

We first review the characterisation of Swhidgets in mobile interfaces, before discussing the approaches our work builds and expands upon to improve discoverability of Swhidgets: visual signifiers, and animated transitions.

5.2.1 Swipe-Revealed Hidden Widgets



(a) *System swhidget* ***(b)*** *View swhidget* ***(c)*** *Item swhidget*

Figure 5.2 *– Visualisation examples of different swipe-revealed hidden widgets. The swhidgets are coloured in green. If other elements from the main view are moved with the swipe, they are coloured in blue. (a) Swipe from the top or bottom of the screen reveals a system swhidget, which appears on top of the general view. (b) Swipe from the screen edge moves the majority of the main view to reveal a swhidget in the created space. (c) Swipe horizontally on a list item moves the entire item to reveal an item swhidget beneath it.*

Swipe-revealed hidden widgets, or Swhidgets, are a technique employed in touch interfaces to provide users access to controls such as buttons and actions through a swipe gesture, while not taking up screen real estate [177]. More specifically, we consider a UI element to be a Swhidget if it is an interactive element that (i) is hidden by default, (ii) introduces a new component when revealed through swiping, and (iii) does not permanently change the main view of the interface, which often remains consistent "underneath" or "above" the swhidget (Figure 5.2). Pong and Malacria [177] classified three types of swhidgets — system, view, and item. We follow their classification but add our own considerations regarding swipe direction, implementation, and the relation to existing screen elements.

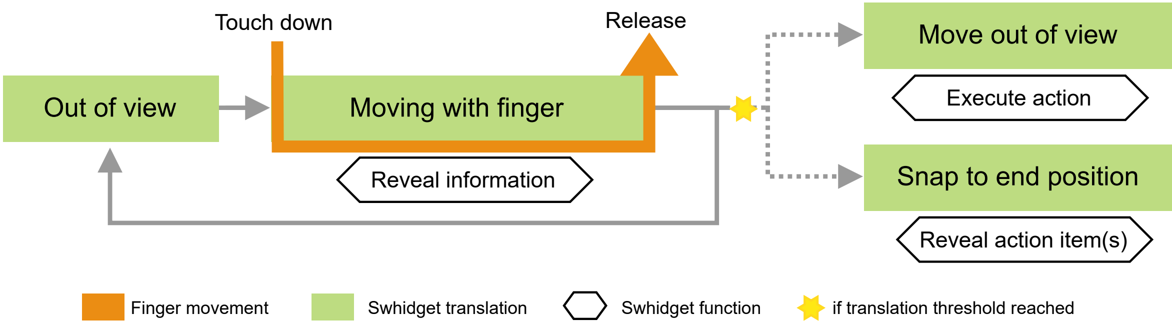


Figure 5.3 – *Chart showing the progression of swhidget. The swhidget moves (green) with the finger movement on the screen (orange). If a threshold is reached the swhidget, upon release, either executes an action and moves out of view, or snaps to an end position revealing action items (i.e. buttons) on screen. If the translation threshold is not reached, the swhidget moves out of view upon release. For certain swhidgets (i.e. time on messages) there are no actions and subsequently no translation threshold, as their sole aim is to reveal information through the swhidgets temporary visibility.*

System swhidgets are revealed from the top or bottom of the screen through vertical swiping, presenting an overlay of the current main view regardless of what the current application or view displays. Key examples are the notification and control panels in both Android and iOS [1]. As illustrated in Figure 5.2a, no part of the main view moves upon revealing the hidden widget, since the widget appears *on top* of the view as an overlay. To distinguish these widget overlays from the main view, Android progressively decreases the main views opacity whereas iOS blurs it.

View swhidgets are revealed from any side of the main view or container through swiping from the screen edge towards the center. The main view moves with the swipe motion, revealing a widget in the created space between the screen edge and the view. Usually the main view does not consist of the entire screen, leaving some elements such as top menu bars in place (see Figure 5.2b). Examples include the email view on iOS, search in email list views or Spotify playlists, message time on Instagram as well as filtering messages on Signal.

Item swhidgets are revealed from the left or right side of the screen through horizontal swiping which moves one individual item or list item on screen with the motion of the finger, revealing the hidden widget on the edge of the screen (see Figure 5.2c). Item swhidgets are the most commonly implemented swhidgets aside from system swhidgets, with examples including deleting, archiving or editing list items in email applications, messaging apps, memos, or reminders as well as reacting or replying to messages.

Regardless of classification, swhidgets generally perform one or multiple of three functions: revealing information, revealing action items, or executing an action. To do so, they follow a movement pattern in which they are out of view, move with the finger once a swipe in the correct area is performed and either move back out of the view or snap to an end position on release (see Figure 5.3). Snapping to an end position as well as executing an action on release depends on whether the swipe movement goes past a certain threshold (e.g. horizontally halfway across the screen). If the threshold is not reached, the swhidget moves

1. The illustrative examples which we discuss in this section are representative of the state-of-the-art animations found in Android version 13 and iOS version 16.3.

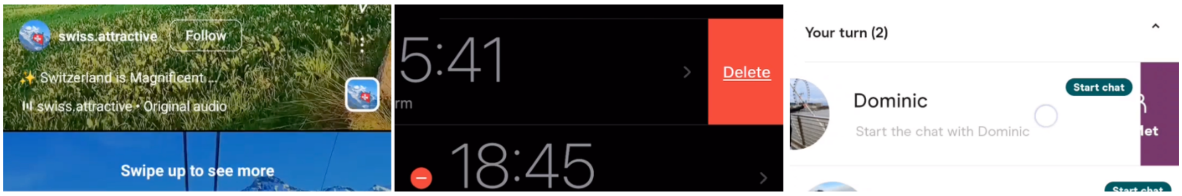


Figure 5.4 *– Examples of swhidgets being temporarily signified through animation. Left: Instagram Reels animation suggesting users can swipe up to see more. Middle: iOS alarm 'edit' page featuring a swipe animation. Right: Hinge match page showing preview animation of swipe revealed menu when dwelling on the list item.*

back out of view without executing an action. Figure 5.3 indicates the possible opportunities where functions can occur. Notably, all examples of vertical swhidgets that we could find revealed action items rather than executing an action immediately upon release.

While they rely on a small set of core principles, swhidgets span a rich design space, with dimensions including type of swhidget (Figure 5.2), associated functions (i.e. buttons, actions, ...), as well as how they respond to user input (i.e. trigger action upon release, thresholds, ...). As a result, it is not trivial for users to be aware which swhidgets may be available in an interface, nor what their specifics are. This challenge is further exacerbated by the lack of standards and conventions, as existing swhidgets are currently inconsistent across different operating systems and applications. For example, in Whatsapp (Version 2.23.2.76) swiping *right* on a message allows users to 'reply' directly to the message, whereas in Telegram (Version 9.4.2), which bears a strong resemblance to the Whatsapp interface, swiping right returns users to the message overview and swiping *left* allows the 'reply' option by default. While Android 13 allows access to a control panel and notifications by swiping down from the top of the screen, iOS differentiates between downward swipes from different edges at the top of the screen, revealing notifications when swiping down from the left-hand corner and a control panel when swiping down from the right.

Swhidgets are a powerful tool to augment the range of possible in-context interactions in touch interfaces, but their discoverability is hindered by their inherent hidden nature and the lack of consistency across applications and systems, which is what we aim to address in this chapter by introducing visual signifiers temporarily exposing swhidgets.

5.2.2 Communicating 'swipe-ability'

There are design patterns which are commonly used to convey the availability of swipe interactions. Notable examples include partially visible content at the view edges, carousels previewing content, tabs and menu bars highlighting the selection of a currently visible view alongside the existence of other views to their sides as well as animations telling us to swipe to unlock or open. These signifiers are essential in communicating to users which type of interaction (swipe) is available where in the interface to achieve something.

While similarly relying on swipe interactions, swhidgets, due to their inherently hidden design, typically do not provide any visual cues before the feedback once the swipe gesture is initiated. This, in addition to their inconsistent implementation, makes it hard for users to discover them or determine when they are available as well as which function they serve.

Nonetheless, there exist some notable examples of "swipe-ability" being visually communicated for content beyond the screen bounds. In Instagram Reels (Version 282.0.0.22.119), the availability of swiping upwards to switch to the next reel is shown through an animation

moving the current content upwards, showing a peek of the following reel with overlaid text stating "Swipe up to see more" (see Figure 5.4 left). Two other examples signify "swipe-ability" once a swhidget is being interacted with through other means. In iOS 16.3 the alarm application allows users to delete alarms by swiping on the alarm, as well as through an edit button that then reveals individual delete icons for each alarm. In this edit view, an animation shows one alarm list item moving to the left, revealing a red block with 'Delete' written on it (see Figure 5.4 middle). Ironically, in this edit view swiping on alarm items (including the one featured in the animation) is not possible. Swiping on the alarms is only possible once users exit the editing mode, that shows them the theoretical availability of the swhidget. In the Hinge application (Version 9.25.1) swiping on existing matches in a list view reveals a menu. These options are not accessible through other means, however when dwelling on one of these matches, a short animation of the list item moving to the left is shown (see Figure 5.4 right). Since this animation is temporary and does not fully reveal any option in the menu, it is purely implemented to signify the possibility of swiping.
We take inspiration from these examples and explore the potential of temporarily displaying visual signifiers that communicate the availability of swhidgets. By displaying signifiers only temporarily, this sustains the advantage of swhidgets not adding visual clutter, while subtly supporting their discovery.

5.2.3 Animated Transitions

Transitions between different UI views describe the switch from view A to view B. To improve usability and navigability, these transitions are commonly smoothly animated to provide context of the location and relationship between different views [131]. Animations are well established as tools to keep users oriented during navigation and to support tracking during layout changes [40]. As such, animated transitions have been found to increase the perception of continuity [97], guide user attention and highlight changes [148], and support the formation of accurate mental models [112]. Animated transition have also been found to increase understanding of the causal connection between states while also being considered visually pleasing [209].
When considering the style of animated transitions, there is a virtually infinite number of ways that one can animate elements between two views. The trajectories that elements follow, the staging, duration, as well as pacing of the animation are all dimensions a designer has to consider when creating an animated transition. For interface design, Apple's developer design guidelines features vague recommendations like *"design a launch screen that smooths the transition to the first screen"* [57] and to *"avoid jarring transitions"* [56], but do not offer a comprehensive record of the transition types currently implemented in their system. Material Design offers a transitions help guide [131] which examines transitions in both Android and iOS considering different transition patterns and when they are suitable.
One example, is the transition that occurs when application icons are tapped (see Figure 5.5), triggering a transition from the homescreen to the application in question. In both Android 13 and iOS 16.4.1 this transition follows a container transform pattern [131] in which the icon slowly scales up to the full screen and transforms into the application view. Despite following the same general idea, there are some differences between platforms. While on both platforms the icon grows and slowly fades into the app view as it scales up, the execution differs. On iOS the view appears in full and scales up to eventually fit the full screen, similar to a zoom. Meanwhile, on Android the icon reveals the view as it scales up with its outer

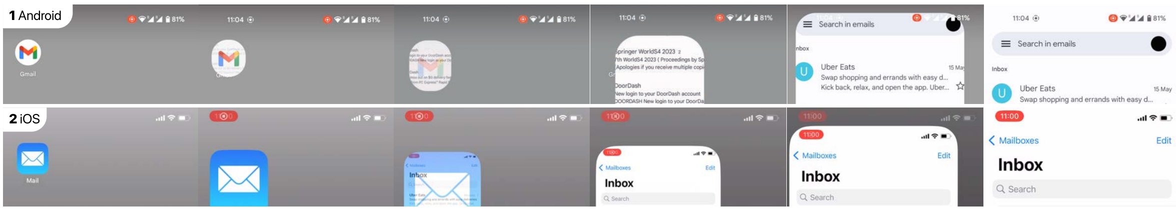


***Figure 5.5** – Animated transition between homescreen icons and the application they open in Android (1) and iOS (2). In both systems, this transition follows a "Container Transform" pattern in which an icon in view A scales up and transforms into view B, which then continues to scale up and fill the screen.*

corners out of view initially. Additionally, on iOS the background darkens and blurs as the icon transforms. Finally, on Android, the top status bar displaying the time, battery status, and so forth, remains a consistent top level layer, while the status bar on iOS appears as part of the new view, eventually scaling up to cover the previous status bar.

These patterns make it clear that animation has become an established part of transitions between views and has successfully been utilised to improve the user experience as well as understanding of the system.
When examining the roles of animation, Chevalier et al. identified one to be specifically targeted at providing teaching aid through affordance and preview [40]. Given the lack of discoverability of swhidgets and the success of visual signifiers and animated transitions as well as the potential of using animation as an affordance preview, we create animated transitions including visual signifiers for "swipe-ability" in the form of transient interaction hints. In doing so we create signifiers that, following the design space introduced in section 4.4, are temporary, on and in the vicinity of the point of interaction and mirror established digital signifiers.

5.3 Animated Transitions displaying transient interaction hints

Animated transitions are already commonly used to establish a coherent spatial model of applications [112] (see Section 5.2.3). We see an opportunity to leverage these animations to their fullest potential, as a means to increase exposure to swhidgets. We propose to address the problem of lacking visual signifiers for swhidgets while maintaining an uncluttered interface, by extending the function of animated transitions to include transient interaction hints. Doing so requires minimal changes in current interfaces: adding visual components that show the presence of swhidgets beyond the usual screen bounds can be achieved while maintaining the animation style that provides context and help to navigate a system.
We focus on view and item swhidgets (see subsection 5.2.1), as they are most prone be inconsistent and overlooked. System swhidgets, in contrast, are well established and an integral part of mobile user interfaces. They are often communicated through product advertisements and keynotes [144]. Therefore, it is likely users would already be aware of their existence and confident in their use. Additionally, since their implementation has been constricted to a specific use, adapting system swhidgets to other use cases would be a cause for confusion that would hamper the potential increase in discoverability achieved through transient interaction hints.

5.3.1 Transition Type

Since we aim to extend the function of existing transitions, we lean on currently implemented animation types to expand upon. Leaning on the Material Design help guide [131], we adapted transition patterns to incorporate transient interaction hints for swhidgets, where appropriate. We describe our modified transitions in the following. Short clips are also provided in the supplementary material and video figure.

Container transform transitions (see Figure 5.5) were adapted to include swhidgets beyond the screen bounds in the early stages of the transformation, when the outer edges of the shape are still fully visible (see Figure 5.6). The visible swhidgets then move beyond the screen bounds naturally as the container scales up from view A, to transform into the end position at view B.

Figure 5.6 *– Adapted 'Container Transform' animation of an email application opening, showing the available swhidgets on both sides as the icon scales up.*

Forward and backward transitions use a horizontal sliding motion indicating forward and backward movement between screens. Similarly to the *Container transform* transitions, they were adapted to show swhidgets before they move to their position beyond the screen bounds (see Figure 5.7). As view B moves into the screen from the left, we show the swhidgets to the right of view B before they disappear on the right side of the screen in the final view B.

Figure 5.7 *– Adapted 'Forward and backward' animation of an email application opening, with the application sliding in from the right side, revealing the swhidgets to the right of the usual application view before reaching its end position.*

Panels transitions combine *Top Level* and *Enter and Exit* transition patterns. Individually, top level transitions are not suitable to communicate the existence of swhidgets as they do not include motion and immediately arrive at the end state, view B. However, this near instantaneous transitions between states A and B allows additional time to be dedicated to transient interaction hints. We implement these by adding the swhidgets, which would typically not be visible in view B, to the initial view loaded through the top level transition and then move them to their location outside the screen bounds through an exit transition. As this results in panels with the swhidgets moving out of frame, we subsequently use 'Panels' as shorthand for this transition.

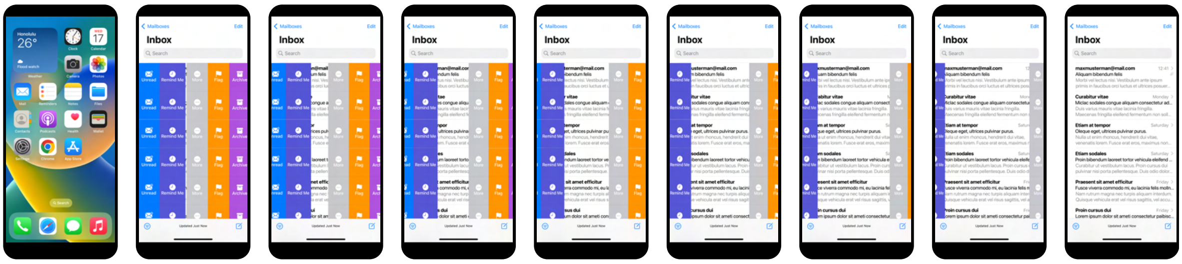


Figure 5.8 *– Adapted 'Combination Top Level + Enter and Exit' animation of an email application opening, with the application opening instantaneous and the available swhidgets slowly sliding out through an exit animation.*

All transitions are designed using a slow-out animation pacing, which has been found to improve user performance in comparison to other pacings [64], and is in line with industry standards [129].

5.3.2 Transition Duration

Previous work suggests that the minimum duration for conscious perception is shortly after 200ms [31,191]. This is in line with current conventions for transitions, with Material Design guidelines suggesting durations between 200ms and 500ms for different transition types, and offering default values for motion durations up to 1000ms in their 'Design tokens' [129]. This maximum duration matches the suggestion that for user interface responses, 1000ms is the maximum time for the user's flow of thought to stay uninterrupted [160]. Following these considerations and given the aim for the transitions to be consciously perceived, we implemented transitions with an initial duration of 1000ms.

5.4 Study 1 - Noticeability

We first run a study investigating which TRANSITION TYPE and DURATION are noticeable to users. We do so before investigating their impact on discoverability, to verify that the visual signifiers embedded in animated transitions are noticeable, which we consider a prerequisite for improving discoverability.

5.4.1 Experimental Design and Procedure

Our independent variables were TRANSITION TYPE (see section 5.3.1) and DURATION (see section 5.3.2). To avoid learning effects, we used a between-subjects design where each participant only saw one of the TRANSITION TYPE × DURATION combinations.
The study followed the procedure depicted in Figure 5.9 and took an average of 5 minutes to complete. Participants were handed a phone and asked to press the email icon, which

triggered an animated transition of the mail application opening (see section 5.3.1). They were then asked to describe what they had seen. Subsequently, they were shown a slowed down video of the animated transition and asked whether this was a slowed down version of what they had just seen.

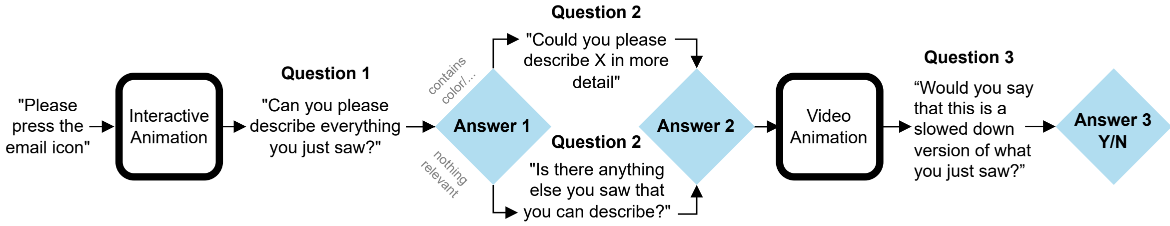


Figure 5.9 – Procedure for noticeability study. Questions and instructions from the experimenter are shown in quotation marks. Encounters with the animated transition on a phone are shown with black borders. Recorded answers are highlighted in blue, they are coded as described in section 5.4.3.

Both animations (Interactive and Video) were played on an iPhone XS running iOS 16.4. In an effort to ensure the novel component of the interaction would be the transition, we opted to use a familiar interface as the basis of the transition animations tested (i.e. iOS homescreen and e-mail).

The transition animations were created using Adobe After Effects. For the "Interactive Animation", the animations were then exported as image sequences with 60 images per second. We built a web application using PixiJS to load these images and play them back when the user touched the email icon on the screen. To ensure consistent and precise durations of the animation, images are displayed according to the passed time, skipping some frames if required depending on the load of the system. To avoid biasing responses, neither question about this animation (Question 1 & 2) explicitly mentions transitions.

The "Video Animation" was the same "Interactive Animation" transition the participant had just seen, but this time as a video playing on the same phone with an extended DURATION of 5 seconds . Participants were then asked whether the video was a slowed-down version of what they had seen previously (Answer 3).

5.4.2 Participants

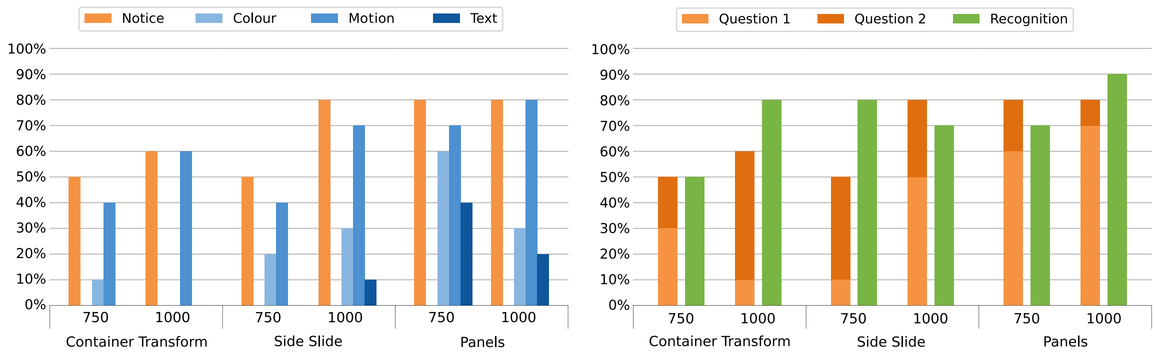
We recruited 60 participants via opportunistic street sampling and word of mouth. They were aged 18 to 65 (Mean=27.5, SD=9.1) with 33 identifying as men, 25 as women, 1 as non-binary, and 1 preferring not to disclose their gender.

5.4.3 Data Coding

Considering responses to "Question 1" and "Question 2", we manually coded responses that included descriptions of the transient interaction hint in the animated transitions. This revealed three unique characteristics that were described: colour, motion, and text/icons. If either characteristic was described in connection to the interaction hint (i.e. *"there was blue and orange on the side"* or *"the view moved towards me and sort of got bigger"*), we recorded the participant as noticing this characteristic. We then created an aggregate score of participants who had described any of the three characteristics and recorded it as a percentage of overall *notice* of the transition. Finally, we recorded the responses to "Question 3" as "Recognition", with participants responding with "Yes" for "Answer 3" considered to have recognised the transition. However, responses to this question may be a more successful indicator of what people definitely *did not* notice, given that the video and question test recognition rather than independent recall. A 'No' response to the recognition question suggests that participants did not notice the transition at all.

5.4.4 Results

This study revealed differences in the noticeability of the different characteristics considered depending on TRANSITION TYPE and DURATION (see Figure 5.10).



***(a)** Noticeability of characteristics* ***(b)** Overall noticeability separated by question; recognition*

***Figure 5.10** – Percentage of participants who noticed the transition overall (orange), noticed particular details (blue) and recognised the transition (green) separated by transition* DURATION *(in ms) and* TRANSITION TYPE.

The TRANSITION TYPE "Panels" achieved the highest noticeability overall, as well as the highest percentage of notices of colour, motion, and text/icon, suggesting it is the most noticeable. In contrast, "Container Transform", which mirrors the existing transition for the presented interaction most closely, achieved the lowest noticeability overall with low percentages of notices of the individual characteristics.
Out of 60 participants, only 4 explicitly referred to the swhidgets when describing the screen—3 of which were shown the "Panels" transition (2 × 1000ms, 1 × 750ms) and one was shown the "Side Slide" transition (1000ms).

5.4.5 Discussion

Given the comparatively higher description rate for transition characteristics like colour, motion and text/icons in response to question 1 and 2, it appears "Panels" was the most noticeable TRANSITION TYPE. Interestingly, the "Container Transform" TRANSITION TYPE was overall described in less detail suggesting it was less noticeable. However, the high rate of question 2 notices for "Container Transform"×1000ms may indicate that this lack of characteristic descriptions may be partially attributed to participants not considering the transition noteworthy rather than them failing to notice it. Since the transition most closely resembles current commercial interfaces, it is the least "different" which may detract from its noticeability while other transitions may stand out by varying from the established norm.

When examining DURATION, the 1000ms transitions were remarked upon repeatedly as noticeably slow, with participants stating things like *"It was kind of slow"* (P5), *"A very slow mailbox opening up... super slow"* (P20) and *"It was coming very slowly I think, the moving generally, my phone it appears right away"* (P25). While some participants also commented on the 750ms being slow, other participants commented on the transition happening fast, stating *"I couldn't remember it, because it went away so fast"* (P36) and *"actually it was like pretty fast, so I could just see like the title of the emails, some of the emails, not all of it"* (P38).

The decreases in some noticeability measures suggest that for the purpose of a discoverability study, further decreasing the DURATION of the transition is not advisable. However, the increase in colour and text notices for the "Panels" transition when decreasing the DURATION suggests that noticeability varies widely between users and a lower transition DURATION may prove to be noticeable to some users, especially when implemented permanently, allowing repeated exposure.

Given these results, we proceed to run a discoverability study using the most noticeable transition (Panels) as well as the one most closely resembling current interfaces (Container Transform) at a DURATION of 750ms. We do not include the 1000ms DURATION due to its negligible increase of noticeability and the repeated references to it's unusually slow movement.

5.5 Study 2 - Discoverability

We examine the impact of transient interaction hints on the discoverability of swhidgets, by comparing two of the previous TRANSITION TYPES ('Container Transform', 'Panels') to a baseline transition with no interaction hints embedded.

5.5.1 Experimental Design and Procedure

The experiment followed the procedure outlined in Figure 5.11. To test the discovery of swhidgets, we created a web application featuring 6 tasks with different swhidgets implemented in the UI (see Figure 5.11 and section 5.5.1.1). All tasks were accessible from a "Homescreen" featuring 6 buttons with the corresponding task numbers on them. Tasks 1–4 were pictionary tasks, where participants had to draw a given word or phrase, whereas tasks 5 and 6 were questionnaires in which participants were asked to indicate their agreement/disagreement with a statement. The pictionary tasks were designed to make the use of colour increasingly necessary with each task. When tapping a task button, the corresponding task opened playing a 750ms transition, which was implemented in the same way as study 1. The TRANSITION TYPE remained the same for all 6 tasks.

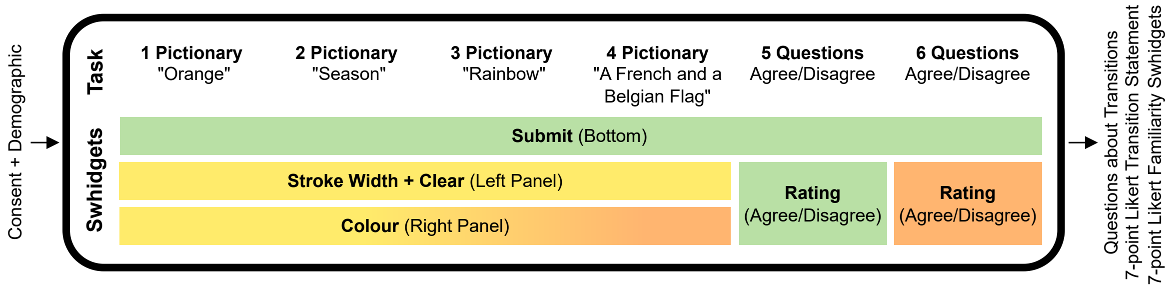


Figure 5.11 *– Outline of the procedure for study 2 and overview of tasks and the swhidgets available for them. Swhidgets in green are necessary to complete the task, but another way to achieve the function is available in the interface. Swhidgets in yellow are not necessary to complete the task but are the only way to access this particular function. Swhidget in orange are necessary to complete the task and the only way to access this particular function. Figure 5.12 depicts the swhidgets and their position on the screen bounds of the phone.*

Following a split-plot design, each participant experienced only one TRANSITION TYPE (section 5.5.1.2), while all participants were presented with the same TASKS featuring the same SWHIDGETS.

The interaction with the web application was followed by questions about the participant's experience. First, participants who used the swhidgets available while completing tasks 1-5 (i.e. swhidgets they were not *forced* to discover) were asked how they knew they could swipe from the screen edge to reveal additional functions. Participants who had not used the swhidgets were asked whether they were aware of them. Afterwards, participants were asked to rate the noticeability of the transition on a 7-point Likert scale (1: Not at all noticeable – 7: Extremely noticeable). Finally, participants were asked to indicate their agreement/disagreement with 4 statements to gauge how they perceived the animated transition, as well as their familiarity with different swhidget types (1: Not at all – 7: Extremely high).

5.5.1.1 Swhidgets

We integrated different SWHIDGETS into the UI of our study application. For the pictionary tasks, we made drawing tool menus available through swiping from the left (stroke width + clear canvas) or the right (colour selection) edge of the screen (Figure 5.12 left). Task 4 substantially increases the need for colour through the prompt of two drawings that differ solely in colour, making discovery of the tool menu almost essential to properly complete the task. For the question tasks, participants could indicate their agreement/disagreement with a statement by respectively swiping left or right on the item (Figure 5.12 right). While the first question task (task 5) also allowed agreement/disagreement to be registered through tapping on the item and selecting agree or disagree on a pop-up screen, this option was removed for the final task 6, forcing discovery to complete it. Additionally, a SWHIDGET to 'Submit' was implemented across all tasks, triggered by swiping upwards from the bottom of the screen. However, closing the tasks was also possible through a three-dot menu in the top right corner, which revealed a menu with a sole 'Submit' option.

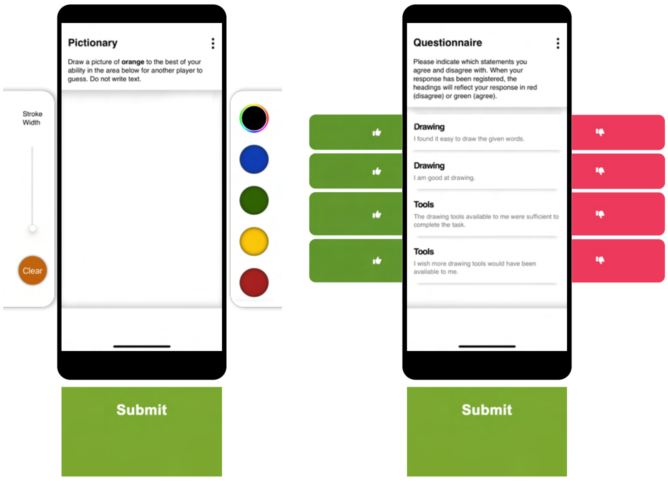


Figure 5.12 – SWHIDGETS *implemented in the discoverability study for the drawing task (left) and the question tasks (right).*

The two drawing tool menus (left [stroke width + clear canvas], right [colour selection]) follow the "Snap to end position" pattern described in Figure 5.3, meaning they reveal action items once a translation threshold is reached, i.e. they can be "opened". In contrast, the 'Submit' and 'Rating' SWHIDGETS immediately execute actions when a given translation threshold is reached. Finally, it should be noted that, from a UI component perspective, the SWHIDGETS at the bottom, as well as the right and left edges of the screen for tasks 1-4, are novel, while the 'Agree/Disagree' rating SWHIDGET for tasks 5 and 6 mirrors the behaviour of existing list views such as emails.

5.5.1.2 Transitions

In addition to the 'Container Transform' (Figure 5.14) and 'Panel Exit' (Figure 5.15) TRANSITION TYPES from the noticeability study, we include a baseline without transient interactions hints. Following the same patterns as 'Container Transform', which is the standard pattern for this type of transition, it does not depict the swhidgets at any point (Figure 5.13).

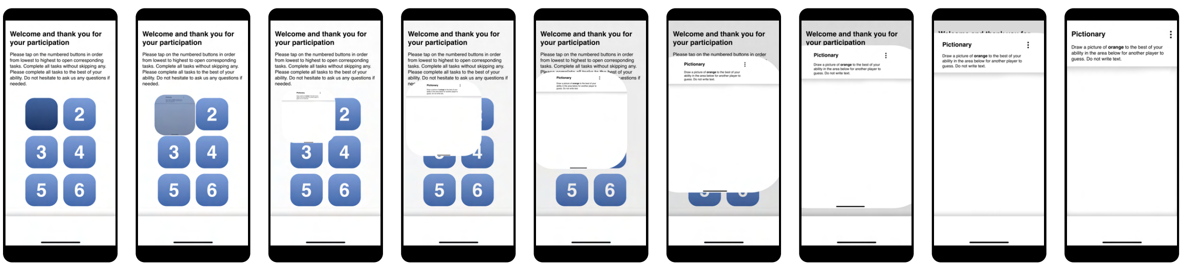


Figure 5.13 – *Baseline transition between the task 'homescreen' and the individual tasks, based on a container transform pattern without transient interaction hints.*

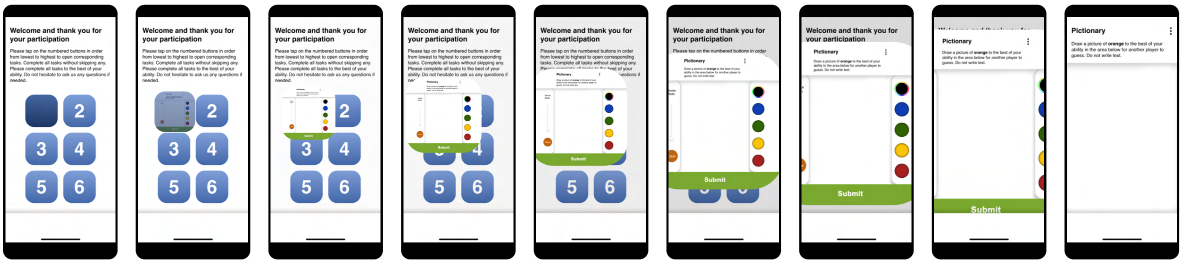


Figure 5.14 – Container transform transition between the task 'homescreen' and the individual tasks, based on a container transform pattern including transient interaction hints at the edges of the container, that move out of view as the end state of the transition is reached.

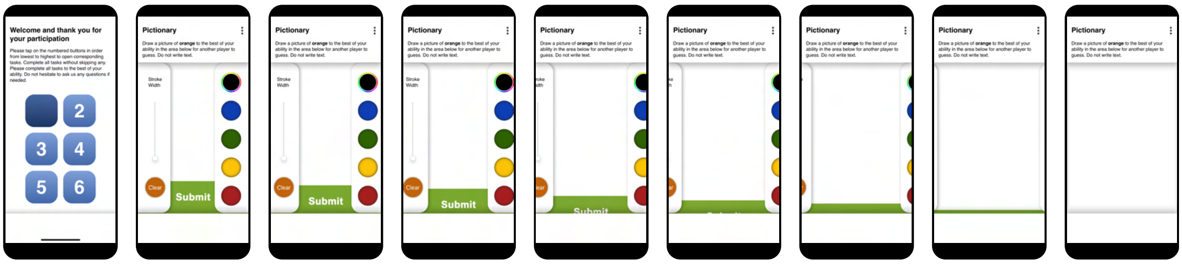


Figure 5.15 – Panel exit transition between the task 'homescreen' and the individual tasks, showing the swhidgets showing up on the edges of the screen as 'Panels' before exiting the screen in the direction in which they are accessible.

5.5.2 Participants

We recruited 33 individuals (23–57 years old, Mean=30.45, SD=7.43; 22 self-identified as men, and 11 as women) to participate in the study, each completing the 6 tasks with one consistent TRANSITION TYPE throughout the tasks. None of the participants participated in study 1 or otherwise previously encountered the transitions.

5.5.3 Results

We employ two metrics as indicators of discoverability. First, we use the percentage of participants that used, and therefore discovered a given swhidget. Second, we examine the time from the point a swhidget was available to participants, to their first use, therefore measuring the amount of time it took to discover a given swhidget.

5.5.3.1 Usage/Discovery Percentage

Between the two pictionary swhidgets, the right swhidget (colour tool) was used by more participants. As Figure 5.16 illustrates, the majority of participants discovered and used the right swhidget by the final pictionary task (task 4) which made colour essential to completing it (drawing a French and a Belgian flag). Nonetheless, there were 7 participants (2 'None', 4 'Container Transform', 1 'Panels') who used neither the left nor the right swhidgets at any point. When questioned, all of them indicated that they had not been aware of the swhidgets.

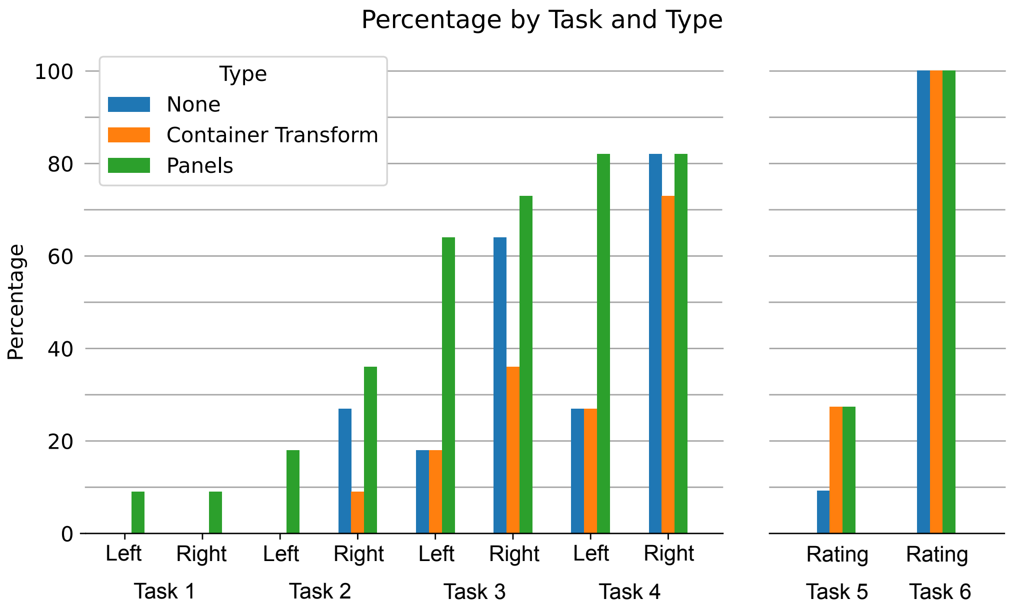


Figure 5.16 *– Plot showing percentage of participants that used a given swhidget per task and transition type. In task 1-4 left and right menu swhidgets were available, while task 5 and 6 included swhidgets to agree/disagree with a statement. Task 6 made the swhidget necessary to complete the task.*

When examining the usage of the 'Agree/Disagree' rating swhidget in tasks 5 and 6, very few participants discovered and used the swhidget in task 5. This was despite the increased effort presented by the alternative way of completing the task, which was a popup that appeared when tapping the statement that then offered an agree and a disagree button. Aside from increasing the steps required to agree or disagree with a statement, this popup appeared on top of the statements, leading to participants opening and closing the popup repeatedly to read the statement again. Notably, of the participants who *did* discover the swhidget in task 5, none returned to using the popup, indicating that the swhidget was the preferred interaction method after it was discovered. In task 6, which made discovery of the swhidget necessary due to the removal of the popup, all participants were able to independently discover the swhidget and complete the task.

Observation during the study revealed, that participants who had not discovered the swhidget in task 5, largely attempted other interaction methods before discovering the availability to swipe in task 6. Upon the removal of the 'Tap popup rating' in the final task, participant repeatedly attempted to tap nonetheless, as well as tapping different locations and in specific orders, using force and dwelling. Despite swipe being a common interaction in list views and the majority of participants indicating general familiarity with list view swhidgets (Figure 5.17), it was not the first interaction attempt following tap. Nonetheless, all participants did discover the swhidget by themselves during task 6. One in particular only noticed the transitions for the final two tasks. After discovering the swhidget at the end of task 5, they noted upon seeing the transition at the beginning of task 6: *"Oh that's why there was a slide animation."* [P26 - Panels], indicating that they had noticed the animation but did not connect it to the swhidget or the possibility of swiping until they had accidentally discovered the interaction.

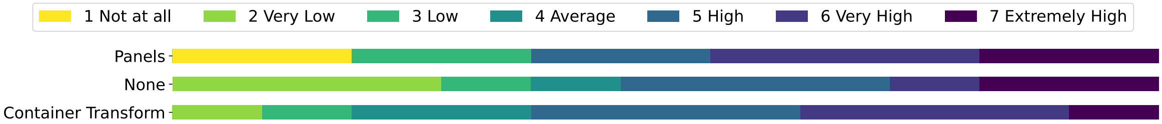


Figure 5.17 *– Plot showing participants rating of their own familiarity with list item swhidgets on a 7 point Likert scale, separated by transition type.*

The 'Bottom Submit' swhidget was only used by a total of three participants, one in each TRANSITION TYPE condition. Multiple participants tried to either 'tap' the bar at the bottom or attempted to swipe up from a wrong position. However, upon questioning their reasoning, it became clear that none of the 10 participants who attempted to swipe up in the wrong position did so to submit the task or because they had noticed the option in the transition. Their reasoning varied from searching for further tools (*"I'm looking for an undo, there must be one...."* [P26 - Panels]) and testing all swipe directions (*"I saw that there was something hidden [left and right] so I tried all [the directions from the screen edge]."* [P21 - Container Transform]), to familiarity with the general availability of swiping up or a lack of familiarity with the iOS interface in particular (*"I didn't know what that [iOS bottom bar] does, so I thought it might be used for that."* [P24 - Container Transform]).

5.5.3.2 Timing

We consider the time until a SWHIDGET was first used from when it was first available to the participant (i.e. when the first task in which the swhidget is available was opened). After filtering out participants who never used a given swhidget, there is no significant effect of TRANSITION TYPE, for any of the SWHIDGETS implemented (Figure 5.18). For the pictionary

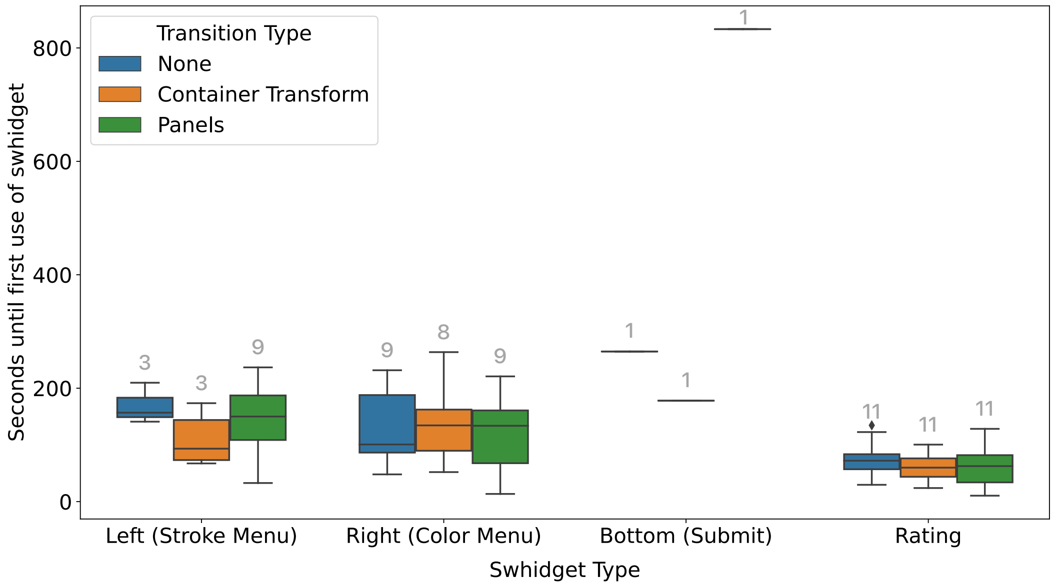


Figure 5.18 *– Time in seconds until a given swhidget was used for the first time from the moment it became available, separated by the swhidget type and transition type viewed. Since number of observations varies between* TRANSITION TYPES *and* SWHIDGETS*, it is indicated as the number above the plot.*

tasks, neither the left ($\chi^2(2) = 2.447, p = 0.29$), nor the right ($\chi^2(2) = 0.407, p = 0.82$) menu SWHIDGET vary significantly in the time required until the first use of the swhidget between TRANSITION TYPES. Similarly, the time until the 'Agree/Disagree' rating SWHIDGET was first used in the question tasks, does not vary significantly between TRANSITION types ($\chi^2(2) = 1.145, p = 0.56$). The number of data points varies between the different SWHIDGET and TRANSITION types, as some participants did not use either of the swhidget menus in the drawing task, while all participants were forced to discover the rating swhidget in the final task, as the only means to complete it.

5.5.3.3 Transition Perception

When examining the agreement with the different transition statements, we did not find significant differences for any of the questions between transition types.

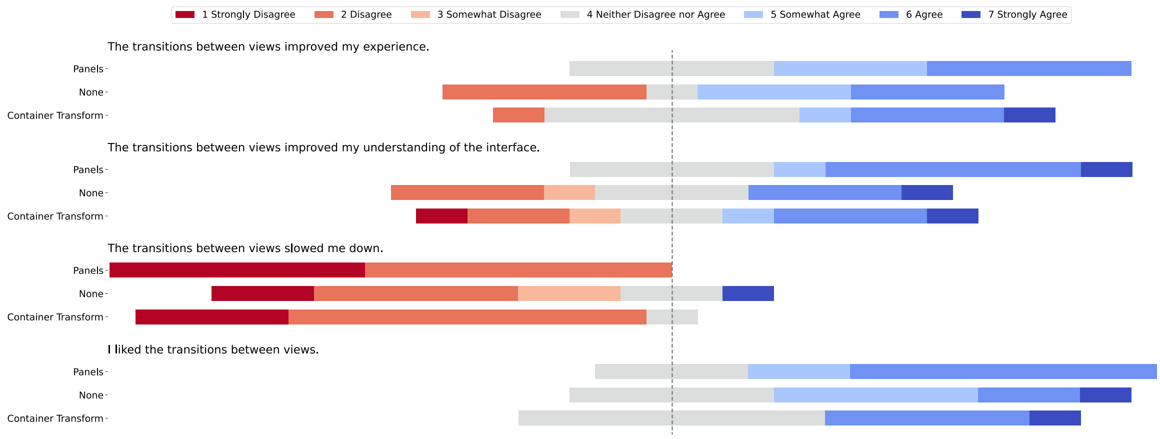


Figure 5.19 *– Plot showing participants agreement rating with the statements "The transition between views improved my experience", "The transition between views improved my understanding of the interface", "The transition between views slowed me down" and "I liked the transition between views" on a 7-point Likert scale, separated by transition type.*

Participants who had used the available swhidgets throughout the tasks, were prompted to explain how they knew they could swipe from the screen edge to access additional menus and features. Most participants reported coming across the swhidgets 'accidentally' (P1, P4, P6, P8, P15, P19, P20, P26, P28) or 'by chance' (P2, P13, P21, P22, P30). Some participants reported they assumed such functionality through familiarity with other interfaces featuring similar swhidgets (P12, P16, P17, P23, P31) or through intuition (P18), while others stated they simply 'guessed' (P11, P29).

Notably, only participants who were shown the 'Panels' transition mentioned the animation at all. P8 noted that *"at first, it was accidentally. Once I realised about this, I noted that it appears at the beginning of the 'game' but very fast"*. Similarly, P14 stated that they knew about the swhidgets *"thanks to the preview at the beginning of the task and also because I accidentally touch a border"* and P20 reported that *"when starting the task, the side menus were apparent and collapsed. This let me know they existed. But I only knew how to access them by error when trying to draw"*. The sole participant who appeared to discover the swhidgets exclusively due to the animation (P32), was hesitant and asked whether *"the colour in the animation, can it be used?"*, stating *"I did not try it myself though I thought about it, but thanks to the animation at the beginning"*. Conversely, another participant in the 'Panels' condition stated they became aware of the swhidgets, because they *"guessed after a few tasks but there was no sign"*, corroborating the findings of the noticeability study, that perception of the transition animation varies substantially between individuals.

Despite this variance between participants, qualitative analysis of the responses suggest, that the 'Panels' TRANSITION TYPE was the most noticeable, given that it was the only TRANSITION TYPE participants cited as a source of their awareness of the swhidgets. However, when considering the reported noticeability ratings for the different TRANSITION TYPES, there is no significant difference in how noticeable the transition is reported to be between groups ($\chi^2(2) = 0.476, p = 0.79$, see Figure 5.20).

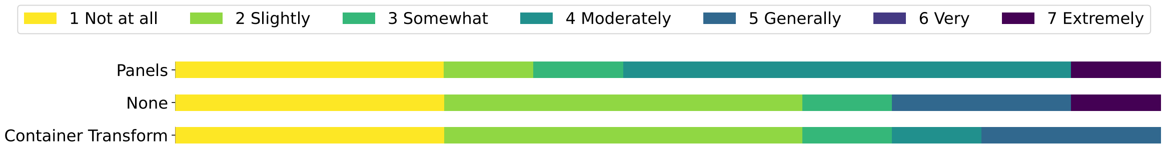


***Figure 5.20** – Plot showing the reported noticeability of the transition on a 7-point Likert scale separated by transition type.*

5.6 Discussion

The results of the studies in this chapter indicate that transient interaction hints are noticeable when the animation has a duration of 750 ms, but do not improve the discoverability of swhidgets. This presents a departure from the prevailing assumption that increased visual depiction increases awareness of a given feature or interaction method which was supported by our studies in the chapter 4.
That our results suggest no improvement through added visualisation raises the question of whether visualisations are not noticeable in certain contexts as well as whether increased noticeability actually translates to increased discoverability.

5.6.1 Are Transient Interaction Hints Noticeable?

The results from study 1 suggest that, at a minimum the 'Panels' transition was noticeable at a duration of 750 ms since participants not only noticed multiple features of the transition, but were able to describe individual characteristics of the transition. In contrast, when considering the subjective ratings of noticeability in study 2, no group rated the transition they were seeing higher than an average of '3 - Somewhat'. While these measures do not allow a direct comparison, they suggest the transitions, were, at the very least, perceived to be less noticeable in the second study. However, individual comments during study 2, contradict the assumption that the transitions were actually less noticeable, since even participants who did not use the swhidgets and indicated they were not aware of them, were able to describe the transitions (e.g. P5 (Panels) *"the screen splits in the middle to give way for the task space"*). Nonetheless, the context of study 2 may have changed the perception and subsequent conscious perception of the transitions. While multiple participants in study 1 independently commented on the transitions being slow at 750 ms, participants in study 2 largely disagreed with a statement that the transitions slowed them down.

Additionally, study 2 presented tasks to the participants, removing their focus from the transitions. This is supported by statements like *"I was so focused on the drawing part and the changes in the interface that I did not realise there were transitions"* (P6 - Container Transform) and *"I didn't pay attention to it at the time"* (P29 - Panels). The participants' notion of what is important in the study may contribute to a cognitive focus on the tasks themselves, disregarding anything else and therefore not considering the transitions due to inattention and change blindness [138].
As the previous comments from P6 and P29 indicate, participants did not pay attention to the transition animation, going so far as to state that they were focusing on *"changes in the interface"* and *"did not realise there were transitions"*, indicating that they did not consider the transition animation to be a part of the interface. Previous studies and state of the art suggest that animated transitions are successful in conveying information about the layout of applications and can support users in building more accurate mental models [112,131]. However, these comments suggest that the participants' mental model of the task and transitions in general may have led them to consider transitions as superfluous. If they considered interface transitions to be purely decorative, they may ignore them in favour of focus on a given task. This suggests that it may be difficult to adapt previously decorative features to include and transfer information as the users' previous experience may prejudice them to pay the feature no attention.
Nonetheless, given the ability of participants to describe the transitions and the increasing need of the tools provided in the swhidgets to complete the task, it is surprising that the perception of the transitions does not appear to translate into increased discovery of the swhidgets or exploration of the interface.

5.6.2 Why doesn't Noticeability Translate to Improved Discoverability?

While discoverability of swhidgets was not improved through transient interaction hints, once they were discovered they were the preferred interaction method over alternatives in menus and pop-ups, confirming the finding of previous studies [177]. This suggests that the delay or lack of use of swhidgets was not due to a preference for alternatives, but rather a lack of discovery.
One potential explanation is that, while noticeable, animated transitions are not perceived as relevant to the interaction and observations of signifiers embedded in them are discarded without further consideration.
Additionally, even participants who not only noticed the transitions, but listed them as part of why they were aware of the swhidgets, did not infer the availability of the input. For instance, P20 states *"when starting the task, the side menus were apparent and collapsed. This let me know they existed. But I only knew how to access them by error when trying to draw"*, and P32 was hesitant and not trying to access the swhidget independently, despite seeing the animation and wondering whether *"the colour in the animation can be used"*. This indicates that even noticed visual depictions may not invariably result in improved discoverability of interactions.
To explore this, we informally tested transitions with an extraordinarily long duration of 5000ms showing a slow-motion animation [214] with 4 participants (2 "Container Transform", 2 "Panels"). Supporting our hypothesis that transitions are not considered relevant, the 2 "Container transform" participants did not discover the swhidgets through the animation. While both reported to have seen the transition animation, they stated they were "too focused on the task" and "didn't really pay attention". In contrast, the 2 "Panels" partici-

pants did discover the swhidgets earlier and directly cited the transition animation as the cause stating *"I saw this when the app started"* and *"I didn't notice it at first [...] but [for the second task] I paid more attention at the beginning when it appears"*. This suggests that transitions that significantly differ from the established norm may be more likely to successfully convey interaction. However it should be noted that none of the participants discovered the "Bottom (Submit)" swhidget, with 3 not noticing it at all in the transition and 1 ("Panels") stating *"my view was drawn towards the top because I noticed that's were the instructions were, so I didn't see what it was, just that there was something"*. Additionally, all participants began reading the task as the text became legible during the transition animation, suggesting they were not inclined to spend time paying attention to the transition itself.

One possible reason noticeability may not translate to discoverability, is that the depiction of the available swhidget shows the features' availability but not the associated interaction. While the interaction to access the swhidget is implied through the directional movement of the animation, it was not immediately clear to participants, who failed to deduce the "swipeability" of screen elements. This suggests that the animated transitions did not support the generation of an accurate mental model of the application.

Another possible cause may be that the transient interaction hints are displayed before the interaction is available, and most importantly needed, rather than concurrent to it. While the swhidgets are available immediately after the transition, participants were primarily occupied with gauging their task in the first moments of each task. Potentially, depictions of the swhidgets once participants had determined their strategy to draw a given word or had read the statements they were indicating their agreement with, would have garnered more success by showing their availability as they were needed. This is supported by previous work, finding that interaction signifiers concurrent to the interaction were more successful [135].

Finally, swhidgets may be largely associated with their command, rather than with the component in which they are embedded. Users may be aware that they can, for example, delete a list item by swiping on it and therefore only consider this interaction in this exact context and with this specific aim. This is supported by previous work [177] finding that users suspected the presence of swhidgets in screen layout mirroring those in which they were familiar with swhidgets, making it unlikely for them to discover swhidgets in new contexts. Since the list view in our experiment does not allow users to delete the items, but rather asks them to agree or disagree with the statements, participants may not associate the swhidget with the desired command and therefore discard the input because they deem it to be improbable.

5.7 Conclusion

Increased noticeability through visual signifiers implemented as transient interaction hints in animated transitions does not increase the discoverability of swhidgets with our sample. This raises questions of how transitions are conceptualised in users' minds, whether decorative UI elements are automatically disregarded, if interaction hints need to be concurrent to the interaction as well as whether increasing noticeability fosters discoverability in general. Given the previously supported hypothesis that increased visualisation improves discoverability, this provides a compelling indication of how interaction availability may be conceptualised differently than assumed and how inputs are discovered while offering multifold paths for future work.

6 General Discussion and Perspectives

This thesis was driven by the goal of improving the discoverability of interactions on mobile touchscreens. In particular, the core thesis was that

> Interaction discoverability can and *should* be supported in interactive systems, for which visual signifiers provide a valuable resource which is critically underutilised.

To support this, we first clarified the terminology and examined impacting factors before isolating visual signifiers as a resource. Investigating them in more detail, we proposed a design space of visual signifier characteristics and implemented various signifier designs to validate their impact on interaction discoverability in multiple user studies. Below the progress towards the objective of this thesis and its contributions are summarised before the implications of our findings are discussed and directions for future work are outlined.

6.1 Summary of Contributions

This thesis provides a fundamental overview of discoverability in human computer interaction and examines the potential of visual signifiers in the interface to improve interaction discoverability. The following sections summarise these contributions.

6.1.1 Theoretical - Clarification and Exploration of Discoverability

There is a lack of consistency and clarity when it comes to the definition of discoverability and related concepts such as learnability, which we hypothesis is once cause of its deficient consideration. Thus we examine existing definitions of discoverability in the context of interaction and clarify discoverability by providing a comprehensive definition and separating system, feature and interaction discoverability. Additionally, we clearly distinguish discoverability from related concepts and highlight how these concepts interrelate and differ.

We then examine established human computer interaction concepts and provide a framework illustrating how they interrelate and impact initial interactions and discoverability in particular. In this framework we identify factors impacting the interactions to be part of one of three *actors* - the system, user or environment.

Finally, after identifying visibility and signifiers in the framework as factors with substantial potential for improving the discoverability of interactions, we propose a design space of visual signifier characteristics, considering their *timing* and *proximity* in relation to the point of interaction as well as their *explicitness*. Through the characteristic *explicitness* we examine the level of interpretation and mental effort expected from the user to comprehend the signifiers intended meaning.

Altogether, this provides a much needed basis for future considerations of discoverability and visual signifiers. Establishing a shared vocabulary allows researchers to clearly communicate focal points while the framework provides an understanding of relevant concepts, as well as a basis of factors that may be leveraged to improve discoverability or examined to determine the cause of its failure. Similarly, the signifier design space offers both a catalyst for the future development and design of visual signifiers as well as a framework through which existing signifiers can be classified and compared.

6.1.2 Methodological

Studying discoverability presents inherent difficulties of avoiding increased discoverability through the study setup itself. Things inherently can only be discovered once and given feasible study durations, are unlikely to be forgotten and "rediscovered". The discovery of one thing, be that a feature or interaction method, primes users for further discoveries, obscuring the source of discovery. Additionally, requiring discovery to complete a task incentivised participants, while not requiring it may lead to ineffectual research in which participants do not engage with the thing to be discovered.
Consequently, in the context of reviewing discoverability and related concepts, we briefly review how discoverability has been studied in previous work in subsection 2.2.5.

We then reflect on methods employed and their varying success and employ different methods within our own studies. As the methods employed in the studies in subsection 4.5.1 are designed to capture the effect of various signifier characteristics, they do not measure discoverability but whether specific signifier designs communicate the intended input. In doing so, they more closely align with *guessability* studies. Similarly, the method in section 5.4 was specifically employed to measure *noticeability*. While both these concepts impact discoverability, their measurement does not fully capture discoverability. In contrast, the user study in section 5.5 does examine discoverability and subsequently presents an example of a possible methodological approach while also providing a basis for further reflections (see subsection 6.3.1).

6.1.3 Practical

To investigate the potential of visual signifiers for interactions on mobile phones, we created two sets of signifier designs. While they provide a basis for our examination which resulted in multiple practical takeaways, they additionally act as a starting point, guideline and inspiration for future implementations of visual interaction signifiers.
For example, the designs introduced in subsection 4.4.2 form the basis for signifidgets [35], which is a reflection on how programming APIs could approach the use of widgets, providing the opportunity to add possible user inputs and concurrently modify the widgets behaviour and appearance to visually signify the change to the user.

As illustrated in Figure 6.1 the visual depiction of the button changes as input listeners and callbacks are added to the code. When a widget is added it is visualised as is seen in (a), as no interaction is available. Once any type of input is added, the widget changes its appearance to communicate interactivity, in this case it changes from grey to a coloured button. For 'Tap' (a) there is no added visualisation beyond the colour. For 'Double Tap' (c) the widget is displayed with two circles that get filled when the button is clicked. Similarly, for 'Dwell'

(a) *No Interaction* ***(b)*** *Tap* ***(c)*** *Double Tap* ***(d)*** *Dwell* ***(e)*** *Dwell Levels* ***(f)*** *Combination*

Figure 6.1 *– The different visual states of signifidgets as interactions are added to a widget programmatically.*

(d) a scale is displayed above the widget that fills as the dwell duration is reached. This can be expanded to include multiple thresholds of dwells (e) or a combination of various inputs (f).

In addition to this concrete implementation, subsection 4.7.1 summarises practical guidelines extracted from our study results regarding signifier design that can be applied to different designs and contexts. These include considerations of which type of design may be most suitable and when and where it should be displayed.
Building on this, chapter 5 reveals that some user interface elements may not be suitable to provide visual interaction signifiers and considers the causes for this, allowing designers to circumvent ineffective visualisations in the future.

6.2 The Importance of Discoverability

One key premise of this thesis is the importance of, not only discoverability, but in particular its increased consideration when examining or introducing interactive systems.
This thesis focuses specifically on the discoverability of touch interactions, as they are widely implemented in commercial devices and yet so diverse that the discovery of *which* input is available continues to be a struggle.
For one, there appears to be an assumption that the interactivity of commercially established devices is inherently evident to users already. However, there is an ever-expanding interaction vocabulary for touch screens with complex inputs like gestures and patterns that are increasingly not only available for a single finger, but also multi finger interaction, that necessitate perpetual discovery. Additionally, even if users are theoretically aware of different input methods on smartphones such as dwell or double tap, *where* and *when* these inputs are available is inconsistent at best, seemingly random at worst.
Furthermore, given the number of features available on modern smartphones, it is not feasible for users to first discover and then recall all of the times particular interactions are available to them without any guidance. If touchscreens are not designed to cater specifically to expert users, which given their widespread adaptation is both unlikely and inadvisable, the lack of discoverability support inherent in the system does a disservice to a majority of their user-base.

Nonetheless, this reflects only a fraction of discoverability which at large remains a challenge. Albeit circumstantial, this is exemplified by the unanimous reaction to discussion about the topic of this thesis being personal examples of interactions one had failed to discover or the sharing of interactions people consider to be unclear.
While a focus on technical possibility and validity or aesthetically pleasing and appealing

interfaces is valid and crucial, it is myopic to simply assume them to be "intuitive". Given that technology is perpetually updated and its capability expanded, the discoverability of interactions is not trivial but rather one of the fundamental problems of human computer interaction.

6.3 Measuring Initial Interactions and Discoverability

Studying discoverability presents an inherent tension between meaningful results and avoiding discovery triggered through the study setup itself and corresponding instructions.

Participants

Since discovery is a binary measure, it can only happen once - either an interaction is discovered, or it is not. This makes repeated measures impossible, leading to an increased number of participants required. Furthermore, as users might infer the goals of the study, examining interaction discoverability limits the interactions whose discovery can be studied simultaneously. This is reflected in our study in section 5.5 by participants who had discovered one swhidget inferring the existence of another, searching for swhidgets on all screen edges well as testing different interactions at random in search of additional tools, assuming they would be similarly hidden. The users' potential inference of the goals of the study additionally makes it inadvisable to include participants who are familiar with the research subject or have participated in previous associated studies. This limits the pool of potential participants and further complicates user studies.

Study Setup

To study the discoverability of an interaction method, users need to, in some way, be encouraged to interact with the system it is implemented in, as well as the interaction method in question. If the interaction method is at no point beneficial to the user, they may never discover it. On the other hand, if users are presented with a task that requires them to discover an interaction method, the study context and wish to complete a task bias the user and results do not reflect whether an interaction method is independently discoverable. Therefore, when studying the discoverability of an interaction, the interaction should *not* be necessary to complete any given task a participant is presented with. However, this makes measuring discoverability challenging, as users may suspect the availability of interactions but never verify their assumptions and discover them, simply because they do not need to use them.
To make an interaction available and likely to be discovered, yet not necessary, it can either be (1) not the only way to achieve something or (2) not necessary to complete a task.
This is reflected in the work of Fennedy et al. [69], who employ method (1) to examine a "spontaneous discovery rate", then remove the alternative way of achieving something to measure an "enforced discovery rate".

6.3.1 Reflecting on Methods in this Thesis

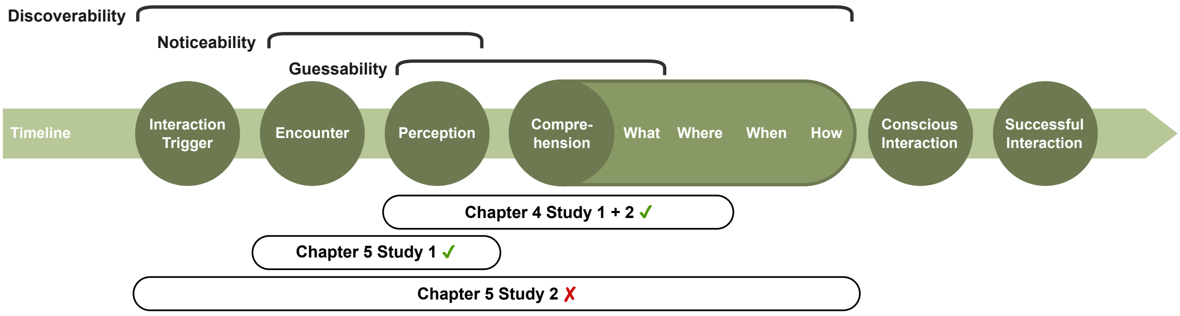


Figure 6.2 – *Studies throughout this thesis reflect on the timeline of initial interactions introduced in chapter 2.*

The methods employed throughout this thesis examine different aspects of initial interactions and discovery, as shown in Figure 6.2.

Both the online survey and interactive study employed in chapter 4 primarily examine the perception and interpretation of different visual signifiers. They focus on whether particular characteristics impact the perception of visual signifiers and examine whether comprehension can be aided through specific timing and proximity, but primarily explicitness of the signifier. The online survey does so by providing the visual signifier in form of a video and examining the interpretation of the visualisation. Similarly, the interactive study directly confronts participants with the interactive widget and examines how the perception and interpretation of available interactivity can be guided through different visual signifiers. In doing so, these studies examine which designs successfully communicate their intended meaning and therefore support comprehension. Consequently, these studies primarily examine guessability rather than full discoverability. Nonetheless, they provide important insights in how signifier characteristics may impact discoverability.
In chapter 5, the first study performed in section 5.4 explicitly aims to investigate noticeability as a prerequisite for discoverability. It does so by examining whether animated transitions incorporating transient interaction hints were perceivable upon a first encounter for different durations. Finally, the study in section 5.5 aims to study discoverability as a whole. To do so we base our setup on previous work and examine primarily "spontaneous" discovery, but cover "enforced" discovery as well. The categories in which each swhidget falls are summarised in Table 6.1. Cells A, B and D reflect spontaneous discovery, while cell C, through the necessity of interaction discoverability to complete the task depicts enforced discovery. The drawing tool swhidgets in tasks 1,2,3 and 4 cover spontaneous discovery as they are not necessary to complete the task, while the agree/disagree swhidget in task 5 and the submit swhidget throughout are necessary to complete the task but are not the only interaction achieving this outcome. The agree/disagree swhidget then transforms to reflect enforced discovery as the alternative way of rating is removed for task 6. The 100% enforced discovery of the rating swhidget in task 6 elucidates that task essential inputs are likely to be discovered, however, since most interactions with smartphones do not follow strict tasks in a laboratory setting, this may not reflect the likelihood of real life discovery.

Our study does not cover case B in table 6.1. Since this study was focused on testing whether signifiers improve the discoverability of swhidgets, we utilise cases in which discovery is not

	No alternative way of completing task	Alternative way of completing task
Not necessary to complete task	**A Discovery is not forced** Task 1,2,3,4 Swhidget Color (Right) Task 1,2,3,4 Swhidget Stroke (Left)	**B Discovery is not forced and not encouraged**
Necessary to complete task	**C Discovery is forced** Task 6 Swhidget 'Agree/Disagree'	**D Discovery is not forced** Task 5 Swhidget 'Agree/Disagree' Task 1,2,3,4,5,6 Swhidget 'Submit'

***Table 6.1** – Task discoverability classification.*

forced, but incentivised, by either providing features beneficial but not necessary for a task (A) or reducing temporal cost to execute an action (D).

While not the focus of this study, case B, which examines spontaneous discovery of different ways to access features already available to users through other means, provides an interesting perspective for future work. It additionally highlights a key difficulty when studying discoverability: Discovery that is not incentivised is difficult to capture in laboratory experiments. If participants are not incentivised to discover particular inputs within a given task, they are unlikely to do so given the general focus to complete the task. On the other hand, the absence of a task while being observed by researchers incentivised participants to explore the interface to a degree that is unlikely to reflect real life behaviour.
Given this difficulty within lab settings, previous research had employed diary studies to examine the triggers of discovery [156,157]. However, this similarly generates multiple issues. First, diary studies only cover *successful* discovery since participants are unable to identify that they failed to discovery something. They therefore do not reflect instances in which participants encounter interactivity but do not perceive it, or perceive interactivity but do not comprehend it. Additionally, even successful discovery may not be captured fully. As Murphy et al. discuss for their diary study, participants may fail to accurately classify their experience as well as the cause of discovery [157]. For example, in the context of discoverability, a user may observe someone else interacting through an unfamiliar input and, at a later point, replicate this input. Furthermore, diary studies require highly motivated and perceptive participants that 1) are willing to continuously interrupt their normal behaviour to log instances of discovery and 2) are able to identify each discovery as it happens and reflect on the process preceding it. This is a problem we encountered in the process of this thesis as a pilot diary study we ran to identify the circumstances of interaction discoverability with different systems (macOS, Android). While participants were highly motivated and logging discoveries for the first few days of using a new system, as they became more familiar with the system and discovery became less frequent, they abandoned logging their discoveries since the study was less present in their mind.
In summary, all measures employed to investigate discoverability have shortcomings. Given that "Trueman show"-like setups that allow true observation of organic discovery are not feasible at an academic level, the benefits and drawbacks of existing methods need to be considered and reflected upon on a case-by-case basis.

6.4 The Potential of Visual Signifiers in the Interface

The results of the studies in chapter 4 indicate that visual signifiers can successfully communicate the availability of various inputs and in doing so decrease the mental effort required to identify appropriate actions. This suggests a largely untapped potential to improve interaction discoverability in the interface, easing understanding and supporting efficient interaction for non-expert users. By providing information in the interface, visual signifiers can support interaction discoverability spatially and temporally concurrent to the interaction opportunity.

In doing so, they can aid discovery of interactions users otherwise would have been entirely unaware of. Additionally, if users *do* know something is possible but are not certain about where or how, visual signifiers provide the information when it is needed, therefore removing a barrier that would require users to be motivated enough to independently seek out information externally or investigate functionality through trial and error. Finally, in contrast to approaches like onboarding [104], signifiers, as an inherent part of the interface, are able to support discovery continuously while temporary one-time hints may be dismissed due to inopportune timing and subsequently lost.

6.4.1 The Clutter Argument

The main argument against increased visual signifiers in the interface appears to be clutter. Not only is increased visualisation not in line with design aesthetics and guidelines, but some research suggests that too much visual information may prove detrimental as signifiers fight for the users' attention [45]. However, this critique largely pertains to the amount of visual guidance rather than suggesting its complete absence. Furthermore, there is research, that suggests increasing visualisation improved user understanding and performance [193, 194]. This is also reflected in our findings in chapter 4 in which added visual signifiers at no point *decreased* the performance of participants and in contrast, showed significant potential of improving interaction discoverability. Moreover, participants throughout the studies presented in this thesis did not consider the added visual signifiers to be distracting. In chapter 4 participants largely preferred the designs featuring added visual signifiers over a baseline that reflects current designs standards. Similarly, even though the added signifiers did not improve discoverability, participants in section 5.5 did not consider the shown animation detrimental to their performance.

This suggests that there needs to be a more careful negotiation between "clean" interfaces and the presence of visual signifiers since the current standard compromises discoverability for a default that, given our study results, neither appears to reflect the users' preference nor optimise their performance.

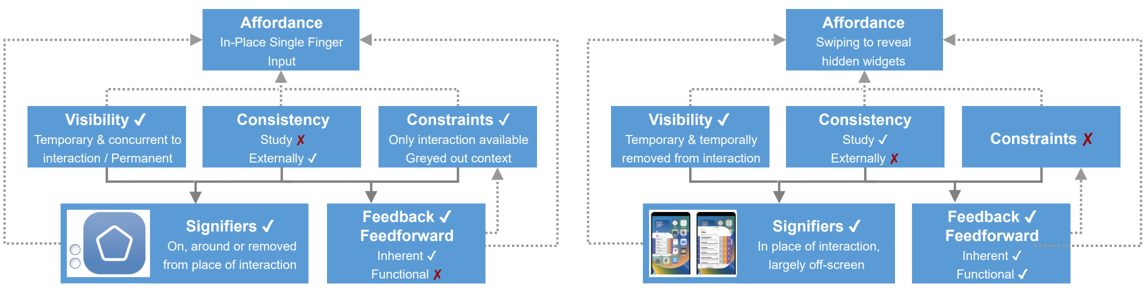
While it may not be feasible or desirable to include visual signifiers universally and permanently, adaptive or customisable user interfaces could provide signifiers if and where they are needed. Since our results suggest that users may prefer added visual signifiers over the current minimalist standard, user interface design and implementation should consider providing alternatives through customisation options, which are available for other interface components (i.e. gesture vs 3-button navigation on Android). This would return some agency to users by leaving the choice to them rather than presuming their preferences and needs while still maintaining the option of supposedly more aesthetic yet less accessible op-

tion.
Further, adaptive interfaces could provide signifiers depending on user behaviour. This could take the approach of progressively removing signifiers as interactions are discovered, performed and retained, similarly to the mark mode in marking menus [116]. Conversely, adaptive systems could feature visual signifiers progressively more prominently as interaction opportunities go unutilised. This may be achieved through any of the dimensions considered in the design space introduced in section 4.4, e.g. through extended durations, increased proximity or a changed level of explicitness. Notably, these approaches conversely mirror the interaction design pattern of progressive disclosure [161] in which options are limited during initial interactions and added options are progressively introduced as users become more familiar with the interface. In contrast, we suggest an approach of *progressive concealment* as well as progressive disclosure applied to signifiers rather than functionality which allows users to operate the entirety of available interactions and functionalities from the beginning. It should be noted that progressive signifier concealment diverges from the progressive disclosure pattern by taking an inverse approach of providing *more* information in the beginning rather than less.

6.4.2 Examining Factors Influencing the Success of Visual Signifiers

Nonetheless, our work also highlights that increased visibility does not universally increase the discoverability of interaction and that the noticeability of signifiers does not necessarily result in increased discoverability. To reflect on this discrepancy of results, we refer back to both the timeline of initial interactions presented in chapter 2 as well as the framework in chapter 3.

As highlighted in subsection 6.3.1, the focal point of our studies within the sequence of actions in initial interactions varies. However, both studies in chapter 4 and the second study in chapter 5 consider moving from perception to comprehension, which the signifiers in the latter largely failed to support.
Examining the signifiers from both studies and our results in the context of our framework of impacting factors provides both an opportunity for reflection and a guideline of possible causes.



(a) *Signifiers chapter 4* ***(b)*** *Signifiers chapter 5*

Figure 6.3 *– Examination of the visual signifier implementation of the previous chapters in the context of the framework of impacting factors with a focused on the system*

6.4.2.1 On the system side.

We first reflect on the system as presented in subsection 3.1.1, which is visualised in the context of our approaches in Figure 6.3. In both approaches we increase *signifier visibility* in an aim to improve their communication of *affordances*. While the signifiers in chapter 4 vary in how long they are visible, even the temporary signifiers considered in this context were largely visible for longer periods of time than those in chapter 5, suggesting one possible reason being the extended visibility of signifiers. However, the informal study with 5000 ms long transitions performed in examination of this as a part of the discussion provided in subsection 5.6.2 suggest this to be an unlikely cause for our differing results. Arguably, the setup in chapter 5 provided more *consistency* through the permanent availability of swhidgets, in comparison to chapter 4 in which the available input varied between individual designs. However, the opposite is the case when considering the widgets at the basis of both approaches outside of the study context. The homescreen buttons imitated in chapter 4 consistently allow input beyond "tap", with iOS permitting both "dwell" and "force", while Android devices support "dwell". In contrast, the availability of swhidgets varies between not only operating systems and applications but also within applications. This suggests that potentially, consistency of visual interaction cues is not sufficient to communicate *in*consistency of the inputs they signify. Further, *constraints* vary significantly between the studies. While in chapter 4 interaction on screen was constrained to the widget in question, which was highlighted through an otherwise grey interface, the signified interaction in chapter 5 was intentionally created to be part of a larger task interface in an effort to truly capture discoverability. Constraining the interaction in subsection 4.5.2 is likely to have significantly increased user comprehension by providing an immediate focal point. This as a cause for varying results is further supported by participants in section 5.5 stating their focus on the task and description contributed to their disregard of the visual signifiers included in the transitions. Finally, both permanent signifiers in chapter 4 and the embedded signifiers in chapter 5 provided *inherent feedforward*. The animation in permanent signifiers and all temporary signifiers in chapter 4 as well as the task completing acted as feedback, which is inherently provided in swhidgets which are displaced with the finger movement.

6.4.2.2 On the user side.

Following this examination of system factors, we consider the user, as presented in subsection 3.1.2 before drawing conclusions based on the combination of both actors. While we directly examine the factors within the system for our use cases as presented in Figure 6.3, we refrain from similar visualisations for the user. Although we speculate about internal factors such as motivation, these may vary widely between users, making a generalised visualisation invalid.

The self-reported *familiarity* of participants with both input methods and widgets did not significantly impact our measures, suggesting that sufficient visual signifiers may negate the impact of familiarity. However, the individual *expectations* described by participants in chapter 4 of interaction availability were largely attributed to familiarity with other interaction contexts. This suggests that the *perception of affordances* depending on familiarity may be mediated by visual signifiers, but the expectations created through said signifier are still informed by familiarity. The *goal and objectives* in the studies vary through their difference in constraints. While the goal in subsection 4.5.2 was solely to interact with a button through

the correct input, in section 5.5 participants' goal was to complete the set task to the best of their ability. This goal consequently moved their attention away from the transitions providing interaction hints, as their motivation to complete the tasks quickly and efficiently led them to concentrate on the instructions as soon as possible. This led to a lack of attention on the transitions, the user interface element providing visual interaction signifiers and a lack of *awareness* of both the signifier and the interactions they communicate.
Finally, we do not further reflect on the environment, as all experiments were performed within a lab setting with individual participants who did not encounter one another and discoverability through the physical or social environment was not the focus of the work in this thesis.

In summary, reflection based on the framework of impacting factors suggests that various factors may contribute to the discrepancy in the success of visual signifiers between our two approaches. While the signifiers in chapter 5 were noticeable, it appears they did not demand the attention required from users to perceive them as relevant to the interaction. One potential approach would be increased or extended visibility grab the users attention. Additionally, the lack of attention appears to be exacerbated if the interaction being signified is inconsistently implemented and interaction is not constraint. In the context of our approach in section 5.5 for example, constraining the availability and visibility of other screen elements during the transition may bring increased attention to the transient interaction hints embedded in the animated transition.
In addition, tasks that motivate users to work towards the completion of specific goals like the study in section 5.5 appear to reduce awareness of the overall interface in favour of attention to the task. This suggests these contexts may require stronger signifiers to facilitate the perception of available interactions and features. This highlights, that discoverability of interactions in task focused contexts may be heavily impacted by the paradox of the active user [34] because users are too focused on completing their task to process interaction signifiers or explore an interface independently.

6.4.3 Limitations

Our work highlights that visual signifiers do have the potential to increase the discoverability of interaction, however it also shows that this is not universally the case and that the noticeability of signifiers does not necessarily result in increased discoverability.

The results presented in chapter 5 show that interaction signifiers embedded into animated transitions are not successful in conveying interactivity, even to users that explicitly note their presence. We examine potential causes through reflection of the implementation in the context of a framework of impacting factors in subsection 6.4.2. Outside of these factors, the results of section 5.5 further suggest that not all user interface elements are suitable to signify interaction to users. This lays the foundation for future research examining the perception of individual UI elements and the impact it may have on their ability to communicate information to the user.

Finally, it should be noted that while we explore different types of signifiers as well as various individual designs in the context of this thesis, the widgets, UI elements and input methods represented are limited. Subsequently it is yet to be confirmed whether the impact (or lack thereof) of different visual signifiers may vary depending on these variables.

6.5 Directions for Future Work

Given the findings presented within this thesis and the relative lack of research focused on interaction discoverability, there are multiple paths that warrant further exploration.

6.5.1 Studying Discoverability

As we reflect upon in section 6.3, studying discoverability inherently presents various hurdles. However, these difficulties have not been generally examined. Given the challenges presented to researchers when studying discoverability, a comprehensive review of existing methodology as well as their categorisation in relation to the steps of initial interaction as presented in subsection 6.3.1 should be performed.

Additionally, there is a distinct lack of knowledge of how discovery actually occurs outside a laboratory setting. Murphy-Hill et al. [157] attempt quantification of discovery sources through a diary study, but due to their focus on social discovery do not holistically examine discoverability. An ethnomethodological examination of users engaging with technology that is new to them may yield important insights on how users discover both interactions and features.

6.5.2 Signifiers and the Perception of UI Elements

The results of chapter 5 suggest transient interaction hints in animated transitions do not work as interaction signifiers, which provides interesting questions for future work. This includes exploring both the perception of users as well as the characteristics and limits of temporary visualisations that may aid discoverability. Furthermore, noticeability did not appear to translate to discoverability, raising the question of what may be necessary to do so.

Mental Models

Since our results suggest that some participants do not consider transitions as an element potentially providing relevant information about interaction, future work should investigate how participants conceptualise animated transitions. While previous work [112] and current design recommendations [130] suggest that transitions can effectively be used to communicate a coherent spatial model of applications to users, our results in chapter 5 suggest this can not be expanded to include information about available interactions. It appears the users mental model of transitions may lead them to exclude transitions from any further considerations if they are perceived to be purely spatial or decorative. This additionally raises the question of whether decorative features in interfaces are generally assessed as irrelevant and subsequently ignored and how much they may need to be altered to elude this categorisation.

Furthermore future work should examine how animated transitions impact the users mental model of the interface, since even conscious perception of the swhidgets displayed in the transition did not translate to interaction and discovery of swhidgets. This is especially interesting since this contrasts previous work, which suggests that animated transitions successfully improved the mental model of the app structure [112]. Therefore there appears to be a limitation to when animated transitions support accurate mental models that may depend on context which is worth exploring. As they are widely employed to support navigational understanding, can they only be used to support top level structural mental models?

Temporary Visualisation to Improve Discoverability

Given that the animated transitions we implemented in chapter 5 did not significantly improve discoverability, the potential of alternative transitions as well as general temporary visual signifiers should be examined.
First, transitions with extremely long durations could be explored further. This may prove particularly interesting in combination with considerations of the mental model of transitions, as extremely slow transitions may lead to a change in the perception of transitions and morph into something akin to a loading screen, which are more established to include interaction hints. However, it should be noted that, when tested informally, 5 seconds did not appear to be long enough for established transition styles to be perceived differently, which additionally raises the question of what, beyond noticeability, is required to successfully improve discoverability. While not feasible permanently, if overly long transitions prove to be beneficial they could be implemented the first time an application is opened as an alternative or complimentary to onboarding [104].
Alternatively, it could be explored whether highlighting the swhidgets in the transitions more clearly may improve their discoverability. This may be achieved by displaying only them and introducing the original app content at the end and therefore constraining the interaction, or through added animation like blinking, drawing the attention of users to that specific part of the screen.
Finally, if transitions prove to fall into a category of UI components that are inherently disconnected from interaction in people's minds, other temporary visualisations may prove to be more beneficial. Since our results in chapter 4 and other previous work [14] are inconclusive about the importance of temporal placement for interaction signifiers concerning single finger, in-place inputs, future work could investigate animation concurrent to the need for an action embedded in a swhidget and whether this improves discoverability. Additionally, animation visualising the existence of the swhidget if the action (or a related action) is performed in an alternative way could signify the availability of swipe interactions.
Furthermore, as touched on in subsection 6.4.1, adaptive interfaces could take an approach of progressive concealment of signifiers as users discover inputs in new interfaces, or progressive disclosure of signifiers if users display inefficient behaviour patterns or fail to discover inputs altogether.

6.5.3 Noticeability and Discoverability

While noticeability has been widely differentiated from *comprehension* [219,233], it is interesting that it also appears to fail to create awareness or arouse curiosity. Although participants noticed the swhidget signifiers in the transitions in chapter 5, this did not translate to awareness of the swhidget. Future research should investigate where this disconnect originates. First, it may be attributed to a lack of mental connection between the transition and the widget. Potentially it is necessary for perception to turn into comprehension that the signifier and interaction are more explicitly linked. Although the embedded signifier in the animated transition are spatially placed where the interaction happens, their temporal removal from the interactions may be a contributing factor worth investigating. Additionally, the experimental structure may contribute to a lack of curiosity or motivation to explore the meaning of noticed visualisations. Further, the lack of exploration may partially be due to the participants task focus as well as their awareness of being examined in a user study context and subsequent wish to perform well. This suggests that future work should examine whether designs created to support discoverability need to stimulate curiosity to a certain degree to be successful in task focused contexts. Additionally it further supports our assertion that methodology to examine discoverability needs to be investigated and optimised.

6.5.4 Visual Signifiers Providing Feedforward

While our results in chapter 4 suggest that signifiers can successfully increase the perception of *inputs* and this additional guidance reduces mental effort, these studies did not consider the outcome of these *inputs* when utilised. In real world applications it may be pertinent to not only signify that an interaction is possible, but also what may be its outcome. The combination of meaningful feedforward and interaction signifiers presents a unique challenge of communicating both the intended outcome and the means to achieve it. Future work should investigate how to signify the outcome of an action without taking precedence over signifying the availability of the input method itself.

6.6 Conclusion

Considering the frequency at which users have to discover novel interactive systems or input methods, the fact that discoverability receives such little attention is troubling. Potentially, the problem is deemed as not significant enough, or interaction designers still believe that users will *"eventually figure it out"*. However, research and commercial use suggest that the assumption of discovery being inevitable is unsubstantiated [15,76,177]. We suggest the HCI research community should avoid the continued postponing or marginalizing of discoverability problems and rather, allocate it the attention and significance it necessitates.
In this thesis we offer the groundwork for future considerations by clarifying discoverability and associated initial interaction concepts and providing a framework of impacting factors. We then investigate the potential of visual signifiers in the interface to improve the discovery of interactions. Due to their omnipresence, we focus in particular on mobile touch screens and touch-based interactions. First, we propose a signifier design space examining the characteristics timing, proximity and explicitness. Through this, we create signifier designs for in-place single finger inputs which we then examine through an online survey and interactive study, with the results suggesting visual signifiers to be not only successful in conveying interactivity but also reducing mental effort. Subsequently, we investigate the capacity of signifiers embedded in animated transitions to convey the availability of swipe-revealed hidden widgets. In contrast to our first implementation of visual signifiers, the results of two user studies examining them suggest that, while they are noticeable, the added signifiers do not succeed in increasing discoverability.
Finally, we reflect on the importance of an increased focus on discoverability, our diverging results, the limitations of visual signifiers and the methodology employed throughout our studies. Through this, we emphasise the core thesis that interaction discoverability can and *should* be supported in interactive system and highlight the potential of visual signifiers, which we find to be a multifaceted but underutilised resource meriting further research.

Associated Publications

Casiez, Géry, Sylvain Malacria, and Eva Mackamul. "Signifidgets: What you see is what widget!" In IHM 2023-34e Conférence Internationale Francophone sur l'Interaction Humain-Machine, 2023.

Mackamul, Eva. "Improving the Discoverability of Interactions in Interactive Systems." In CHI Conference on Human Factors in Computing Systems Extended Abstracts, pp. 1-5. 2022. https://doi.org/10.1145/3491101.3503813.

Mackamul, Eva, Géry Casiez, and Sylvain Malacria. "Exploring visual signifier characteristics to improve the perception of affordances of in-place touch inputs." Proceedings of the ACM on Human-Computer Interaction 7, no. MHCI (2023): 1-32. https://doi.org/10.1145/3604257.

Bibliography

[1] Marina A. Kosycheva and Elena Victorovna Tikhonova. Students' Self-Efficacy and Motivation in Emergency Remote Learning. In *2021 12th International Conference on E-Education, E-Business, E-Management, and E-Learning*, IC4E 2021, page 157–162, New York, NY, USA, 2021. Association for Computing Machinery.

[2] Anand Agarawala and Ravin Balakrishnan. Keepin' It Real: Pushing the Desktop Metaphor with Physics, Piles and the Pen. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, page 1283–1292, New York, NY, USA, 2006. Association for Computing Machinery.

[3] David Ahlström, Andy Cockburn, Carl Gutwin, and Pourang Irani. Why It's Quick to Be Square: Modelling New and Existing Hierarchical Menu Designs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, page 1371–1380, New York, NY, USA, 2010. Association for Computing Machinery.

[4] Mendoza G. Alfredo, J. Alvarez R. Francisco, Muñoz A. Jaime, Cristian Rusu, Acosta E. Francisco, and Mendoza G. Ricardo. A Cooperative Process for a Learnability Study with Down Syndrome Children. In *Proceedings of the XVII International Conference on Human Computer Interaction*, Interacción '16, New York, NY, USA, 2016. Association for Computing Machinery.

[5] Mendoza G. Alfredo, J. Alvarez R. Francisco, Mendoza G. Ricardo, Acosta E. Francisco, and Muñoz A. Jaime. Analyzing Learnability of Common Mobile Gestures Used by Down Syndrome Users. In *Proceedings of the XVI International Conference on Human Computer Interaction*, Interacción '15, New York, NY, USA, 2015. Association for Computing Machinery.

[6] Bruce G Allen and Elizabeth Buie. What's in a word? The semantics of usability. *Interactions*, 9(2):17–21, 2002.

[7] Sherman R Alpert. Self-describing animated icons for human-computer interaction: a research note. *Behaviour & Information Technology*, 10(2):149–152, 1991.

[8] Florian Alt, Stefan Schneegaß, Albrecht Schmidt, Jörg Müller, and Nemanja Memarovic. How to Evaluate Public Displays. In *Proceedings of the 2012 International Symposium on Pervasive Displays*, PerDis '12, New York, NY, USA, 2012. Association for Computing Machinery.

[9] Rui Alves and Nuno Jardim Nunes. Ceiling and threshold of paas tools: the role of learnability in tool adoption. In *Human-Centered and Error-Resilient Systems Development*, pages 335–347. Springer, 2016.

[10] Jessalyn Alvina, Andrea Bunt, Parmit K. Chilana, Sylvain Malacria, and Joanna McGrenere. Where is That Feature? Designing for Cross-Device Software Learnability. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference*, DIS '20, page 1103–1115, New York, NY, USA, 2020. Association for Computing Machinery.

[11] Axel Antoine, Sylvain Malacria, and Géry Casiez. ForceEdge: Controlling Autoscroll on Both Desktop and Mobile Computers Using the Force. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, page 3281–3292, New York, NY, USA, 2017. Association for Computing Machinery.

[12] Caroline Appert, Olivier Chapuis, and Emmanuel Pietriga. Dwell-and-Spring: Undo for Direct Manipulation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, page 1957–1966, New York, NY, USA, 2012. Association for Computing Machinery.

[13] Caroline Appert, Olivier Chapuis, Emmanuel Pietriga, and María-Jesús Lobo. Reciprocal Drag-and-Drop. *ACM Trans. Comput.-Hum. Interact.*, 22(6), sep 2015.

[14] Liv Arleth, Emilie Lind Damkjær, and Hendrik Knoche. "But Wait, There's More!" a Deeper Look into Temporally Placing Touch Gesture Signifiers. In *Interactivity, Game Creation, Design, Learning, and Innovation: 8th EAI International Conference, ArtsIT 2019, and 4th EAI International Conference, DLI 2019, Aalborg, Denmark, November 6–8, 2019, Proceedings 8*, pages 290–308. Springer, 2020.

[15] Jeff Avery and Edward Lank. Surveying Expert-Level Gesture Use and Adoption on Multi-Touch Tablets. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*, DIS '16, pages 577–581, New York, NY, USA, 2016. ACM.

[16] Jeff Avery, Sylvain Malacria, Mathieu Nancel, Géry Casiez, and Edward Lank. Introducing Transient Gestures to Improve Pan and Zoom on Touch Surfaces. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 1–8, New York, NY, USA, 2018. Association for Computing Machinery.

[17] Nick Babich. Tips to Improve Discoverability in UX: Adobe XD Ideas. Retrieved January 27th, 2022 from https://xd.adobe.com/ideas/process/information-architecture/tips-to-improve-discoverability-in-ux/, Apr 2020.

[18] Klaus B. Bærentsen and Johan Trettvik. An Activity Theory Approach to Affordance. In *Proceedings of the Second Nordic Conference on Human-Computer Interaction*, NordiCHI '02, page 51–60, New York, NY, USA, 2002. Association for Computing Machinery.

[19] Shruti Baijal and Narayanan Srinivasan. Types of attention matter for awareness: A study with color afterimages. *Consciousness and cognition*, 18(4):1039–1048, 2009.

[20] David Benyon. *Designing Interactive Systems: A Comprehensive Guide to HCI, UX and Interaction Design*. Pearson, 2013.

[21] Scott Berkun. The myth of discoverability. Retrieved December 9th, 2021 from https://scottberkun.com/essays/26-the-myth-of-discoverability/, Dec 2012.

[22] Alethea L Blackler, Rafael Gomez, Vesna Popovic, and M Helen Thompson. Life is too short to RTFM: how users relate to documentation and excess features in consumer products. *Interacting with Computers*, 28(1):27–46, 2016.

[23] David Bonnet, Caroline Appert, and Michel Beaudouin-Lafon. Extending the Vocabulary of Touch Events with ThumbRock. In *Proceedings of Graphics Interface 2013*, GI '13, Regina (Saskatchewan), Canada, May 2013. CHCCS.

[24] Jelmer P. Borst, Niels A. Taatgen, and Hedderik van Rijn. What Makes Interruptions Disruptive? A Process-Model Account of the Effects of the Problem State Bottleneck on Task Interruption and Resumption. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, page 2971–2980, New York, NY, USA, 2015. Association for Computing Machinery.

[25] Jeremy Boy, Louis Eveillard, Françoise Detienne, and Jean-Daniel Fekete. Suggested interactivity: Seeking perceived affordances for information visualization. *IEEE transactions on visualization and computer graphics*, 22(1):639–648, 2015.

[26] Harry Brignull and Yvonne Rogers. Enticing people to interact with large public displays in public spaces. In *INTERACT*, volume 3, pages 17–24, 2003.

[27] David M Cades, Deborah A Boehm Davis, J Gregory Trafton, and Christopher A Monk. Does the difficulty of an interruption affect our ability to resume? In *Proceedings of the human factors and ergonomics society annual meeting*, volume 51, pages 234–238. SAGE Publications Sage CA: Los Angeles, CA, 2007.

[28] David M Cades, Nicole Werner, Deborah A Boehm-Davis, J Gregory Trafton, and Christopher A Monk. Dealing with interruptions can be complex, but does interruption complexity matter: A mental resources approach to quantifying disruptions. In *Proceedings of the human factors and ergonomics society annual meeting*, volume 52, pages 398–402. Sage Publications Sage CA: Los Angeles, CA, 2008.

[29] Francesco Cafaro, Leilah Lyons, and Alissa N. Antle. Framed Guessability: Improving the Discoverability of Gestures and Body Movements for Full-Body Interaction. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 1–12, New York, NY, USA, 2018. Association for Computing Machinery.

[30] Francesco Cafaro, Milka Trajkova, and A'aeshah Alhakamy. Designing Embodied Interactions for Informal Learning: Two Open Research Challenges. In *Proceedings of the 8th ACM International Symposium on Pervasive Displays*, PerDis '19, New York, NY, USA, 2019. Association for Computing Machinery.

[31] Stuart K Card. *The psychology of human-computer interaction*. Crc Press, 2018.

[32] Stuartk Card, Thomas P. Moran, and Allen Newell. The model human processor- An engineering model of human performance. *Handbook of perception and human performance.*, 2(45–1), 1986.

[33] Jennifer Cardello. Low findability and discoverability: Four testing methods to identify the causes. Retrieved December 9th, 2021 from https://www.nngroup.com/articles/navigation-ia-tests/, 2014 updated 2019.

[34] John M Carroll and Mary Beth Rosson. Paradox of the active user. In *Interfacing thought: Cognitive aspects of human-computer interaction*, pages 80–111. MIT Press, 1987.

[35] Géry Casiez, Sylvain Malacria, and Eva Mackamul. Signifidgets: What you see is what widget! In *IHM 2023-34e Conférence Internationale Francophone sur l'Interaction Humain-Machine*, 2023.

[36] Victor Cheung. Improving Interaction Discoverability in Large Public Interactive Displays. In *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces*, ITS '14, page 467–472, New York, NY, USA, 2014. Association for Computing Machinery.

[37] Victor Cheung. *Increasing passersby engagement with public large interactive surfaces*. PhD thesis, University of Waterloo, 2016.

[38] Victor Cheung and Stacey D. Scott. Investigating Attraction and Engagement of Animation on Large Interactive Walls in Public Settings. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '13, page 381–384, New York, NY, USA, 2013. Association for Computing Machinery.

[39] Victor Cheung and Stacey D. Scott. A Laboratory-Based Study Methodology to Investigate Attraction Power of Large Public Interactive Displays. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '15, page 1239–1250, New York, NY, USA, 2015. Association for Computing Machinery.

[40] Fanny Chevalier, Nathalie Henry Riche, Catherine Plaisant, Amira Chalbi, and Christophe Hurter. Animations 25 years later: New roles and opportunities. In *Proceedings of the international working conference on advanced visual interfaces*, pages 280–287, 2016.

[41] Angela Chow and Nancy Law. Measuring Motivation in Collaborative Inquiry-Based Learning Contexts. In *Proceedings of Th 2005 Conference on Computer Support for Collaborative Learning: Learning 2005: The next 10 Years!*, CSCL '05, page 68–75. International Society of the Learning Sciences, 2005.

[42] Josh Clark. *Designing for touch*. A Book Apart New York, 2015.

[43] Andy Cockburn, Carl Gutwin, Joey Scarr, and Sylvain Malacria. Supporting Novice to Expert Transitions in User Interfaces. *ACM Comput. Surv.*, 47(2), nov 2014.

[44] Alan Cooper, Robert Reimann, David Cronin, and Christopher Noessel. *About Face: The Essentials of Interaction Design*. Wiley, 4th edition edition, Sep 2014.

[45] Eric Corbett and Astrid Weber. What Can I Say? Addressing User Experience Challenges of a Mobile Voice User Interface for Accessibility. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '16, page 72–82, New York, NY, USA, 2016. Association for Computing Machinery.

[46] Travis Cox, Marcus Carter, and Eduardo Velloso. Public DisPLAY: Social Games on Interactive Public Screens. In *Proceedings of the 28th Australian Conference on Computer-Human Interaction*, OzCHI '16, page 371–380, New York, NY, USA, 2016. Association for Computing Machinery.

[47] Lucia D'Acunto, Jorrit van den Berg, Emmanuel Thomas, and Omar Niamut. Using MPEG DASH SRD for Zoomable and Navigable Video. In *Proceedings of the 7th International Conference on Multimedia Systems*, MMSys '16, New York, NY, USA, 2016. Association for Computing Machinery.

[48] Emilie Lind Damkjær, Liv Arleth, and Hendrik Knoche. "I Didn't Know, You Could Do That"-Affordance Signifiers for Touch Gestures on Mobile Devices. In *Interactivity, Game Creation, Design, Learning, and Innovation: 7th EAI International Conference, ArtsIT 2018, and 3rd EAI International Conference, DLI 2018, ICTCC 2018, Braga, Portugal, October 24–26, 2018, Proceedings 7*, pages 206–212. Springer, 2019.

[49] Lavinia Andreea Danielescu. *Discoverable Free Space Gesture Sets for Walk-Up-and-Use Interactions*. PhD thesis, Arizona State University, 2019.

[50] Thomas Davies and Ashweeni Beeharee. The Case of the Missed Icon: Change Blindness on Mobile Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, page 1451–1460, New York, NY, USA, 2012. Association for Computing Machinery.

[51] Clarisse Sieckenius De Souza, Bonnie A Nardi, Victor Kaptelinin, and Kirsten A Foot. *The semiotic engineering of human-computer interaction*. MIT press, 2005.

[52] Mário Rosado de Souza, Robert Haines, Markel Vigo, and Caroline Jay. What Makes Research Software Sustainable? An Interview Study with Research Software Engineers. In *Proceedings of the 12th International Workshop on Cooperative and Human Aspects of Software Engineering*, CHASE '19, page 135–138. IEEE Press, 2019.

[53] Lambert Deckers. *Motivation: Biological, psychological, and environmental*. Routledge, 2018.

[54] Stanislas Dehaene, Jean-Pierre Changeux, Lionel Naccache, Jérôme Sackur, and Claire Sergent. Conscious, preconscious, and subliminal processing: a testable taxonomy. *Trends in cognitive sciences*, 10(5):204–211, 2006.

[55] William Delamare, Thomas Janssoone, Céline Coutrix, and Laurence Nigay. Designing 3D Gesture Guidance: Visual Feedback and Feedforward Design Options. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '16, page 152–159, New York, NY, USA, 2016. Association for Computing Machinery.

[56] Apple Developer Design. Apple Developer - Design - Human Interface Guidelines - Going Full Screen, 2023.

[57] Apple Developer Design. Apple Developer - Design - Human Interface Guidelines - Launching, 2023.

[58] Sebastian Deterding. Gamification: Designing for Motivation. *Interactions*, 19(4):14–17, jul 2012.

[59] Anna Dickinson, Roos Eisma, and Peter Gregor. The barriers that older novices encounter to computer use. *Universal Access in the Information Society*, 10(3):261–266, 2011.

[60] Merriam Webster Dictionary. Awareness, 2021. Retrieved January 27th, 2022 from `https://www.merriam-webster.com/dictionary/awareness`.

[61] Alan John Dix, Janet Finlay, Gregory D Abowd, and Russell Beale. *Human-computer interaction*. Pearson Education, 2003.

[62] Tom Djajadiningrat, Kees Overbeeke, and Stephan Wensveen. But How, Donald, Tell Us How? On the Creation of Meaning in Interaction Design through Feedforward and Inherent Feedback. In *Proceedings of the 4th Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*, DIS '02, page 285–291, New York, NY, USA, 2002. Association for Computing Machinery.

[63] Tao Dong, Mira Dontcheva, Diana Joseph, Karrie Karahalios, Mark Newman, and Mark Ackerman. Discovery-Based Games for Learning Software. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, page 2083–2086, New York, NY, USA, 2012. Association for Computing Machinery.

[64] Pierre Dragicevic, Anastasia Bezerianos, Waqas Javed, Niklas Elmqvist, and Jean-Daniel Fekete. Temporal Distortion for Animated Transitions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, page 2009–2018, New York, NY, USA, 2011. Association for Computing Machinery.

[65] Sebastian Draxler, Gunnar Stevens, Martin Stein, Alexander Boden, and David Randall. Supporting the Social Context of Technology Appropriation: On a Synthesis of Sharing Tools and Tool Knowledge. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, page 2835–2844, New York, NY, USA, 2012. Association for Computing Machinery.

[66] Ekwa Duala-Ekoko and Martin P. Robillard. Using Structure-Based Recommendations to Facilitate Discoverability in APIs. In *Proceedings of the 25th European Conference on Object-Oriented Programming*, ECOOP'11, page 79–104, Berlin, Heidelberg, 2011. Springer-Verlag.

[67] Mica R Endsley. Design and evaluation for situation awareness enhancement. In *Proceedings of the Human Factors Society annual meeting*, volume 32, pages 97–101. Sage Publications Sage CA: Los Angeles, CA, 1988.

[68] Elias Fares, Victor Cheung, and Audrey Girouard. Effects of Bend Gesture Training on Learnability and Memorability in a Mobile Game. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*, ISS '17, page 240–245, New York, NY, USA, 2017. Association for Computing Machinery.

[69] Katherine Fennedy, Angad Srivastava, Sylvain Malacria, and Simon T. Perrault. Towards a Unified and Efficient Command Selection Mechanism for Touch-Based Devices Using Soft Keyboard Hotkeys. *ACM Trans. Comput.-Hum. Interact.*, 29(1), jan 2022.

[70] ISO International Organization for Standardization. Ergonomics of human-system interaction — Part 110: Interaction principles, 2020.

[71] Anushay Furqan, Chelsea Myers, and Jichen Zhu. Learnability through Adaptive Discovery Tools in Voice User Interfaces. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '17, page 1617–1623, New York, NY, USA, 2017. Association for Computing Machinery.

[72] Michelle Gantt and Bonnie A. Nardi. Gardeners and Gurus: Patterns of Cooperation among CAD Users. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '92, page 107–117, New York, NY, USA, 1992. Association for Computing Machinery.

[73] James J Gibson. *The ecological approach to visual perception*. Houghton Mifflin, Boston, 1979.

[74] Tony Gillie and Donald Broadbent. What makes interruptions disruptive? A study of length, similarity, and complexity. *Psychological research*, 50(4):243–250, 1989.

[75] Marco Gillies. What is Movement Interaction in Virtual Reality For? In *Proceedings of the 3rd International Symposium on Movement and Computing*, MOCO '16, New York, NY, USA, 2016. Association for Computing Machinery.

[76] Vindu Goel. Apple Piles On the Features and Users Say, 'Enough!', May 2017. Retrieved December 9th, 2021 from `https://www.nytimes.com/2017/06/02/technology/apple-iphone-developer-conference.html`.

[77] Alix Goguey, Géry Casiez, Daniel Vogel, Fanny Chevalier, Thomas Pietrzak, and Nicolas Roussel. A Three-Step Interaction Pattern for Improving Discoverability in Finger Identification Techniques. In *Proceedings of the Adjunct Publication of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST'14 Adjunct, page 33–34, New York, NY, USA, 2014. Association for Computing Machinery.

[78] Alix Goguey, Sylvain Malacria, and Carl Gutwin. Improving Discoverability and Expert Performance in Force-Sensitive Text Selection for Touch Devices with Mode Gauges. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 1–12, New York, NY, USA, 2018. Association for Computing Machinery.

[79] Alix Goguey, Daniel Vogel, Fanny Chevalier, Thomas Pietrzak, Nicolas Roussel, and Géry Casiez. Leveraging finger identification to integrate multi-touch command selection and parameter manipulation. *IJHCS*, 99:21–36, 2017.

[80] Jun Gong, Aakar Gupta, and Hrvoje Benko. Acustico: Surface Tap Detection and Localization Using Wrist-Based Acoustic TDOA Sensing. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, UIST '20, page 406–419, New York, NY, USA, 2020. Association for Computing Machinery.

[81] Tovi Grossman and George Fitzmaurice. ToolClips: An Investigation of Contextual Video Assistance for Functionality Understanding. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, page 1515–1524, New York, NY, USA, 2010. Association for Computing Machinery.

[82] Tovi Grossman, George Fitzmaurice, and Ramtin Attar. A Survey of Software Learnability: Metrics, Methodologies and Guidelines. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, page 649–658, New York, NY, USA, 2009. Association for Computing Machinery.

[83] The Standish Group. Modernization: Clearing a Pathway to Success, 2010.

[84] Frank Halasz and Thomas P. Moran. Analogy Considered Harmful. In *Proceedings of the 1982 Conference on Human Factors in Computing Systems*, CHI '82, page 383–386, New York, NY, USA, 1982. Association for Computing Machinery.

[85] Chris Harrison, Julia Schwarz, and Scott E. Hudson. TapSense: Enhancing Finger Interaction on Touch Surfaces. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, page 627–636, New York, NY, USA, 2011. Association for Computing Machinery.

[86] Björn Hartmann, Meredith Ringel Morris, Hrvoje Benko, and Andrew D. Wilson. Pictionaire: Supporting Collaborative Design Work by Integrating Physical and Digital Artifacts. In *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work*, CSCW '10, page 421–424, New York, NY, USA, 2010. Association for Computing Machinery.

[87] Rex Hartson. Cognitive, physical, sensory, and functional affordances in interaction design. *Behaviour & information technology*, 22(5):315–338, 2003.

[88] Denis Helic, Sebastian Wilhelm, Ilire Hasani-Mavriqi, and Markus Strohmaier. The Effects of Navigation Tools on the Navigability of Web-Based Information Systems. In *Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies*, i-KNOW '11, New York, NY, USA, 2011. Association for Computing Machinery.

[89] Seongkook Heo, Jiseong Gu, and Geehyuk Lee. Expanding Touch Input Vocabulary by Using Consecutive Distant Taps. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, page 2597–2606, New York, NY, USA, 2014. Association for Computing Machinery.

[90] Seongkook Heo and Geehyuk Lee. Forcetap: Extending the Input Vocabulary of Mobile Touch Screens by Adding Tap Gestures. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '11, page 113–122, New York, NY, USA, 2011. Association for Computing Machinery.

[91] Luke Hespanhol, Martin Tomitsch, Ian McArthur, Joel Fredericks, Ronald Schroeter, and Marcus Foth. Vote as You Go: Blending Interfaces for Community Engagement into the Urban Space. In *Proceedings of the 7th International Conference on Communities and Technologies*, page 29–37, New York, NY, USA, 2015. Association for Computing Machinery.

[92] Cecilia Heyes. What's social about social learning? *Journal of Comparative Psychology*, 126(2):193, 2012.

[93] Jamie Holst. Users Continue to Double-Click Online, Jul 2013.

[94] Eva Hornecker. “I don’t understand it either, but it is cool” - visitor interactions with a multi-touch table in a museum. In *2008 3rd IEEE International Workshop on Horizontal Interactive Human Computer Systems*, pages 113–120, 2008.

[95] Simo Hosio, Jorge Goncalves, and Vassilis Kostakos. Application Discoverability on Multipurpose Public Displays: Popularity Comes at a Price. In *Proceedings of the 2nd ACM International Symposium on Pervasive Displays*, PerDis ’13, page 31–36, New York, NY, USA, 2013. Association for Computing Machinery.

[96] Shih-Wen Hsiao, Chiao-Fei Hsu, and Yin-Ting Lee. An online affordance evaluation model for product design. *Design Studies*, 33(2):126–159, 2012.

[97] Jussi Huhtala, Ari-Heikki Sarjanoja, Jani Mäntyjärvi, Minna Isomursu, and Jonna Häkkilä. Animated UI Transitions and Perception of Time: A User Study on Animated Effects on a Mobile Screen. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’10, page 1339–1342, New York, NY, USA, 2010. Association for Computing Machinery.

[98] Samia Ibtasam, Hamid Mehmood, Lubna Razaq, Jennifer Webster, Sarah Yu, and Richard Anderson. An Exploration of Smartphone Based Mobile Money Applications in Pakistan. In *Proceedings of the Ninth International Conference on Information and Communication Technologies and Development*, ICTD ’17, New York, NY, USA, 2017. Association for Computing Machinery.

[99] Apple Inc. Buttons, 2023.

[100] Material IO. Material Design Buttons, 2023.

[101] Giulio Jacucci, Ann Morrison, Gabriela T. Richard, Jari Kleimola, Peter Peltonen, Lorenza Parisi, and Toni Laitinen. Worlds of Information: Designing for Engagement at a Public Multi-Touch Display. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, page 2267–2276, New York, NY, USA, 2010. Association for Computing Machinery.

[102] Steve Jobs. Macworld San Francisco 2007 - Steve Jobs introduced the first iPhone, 2007.

[103] Patrick W Jordan, Stephen W Draper, Kirsteen K MacFarlane, and Shirley-Anne McNulty. Guessability, learnability, and experienced user performance. In *People and Computers VI - Proceedings of the HCI ’91 Conference*. Cambridge University Press, 1991.

[104] Alita Joyce. Mobile-App Onboarding: An Analysis of Components and Techniques. Retrieved January 27th, 2022 from `https://www.nngroup.com/articles/mobile-app-onboarding/`, June 2020.

[105] Hyo Jeong Kang, Jung-hye Shin, and Kevin Ponto. A Comparative Analysis of 3D User Interaction: How to Move Virtual Objects in Mixed Reality. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 275–284, 2020.

[106] Victor Kaptelinin and Bonnie Nardi. Affordances in HCI: Toward a Mediated Action Perspective. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, page 967–976, New York, NY, USA, 2012. Association for Computing Machinery.

[107] Kimia Kiani, Parmit K. Chilana, Andrea Bunt, Tovi Grossman, and George Fitzmaurice. “I Would Just Ask Someone”: Learning Feature-Rich Design Software in the Modern Workplace. In *2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 1–10, 2020.

[108] Sangyeon Kim and Sangwon Lee. Smash the dichotomy of Skeuomorphism and flat design: Designing an affordable interface to correspond with the human perceptuo-motor process. *International Journal of Human-Computer Studies*, 141:102435, 2020.

[109] Philipp Kirschthaler, Martin Porcheron, and Joel E. Fischer. What Can I Say? Effects of Discoverability in VUIs on Task Performance and User Experience. In *Proceedings of the 2nd Conference on Conversational User Interfaces*, CUI '20, New York, NY, USA, 2020. Association for Computing Machinery.

[110] Michaela Klauck, Yusuke Sugano, and Andreas Bulling. Noticeable or Distractive? A Design Space for Gaze-Contingent User Interface Notifications. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '17, page 1779–1786, New York, NY, USA, 2017. Association for Computing Machinery.

[111] Christof Koch and Naotsugu Tsuchiya. Attention and consciousness: two distinct brain processes. *Trends in Cognitive Sciences*, 11(1):16–22, 2007.

[112] Jonas F Kraft and Jörn Hurtienne. Transition animations support orientation in mobile interfaces without increased user effort. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*, pages 1–6, 2017.

[113] Jon A Krosnick and Duane F Alwin. An evaluation of a cognitive theory of response-order effects in survey measurement. *Public opinion quarterly*, 51(2):201–219, 1987.

[114] Steve Krug. *Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability*. New Riders, 3rd edition edition, Dec 2013.

[115] Ernst Kruijff, Jason Orlosky, Naohiro Kishishita, Christina Trepkowski, and Kiyoshi Kiyokawa. The influence of label design on search performance and noticeability in wide field of view augmented reality displays. *IEEE transactions on visualization and computer graphics*, 25(9):2821–2837, 2018.

[116] Gordon Paul Kurtenbach. *The design and evaluation of marking menus.* University of Toronto, 1993.

[117] Lynn Kysh. What's in a name? The difference between a systematic review and a literature review and why it matters. In *Poster presented at Medical Library Group of Southern California & Arizona (MLGSCA) and the Northern California and Nevada Medical Library Group (NCNMLG) Joint Meeting, La Jolla, CA. Retrieved from http://libguides. sjsu. edu/c. php*, 2013.

[118] Vincent Lambert, Adrien Chaffangeon Caillet, Alix Goguey, Sylvain Malacria, and Laurence Nigay. Studying the Visual Representation of Microgestures. *Proc. ACM Hum.-Comput. Interact.*, 7(MHCI), 2023.

[119] Victor AF Lamme. Why visual attention and awareness are different. *Trends in cognitive sciences*, 7(1):12–18, 2003.

[120] Daniel Lamprecht, Dimitar Dimitrov, Denis Helic, and Markus Strohmaier. Evaluating and Improving Navigability of Wikipedia: A Comparative Study of Eight Language Editions. In *Proceedings of the 12th International Symposium on Open Collaboration*, OpenSym '16, New York, NY, USA, 2016. Association for Computing Machinery.

[121] Daniel Lamprecht, Florian Geigl, Tomas Karas, Simon Walk, Denis Helic, and Markus Strohmaier. Improving Recommender System Navigability through Diversification: A Case Study of IMDb. In *Proceedings of the 15th International Conference on Knowledge Technologies and Data-Driven Business*, i-KNOW '15, New York, NY, USA, 2015. Association for Computing Machinery.

[122] Edith Law, Ming Yin, Joslin Goh, Kevin Chen, Michael A. Terry, and Krzysztof Z. Gajos. Curiosity Killed the Cat, but Makes Crowdwork Better. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, page 4098–4110, New York, NY, USA, 2016. Association for Computing Machinery.

[123] Huy Viet Le, Sven Mayer, and Niels Henze. Imprint-Based Input Techniques for Touch-Based Mobile Devices. In *19th International Conference on Mobile and Ubiquitous Multimedia*, MUM 2020, page 32–41, New York, NY, USA, 2020. Association for Computing Machinery.

[124] Gun A Lee, Jonathan Wong, Hye Sun Park, Jin Sung Choi, Chang Joon Park, and Mark Billinghurst. User defined gestures for augmented virtual mirrors: a guessability study. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, pages 959–964, 2015.

[125] Joshua Leung. *Understanding the noticeability and distraction of interactive highlighting techniques*. PhD thesis, University of Canterbury, 2017.

[126] Rock Leung, Leah Findlater, Joanna McGrenere, Peter Graf, and Justine Yang. Multi-Layered Interfaces to Improve Older Adults' Initial Learnability of Mobile Applications. *ACM Trans. Access. Comput.*, 3(1), sep 2010.

[127] Jingya Li, Erik D. van der Spek, Xiaoyu Yu, Jun Hu, and Loe Feijs. Exploring an Augmented Reality Social Learning Game for Elementary School Students. In *Proceedings of the Interaction Design and Children Conference*, IDC '20, page 508–518, New York, NY, USA, 2020. Association for Computing Machinery.

[128] Wei Li, Justin Matejka, Tovi Grossman, Joseph A. Konstan, and George Fitzmaurice. Design and Evaluation of a Command Recommendation System for Software Applications. *ACM Trans. Comput.-Hum. Interact.*, 18(2), jul 2011.

[129] Material Design M3. Easing and duration - Applying easing and duration, 2023.

[130] Material Design M3. Transitions - Transitions help guide users as they navigate an app - Applying Transitions, 2023.

[131] Material Design M3. Transitions - Transitions help guide users as they navigate an app - Transition Patterns, 2023.

[132] Tomáš Macek, Martin Labský, Jan Vystrčil, David Luksch, Tereza Kašparová, Ladislav Kunc, and Jan Kleindienst. Interactive Car Owner's Manual User Study. In *Proceedings of the 6th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, AutomotiveUI '14, page 1–4, New York, NY, USA, 2014. Association for Computing Machinery.

[133] Renáta Máchová, Miloslav Hub, and Martin Lnenicka. Usability evaluation of open data portals: Evaluating data discoverability, accessibility, and reusability from a stakeholders' perspective. *Aslib Journal of Information Management*, 2018.

[134] Arien Mack, Irvin Rock, et al. *Inattentional blindness*. MIT press, 1998.

[135] Eva Mackamul, Géry Casiez, and Sylvain Malacria. Exploring Visual Signifier Characteristics to Improve the Perception of Affordances of In-Place Touch Inputs. In *ACM International Conference on Mobile Human-Computer Interaction (MobileHCI 2023)*, 2023.

[136] Wendy E Mackay. Patterns of sharing customizable software. In *Proceedings of the 1990 ACM conference on Computer-supported cooperative work*, pages 209–221, 1990.

[137] Aristides Mairena, Carl Gutwin, and Andy Cockburn. Peripheral Notifications in Large Displays: Effects of Feature Combination and Task Interference. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–12, New York, NY, USA, 2019. Association for Computing Machinery.

[138] Gabriela Mancero, William Wong, and Paola Amaldi. Looking but not seeing: implications for HCI. In *Proceedings of the 14th European conference on Cognitive ergonomics: invent! explore!*, pages 167–174, 2007.

[139] Stefan Marr and Benoit Daloze. Few Versatile vs. Many Specialized Collections: How to Design a Collection Library for Exploratory Programming? In *Conference Companion of the 2nd International Conference on Art, Science, and Engineering of Programming*, Programming'18 Companion, page 135–143, New York, NY, USA, 2018. Association for Computing Machinery.

[140] Andrea Marrella and Tiziana Catarci. Measuring the Learnability of Interactive Systems Using a Petri Net Based Approach. In *Proceedings of the 2018 Designing Interactive Systems Conference*, DIS '18, page 1309–1319, New York, NY, USA, 2018. Association for Computing Machinery.

[141] Paul Marshall, Richard Morris, Yvonne Rogers, Stefan Kreitmayer, and Matt Davies. Rethinking'multi-user' an in-the-wild study of how groups approach a walk-up-and-use tabletop interface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3033–3042, 2011.

[142] Justin Matejka, Tovi Grossman, and George Fitzmaurice. Patina: Dynamic Heatmaps for Visualizing Application Usage. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, page 3227–3236, New York, NY, USA, 2013. Association for Computing Machinery.

[143] Justin Matejka, Wei Li, Tovi Grossman, and George Fitzmaurice. Community-Commands: Command Recommendations for Software Applications. In *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology*, UIST '09, page 193–202, New York, NY, USA, 2009. Association for Computing Machinery.

[144] Sven Mayer, Lars Lischke, Adrian Lanksweirt, Huy Viet Le, and Niels Henze. How to Communicate New Input Techniques. In *Proceedings of the 10th Nordic Conference on Human-Computer Interaction*, NordiCHI '18, page 460–472, New York, NY, USA, 2018. Association for Computing Machinery.

[145] Joanna McGrenere and Wayne Ho. Affordances: Clarifying and evolving a concept. In *Graphics interface*, volume 2000, pages 179–186, 2000.

[146] Fenwick McKelvey and Robert Hunt. Discoverability: Toward a definition of content discovery through platforms. *Social Media+ Society*, 5(1):2056305118819188, 2019.

[147] Ron Mendoza and Kaari Peterson. From Tools to Tasks: Discoverability and Adobe Acrobat 6.0. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '04, page 912–927, New York, NY, USA, 2004. Association for Computing Machinery.

[148] Benedikt Merz, Alexandre N. Tuch, and Klaus Opwis. Perceived User Experience of Animated Transitions in Mobile User Interfaces. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '16, page 3152–3158, New York, NY, USA, 2016. Association for Computing Machinery.

[149] Martin Mihajlov, Mark Springett, and Effie Lai-Chong Law. Learning to Drag: The Effects of Social Interactions in Touch Gestures Learnability for Older Adults. In *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces*, ISS '19, page 205–215, New York, NY, USA, 2019. Association for Computing Machinery.

[150] Christopher A Monk, J Gregory Trafton, and Deborah A Boehm-Davis. The effect of interruption duration and demand on resuming suspended goals. *Journal of experimental psychology: Applied*, 14(4):299, 2008.

[151] Peter Morville. *Ambient findability: What we find changes who we become*. " O'Reilly Media, Inc.", 2005.

[152] Joerg Mueller, Robert Walter, Gilles Bailly, Michael Nischt, and Florian Alt. Looking glass: a field study on noticing interactivity of a shop window. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '12, page 1465–1466. Association for Computing Machinery, May 2012.

[153] Philipp Müller, Sander Staal, Mihai Bâce, and Andreas Bulling. Designing for Noticeability: Understanding the Impact of Visual Importance on Desktop Notifications. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA, 2022. Association for Computing Machinery.

[154] Andreea Muresan, Jess McIntosh, and Kasper Hornbæk. Using Feedforward to Reveal Interaction Possibilities in Virtual Reality. *ACM Trans. Comput.-Hum. Interact.*, jun 2023. Just Accepted.

[155] Emerson Murphy-Hill. Continuous Social Screencasting to Facilitate Software Tool Discovery. In *Proceedings of the 34th International Conference on Software Engineering*, ICSE '12, page 1317–1320. IEEE Press, 2012.

[156] Emerson Murphy-Hill and Gail C. Murphy. Peer Interaction Effectively, yet Infrequently, Enables Programmers to Discover New Tools. In *Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work*, CSCW '11, page 405–414, New York, NY, USA, 2011. Association for Computing Machinery.

[157] Emerson Murphy-Hill, Gail C Murphy, Joanna McGrenere, et al. How do users discover new tools in software development and beyond? *Computer Supported Cooperative Work (CSCW)*, 24(5):389–422, 2015.

[158] Emerson Murphy-Hill, Edward K. Smith, Caitlin Sadowski, Ciera Jaspan, Collin Winter, Matthew Jorde, Andrea Knight, Andrew Trenk, and Steve Gross. Do Developers Discover New Tools on the Toilet? In *Proceedings of the 41st International Conference on Software Engineering*, ICSE '19, page 465–475. IEEE Press, 2019.

[159] Jaye Nias. Guessability as an Ethnographic Study of Mobile Technology Usage in Kenya. In *Proceedings of the Seventh International Conference on Information and Communication Technologies and Development*, ICTD '15, New York, NY, USA, 2015. Association for Computing Machinery.

[160] Jakob Nielsen. *Usability engineering*. Morgan Kaufmann, 1994.

[161] Jakob Nielsen. Progressive Disclosure. Retrieved October 23rd, 2023 from https://www.nngroup.com/articles/progressive-disclosure/, 2006.

[162] Jakob Nielsen. Usability 101: Introduction to usability. Retrieved January 27th, 2022 from https://www.nngroup.com/articles/usability-101-introduction-to-usability/, 2012.

[163] Don Norman. Affordances and design. *Unpublished article, available online at: http://www. jnd. org/dn. mss/affordances-and-design. html*, 2004.

[164] Donald A Norman. *The psychology of everyday things*. Basic books, 1988.

[165] Donald A. Norman. Affordance, Conventions, and Design. *Interactions*, 6(3):38–43, may 1999.

[166] Donald A. Norman. THE WAY I SEE IT Signifiers, Not Affordances. *Interactions*, 15(6):18–19, nov 2008.

[167] Donald A. Norman. *The design of everyday things: Revised and expanded edition*. Basic books, 2013.

[168] Jan Noyes. Expectations and their forgotten role in HCI. In *Encyclopedia of Human Computer Interaction*, pages 205–210. IGI Global, 2006.

[169] Arfika Nurhudatiana and Alixia Shafira Caesarion. Exploring User Experience of Massive Open Online Courses (MOOCs): A Case Study of Millennial Learners in Jakarta, Indonesia. In *Proceedings of the 2020 9th International Conference on Educational and Information Technology*, ICEIT 2020, page 44–49, New York, NY, USA, 2020. Association for Computing Machinery.

[170] Timo Ojala, Vassilis Kostakos, Hannu Kukka, Tommi Heikkinen, Tomas Linden, Marko Jurmu, Simo Hosio, Fabio Kruger, and Daniele Zanni. Multipurpose interactive public displays in the wild: Three years later. *Computer*, 45(5):42–49, 2012.

[171] Alex Olwal, Jon Moeller, Greg Priest-Dorman, Thad Starner, and Ben Carroll. I/O Braid: Scalable Touch-Sensitive Lighted Cords Using Spiraling, Repeating Sensing Textiles and Fiber Optics. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, UIST '18, page 485–497, New York, NY, USA, 2018. Association for Computing Machinery.

[172] Martin T Orne. Demand characteristics and the concept of quasi-controls. *Artifacts in behavioral research: Robert Rosenthal and Ralph L. Rosnow's classic books*, 110:110–137, 2009.

[173] David Oswald. Affordances and metaphors revisited: testing flat vs. skeuomorph design with digital natives and digital immigrants. In *Proceedings of the 32nd International BCS Human Computer Interaction Conference 32*, pages 1–11, Swindon, GBR, 2018. BCS Learning & Development Ltd.

[174] Jamie Pearson, Jiang Hu, Holly P. Branigan, Martin J. Pickering, and Clifford I. Nass. Adaptive Language Behavior in HCI: How Expectations and Beliefs about a System Affect Users' Word Choice. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, page 1177–1180, New York, NY, USA, 2006. Association for Computing Machinery.

[175] S. Camille Peres, II Franklin P. Tamborello, II Michael D. Fleetwood, II Phillip Chung, and II Danielle L. Paige-Smith. Keyboard Shortcut Usage: The Roles of Social Factors and Computer Experience. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 48(5):803–807, 2004.

[176] Annuska Perkins and Tira Cohene. The Impact of User Research on Product Design Case Study: Accessibility Ecosystem for Windows Vista. In *Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility*, Assets '06, page 281–282, New York, NY, USA, 2006. Association for Computing Machinery.

[177] Nicole Ke Chen Pong and Sylvain Malacria. Awareness, Usage and Discovery of Swipe-Revealed Hidden Widgets in IOS. In *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces*, ISS '19, page 193–204, New York, NY, USA, 2019. Association for Computing Machinery.

[178] Nicole Ke Cheng Pong. *Interacting with signifier-less designs - the case of swhidgets*. Theses, Université de Lille (2018-....), Oct 2020.

[179] Venkatesh Potluri, Priyan Vaithilingam, Suresh Iyengar, Y. Vidya, Manohar Swaminathan, and Gopal Srinivasa. CodeTalk: Improving Programming Environment Accessibility for Visually Impaired Developers. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–11, New York, NY, USA, 2018. Association for Computing Machinery.

[180] Janet T. Powell and Mark J. W. Koelemay. Systematic Reviews of the Literature Are Not Always Either Useful Or the Best Way To Add To Science. *EJVES Vascular Forum*, 54:2–6, 2022.

[181] Alisha Pradhan, Kanika Mehta, and Leah Findlater. "Accessibility Came by Accident": Use of Voice-Controlled Intelligent Personal Assistants by People with Disabilities. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–13, New York, NY, USA, 2018. Association for Computing Machinery.

[182] Raquel O Prates, Simone DJ Barbosa, and Clarisse S de Souza. A case study for evaluating interface design through communicability. In *Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques*, pages 308–316, 2000.

[183] Jennifer Preece, Helen Sharp, and Yvonne Rogers. *Interaction design: beyond human-computer interaction*. John Wiley & Sons, 2019.

[184] Silvia Ramis, Francisco J Perales, Cristina Manresa-Yee, and Antoni Bibiloni. Usability study of gestures to control a smart-tv. In *Iberoamerican Conference on Applications and Usability of Interactive TV*, pages 135–146. Springer, 2014.

[185] Jef Raskin. *The humane interface: new directions for designing interactive systems*. Addison-Wesley Professional, 2000.

[186] Elaine M. Raybourn and Annika Waern. Social Learning through Gaming. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '04, page 1733–1734, New York, NY, USA, 2004. Association for Computing Machinery.

[187] Mark S Reed, Anna C Evely, Georgina Cundill, Ioan Fazey, Jayne Glass, Adele Laing, Jens Newig, Brad Parrish, Christina Prell, Chris Raymond, et al. What is social learning? *Ecology and society*, 15(4):r1, 2010.

[188] The Interaction Design Foundation. Make it Easy on the User: Designing for Discoverability within Mobile Apps. Retrieved January 27th, 2022 from https://www.interaction-design.org/literature/article/make-it-easy, Sep 2020.

[189] Christian Rendl, Patrick Greindl, Kathrin Probst, Martin Behrens, and Michael Haller. Presstures: Exploring Pressure-Sensitive Multi-Touch Gestures on Trackpads. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, page 431–434, New York, NY, USA, 2014. Association for Computing Machinery.

[190] Anne Roudaut, Eric Lecolinet, and Yves Guiard. MicroRolls: Expanding Touch-Screen Input Vocabulary by Distinguishing Rolls vs. Slides of the Thumb. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, page 927–936, New York, NY, USA, 2009. Association for Computing Machinery.

[191] Renate Rutiku, Jaan Aru, and Talis Bachmann. General markers of conscious visual perception and their timing. *Frontiers in human neuroscience*, 10:23, 2016.

[192] Paulo J Santos and Albert Badre. Discount learnability evaluation. Technical report, Georgia Institute of Technology, 1995.

[193] Joey Scarr, Andy Cockburn, Carl Gutwin, and Andrea Bunt. Improving Command Selection with CommandMaps. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, page 257–266, New York, NY, USA, 2012. Association for Computing Machinery.

[194] Joey Scarr, Andy Cockburn, Carl Gutwin, Andrea Bunt, and Jared E. Cechanowicz. The Usability of CommandMaps in Realistic Tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, page 2241–2250, New York, NY, USA, 2014. Association for Computing Machinery.

[195] Joey Scarr, Andy Cockburn, Carl Gutwin, and Sylvain Malacria. Testing the Robustness and Performance of Spatially Consistent Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, page 3139–3148, New York, NY, USA, 2013. Association for Computing Machinery.

[196] Katherine Schramm, Carl Gutwin, and Andy Cockburn. Supporting Transitions to Expertise in Hidden Toolbars. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, page 4687–4698, New York, NY, USA, 2016. Association for Computing Machinery.

[197] Mindy Seto, Stacey Scott, and Mark Hancock. Investigating Menu Discoverability on a Digital Tabletop in a Public Setting. In *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '12, page 71–80, New York, NY, USA, 2012. Association for Computing Machinery.

[198] Ross Shannon, Aaron Quigley, and Paddy Nixon. Software considerations for automotive pervasive systems. *Workshop on Software Engineering Challenges for Ubiquitous Computing*, 01 2006.

[199] Ben Shneiderman, Catherine Plaisant, Maxine Cohen, Steven Jacobs, Niklas Elmqvist, and Nicholas Diakopoulos. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Pearson, USA, 6th edition, 2016.

[200] Adish Singla and Ingmar Weber. Tagging and Navigability. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, page 1185–1186, New York, NY, USA, 2010. Association for Computing Machinery.

[201] Sidney L Smith and Jane N Mosier. *Guidelines for designing user interface software*. Citeseer, 1986.

[202] McKay Moore Sohlberg and Catherine A Mateer. Effectiveness of an attention-training program. *Journal of clinical and experimental neuropsychology*, 9(2):117–130, 1987.

[203] Konstantinos Spiliotopoulos, Maria Rigou, and Spiros Sirmakessis. A comparative study of skeuomorphic and flat design from a UX perspective. *Multimodal Technologies and Interaction*, 2(2):31, 2018.

[204] Arjun Srinivasan, Mira Dontcheva, Eytan Adar, and Seth Walker. Discovering Natural Language Commands in Multimodal Interfaces. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, IUI '19, page 661–672, New York, NY, USA, 2019. Association for Computing Machinery.

[205] Jeremiah D. Still and Veronica J. Dark. Cognitively describing and designing affordances. *Design Studies*, 34(3):285–301, 2013.

[206] Hironobu Takagi, Shin Saito, Kentarou Fukuda, and Chieko Asakawa. Analysis of Navigability of Web Applications for Improving Blind Usability. *ACM Trans. Comput.-Hum. Interact.*, 14(3):13–es, sep 2007.

[207] Suja Thomas. 2019 Feature Adoption Report, 2019.

[208] Gerard Torenvliet. We Can't Afford It! The Devaluation of a Usability Term. *Interactions*, 10(4):12–17, jul 2003.

[209] Marcus Trapp and René Yasmin. Addressing animated transitions already in mobile app storyboards. In *Design, User Experience, and Usability. Web, Mobile, and Product Design: Second International Conference, DUXU 2013, Held as Part of HCI International 2013, Las Vegas, NV, USA, July 21-26, 2013, Proceedings, Part IV 2*, pages 723–732. Springer, 2013.

[210] user&i. Do you consider Discoverability when designing? Retrieved January 27th, 2022 from http://www.userandi.com/do-you-consider-discoverability-when-designing/, March 2021.

[211] Priyan Vaithilingam and Philip J. Guo. Bespoke: Interactively Synthesizing Custom GUIs from Command-Line Applications By Demonstration. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, UIST '19, page 563–576, New York, NY, USA, 2019. Association for Computing Machinery.

[212] Robert J Vallerand, Luc G Pelletier, Marc R Blais, Nathalie M Briere, Caroline Senecal, and Evelyne F Vallieres. The Academic Motivation Scale: A measure of intrinsic, extrinsic, and amotivation in education. *Educational and psychological measurement*, 52(4):1003–1017, 1992.

[213] Sandra A. Vannoy and Prashant Palvia. The Social Influence Model of Technology Adoption. *Commun. ACM*, 53(6):149–153, jun 2010.

[214] Jo Vermeulen, Kris Luyten, Karin Coninx, and Nicolai Marquardt. The Design of Slow-Motion Feedback. In *Proceedings of the 2014 Conference on Designing Interactive Systems*, DIS '14, page 267–270, New York, NY, USA, 2014. Association for Computing Machinery.

[215] Jo Vermeulen, Kris Luyten, Elise van den Hoven, and Karin Coninx. Crossing the Bridge over Norman's Gulf of Execution: Revealing Feedforward's True Identity. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, page 1931–1940, New York, NY, USA, 2013. Association for Computing Machinery.

[216] J Vieira, Rosane Sampaio, Rafael Nascimento, João Pedro Ferreira, Sara Machado, Nuno Ribeiro, Estevão Silva, and S Mouta. Gestures while driving: A guessability approach for a surface gestures taxonomy for in-vehicle indirect interaction. pages 249–263, 2016.

[217] Daniel Vogel and Ravin Balakrishnan. Interactive Public Ambient Displays: Transitioning from Implicit to Explicit, Public to Personal, Interaction with Multiple Users. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, UIST '04, page 137–146, New York, NY, USA, 2004. Association for Computing Machinery.

[218] I. Wagner, M. Kompast, and R. Lainer. Visualization Strategies for the Design of Interactive Navigable 3-D Worlds. *Interactions*, 9(5):25–34, sep 2002.

[219] Robert Walter, Gilles Bailly, and Jörg Müller. StrikeAPose: Revealing Mid-Air Gestures on Public Displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, page 841–850, New York, NY, USA, 2013. Association for Computing Machinery.

[220] Hongling Wang and Joseph K. Kearney. A Parametric Model for Oriented, Navigable Surfaces in Virtual Environments. In *Proceedings of the 2006 ACM International Conference on Virtual Reality Continuum and Its Applications*, VRCIA '06, page 51–57, New York, NY, USA, 2006. Association for Computing Machinery.

[221] Ruijie Wang, Liming Chen, Ivar Solheim, Trenton Schulz, and Aladdin Ayesh. Conceptual Motivation Modeling for Students with Dyslexia for Enhanced Assistive Learning. In *Proceedings of the 2017 ACM Workshop on Intelligent Interfaces for Ubiquitous and Smart Learning*, SmartLearn '17, page 11–18, New York, NY, USA, 2017. Association for Computing Machinery.

[222] Taylor W Webb, Hope H Kean, and Michael SA Graziano. Effects of awareness on the control of attention. *Journal of Cognitive Neuroscience*, 28(6):842–851, 2016.

[223] Daniel M Wegner and Toni Giuliano. The forms of social awareness. In *Personality, roles, and social behavior*, pages 165–198. Springer, 1982.

[224] S. A. G. Wensveen, J. P. Djajadiningrat, and C. J. Overbeeke. Interaction Frogger: A Design Framework to Couple Action and Function through Feedback and Feedforward. In *Proceedings of the 5th Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*, DIS '04, page 177–184, New York, NY, USA, 2004. Association for Computing Machinery.

[225] Christopher D Wickens. Multiple resources and mental workload. *Human factors*, 50(3):449–455, 2008.

[226] Daniel Wigdor and Dennis Wixon. *Brave NUI world: designing natural user interfaces for touch and gesture*. Elsevier, 2011.

[227] Darren Wilson. The A to Z of UX — D is for Discoverability: 15 tips to ensuring users find what they need. Retrieved January 27th, 2022 from https://uxdesign.cc/the-a-to-z-of-ux-d-is-for-discoverability , July 2019.

[228] Jacob O. Wobbrock, Htet Htet Aung, Brandon Rothrock, and Brad A. Myers. Maximizing the Guessability of Symbolic Input. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '05, page 1869–1872, New York, NY, USA, 2005. Association for Computing Machinery.

[229] Jacob O. Wobbrock, Meredith Ringel Morris, and Andrew D. Wilson. User-Defined Gestures for Surface Computing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, page 1083–1092, New York, NY, USA, 2009. Association for Computing Machinery.

[230] Qiong Wu and Chunyan Miao. Curiosity: From Psychology to Computation. *ACM Comput. Surv.*, 46(2), dec 2013.

[231] Te-Yen Wu, Lu Tan, Yuji Zhang, Teddy Seyed, and Xing-Dong Yang. Capacitivo: Contact-Based Object Recognition on Interactive Fabrics Using Capacitive Sensing. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, UIST '20, page 649–661, New York, NY, USA, 2020. Association for Computing Machinery.

[232] Koji Yatani, Kurt Partridge, Marshall Bern, and Mark W. Newman. Escape: A Target Selection Technique Using Visually-Cued Gestures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, page 285–294, New York, NY, USA, 2008. Association for Computing Machinery.

[233] Stephen L Young. Increasing the noticeability of warnings: Effects of pictorial, color, signal icon and border. In *Proceedings of the Human Factors Society Annual Meeting*, volume 35, pages 580–584. Sage Publications Sage CA: Los Angeles, CA, 1991.

[234] Debora Zanatto, Massimiliano Patacchiola, Jeremy Goslin, Serge Thill, and Angelo Cangelosi. Do Humans Imitate Robots? An Investigation of Strategic Social Learning in Human-Robot Interaction. In *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, page 449–457, New York, NY, USA, 2020. Association for Computing Machinery.

[235] Fred RH Zijlstra, Robert A Roe, Anna B Leonora, and Irene Krediet. Temporal factors in mental work: Effects of interrupted activities. *Journal of Occupational and Organizational Psychology*, 72(2):163–185, 1999.

[236] J. Zimmerman. Video sketches: exploring pervasive computing interaction designs. *IEEE Pervasive Computing*, 4(4):91–94, 2005.

Appendices

A II In-Place Single Finger Inputs

A.1 Finite State Machine

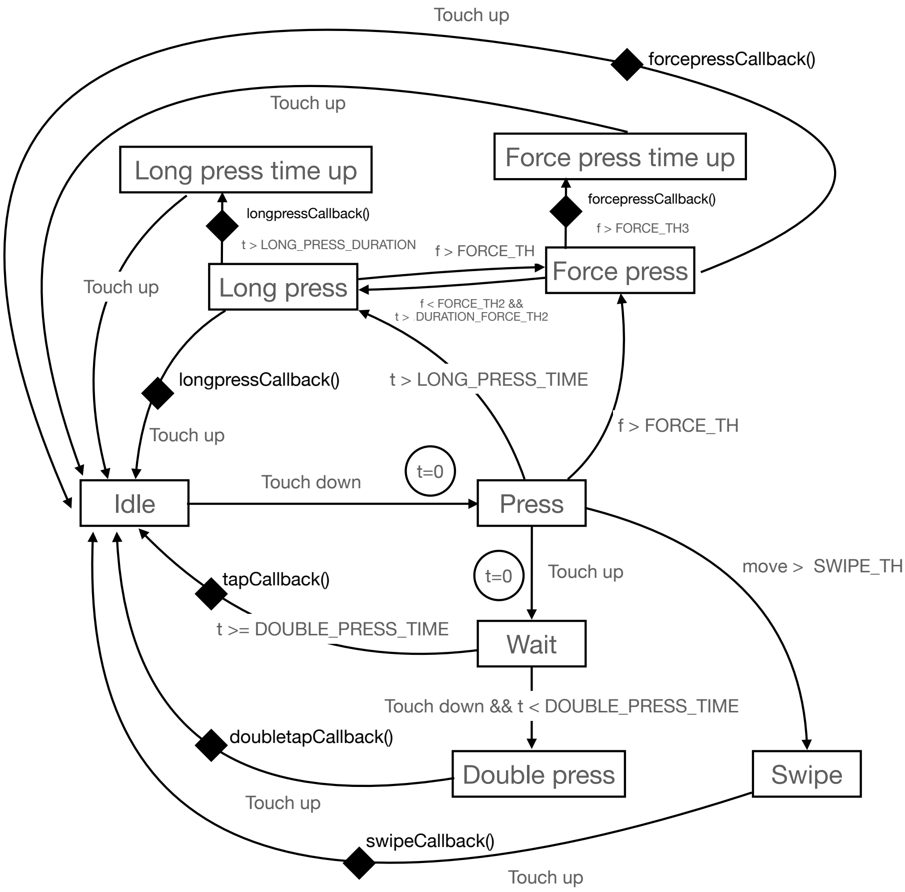


***Figure A.1** – Finite State Machine used to classify input states in real time during the laboratory experiment. We used the thresholds 0.3s for DOUBLE_PRESS_TIME and 0.75s for LONG_PRESS_DURATION, as defined by default on iOS; 10 pixels for SWIPE_TH; 0.9 for FORCE_TH, 0.7 for FORCE_TH2 and 0.1 s for DURATION_FORCE_TH2. The thresholds for force have arbitrary units but range from 0 to 1 as provided by Pressure.js 2.2.0 and were set using informal tests. Each design is associated with a callback that is considered as correct. Any call to another callback results as an error.*

A.2 Touch Interaction Categorisation

Touch Gesture	Contact Points	Position	Speed	Motion Pattern	Duration	Frequency	Pressure
Tap	1	static	-	-	short	1	regular
Double Tap	1	static	-	-	short	2	regular
Long Press	1	static	-	-	extended	1	regular
Strong Tap	1	static	-	-	short	1	high
Scroll	1	dynamic	medium	Move from contact point position in straight line to other point	varied	1	regular
Swipe / Flick	1	dynamic	high	Move from contact point position in straight line to other point	short	1	regular
Pinch	2	dynamic	varied	Move the 2 contact points straight towards one another	varied	1	regular
Spread	2	dynamic	varied	Move the 2 contact points straight away from one another	varied	1	regular
Rotate	2	dynamic	varied	Move the 2 contact points in clockwise or counterclockwise curve	varied	1	regular
Stroke Gesture	1	dynamic	varied	Move in predefined pattern	varied	1	regular

A.3 Table visualising study 1 responses

Table showing the *Selection%* for each DESIGN and *input*. Cells represented with a green background represent inputs that were assumed to be signified for the design. Others appear with a blue background. The intensity of the background colour is proportional to *Selection%*.

	Tap	Double Tap	Dwell	Force Press	Scroll	Swipe	Pinch	Rotate	Pattern
01 Basic	94	47	56	31	3	6			16
02 Text Countdown	81	81	44	38					
03 Text Timer	84	38	94	50		3			6
04 Text Counters	91	81	91	84					3
05 Text Small	94	25	88	84					
06 Text Temporary Attached	94	78	88	75			3		
07 Text Temporary Removed	97	81	84	78		6	3		3
08 Text Permanent Attached	94	84	94	75					
09 Text Permanent Removed	91	75	91	81		3			
10 Color Change (Steps)	94	56	47	38	3	6			3
11 Circumferential Steps	88	47	81	50	3	9		16	22
12 Scale (Check) Distant Temp	88	56	66	53	3	6			9
13 Checkmarks (Radio)	91	88	41	25			3		3
14 Checkmarks (Tick)	94	91	34	25	6				
15 Circles Around	88	75	53	53		3	3		
16 Scale (Check) Distant	97	59	69	44	3	12			6
17 Folding	72	28	84	31	12	38		3	3
18 Leaf Through	78	34	53	31	9	44			9
19 Scale Vertical Internal	78	25	94	78	3				
20 Color Change (Gradual)	97	25	66	53		3			6
21 Scale Circular Distant	81	22	88	62	3	3	3	9	3
22 Scale Radial	81	19	94	62	3	9	3		9
23 Scale Distant Temporary	72	28	94	66		6			3
24 Arrow	84	28	53	28	3	41	3		31
25 Scale Circular	72	25	97	66				3	
26 Scale Horizontal	75	25	91	72		3			
27 Scale Distant Permanent	84	22	94	69		3			
28 Sinking	81	28	59	62	12	12	6	3	12
29 Filling Balloon	75	28	84	59	9	16			9
30 Filling Water	91	25	91	62	9	6			6
31 Hourglass	91	41	72	28		3		6	
32 Scale Vertical	69	28	94	72	3	3			3
33 Squeeze	69	25	84	84	3	6			9
34 Ripples	91	41	53	66	6	9	3		9
35 Material (Concrete)	91	56	59	72	6				6
36 Highstrike	75	28	72	84	9	3			3

A.4 Table visualising study 2 responses

Table showing the *Attempt%* for each DESIGN and *input*. Cells represented with a green background represent inputs that correct for the design. Others appear with a red background. The intensity of the background colour is proportional to *Attempt%*.

	Tap	Double Tap	Dwell	Force	Swipe
01A	75	88	42	29	38
01B	42	21	79	25	21
01C	79	58	50	79	29
02	25	100	0	0	12
03	54	33	88	25	21
04A	46	100	4	4	4
04B	29	21	79	29	12
04C	8	8	8	100	4
05A	67	100	21	17	4
05B	46	4	100	8	17
05C	75	4	29	100	8
06A	42	100	62	0	4
06B	62	8	92	21	12
06C	54	17	54	96	0
07A	46	100	42	8	4
07B	46	4	100	8	4
07C	33	8	62	96	4
08A	4	100	0	0	0
08B	4	8	92	12	8
08C	8	4	17	100	0
09A	12	100	17	0	4
09B	17	4	100	4	8
09C	4	0	42	100	4
10	67	92	54	46	21
11	54	92	75	21	21
12	71	92	62	17	8
13	17	96	21	0	8
14	8	100	8	0	4
15	17	96	62	42	8
16	42	88	54	21	25
17	67	58	50	62	67
18	58	79	54	38	38
19	62	21	71	46	12
20	67	46	58	92	29

	Tap	Double Tap	Dwell	Force	Swipe
21	33	17	79	96	21
22	46	21	83	38	4
23	54	21	83	21	12
24	42	17	83	38	79
25	25	50	88	50	17
26	8	0	58	100	8
27	50	29	92	42	8
28	71	38	50	79	46
29	54	38	75	38	17
30	54	21	79	38	17
31	17	4	96	4	8
32	4	12	71	96	8
33	50	50	58	75	33
34	58	21	71	83	29
35	50	71	38	83	17
36	17	12	54	96	33

A.5 Figure visualising the input attempts in the interactive study

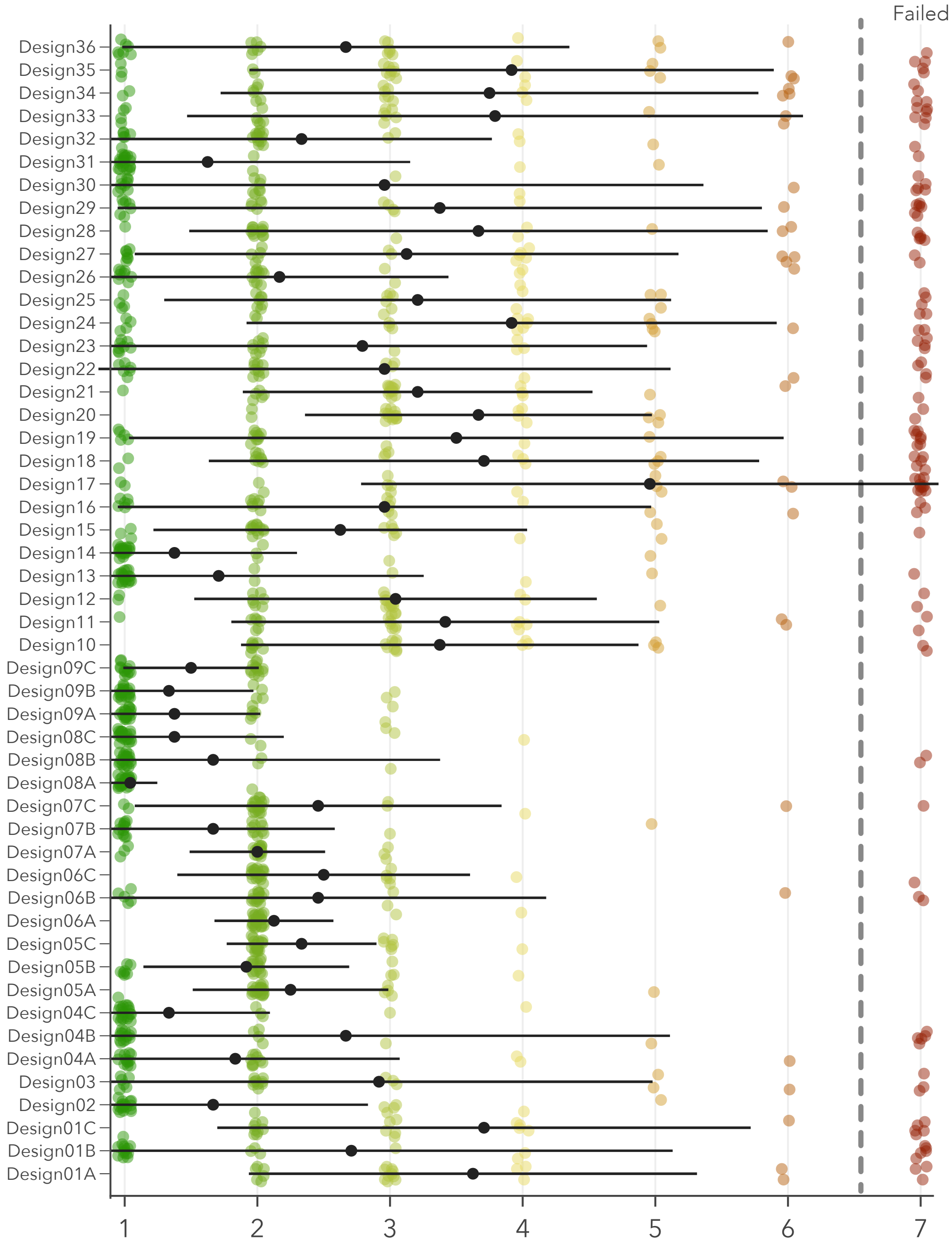


Figure A.2 – Trials required to achieve the correct interaction.