



UNIVERSITÀ DEGLI STUDI DI URBINO CARLO BO

Dipartimento di Scienze Pure e Applicate
Scuola di Scienze e Tecnologie dell'Informazione

Ph.D. thesis

**ON USABILITY OF DATA AND SERVICES
THROUGH MOBILE MULTITOUCH INTERFACES**

Tutor:
Chiar.mo Prof. Alessandro Bogliolo

Candidate:
Silvia Malatini

Dottorato in Scienze di base e applicazioni
Curriculum Scienza della Complessità
Ciclo XXIX ciclo- A.A. 2015 - 2016
Settore Scientifico Disciplinare INF/01

To Simone
“Forever with me ’till the end of time”

Contents

Abstract	1
Introduction	3
1 Mobile Usability	8
1.1 Usability definitions: scope	9
1.2 Evaluation methods	15
1.2.1 10 heuristics of Nielsen (and Molich)	21
1.2.2 Web usability guidelines	24
1.3 Web vs. mobile: mobile limitations and strengths	30
1.3.1 Limitations	30
1.3.2 Strengths	32
1.4 The mobile apps spread and usage: problem definition	32
2 A gamification approach to usability measures	36
2.1 Gamification, crowd-produced data and field trials	37
2.2 Gamification elements	39
2.2.1 Players	40
2.2.2 Game mechanics	43
2.2.3 Fields of applications	47
2.2.4 Pros. and cons.	50
2.3 Reachability problem	53
2.3.1 Smartphones evolution	54
2.3.2 Literature review	56
3 Usability game	63
3.1 Application design	64
3.2 Game design elements	67

3.2.1	Identity	67
3.2.2	Points, levels, achievements and scoreboard	68
3.2.3	Bonus	69
3.3	Data collection - The web-server	70
3.3.1	Architecture	70
3.3.2	Database	72
3.3.3	Web application - server side	73
4	Results and discussion	78
4.1	Deployment and usage	78
4.2	Evaluation	80
4.2.1	Device grips	81
4.2.2	Screen size and distance	85
4.3	Discussion	88
4.3.1	Limitations and future work	89
5	Beyond mobile interfaces	91
5.1	Evolution of user interfaces	92
5.1.1	Physical Devices	92
5.1.2	Input/output devices	94
5.1.3	Graphical interfaces	95
5.1.4	Outline	96
5.2	A deeper look at conversational interfaces	96
5.3	History of conversational interfaces	99
5.3.1	The dawn of "intelligent agents"	99
5.3.2	Spoken interaction	102
5.3.3	Virtual private assistants	104
5.3.4	Bot platforms	106
5.4	Modern bots	107
5.5	Overview of bot platforms	112
5.5.1	Interface features	117
5.5.2	Advantages of Bots for users	120
5.5.3	Advantages of Bots for developers	122
5.6	Apps vs Bots	123
5.7	Usability in the third wave of HCI	125
5.7.1	Can traditional Usability metrics and guidelines be used for bots?	127

5.7.2	Applying traditional metrics	128
5.7.3	New issues beyond usability	140
5.7.4	Discussion	142
	Conclusions	144
	Online references	147
	Acknowledgments	165

List of Figures

1.1	Learning curves	11
1.2	Most used mobile apps	12
1.3	Lab studies	17
2.1	A screenshot of Foursquare.	47
2.2	IBM Simon	55
3.1	Usability game logo	63
3.2	Application screenshots	65
3.3	Hand postures	66
3.4	More app screenshots	68
3.5	More app screenshots	69
3.6	Claiming bonus	70
3.7	The communication scheme between app and server	71
3.8	Delay heatmap on the web server	74
3.9	The possible filters.	75
3.10	Mobile version of the user website	77
4.1	Distribution of game sessions by game mode.	79
4.2	Total collected data points by device screen size.	80
4.3	Delay heatmaps	81
4.4	Average delay by position on screen.	83
4.5	Average distance from target by position on screen.	84
4.6	Delay by screen size.	86
4.7	Delay by distance between subsequent taps.	87
4.8	Average delay and average distance from target by target size.	88
5.1	Nokia communicator	93
5.2	Microsoft surface	94

5.3	Users of online messaging applications	97
5.4	A.L.I.C.E. brain	101
5.5	Main VPA logos	105
5.6	Main bot platforms	106
5.7	Quick replies	117
5.8	Structured commands	118
5.9	Structured messages	119
5.10	Telegram command list	129
5.11	Visibility of system status	132
5.12	Speak user language	133
5.13	Delayed message	135
5.14	Clear preferences	136
5.15	WeatherBot buttons inconsistency	136
5.16	Weatherbot	137
5.17	Hipmunk flexibility	138
5.18	Chatty bots	139
5.19	Hipmunks lack of errors indication	139
5.20	Helpful bots	140

Abstract

Usability of digital interfaces is a crucial point for their success, but, if a lot of efforts have been devoted to the development of usable desktop Web in the last decades, more has to be done for mobile environment. Mobile apps, mobile web sites, conversational interfaces, wearable devices, ubiquitous computing: a lot of new technologies are rapidly emerging and traditional usability studies are barely able to adapt to them, addressing the new challenges that arise because of fragmentation. The aim of this work is to investigate the state of the art in mobile usability techniques and researches, and try to address the problem of a lack of suitable approaches to usability studies. In this work, the application of game mechanics to non-gaming contexts—such as usability evaluation—is investigated. A simple mobile game application has been developed in order to collect a large amount of usage performance data produced by end-users in real world contexts. An analysis of the collected data is presented, comparing them with previous studies, and clarifying unexplored effects of device grip and screen size on usability metrics. Results show, as expected, that the increasing screen size negatively affects users performance in terms of speed and accuracy, and that device grips should be taken into consideration when designing interfaces. Results demonstrate that this data collection approach can be used to validate existing guidelines and renew them according to changes due to the evolution of hardware and software trends. Further studies have been conducted to understand what the emerging trends of mobile interfaces are, in order to figure out which new challenges usability studies will have to face up. Among the plethora of emerging hardware and software interfaces, a great, increasing, and renewed interest has been observed for

conversational interfaces, notably bots. Hence this old, yet new, kind of interface has been chosen as an example of technology that would need detailed studies of usability, in order to be fully exploitable. Conversely to bots developed over the last decades, in the last couple of years, bots have been acquiring increasing capabilities and purposes more related with everyday tasks. The possibility of using online messaging applications as real development environments is one of the main reasons for the current spread of bots. Bots reside inside these messaging applications instead of being standalone systems with their own interface, letting users interact with them by means of already known interfaces. An overview of these online messaging platforms, highlighting bot characteristics, advantages, and disadvantages, is also provided, comparing them with mobile applications, in order to understand whether or not bots could possibly be a replacement of apps. Discussing the different characteristics of these technologies, it can be argued that bots will not substitute apps in the near future, but that for now they can be a valuable alternative in some cases, and that in the future of mobile and ubiquitous computing, additional interface changes will undoubtedly have to be faced by designers and developers. An example of usability study of modern bots is presented, using a heuristic evaluation, in order to understand advantages and lacks of traditional methods, when related to emerging technologies. Both the study of interaction modes for mobile apps and the study of textual bots on messaging platforms prompt the development of new approaches to usability, tailored to emerging trends and technologies, in order to capture their peculiarities. In this context, gamification can be an effective way of gathering large amounts of data to support usability studies.

Introduction

Over more than thirty years of Internet, personal computers, mobile phones, and social networks, the way in which people get access to data and services has radically changed. From mere working tools, technological devices have acquired a predominant position in people everyday life and tasks. What has become evident is the centrality of the user in the process of technology commoditization: the possibilities offered by this evolution should be easily accessible to users in order to be fully valuable. Nowadays, the importance of the development of usable digital interfaces is largely renowned by web designers, developers, and human-computer interaction experts [94].

Another fact in this panorama is the continuous and rapid change of these technologies, besides the emergence of new ones. Not just hardware and software interfaces have been changing over the years, but also the user's propensity towards technology and the needs that technology has to satisfy have transformed. Because of the pervasiveness of the Web, users expectations have changed: people expect to find whatever they search, pretend that web sites work, and are generally less tolerant to faults and bad design [94]. Hence, challenges for usability studies have been abundantly increasing, are still growing, and will continue to grow, as this evolution is far from stopping [10]. .

After the emergence of a new technology trend, some time is needed before it becomes mature, usable, and accepted by users. Several studies and efforts are needed for the technology improvement and to understand the real users desires. For instance, smartphones have been out there since more than twenty years, but only ten years ago, with the iPhone, they have gained their success: at that time, technologies for the devices were mature, the use of digital devices in everyday tasks was already accepted, and using a phone for purposes different—and far—from simply calling or texting

someone have become fairly normal [122]. How was the iPhone different from other smartphones in order to persuade people to spend hundreds of euros (or dollars) for something not so fundamental in a person's life? Not just the more charming design of Apple's iPhone, if compared with existing smartphones, has ratified its success, but most of all, Apple's conception of the mobile Internet as being another modality of the existing wired Internet, and its leveraging of existing systems competencies [122]. Apple and the iPhone, have created in the people the need of owning a smartphone—especially that one—opening the doors to the spread of smartphones how we know them today [11].

Touchscreens, before that time, were not so popular too, whereas now it is difficult to find a smartphone with a physical keyboard. Resistive touchscreens of older PDAs (Personal Digital Assistants) could register only a single touch event at a time and needed significant more pressure than modern capacitive ones, making the use of a stylus almost necessary, and different kind of gestures very impractical [57].

In contrast to most screens of 2007, the iPhone's capacitive touchscreen was much more accurate, cheaper and multi-touch: different and more "natural" types of gestures—such as swiping for sliding—were possible, making it much more pleasant and easier for users to interact with the device. A more mature touchscreen technology, together with a set of dedicated applications, specifically oriented to the Web—unlike most other smartphones, the iPhone required a mobile data plan—and entertainment—with the iTunes music and video service—made Apple's strategy successful in turning the smartphone market into the most valuable in digital technologies in the last decade [122].

Hence, it is comprehensible how the usability of an interface can decree its success and how the importance of good and modern usability studies cannot be forgotten. Usability studies should follow emerging technologies and possibly be suitable for the new ones, easily adapting to them.

A lot of efforts to help designers and developers to build usable interfaces have been made through the years: usability guidelines and studies have covered ample areas of usability issues, resulting in improvements on different technologies, primarily web sites [66]. However, the time and efforts needed to have a deep knowledge of web usability have been huge so far, and are still not sufficient: not all the websites built today are usable,

even though the necessary tools are highly available.

The process that has brought to have an acceptable degree of usability of web sites, is being gone through by mobile applications since their spread, over the last ten years. The time needed to reach a good level of usability in mobile applications has been lower than the time needed by web sites to accomplish a similar level. One of the reason is that previous studies, regarding desktop web sites, have been adapted to mobile applications, hence existing knowledge has been reused, having more clear which aspects need to be investigated. However, as all with new technologies, the mobile environment has originated new challenges to be addressed, making necessary further studies, that are far from being completed [93]. Mobile devices have inherent physical limitations, that need to be taken into consideration when designing for mobile [18].

Even though existing usability studies can be adapted to emerging trends, efforts needed to have a deep knowledge of the technology, and to make it usable, mature, and successful are still quite high. Considering the rapid changes in users needs and interfaces, these efforts can become unaffordable and the time needed unbearably long. Traditional study methods, used to collect user data, to validate existing guidelines, and to develop new ones, need to be supported by new techniques, in order to minimize the efforts and time needed to make mature the new technologies [46]. Traditional laboratory experiments on usability are still the preferred method to research on guidelines [62], but new trends in the last ten years, have been increasingly focusing on field trials, collecting data from users in everyday life contexts [17].

A promising and still partially exploited technique to collect large amounts of data, from users in real contexts, is the use of game mechanics in non-gaming contexts, known with the term of *gamification* [30]. In the last couple of years, gamification has drawn the attention in academia and research and is being increasingly used in scientific contexts [123]. The development of small mobile games, in order to study users behavior while using mobile devices, seems a cheap and fairly fast way to collect large sets of meaningful data, that can be used in mobile guidelines studies [47].

In this work, a mobile application game—*Usability game*—we developed in order to study the influence of smartphone screen size and hand posture, used to hold the device, on users performance. Initially collected data,

on a two months basis, have been analyzed and results are presented and discussed.

Further studies have been conducted to investigate what the emerging trends in terms of interfaces are, and how traditional inspection and usability evaluation techniques are able to adapt to these new trends is also discussed.

Several emerging wearable devices—such as smart-watches, smart-glasses, wristband/armband fitness trackers, sensing jackets and so forth—have gained popularity, supported also by advances made in virtual and augmented reality, and in general what is called natural and ubiquitous computing, which are slowly advancing and gaining more interest in scientific research. A faster emerging—or re-emerging—trend, in the last couple of years, is represented by the renewed interest for older chatterbots. Over the last decades, conversational interfaces have received different degrees of interest: from the idea of developing intelligent devices, able to converse with real people, while fooling them into believing to interact with another human, today, “modern” bots are overtly artificial [76].

Indeed, one of the main differences between modern bots and traditional ones is their changed purpose: from mere research ambitions, now most of them are more focused on information tasks, and in general to supply users with useful data and services about diverse contexts, from weather forecast, to trip organization.

In the last couple of years, the mobile applications spread has suffered an abrupt trend reversal: most used apps are online messaging apps and users do not download tons of new apps as it was in the beginning [123]. Moreover, the main online messaging applications, have turned into real development environments, giving developers facilities for fast and easily develop bots that are able to access the more diverse sets of data and offer diverse services to users. This scenario of the re-birth of bots has raised the question whether modern bots could possibly be a replacement for traditional apps.

In this work, the emergence of this new application-like bots has been investigated, their strength and limitations and their usability has been studied, in order to determine whether traditional methods can be easily applied to new technologies.

Research questions

From the aforementioned open challenges and questions about mobile interfaces usability, and the future of software interfaces, the aim of this thesis is to investigate in these directions.

The research questions that the next chapters will try to address are:

- RQ1** - Exploring new techniques for usability studies: is it possible to evaluate and expand usability guidelines by means of large amounts of crowd-produced data, with the support of gamification?
- RQ2** - Validating the proposed approach by investigating the mobile screen reachability problem. How device grip and screen size affect user performance?
- RQ3** - Beyond conversational interfaces: what are the trends of evolution of mobile interfaces? Are traditional usability study methods suitable for new emerging trends?

Outline

In Chapter 1 an overview of state of the art web usability findings and techniques will be given. A specific highlights is given to the differences between web and mobile environments.

In Chapter 2 the proposed approach to evaluate and expand existing usability guidelines using gamification is examined. Some basic aspects of gamification and crowd-sourcing are introduced, together with an explanation of the reachability problem, and the existing solutions.

In Chapter 3 and 4 the developed mobile game is presented, followed by the analysis and presentation of the collected data.

Finally, in Chapter 5 a survey on hardware and software interfaces is provided, with particular care for conversational interfaces. The new trends on conversational interfaces—*modern bots*—are presented, and a comparison with traditional mobile applications is explained. Ultimately, an attempt to use traditional usability methods with modern bots is given, investigating new challenges and limitations of existing approaches.

Chapter 1

Mobile Usability

In their 2006 book, Hoa Loringer cited Nielsen's book "Designing web usability: The practice of simplicity", as the usability *manifesto*: the turning point in which websites' designers and developers finally understood that the success of a website was much more tied to its usability than to its coolness ([89], [94]).

Nielsen - Norman group is one the main and acknowledged companies studying web usability since the dawn of the web and websites. Until 2006, Nielsen Norman Group had published about 5,000 pages of reports from its usability researches, running experiments with thousands of users interacting with websites. Their usability guidelines have strongly influenced the web, and given a great pulse to abandon bad usability practices in favor of correct ones.

Since the publication of the "9 heuristics of Nielsen and Molich" [95], in 1990, almost thirty years ago, websites have greatly improved, but other technologies have taken hold. Mobile devices, and most of all, mobile applications –generally referred as *apps*–have become the predominant way in which users interact with digital technologies, according to recent statistics [67].

Thus, the attention has much more increased on mobile applications, but since the first *App Store* has been launched, in 2008, even though a lot has been done, much remains to do to improve apps usability.

Regarding websites, literature is full of studies and guidelines for any of their aspects; catalogues of those can be found anywhere and are also spread by the main players of the web, from Google to Microsoft. However, for

mobile applications and mobile websites, the situation is still fragmented and unclear.

This is also due to inherent reasons: mobile devices are much more variable than desktop ones among each others, thus highly rising the necessity for up to date and specialized mobile usability findings.

In the remainder of this chapter, the main aspects and techniques employed in web usability studies, will be depicted. A comparison between web and mobile will be introduced, with an analysis of apps usage, and an overview of state of the art of mobile usability techniques.

1.1 Usability definitions: scope

Since the beginning of studies in *Human - Computer Interaction*, their key goal has been to maximize the usability of interfaces, and several definitions of this term have been given and used over time. Initially thought for software systems, the same concepts have then been adapted to websites, and later to mobile applications.

ISO, the International Standard Organization defines usability as

“the extent to which a product can be used with effectiveness, efficiency, and satisfaction in a specified context of use”

as a broader concept that can be applied to other fields not directly related to digital interfaces [56].

Nielsen, with a more specific definition, talks about usability in terms of five parameters: learnability, efficiency, memorability, errors, and satisfaction, adding the concept of *utility*, as the extent to which a design’s functionality is needed by users [91].

As usual, with term definitions, confusion arises when several different definitions are used, causing assorted strands of research. Also similar terms, with different meanings increase the difficulty of fully analyzing a topic: user experience, accessibility, are different words with different meanings, but often associated to usability.

In this work, the second aforementioned interpretation will be used, in order to define the topic. An explanation of the listed attributes is given in the following sections.

Learnability

Concerns how fast users can understand how to use the system in an efficient and proficient way, meaning how long it takes a user to accomplish a useful task the first time they approach the system [88].

Even though for certain systems users are more inclined to face a slow learning process, because systems are perceived as complicated to use, in other cases users do not stand to spend large amounts of time to accomplish certain tasks. A famous example comes from Donald Norman's book "The design of everyday things", talking about *Norman doors* [99]:

My problems with doors have become so well known that confusing doors are often called "Norman doors." Imagine becoming famous for doors that don't work right. I'm pretty sure that's not what my parents planned for me.

Basically the author explains how disappointing it is for users to find difficulties in understanding easy systems, as doors are: a user trying to learn how to use a cockpit of a modern jet airliner, would not have a bad feeling in wasting several minutes to find how to switch-on the system, but what if for opening a door, one has to use more than some seconds? Fairly certainly they would feel angry, or disappointed, trying, for instance, to open a door by pushing instead of pulling.

In Figure 1.1 the curves show two different types of systems: a system with high learnability, focusing on novice users, and a more complex to learn, especially thought for expert users: a trade-off between learnability and proficiency of use should be carefully chosen.

Hence, it should be considered that certain systems require higher amounts of time than others to be understood, and be started to use with proficiency.

Trying to apply this concept to digital interfaces, the context of use and target users of a system are fundamental. Low learnability can be acceptable in "complex" software systems, but it has been shown not to be tolerable in websites: the average time after which a user gets disappointed and leaves a web page because of dissatisfaction is a question of seconds [72]. Thus, the web is generally perceived as an "easy" task, and learnability is probably the crucial aspect to be considered. As Krug's famous motto says, resuming all these attributes in one sentence, from the user's point of view: "Don't make me think" [65].

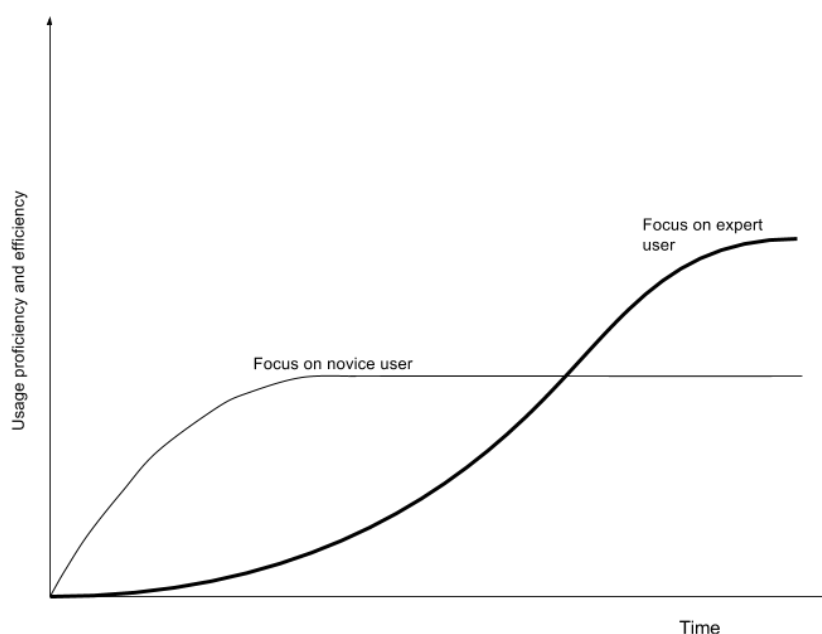


Figure 1.1: Learning curves: a system with fast growing learning curve, can be successful among novice users, but possibly suffers in terms of efficiency over time, limiting it to a certain extent; systems with slow growing curve, require more time to be understood, but allow expert users to reach higher efficiency with the system. Source [88]

When thinking of mobile apps, it has been shown that still a little part of users use them for complex or sensitive tasks, such as purchasing clothes, or online banking. The most used mobile apps are still focused on entertainment—i.e. social networking, music listening, gaming, as shown in Figure 1.2—hence, these should not be too complex tasks [68]. Moreover, the context of use of mobile devices is much more fragmented and diverse than desktop, as they can be used anywhere: apps should be highly learnable.

Efficiency

Efficiency is more focused on experienced users, indicating how well a user can reach their goal with the system, given they know the system.

As noted by Nielsen, it should be considered that not all systems can be fully known by their users: complicated software programs are likely to remain highly unexplored by average users, and partly by expert ones. Hence, efficiency could be measured as how many tasks can be performed by a

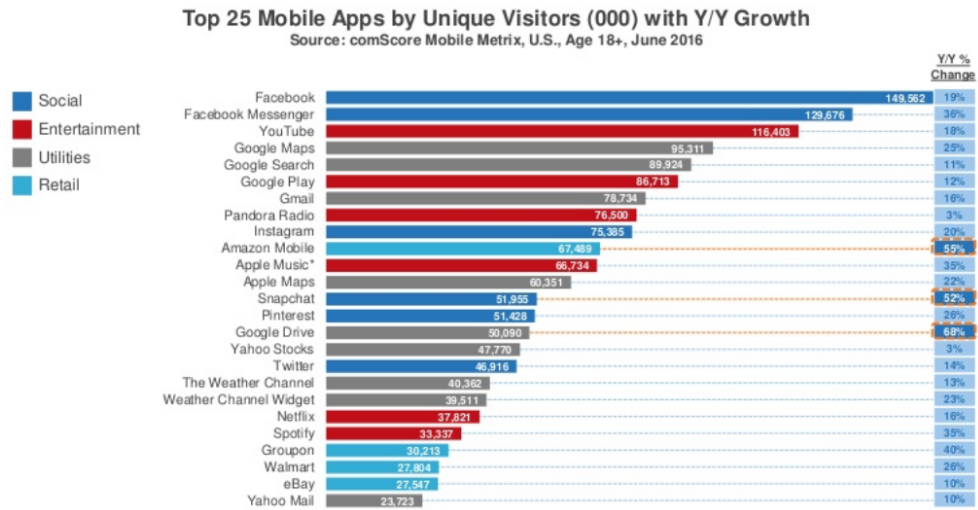


Figure 1.2: In the Figure are shown the 25 top used apps, ranked by number of unique visitors. The most used apps are about social networking and entertainment in general. Source: [68]

user.

On the other hand, an expert user could be considered the one that, knowing some advanced features, can perform a task quicker than average users.

There exists a steady-state level of proficiency, in which users feel themselves to have acquired enough expertise to use the system, and quit learning and improving their efficiency: it corresponds to the curve flattening in Figure 1.1.

Thus, the definition of efficiency of use, is fairly tricky and should be defined according to the context of use, using some metrics to measure the users' expertise.

As for *learnability* a distinction has to be made between the type of systems: the measure of efficiency in a software system should be fairly clear when comparing a novice with an expert user, differently for what can happen with web sites, and more with mobile applications. Tasks performed with web sites and apps, are generally easier than the ones performed with a software program, hence efficiency among users can be barely measured.

In many studies regarding mobile devices, tested with users, an expertise term often used is the period of time in which a user has owned some device. In the next chapters some studies will be introduced. See section on 2.3

Memorability

This attribute refers to *casual* users, meaning how well a user is able to use the system when returning to it after some time of non - usage [88].

Once a user has experienced a system for some amount of time, they can possibly temporarily go away from it: thinking of apps, it is sufficient to compare the number of installed apps in mobile devices, and the number of daily used apps. According to recent researches, considering an average of 60 installed apps, only an average of 3 to 4 are daily used [67]. Hence, the use of apps, and websites is mostly intermittent, more than it can be with a software used for work reasons.

As noted before, the chance to find users prone to spend fairly large amounts of time to learn web sites and apps is generally low. Thus, memorability should be highly boosted when designing such interfaces, possibly through learnability improvements. Nevertheless, a lack of learnability does not always means a lack of memorability, because, for instance, users can remember some features just because of their appeal, even if at first sight they are not so understandable.

Indeed, in these last 30 years of usability studies, one lesson comes from the increased understanding that a good part of the success, of any product including digital ones, comes from emotional design and individual factors, not just from its technical merits [98]. Hence, even though prioritizing "coolness" above usability is a step backward in the usability improvements gathered over decades of studies, it should not be forgotten that interfaces are for humans, and psychological aspects should always be taken into consideration, as a good practice of human-centric design. The emotional side of interfaces is further enforced in the attribute of satisfaction.

Errors

Failing is frustrating. System failures are frustrating too. Nevertheless, there are different levels of frustration, and different ways to mitigate it in users.

A bug-free system is yet to be seen, because even assuming a system free from programming bugs, designers and developers should always keep in mind that users are going to use it in unexpected ways, thus generating unexpected system behaviors and possible breakdowns. In first instance, particular care must be put in avoiding serious errors, causing loss of accomplishment in delicate tasks, or loss of user data.

Smaller errors, such as a broken link, can be sometimes tolerated, provided the respect of two conditions: it must be clear to users that an error has occurred, and they have a visible way to recover from it, possibly avoiding to make them feeling ashamed or guilty for what has happened. For instance, if during the registration process, a user types an incorrect password confirmation and the sign up process cannot be completed, a clear message with a suggestion should be shown in order to help the user finish their task.

Undo, back and home buttons, also, for instance, in apps should always be present and visible, in order to never let the user feel lost to accomplish their goals.

Satisfaction

As mentioned while talking about memorability, simply taking into account technical features of a digital system, in order to decree its success, has become progressively more constraining. The emotional side of using a product should be also taken into consideration, when designing a digital system [98] According to Nielsen, satisfaction indicates how pleasant it is for a user to use the system, and he highlights how this becomes more crucial for systems with an entertainment purpose, as subjective aspects become more important. A more general and user-centric approach is generally the field study of user experience.

A quantitative approach in satisfaction measure is obviously not feasible: in fact, it is often addressed with questionnaires in which users are asked to expose their opinions and rate the system according to some numeric or semantic scale. However, even with personal questionnaires, understanding users' satisfaction is tricky, because it can be related to different aspects: sometimes users find it pleasant to use systems that are "easy" to use, whereas others can find good-looking interfaces pleasant and so on.

Even for satisfaction, the context in which the system is used is crucial: when talking about web and mobile sites and apps, the fact that they are intended mostly for entertaining purposes, user satisfaction is considerably more important than for systems with working functionalities.

For mobile applications, app stores make directly available a system of rating and reviewing, as it is for the Google Play Store, or the Apple Store. According to Henze et al. studies on publishing apps in the stores, these reviews are generally less powerful than it can be thought [46]. Generally,

comments and reviews are more concerned to complain about malfunctioning or missing features, or about a broad appreciation, than as a true expression of users' opinions, hence poorly valuable to improve critical issues.

1.2 Evaluation methods

Once defined the term usability, which are the most used techniques used to first develop and then evaluate usable digital interfaces, should be analyzed. From this point, when talking about systems or interfaces, we refer to web and mobile websites, and to mobile apps, if not differently specified.

The process of designing usable and successful digital interfaces, in fact, could intervene at any of the steps of interface design and development: from evaluating non-functional mock ups, to user testing an up and running interface, with successive re-design and correction, it is an iterative process.

Usability interface problems have started to decrease when programmers and designers have stopped considering users as obstacles, and started to think of them as a resource [94].

Usable systems and interfaces generally require fairly large amounts of effort, in terms of time and money, hence the general trend of companies is to save these resources for other phases of product development, sometimes going towards failure.

The main types of usability analysis can be differentiated principally according to two different factors: who runs the evaluation and when the evaluation is run.

For the first aspect, the evaluation can be done from experts or with the aid of users; the other variable can vary if it is run on a functioning system, or on a not yet implemented one. A good set of possible different types of usability evaluation methods is given in the book of usability guidelines collected by the U.S. Department of Health and Human Services (HHS) [66]. In HHS study, studies conducted by experts are addressed as *usability evaluation*, whereas the term *usability testing* is used for studies conducted with users.

In the following sections a brief classification and description of the different techniques is given.

Run by users

User testing is generally made to evaluate a running system, and after a user testing session, data must be collected and analyzed by usability experts. It is possible anyway, to run testing at early stages of the design, and it is advisable to follow an iterative approach: after a user test has been run and corrections are made, the system should be tested again to evaluate improvements [66].

When running tests with users, a series of tasks to be fulfilled must be specifically designed by evaluators, and notified to users. After the test has been run, quantitative and/or qualitative data have to be collected and analyzed by experts. After summarizing the data, these have to be communicated to designers and developers in order to perform necessary changes.

Laboratory studies: a limited number of users is called in a proper location, adequately set up (see Nielsen for proper lab conditions), with proper equipment—depending on the type of device on which the interface has to be tested—and they are asked to run a set of tasks. Facilitators give users tasks to be run and indications about what is to be tested, while observers take notes on what users do, collecting data. Facilitators and observers can be in the same room with users, or in a separate room, observing video streaming (or recordings) of what users are doing.

Setting up a proper location for lab studies can be expensive, and not all companies have enough room to host users. Moreover, much care has to be taken, as observed by Sonderegger and Sauer, the presence of observers and facilitators can influence user performance: psychological stress responses, decrease of performance on some measures and affection on the emotional state of test participants, have been highlighted [115]. Unwanted bias on testing results are undeniably a deep drawback of this kind of evaluations, even though the interface with real users is an essential element in building its usability.

Thinking aloud testing: is a particular type of lab study, in which users are asked to talk, giving their opinions, impressions, and expressing their feelings while performing the test. The audio and video of testing sessions can be recorded and analyzed by experts, or they can stand near



Figure 1.3: A user running a lab experiment for a usability problem with smartphone devices. Source: [83]

the user taking notes. Another possible way is to ask the user to verbalize what they think. As with typical lab experiments, results are collected and analyzed, and then sent to practitioners.

However, drawbacks of this type of evaluation can possibly be even worse than “normal” lab studies, besides its undoubtedly benefits.

As observed by Nielsen, thinking aloud seems very unnatural to most people, and many users find it difficult to speak and express their ideas while using the system [88]. Moreover, asking to verbalize their thoughts can slow down user performances, or even change their behavior because of the writing.

Nørgaard and Hornbæk also highlighted altered observers’ behaviors: in a specific study, when testing, evaluators seem to seek confirmation of problems that they are already aware of, and the immediate analysis of the think-aloud sessions are rarely done. Rather than asking about experienced problems, evaluators often ask users about their expectations and about hypothetical situations, learning little about the utility

of the tested system [96].

Remote testing: the user and the expert are not in the same location. User evaluation can be done via a post-hoc questionnaire, or via a webinar, or with video recording, and tracking of what users have done.

Another kind of remote testing, that has gained popularity in more recent years, is the *field trial*: people use the system in real conditions for some amount of time and can collect data via diary reports, or with questionnaires. These kind of testing is also referred as *into-the-wild* studies [15]. A more particular type of remote testing, is the one in which users are recruited with *crowdsourcing*. A deeper look on these user testing methods will be given in the next chapter, where the approach used in this work will be presented.

Recommended iterative testing, with test on the before and after, is one of the good practices to always be kept in mind [66].

Run by expert

These are called inspection methods and beside the advantages coming from money and time savings—as they can be run directly inside the company from employee practitioners and not necessarily from usability experts, with low advance planning—, they have possible drawbacks.

Sometimes they identify usability problems without directly providing solutions to solve them [95].

Moreover, several studies have shown that far more problems than actually exist are often detected, whereas others are missed. On average, for every hit there will be about 1.3 false positives and 0.5 misses [66].

Inspection methods can better be used in order to identify problems to be further examined with user testing.

Cognitive walkthrough: designers and developers imagine the steps performed by a user to complete a specific task and then evaluate the system responses to those tasks.

After identifying tasks the user may want to perform, the participants in the evaluations typically ask for questions related to the steps they

imagine the user will need to complete the task. During the walk-through, usability is measured according to if and how well a user could perform the task, collecting data about potential problems. Particular care should be put in designing the tasks and simulating the users' behavior.

Cognitive walkthrough is much about the interface learnability and memory workload, as evaluators do not know how the system is structured, so they act as a user approaching it for the first time.

Automatic evaluation methods: dedicated software are able to produce insights about different metrics of an interface: just simply with Google Analytics it is possible to know which are the most clicked areas in a given web page.

Automatic tools were not feasible in the early 90s [88], but whereas now there are fairly famous examples for the web, few tools can be found for apps, because of technical difficulties ([63], [69]).

Automatic tools for usability measures collect data with user clicking and navigation, some advanced ones can use eyetracking (e.g. Eyequant [u1]). Others are libraries to be inserted in interface code to log events generated by different users' actions (e.g. FLUD - Framework for Logging Usability Data developed by the Visualization and Usability Group in National Institute of Standards and Technology, from US Department of Commerce [u2]).

Data are collected in web servers and can be visualized with heatmaps or in human-readable file formats, or other images formats. Later on they have to be analyzed and evaluated by experts.

Heuristic evaluation: this method will be explained in the dedicated later section 1.2.1, with an analysis of a set of possible heuristics.

Beside annoyance caused by non-usable systems, it should be taken into account that the impact of usability in everyday life can have severe consequences.

As in the example of *Norman doors*, interfaces with usability issues have been demonstrated to possibly make the difference between life and death: Norman also cites the Three Mile Island in 1979, a partial nuclear meltdown

happened in the nuclear power plant on the same name island, that caused the emission of radioactive gas in the environment. Even though the problem was said to be caused by “human errors”, Norman, who was asked to investigate on the accident, highlighted several usability problems in the supervising system [97].

Similar to this one, potential threatening problems have been shown in interfaces of health systems: system for dosing medicines to hospital’s patients with a numerical keyboard was given to nurses, changing from an interface with up and down arrows used before. With the numerical keyboard, nurses were shown to make mistakes of several orders of magnitude in dosing medicines, potentially seriously damaging patients’ health [97].

Even if such severe threats are not so likely to happen with common digital interfaces, these examples just reinforce the assumption that usability is crucial in many fields.

Thus, even though the importance of designing usable systems has been understood in years of experience and studies, still many companies do not apply the necessary techniques because of the necessary high efforts for running such studies, as previously mentioned. On the flip side, Nielsen propose the application of *discount usability* methods, as the application of simplified usability testing techniques, in order to lower costs and attracting companies to run usability studies [86]. According to Nielsen, just by applying simplified usability techniques, instead of complete ones, usability can be improved by a good extent and can be adopted by every company.

The importance of usability in commercial systems, either web sites or apps, has been also shown in many studies: usable websites have been shown to be perceived more trustworthy, with increased loyalty of their users ([94], [36]). Improved usability has a positive influence on user satisfaction, hence on product success. The same benefits can be observed on mobile apps.

Thus, the base rule for usability success is a user-centric design: when developing interfaces, this has to be done trying to understand users’ needs and prioritize them. Bad design has been shown often to be due to designers designing for not real targets: as Nielsen often outlines, designers and developers are not target users of their product, hence they should not be the ones evaluating their usability.

1.2.1 10 heuristics of Nielsen (and Molich)

Originally this set of heuristics was composed of only 9 items, and was developed by Nielsen and Molich, in order to lower the complexity of the big amount of available guidelines, and to give experts a quick way to evaluate the usability of an interface [95]. Subsequently the set was enlarged to 10, by Nielsen, and the heuristic evaluation better explained and tested as a usability evaluation method, that can be used by experts to analyze an interface and rate its usability, using this general principle, without using long time to select a good and complete amount of guidelines from the thousands available [85]. The term experts does not necessarily imply it should be usability experts, but it is used just to distinguish practitioners from users.

In heuristic evaluation, a small set of experts using a limited set of recognized usability principles—“heuristics”—judges how an interface is compliant to this set of heuristics. The best number of evaluators ranges from three to five (according to Nielsen [87]), since in previous studies with Molich, on different projects, evaluation made by only one evaluator, only found about the 35% of usability problems. Thus, it is advisable to run this analysis with multiple experts, and the indicated number, can increase the percent of found problems, maintaining a good trade-off with costs. Nevertheless this remains a cheap and quick method of evaluation, compared to usability testings conducted with users. Another advantage of heuristic evaluation is that it can be run at any time of the process of design and implementation, and with small advance in planning [95].

On the flip side, drawbacks of this method should also be considered, because other studies have shown a low problem detection, and the need for large numbers of evaluators: 16 evaluators have been needed to uncover the 75% of problems in other studies [66]. Other problems can come from the fact that evaluators frequently apply wrong heuristics or guidelines, which can be misleading for designers asked to solve the problem.

This set of heuristics is probably the most known and used ones by usability experts for their reviews; following a description of the heuristics will be given, and for each one, the original heuristic can be found between parenthesis.

Visibility of system status (Feedback): at any time the user is interacting with

the interface, it should be clear what is happening, after reasonable amount of time. At every action of the user, a feedback should always be provided, to inform them what is the effect of such action. If, for instance, a user clicks a search button, and the research takes several seconds to the system in order to perform it, a loader could inform the user to wait for the result to be provided. If the user is not informed on what the system is doing, it is likely they will feel lost, and likely to abandon the web site or the app.

Match between system and the real world (Speak the user's language): in order to make the user feeling comfortable with the interface, the information should not be provided with technical language, but rather with human expressions. Error messages, menu items, buttons' text, should be familiar and straightforward to the user to understand. Moreover, depending on the target users, the type of user that is likely to use the system, a different language should be used. Depending on ethnographic characteristics, system purposes and contexts of use, the system could change its way of expressing, always trying to keep a natural and logical order of the given information.

User control and freedom (Clearly marked exits): support for undo-cancel-redo-go back-home links is essential for users finding themselves in unwanted situations. It often happens that users find themselves in a different situation from the desired one, because of wrong steps: they should find an easy way to exit from that situation and go back to a more comfortable one.

The lack of such "emergency exit" can cause some kind of *panic* in users who find themselves stuck at some undesired point.

Consistency and standards (Be consistent): same words should mean the same concept, and different ones should mean different concepts too. If this distinction is not respected, users get confused and are prone to errors.

Furthermore, consistency with the external "world" should be respected: there is no need to re-invent existing and well-known terms, concepts and procedures: using familiar expressions and conventions, lets users feel more confident in using the system avoiding errors.

Error prevention (Prevent errors): as also mentioned while explaining usability definition, failures are frustrating, hence, preventing errors occurrence is important. Much attention should be put on severe errors, but any possible recognized error should be considered. Clearly, not *all* possible errors can be detected before the system is deployed to users, because users can perform unexpected actions with the system, but preventing known errors is crucial.

For instance, in a registration form requiring a password with special characters, password validation could happen while the user is typing, instead of on submission.

Recognition rather than recall (Minimize user memory load): while talking about memorability, in usability definition, it has been said that it should be easy for casual users to remember where to find what they look for. In addition, it must be said that also while using the interface, users should not be asked to remember previous executed steps or chosen options: objects and actions should be visible, and the user has to be able to carry on their tasks without remembering what they already did.

Flexibility and efficiency of use (Provide shortcuts): as previously seen while talking about learnability, when designing a system, both novice and expert users must be kept in mind. Without compromising the interface learnability, shortcuts for usual users should be provided, in order to increase their efficiency in using the system.

For instance, in an e-commerce web site or app, a registered user should be able to conclude an order without always re-inserting their address or payment information.

Aesthetic and minimalist design (Simple and natural dialogue): even though an interface should be aesthetically pleasant for users, unnecessary information should be avoided at any step of the interaction. A clean design, supplying concise and necessary information to the user, increases users satisfaction and lowers frustration in searching for elements.

Interfaces with redundant information do not allow users to easily find desired elements and appear difficult to use.

Help users recognize, diagnose, and recover from errors (Good error messages):

where error prevention is not possible—as previously said, not all the possible errors can be detected before deploying a system to users—errors should be clearly shown to users, together with a possible way to recover. Users' behavior and interface use is not predictable, hence, any possible exception should be highlighted and explained to users, and wherever possible, a recovery should be offered.

Help and documentation (new): even though a manual should not be necessary for a user to favorably use an interface, at some point help could be needed, hence a visible way to access documentation about the system, should be supplied.

Help, support, and documentation, to avoid the users feeling lost, should be visible and effective, focused on users' questions and task, and also completely but not too largely expressed.

1.2.2 Web usability guidelines

Nielsen claims to have developed the *10 heuristics* in order to provide designers and developers a quick way to design and evaluate usable interfaces, without struggling with thousands of guidelines [88].

As previously mentioned, there exists indeed several sets of structured guidelines for web sites, organized in categories and heuristics, covering all the main aspects of the interface. These guidelines are the result of specific researches on peculiar aspects of the interface and each addresses a specific interface issue.

Examples of structured guidelines for web usability have been developed from different types of institutions, such as governments, private companies, or public authorities. HHS, part of the US government [66] has developed and maintains organized and up-to-date, what is probably the largest set of these guidelines [66]. The W3C (World Wide Web Consortium) has several *Working Groups* that are developing standards and guidelines for usability and accessibility [26]. The Nielsen-Norman Group periodically publishes reports concerning their usability experiments to update already existing guidelines and disclose changes happening on the Web ([92], [94]).

These guidelines are generally divided into categories, each one addressing a macro-issue of interfaces, which is then detailed by single guidelines.

Beside differences among categories of different sets of guidelines, typical macro-areas of interfaces are similar:

Layout: recommendations regarding the pages structure, thus how to place elements on single pages in order to make it easy for users to find, read, and understand the content, and to perform their tasks. Pages layout deeply influences learnability for users approaching the web site, helping them to fast understand where to find what they are looking for.

Indeed, it is recommended to give the right emphasis to important page objects, following an order that reflects their importance: objects needing more visibility should be placed on the top or in the center of the page, where users' attention is firstly attracted.

Anyhow, the layout should not be messy, or too crowded: pages with high elements density, make it difficult for users to find the searched content, lowering memorability and also efficiency for expert users. One of the most important aspects concerning layouts, is compatibility with different screen resolutions. Using fixed or absolute layouts, using fixed dimensions for objects is one of the worst practices to be used. Implementing fluid and responsive layouts, that adapt elements dimensions to diverse screen resolutions, requires great efforts to developers, but it is a necessary requirement for usable web sites. Luckily, this good practice has been increasingly adopted with the web evolution, and several frameworks supporting developers to implement responsive design, have been emerging through years [66].

Guidelines suggest also to minimize pages' length, thus limiting vertical scrolling; horizontal scrolling should be attentively avoided.

Navigation: refers to the way users move inside the web site, hence all the elements providing hyperlinking, such as menus, buttons, links and so forth.

Navigation items have the aim to indicate to users their destination pages and how to reach them. Thus, navigation element labels should be descriptive, and menus clearly organized, with proper hierarchies and grouped elements. Main menus should be on the left panel, because it is the place where users generally expect to find them, or on

the top of the page. It should always be clear the difference between active and visited links, using different graphical effects, such as different colors, or underline.

Guidelines concerning navigation have the aim of helping users to easily identify the necessary steps to accomplish their tasks, without feeling lost, or without an exit. Users should always have clear where they are on the web site and where they should go successively: breadcrumbs and site maps are necessary tools to help users in their tasks. In order to always offer an emergency exit, also back and home buttons should be present, together with a search box in case some links cannot be found.

Information organization: in order to help users to find useful information, this has to be clearly organized and grouped, with proper order and hierarchy, with different web site and page levels: the information structure should reflect user needs and site's goals. Developers have to ensure that necessary information is displayed and important one is highlighted, avoiding unnecessary content. Indeed, users are known to spend more time scanning content instead of really reading it, hence this process should be supported using meaningful, unique and descriptive headings—with appropriate HTML order—and meaningful page titles.

It is also recommended, regarding the quantity of provided information, to minimize the number of pages and clicks.

Common pages: this category contains recommendations regarding web pages that are generally common to all web sites. Obviously the home page is always present, but also other useful pages, like FAQ (Frequently Asked Questions) and site map should also be present, to help users find help whenever they feel lost. Probably the home page has the highest importance among the pages composing a web site, but it is not uncommon people that land to other internal pages, after a web search with search engines.

In first instance, the home page design should clearly communicate site's purposes, giving a positive first impression of the web site. It should visually clearly appear as a home page, with limited written

content, and by any means showing all major web site's available options. Moreover, it should be reachable from any part of the web site, maybe through a clickable logo on the top of all pages.

On the other side, site maps should provide access to all the features of the web site, being up-to-date to reflect the current state of the web site. Not every possible link has to be provided, nor the web directory structure has to necessarily be respected, but all hierarchies and relationships should be clear.

Concerning FAQ page, it should reflect actual frequently asked questions from real users, not just additional information not fitting anywhere else in the web site. Questions should be grouped together if numerous, either by topic, or by popularity, or other types of categorization. If numerous, all the questions should be provided on the top of the page, and answered either on separate pages, or on the remaining part of the page.

Site map and FAQ page, like the home page, should be reachable from all other pages of the web site, but their links are generally placed in the footer part, at the bottom of the page.

Design: this set of guidelines concerns the global appearance of the web site, as well as the graphical and multimedia elements employed in web pages. In a general way, as recommended by one of the Nielsen's heuristics, even though the aesthetic appearance should be pleasant, an overloaded design can be confusing, hence also images, and other graphical elements should be carefully used. A lightweight design should be prioritized in order not to slow the download and response time of the web site.

Background should use simple images, or plain colors; images should be optimized for the web in terms of weight, and if large images have to be served, thumbnails could be used. Moreover, clickable images should be clearly indicated and above all, the use of images and multimedia in general—mainly audio and video—should be limited to the real need of using them in meaningful ways: graphics should facilitate learning and not just be aesthetic embellishments, or a way to convey user attention.

Audio, video, and animations starting without the explicit request of the user, and without a meaningful reason, have to be avoided, because they are generally annoying for users.

One image that should be always clear and visible, is the web site owner logo, possibly clickable, linking to the home page, in order to provide users with an always present emergency exit, and recognizability.

An important aspect of the general design is coherence, both internal, amongst the graphical web site's element, and external, with other existing similar graphical elements, such as icons or graphic buttons.

Content: recommendations about writing text for the web sites, from taglines, to articles, or products descriptions.

As mentioned for others sets of guidelines, users do not generally read the full content, but are more likely to scan it: users do not spend much time to understand a web site, sessions of use are fast, hence texts have to be clear and concise, avoiding unnecessary information, deferring less important ones to secondary pages or expandable paragraphs.

By any means, text has to reflect the users' language, avoiding technical terms, and using appropriate language depending on its target users, but also avoiding jargon. Also abbreviations and acronyms should be moderately used and always be explained

The first sentence of each paragraph should be descriptive, when writing prose, for the rest of the paragraph, helping users to understand what they need. Text appearance should also be bore in mind: reading on a digital screen requires more attention to the user, and it is generally more difficult to understand. Hence, graphical text characteristics have to be exploited to minimize the user cognitive load, and help them to easily differentiate words. Text-background color contrast should be high—black text on white background is generally a winning idea, font size should be appropriate and readable, and also font family should be familiar to users: researches demonstrate that there is no difference in speed reading serif or sans-serif fonts, but using unfamiliar fonts can slow down user performance [66].

Other suggested techniques to increase readability in long texts are the cautious use of mixed-case, using appropriate capitalization, and the

highlight of important information, with bold or underlined words or phrases, sparingly used.

Accessibility: special guidelines for people with disabilities. W3C has a special initiative called WAI [u3]—Web Accessibility Initiative—that develops guidelines and other material to help make the web accessible to people with limited viewing and hearing capabilities. WAI is composed of 3 working groups developing guidelines regarding web content, authoring tools and user agents, all of which are standards (ATAG - Authoring Tool Accessibility Guidelines, WCAG - Web Content Accessibility Guidelines, UUAG - User Agent Accessibility Guideline).

According to W3C, improving accessibility, also helps usability in mobile devices, as generally highly accessible web sites, easily adapt to different browsers and are highly usable. Whereas UUAG and ATAG regard the development of technical tools—notably more accessible user agents and authoring tools—WCAG indicate how to produce accessible web sites, based on principles that resemble the usability attributes developed by Nielsen: operability, perceivability, understandability and robustness.

In general, accessibility of web sites is based on the concept that the content should be available also if users do not have full capabilities of fruition. Limited possibilities could not be just physical impairments: slow connections, limited screen resolutions, malfunctioning devices, and other technical limitations could possibly be present too.

Thus, in order to produce highly accessible content, careful attention has to be put in using multimedia content: alternative texts or captions should be provided instead of images, or videos, or audio, and where not possible, substituting elements with the same meaning should be served.

All functionalities should be made available even in absence of a pointing device. Guidelines regarding text should be more stressed: conciseness, high-contrast, and correct font size and family should be studied in deeper way. Compatibility with different browsers and other tools should be also maximized.

1.3 Web vs. mobile: mobile limitations and strengths

When talking about mobile usability, inherent device limitations should be taken into deep consideration, because they strongly influence users experience of use [19].

With *mobile* also different types of interfaces are possible, the first choice is between a mobile web site or an app. Moreover, in the first case, another choice that has to be made is whether to design a responsive full website, or a dedicated mobile web site.

According to Nielsen it is generally better to have a separate web site, and even better, a dedicated app, mostly for complex tasks [84].

Furthermore it is necessary to know for which mobile device the app has to be designed: depending on which device is considered—a smartphone, rather than a tablet, or other—apps should have different interfaces and purposes. Indeed smartphones are more personal devices, whereas tablets are shared between people. The size of the touchscreen is different and tablets are generally used for easier tasks: playing games, checking email, social networking, watching videos, reading news or books.

Some limitations are common to all mobile devices, and whatever is the choice of having a mobile web site or an app, designers have to deal with them.

1.3.1 Limitations

Small screen size

In mobile devices the content view is much more limited than on desktop computers, as the available screen area is much smaller. Hence a limited amount of information can be displayed at any time to the user: small fonts often make reading on small devices an awkward experience.

Moreover, multiple windows cannot be displayed at the same time, making multi-tasking not possible. Users are forced to close and re-open the app whenever they need to temporarily switch task. Hence, a design should be *self-sufficient*: any mobile task should be possible within a single app or web site, in order to help users to fully understand needed actions [19]

Scrolling around, to have a complete and general view of the content is also necessary, increasing the cognitive load required to users: interaction is more complex and error prone.

Context of use/environment

The main difference of mobile devices is their portability. They can be used in any location, indoor or outdoor, with different weather conditions, surrounded by other people.

This conditions and context variability makes the attention on mobile often fragmented and generally sessions of use are pretty short. The average mobile session duration is 72 seconds, compared to the 150 seconds of desktop [18].

While using mobile, people are likely to be interrupted, and forced to temporarily give attention to other external factors, and go back later to finish their task. Hence designers should put much care in saving user's progress in their tasks, and make it easy to retrieve the context of interrupted tasks.

Furthermore it is even more important to minimize the information given to users: it is necessary to identify important information and give them as soon as possible, prioritizing essential yet complete design.

Touchscreen

Nowadays the largest part of mobile devices totally relies on this technology, and many are totally free of physical buttons. If several gestures, different from the single click, are efficiently exploited, the experience of use of touchscreen can be fluid and faster than with physical keyboards.

On the flip side, gestures suffer of low memorability and discoverability, they are sometimes hidden and fairly complex to remember [18].

Probably the biggest problem is for typing: using soft-keyboards requires the user to continuously split their attention between the keyboard and the writing area. Target buttons are pretty small and are subject to the so-called *fat finger problem*: whenever target sizes are smaller than the size of the person's finger, the tapping can lead to misleading actions, because of the wrong touched area. This problem is worsened because, while tapping, the clickable area is occluded to the user and tapping feedback are not always present, nor clear [53].

There exists a contrast between the reduced screen area and the target size of interface objects: it is necessary to find a good trade-off between this two needs has to be found, in order to minimize errors coming from mistyping, and avoid it.

Variable connectivity and download delays

According to Nielsen's researches, a maximum of 10 seconds is the time limit that users are willing to wait, in order to get the requested content from a web site [72]. Other researches increase this limit to 12 seconds, but the evidence is that users dislike waiting, and are likely to abandon the system if it requires long time to respond [52].

For web pages this time depends mostly on how light the web site interface is and on the amount and size of the assets to be downloaded, as generally good Internet connections are largely available [72]. On the flip side, when talking about mobile, connectivity can be an issue, because of variable coverage and speed. Hence, more than with web sites light interfaces and minimized number of page loads are crucial when designing for this scenario.

1.3.2 Strengths

Context awareness

If portability is a limitation because of attention and environmental conditions, it adds possibilities in the developed apps. Thanks to GPS sensors, beyond using mobile devices as navigation apparatus, a plethora of apps exploiting user's location have been and can be developed, giving enlarged possibilities to a variety of contexts.

From gaming purposes, to exploring nearby locations, users are given plenty of opportunities to take advantage of increased awareness.

Added sensors

Not only GPS sensors are present on mobile devices, but also accelerometers and gyroscopes add opportunities to data collection on a variety of physical measures thus increasing possibilities on app development.

1.4 The mobile apps spread and usage: problem definition

The pervasiveness of mobile devices in everyday life is something visible to everybody and statistics of recent years demonstrate it clearly. According to the US Mobile App Report, in 2014, mobile usage has overtaken desktop

usage, in terms of time spent, and the digital market to be dominated by mobile instead of desktop [67].

The number of available apps on Google Play Store is estimated to be 2.2 million, and 2 millions are the ones available on the Apple Store, as of statistics from June 2016 [u4].

The birth of app stores is dated back in 2008, almost ten years ago, after the spread of smartphones among the big public. Since their appearance, thousands of apps have been developed and deployed to these markets, for any possible purpose: entertainment, communication, work aid, health, education and so forth.

The average user has around 26 apps installed on their mobile phone, even though the number of downloaded and used apps on daily basis, is decreasing in these last years, as people tend to use mostly always the same small set of preferred apps, instead of installing new ones [123]. Other researches studying user behavior with apps have shown that on average, smartphones are used at least 1 hour per day, with very brief sessions, lasting less than a minute, and that the most used type of apps is still composed of communication apps ([13], [31]).

It can be fairly easy to understand the need for up to date usability guidelines specific for apps, as their extended availability. Also the big players of the app stores try to standardize and give advice on interface design: Google Material Design <https://material.io/> and Apple Guidelines [u6] are examples of how it is now important to prioritize mobile usability.

However, it must be taken into consideration that what happened with websites is happening with apps, at a more extended rate: in the dawn of websites about thirty years ago, the lack of structured sets of guidelines, the poor available technology, often resulted in poor user experiences, and awkward design.

Bad design practices, from Flash [u7] content, to heavy web pages were fulfilling the Internet. With thirty years of experience and researches, surfing the web is nowadays generally a pleasant experience—even if old-fashioned web sites are still out there—and structured sets of scientific guidelines can be extensively found.

Apps are relatively new, with less than ten years of life, and are facing the same problems of web sites experienced years ago. Poorly usable apps are still fairly numerous, and moreover, structured sets of guidelines are still

lacking [114].

The Nielsen–Norman group periodically publishes large studies on web and mobile usability, based on lab experiments with (not so) limited number of users, with varied backgrounds [93]. Beside their utility, completeness and reliability, they are expensive and require large amounts of time to be carried out and explained: the last study is dated back to three years ago.

Findings from the last mobile report indicates many still popular bad practices, that cause users to fail more than they succeed. Splash screens are still badly used, download times are the major factor of slow interactions, video and advanced multimedia are still very buggy, and there are still plenty of overstuffed pages, that make users feel lost [67].

If on mobile devices, gestures performed with different numbers of fingers, in different directions, composing them in different ways—such as pinching with thumb and index, or swiping with more fingers, and so forth—can replace common buttons for certain operations and open new possibilities of navigation, users have shown to remember only a restricted set of gestures, and that their discoverability is still low.

If on mobile gestures can replace common buttons and links for certain operations, being composed among them, resulting in large numbers of possibilities,

As in previous studies for the web, Nielsen has always shown how users are rushed, when dealing with digital interfaces: sessions of use, as previously mentioned, tend to last less than 1 minute, as people generally use mobile while on-the-go, or for killing time purposes. Less than with desktop, users do not have time to relax and explore digital interfaces:

“Users spend 99% of their time on other websites and apps” [90].

Meaning users are not willing to discover “your” system, but just want to get things done, so maximizing learnability is also crucial.

Finally, some considerations on mobile trends have also been illustrated. Firstly, a substantial difference between devices according to personal use: tablets are shared devices, whereas smartphones are perceived as much more individual [19]. Moreover a general trend to move to apps instead of mobile web sites have been shown, with a higher success of the former on the latter ones.

It must be also noted that mobile environments are constantly and fastly

changing, and the need for up-to-date guidelines, adapted to different devices, with different screen resolutions and dimensions, as well as different purposes is increasing.

Other efforts have been made in order to collect mobile usability guidelines and give them a structure, as it has extensively done for web sites, for easy consultation and use, but the problem of keeping these set up-to-date with reasonable costs in terms of time, money and human resources still remains ([114], [126]).

Other studies have tried to reinforce the definition of usability, adding to the usual definitions the support for mobile limitations, like the context of use [44]. Harrison et al. introduce a new usability model called *PACMAD (People At the Centre of Mobile Application Development)*: taking into consideration the three main factors affecting usability—users, tasks, and context of use—the two definitions from ISO ([56]) and Nielsen ([91]) are merged, with an extended description of errors and adding the cognitive load as a metrics.

Thus, it is necessary to find alternative ways to faster and cheaper adapt usability guidelines to a diversity of devices, and to develop new ones as soon as new trends on mobile emerge, trying to develop an environment populated by usable apps.

Chapter 2

A gamification approach to usability measures

Considering the problems presented at the end of the previous chapter, concerning mobile usability guidelines, in this work a possible approach to address these problems is presented.

The rapid and continuous evolution of devices and interface evolution makes it difficult to ensure that existing guidelines are up to date and uniform: those derived for particular types of devices, can become obsolete and sometimes not suitable for newer devices and technologies.

Moreover, the pervasiveness of mobile technology in everyday life possibly makes the low generalizability of lab studies more problematic than before: more “natural” approaches, not requiring to build artificial environments, should be taken into deeper consideration. The need of different research methods, to support traditional ones, is becoming crucial, considering also the increased effects of emotional and personal aspects of the use of technology [11].

Two great phenomena have been emerging in recent years and raised the attention of the great public: *crowdsourcing* and *gamification*.

Crowdsourcing is one of the major application areas for gamification, and it has been shown to be very useful to coordinate work for tasks that can benefit from a collective intelligence ([80], [43]).

Into-the-wild studies have become a nearly “standard” method in HCI, testing new systems with groups of users in uncontrolled conditions outside the lab, but the use of gamification to collect crowdsourced data for usability

studies, has not been fully exploited yet [17].

In the last two years, gamification has been used by at least 50% of organizations, as a mean to engage new users and re-engage existing customers [43].

Developing games for purposes different from simply playing and applying game mechanics to contexts outside gaming has gained, and is still gaining, increasing attention from industry and academia.

Nevertheless a large and accepted gamification definition is fairly recent, and not uniformly used yet.

In the next sections, a presentation about the use of gamification in usability context will be given, and the used approach will be presented.

In this work, a game application has been developed, to deepen the knowledge of a particular usability problem, i.e. of the effect of screen size and device grips on user performance, and it will be described in the next chapter.

Thus, the aim is to show how this approach can be a promising one to validate existing guidelines and create new ones with large amounts of crowd-produced data, in "natural environments" with the support of gamification.

In the remainder of this chapter gamification for crowdsourcing and the reachability problem will be presented.

2.1 Gamification, crowd-produced data and field trials

The term crowdsourcing has been introduced in 2006 by Howe and Robinson, as a new way for companies of outsourcing internal tasks, previously performed by internal employees, to large networks of potential collaborators by means of an open call format [51].

Brabham presents some examples of successful cases of crowdsourcing use as a business model: Threadless [u8], a web-based t-shirt company that crowdsources the design process of their t-shirt with an online competition; iStockPhoto [u9], an online, royalty free, micro stock photography provider, which offers photos, illustrations, clip art and videos, created by the crowd; InnoCentive [u10], an online platform for scientists, to receive professional recognition and financial award for solving R&D challenges [16].

Other famous crowd-based platforms, in more recent days are the ones

like Kickstarter [u11], based on crowd-funding of new ideas and start-ups, where people offer micro-donations to support more promising projects.

Other platforms supply paid users to run usability testing, like, for instance Amazon Mechanical Turk [u12], oDesk [u13], CrowdFlower and so forth [u14], who pay small amounts of money to people running usability testing, or surveys for various genres of researches.

In general, the strength of crowdsourcing, comes from the classical theory of the *wisdom of the crowd* – started to be studied by Galton at the beginning of the twentieth century – according to which, in decision-making situations, groups become particularly smart, often smarter than the smartest subjects inside them.

Hence, if on one hand crowdsourcing is an appealing method for gathering large amount of data on human behavior, reducing operating costs, and enlarging and diversifying the participants' pool, on the other hand, drawbacks arise about data quality and accuracy ([16], [29], [71]).

In "pure" crowdsourcing, the intellectual labor performed by the crowd, is much more valuable than comparable solutions sold by companies. People can possibly lack motivation because of poor rewards, thus providing limited valuable data. This issue is worsened in case people are not deeply interested in performing the requested task (i.e. who would systematically be a tester of a new app or website or technology, just for the sake of fun and curiosity, without a reward?).

Therefore, making use of gamification in order to address the motivation issue, to enlarge users' participation, and to leverage the advantages of crowdsourcing, appears to be an effective approach, which is increasingly used during the last years [80].

In their review, Morschheuser et al. show that the application of gamification, as a support for crowdsourcing, has a general positive impact, with notable effects: the increase of long-term user engagement, quality of the output data and the reduction in cheating, compared to the traditional paid crowdsourcing methods.

If the web is a necessary base-technology for crowdsourcing, the spread of mobile applications and games has given the right stimulus to the boom of gamification.

Henze et al. have extensively studied the effects of publishing experimental apps – some of which are games – to the different app stores, in

order to collect large amount of data, and have used the collected data to better understand existing usability problems, and evaluate existing guidelines ([46], [47], [48]).

Besides its increasing popularity, gamification in scientific literature is still a bit confusing topic, hence in next sections an attempt to give a clearer scope of definition will be given.

2.2 Gamification elements

The term *gamification* has its origin in the digital media industry. Used for the first time in 2008, the term has not seen a large adoption until the second half of 2010 [30].

Actually two are the most used definitions for gamification.

The first one, by Deterding et al. defines gamification as “*the use of game design elements in non-game contexts*”, thus proposing a general point of view [30].

Huotari and Hamari define the term from a service marketing perspective, as “*a process of enhancing a service with affordances for gameful experiences in order to support user’s overall value creation*” [54], highlighting more the goal of gamification, instead of its methods.

The two definitions address the term interpretation from two opposite, but complementary points of view: whereas the first see the inclusion of game mechanics in a non-gaming context as a sufficient conditions to talk about gamification, the latter strongly resides on gaming theory and considers the inclusion of some game elements as a necessary condition to talk about gamification. The main objection Huotari et al. make to Deterding’s definition is the lack of specification of an experiential point of view: the overall goal of gamification should not only be to supply a fun experience to its users, but also to steer players’ behavior by means of incentives and affordances.

Houtari et al. also highlight the fact that there does not exist a set of features concerning exclusively gaming fields.

Moreover, a plethora of similar terms boosts the confusion for this word:

“productivity games”, “playful design”, “behavioral games”, “game layer” and so forth.

Trying to dissolve the confusion determined by such philosophical dissertation, gamification is now commonly considered as “*the use of game thinking and game mechanics to engage users and solve problems*” [128].

Following the explanation of the main users’ motivations and applied game mechanics will be given, in order to have an essential, yet exhaustive overview of the topic.

Studies on gamification have been encouraged by the pervasiveness of gaming in everyday life tasks (see Subsection 2.2.3 for the examples of fields of application): gaming is becoming the way in which people interact with everyday tasks.

2.2.1 Players

The primary goal of gamification, in any of the contexts in which it is applied, is user engagement, namely to encourage users to engage with the task presented, whence the necessity to deeply understand users’ motivations for playing. The core of gamification are players.

Users’ engagement in gamification can find its fundamentals in *Flow theory*, firstly developed by Mihaly Csikszentmihalyi in the mid-70s. The concept of *Flow* represents the feeling of complete and energized focus in an activity, with a high level of enjoyment and fulfillment [28].

The act of performing the activity causes a total focus on it and positive feelings in the user, who loses track of time and worries while they are involved in the activity.

Flow is also called the optimal experience, or being in “the Zone”, and Chen applies this concept to the field of games, because it is the same experience that players feel when immersed in games [24].

According to Chen considerations, the more a game maximizes the duration of the Flow during a gaming experience, the higher is the quality of the Flow experienced by players, hence a key success for a gaming experience.

Besides the Flow and the positive feeling deriving from playing, a deep look should be given to the motivations that lead users to play a game, or perform an action.

According to Zichermann there are four main intermixed motivations for which people play games [128]:

- For mastery - interacting with a system to master an idea
- To de-stress
- To have fun
- To socialize

resulting in four main types of players (mutually inclusive):

Explorers Steered by the desire of discovering new things and show them to their communities.

Achievers People who are attracted by measuring their performances and get awarded.

Socializers Players who want to benefit of social interactions

Killers Or *griefers*. For this kind of players, winning is not enough as it is for achievers: someone else has to lose and others have to see it happened, otherwise it wouldn't be a real win.

Considering mutually inclusive the four categories, it comes out that the vast majority of people are socializers: about 75%.

Hence, motivation is strongly tied to psychological and behavioral outcomes: the result of the gaming experience should lead the user to positive emotions, in their Flow zone for the longest possible periods, and to a changed behavior concerning the task proposed by the game, or in general to generate some specific actions. For instance, games with ecological purposes should give users more consciousness on these problems and make them more respectful of the environment, as in the Swedish game developed by Gustaffsson et al. [41].

A deeper understanding of users' motivations – and hence how to increase their engagement – can be found in the *Theory of 16 basic desires*, developed by Steven Reiss, that has classified people's goals into 16 *motivators*, to identify what are the leading factors of people's activities [106].

Power, curiosity, romance, honor, vengeance are some of these basic desires that people need to satisfy in order to find happiness and satisfaction. According to Reiss, it is interesting to note that the satiation of one of these basic desires is always temporary: as soon as one desire is satisfied, it can reappear within some hours and need to be satisfied again. Moreover, the

most important desires that determines a person's personality are not absolute: sometimes people need to satisfy a desire that is opposite a fundamental one, once the fundamental desire is totally satisfied. For example, when a person experiences more power than they need, the individual is motivated to be submissive for some amount of time, in order to balance the effect of this over-satisfaction.

Generally, people have a quite balanced need for each of the 16 basic desires, and the individual personality can be determined according to the different order in which a person puts them. Reiss observes that the most important desires for explaining a person's behavior, are those that are unusually strong or unusually weak compared with appropriate norms.

Hence, a game's success is greatly based on the understanding of target players, their motivations and expectations: the broader the target, the more difficult to find the Flow Zone for all players, the more motivators should be satisfied.

The Flow Zone is determined by the trade off between challenge of the activity and the player's abilities to address and overcome it: too difficult challenge can cause anxiety, whereas a too easy one, compared to user's abilities, can cause boredom [24].

Hence, the mastery motivation is a leading factor for all players: all of them wish to improve their performance and abilities with the game, starting from a basic level, to reach a top level [128].

As a consequence, the game should be designed for the different levels of mastery:

- Novice: just arrived in the system.
- Problem solver: similar to a novice, but with some information already in hand.
- Expert: has started to learn how the system works.
- Master: has a deep understanding of the system.
- Visionary: a special kind of master, which has a deep understanding of the system and ideas on how to improve it.

In few words, games should take into great account users' motivations, help them to reach their expectations through playing, and try to maximize the period in which they are in their "Zone".

2.2.2 Game mechanics

Mechanics are tools that aim to leverage users' motivations by a system of rewards and call to actions. In this section the most used ones are analyzed, according to Zichermann and Cunningham classification [128].

As also reported by Hamari et al., the most used "motivational affordances" are points, leaderboards, and badges [43].

Points

They are an absolute requirement for most gamified system, whether they are shared among players or not, or among players and developers.

Their aim is to supply achievements to players, and to allow designers to understand how the users are interacting with the system: every move players make must be valued and tracked.

Several points systems have been developed and experienced by most users:

- *Cash score*: the number is not directly visible, but it is shown by status objects the player owns.
- *Video games score*: the score is always present on the screen, keeping the user informed on his progress and on how far he is from next level. More for video games than for gamified systems.
- *Social networking score*: points serve as a means of quantifying social acceptance as it could be the number of followers on social networks.
- *Keeping score*: the higher the score is, the more players are engaged.
- *Composite metrics*: score is determined by evaluating and merging different metrics, in order to give a more immediate comprehension for complex metrics.

Besides the choice of which kind of point system has to be used, the meaning points have is also an important choice:

- *Experience points*: indicate how players are ranked and guided. The more a user plays the game, the more activities they do, the more experience they gain.
- *Redeemable points*: can go up and down. Generally players expect that these points are usable within the system to be exchanged for things, as they are earned and cashed.

- *Skill*: gained by users performing some kind of tasks, they allow players to gain experience or rewards for their activities.
- *Karma*: these are not directly rewarding, but like in voting, the user's contribution to a certain cause, can help the community and the cause to happen. That would be the reward.
- *Reputation*: for systems requiring trust, it is important that players are ranked according to their good behavior.

Levels

Most of the times these are markers for progress, but this is not an exclusive role. Similarly to a full-fledged game, but far from being totally implemented as in video games, they tell the users where they are in the gaming experience over time.

Even though level difficulty is not linear (not doubling from one level to another), the complexity increase, passing from one level to another, makes the user gain confidence and experience, motivating the user to master his experience.

A good game level design should make them logical or easy to understand for players, extensible and flexible to expand and modify for developers.

Leaderboards

Basically leaderboards serve as ranking systems to increase users' competition. However, special care must be reserved in designing leaderboards.

Comparison among users should be an incentive to improve performances, instead of outlining players' weaknesses. Players should see themselves in the middle of the leaderboard, despite their real position: the portion regarding each players should be shown to them, together with players below and above them, instead of the complete one.

Another limitation in the leaderboard view could be given by filtering results according to some parameters: in a leaderboard with thousands of players, a user could choose to visualize scores of local players, comparing their results with users in the immediate vicinity. Other filters could be based on time: daily, weekly, or monthly leaderboards can be a good incentive for different kind of users, from novice to expert ones.

Privacy issues should also be taken into careful consideration: comparing players according to some personal information, or ability, can be a disincentive. Education based games, for instance are not the most suited ones for leaderboards, as they measure very personal users' skills, possibly causing ashaming in some players who are not inclined on publicly showing their intellectual abilities.

Leaderboards should be designed with a win-win proposition for all players.

Badges/Achievements

Badges (or achievements) are signals to show other players the user's characteristics and goals, and in some systems they can replace levels as a progress marker.

However there are several reasons for which players desire badges: some players feel collecting badges as an expression of power, others like the surprise of getting unexpected badges, whereas also others just like them for aesthetic reasons, if badges are visually valuable. It must be kept in mind that users positively respond to good design.

Hence, a valuable badge design should take into consideration psychological objectives of the player, as well as a good visual appeal.

Customization

Most designers believe customization is a powerful tool for commitment and engagement: letting users recognize themselves and differentiate from others can be a good way to value the game experience.

The demand of 3D avatars in gamified systems is fairly lower than full-fledged games, hence there is generally no need to implement a fully customizable avatar system. Letting users choosing their own username and picture, can be enough in several systems, like in social networks.

Moreover, letting too many choices to users, can be sometimes counter-productive: a limited set of choices is generally a good trade-off between user engagement and satisfaction.

Challenges/Quests

Their aim is to serve users as a guide on how to proceed along the game experience: many players come to a game without any idea on what to reach,

and how to proceed. Challenges ensure users approaching the game to always have something new to try, hence to increase long-term engagement.

It must be kept in mind different challenges for different levels of users are needed: for every level of users, a different level of challenge should be proposed.

Challenges involving quests, as, for instance, recruiting a certain number of friends in order to accomplish a certain goal, strongly promote socialization, which, as previously seen, is a very pushing motivation in playing games.

Thus, providing users with proper challenges and quests is a truly valuable way of long-term engagement and loyalty.

Onboarding

It's the action of encouraging players to start playing the game. The first minute spent in playing the game has been shown to be one of the most important reasons for users to keep playing the game: the first minute a player spends with the game is not for explaining everything about the game, but for allowing users to explore the game.

Giving too much information at the beginning discourages users to keep on playing, as well as pretending novice users to act like experienced ones. At a first interaction, players should feel comfortable and winning in playing the game, as there will be plenty of time in explaining how every feature of the game could be exploited.

Players' experience should be gradually guided with increasing levels of complexity, and requested skills to overtake increasing difficult tasks.

Social engagement loops

Viral loop design, like social networks posts: it's the way in which a system makes user go back to it.

The best example comes from social networks: users exposing their thoughts and emotions, whenever this exposure is rewarded with social recognition, they will be encouraged to reiterate this behavior, hence increasing game engagement, in a long-term loop.

As an example, Zichermann and Cunningham propose Twitter: people express themselves on a certain topic; others react to their thoughts and mention them; the user is called to go back again to the system and, if their

opinion is a valuable one, they will gain followers and will be motivated to keep expressing themselves [128].

2.2.3 Fields of applications

As previously mentioned, in the beginning, gamification has mostly received attention from industry and commercial companies, as a marketing tool to attract and engage new and existing users, and only later, by academia.

There have been several successful examples of gamification as a marketing tool, starting from Foursquare [u15] in 2009: every *check-in* is rewarded with some points, *badges*, and mayorships visible in one's public profile, together with possible tangible rewards offered by some companies to mayors of some places [70].

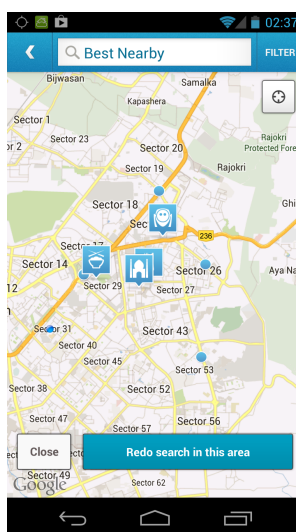


Figure 2.1: A screenshot of Foursquare.

Like Foursquare, many others online services offer reward systems for users' tasks and game mechanics (e.g. TripAdvisor, StackOverflow, HuffingtonPost and so on), whereas others developed their own games, with tangible rewards to attract customers (e.g. KLM, the Dutch Airline with Aviation Empire, a 3D strategy mobile game simulating to be the CEO of an airline).

Outside commercial use, gamification has been applied to several diverse context: an extensive survey can be found in [43].

Commerce: Hamari used a badge system in Sharetribe [u16], an online marketplace for local communities, where users can buy and sell goods and services in a trusted environment. Their aim was to study how user activity changes, according to different configurations of possibilities granted by badges.

The outcome was that users who actively used the badges system were more likely to actively use the service, but the author concludes that implementing gamified features alone, would not lead to significant increase in usage frequency [42]

Education/learning: Fitz et al. developed a mobile application, named Orientation Passport, to help new students at university [35]. The application utilises games achievements to present personalized orientation events to new students, in order to introduce them to the university campus, people and services. The application, used on small set of students, was found useful by the majority of them, and that achievement system added value to their orientation experience.

Health in the largest part of mobile applications dedicated to exercises, a gamification system has been applied: Runtastic, Endomondo and others, all offer systems to share progress on social networks with friend, have badges and achievement systems, and leaderboards for different categories.

Social recommender systems: Farzan and Brusilovsky studied the impact of gamification on user participation in a collaborative and social recommender system [33]. A course rating system, CourseAgent has been implemented, and gamification techniques have been applied to increase the number of course ratings: users' feedbacks are turned in self-beneficial activities. Other famous systems, like TripAdvisor, have also introduced gamification elements, with badges system and user leaderboard. In spite of positive effects on the increase of users' participation, the authors reported a positive rating bias, due to the incentive based on personal needs mechanism: users rated courserers with higher marks, when taking advantage of personal incentives on their career. This is an entailed threat coming from incentive systems in gamification, that have to be carefully used in order not to bias results from

field studies.

Sharing: Montola et al. applied an achievement system to a photo sharing service, to study the effects of achievement systems outside the gaming domain and its potential in teaching users to explore the various features of the service [79]. Even though most users welcomed the system in a good way, and some friendly competition and comparison between users was aroused, some users expressed concerns about achievements. It must be taken into account that the achievement system mentioned here has been applied in very early stages of gamification, hence some concerns from users seem fairly natural.

Sustainable consumption: in Sweden, a pervasive game to encourage teenagers and their families to reduce energy consumption in the home, called Power Agent, has been developed by Gustaffsson et al. [41]. The authors' aim was to promote engagement with a team competition scheme, in power consumption reduction, and to gather information about electricity usage exploiting home environment and its devices.

The results shown high users' motivation and engagement in changing their daily energy-consumption patterns during the game trial, even though no evidences have been gathered on the influence of the game on players, after the trial period. Nevertheless the approach of increasing people's consciousness on social and ethical issues seems also a promising field of application for gamification.

Work: Anderson et al. have studied the badge system in the popular Q&A website StackOverflow, dedicated to practitioners and newbie of software development [1]. The aim of the study was to analyse how badges can boost people in changing their behavior and how to use this system to steer users' behavior in particular ways. A prediction model has been developed and validated with data extracted from the website: their findings show how the optimization of badge placement is crucial in this kind of systems of incentives.

Innovation/ideation: In order to analyze challenges and potential of game mechanics, Witt et al. applied gamification to an online idea competition [124]. Their results show how carefully game mechanics should be designed and applied in all the contexts: besides the typical enjoyment

deriving from common game elements, the authors highlighted poor agreeability deriving from inattentive game mechanics development. Little details can make the difference in the success of the application of gamification in any context.

Usability: some gamification application to usability studies, concerning the reachability problem, and typing performances, will be presented in the next sections .

2.2.4 Pros. and cons

Although the application of gamification to the most diverse fields of research interest is a promising approach, a full knowledge of the topic is far from being reached. Beyond the "promises" of collecting large amount of data with considerable cost reduction, of enlarging users participation and long term engagement, because of motivation, further positive considerations can be underlined:

Pool diversity: there is still a clear tendency towards experiments in laboratory settings in mobile HCI research, while learning from real use of systems in natural settings is not prioritized [62]. Most of the studies are based on controlled experiments, run with a limited amount of subjects, often recruited from a pool of users with similar knowledge and habits with smartphones. This can lead to a bias in collected behaviors because the considered sample is not fully representative of real world users.

Human behavior influence: as shown in other studies, the influence due to the presence of an external examiner and observer in a controlled lab experiment can also lead to a possible bias in the results [115]. Brown et al. [17] investigating field trials, in which a restricted pool of subjects is explicitly asked to use a system in a natural environment, and log results on diaries, or give their opinion in final qualitative questionnaires and surveys, have also shown how bias can threaten gathered results. Users behave as to satisfy practitioners' request, hence forcing their behavior, biasing collected result. "True" into-the-wild studies, can exclude this bias.

Environmental settings: As observed in other studies, experimental conditions in a controlled laboratory environment do not fully reflect the variety of real world smartphone usage contexts: using the smartphone in a crowded square, in a sunny summer day, is not the same as using it in an almost empty room with good lighting [47, 82]. This is a predominant limitation in lab studies, as mobile devices are primarily thought to be used in the more diverse situations and environmental settings.

Nonetheless, several challenges are still open and should be carefully addressed in order to fully exploit this approach:

Cost of app development: designing and developing specialized games applications for mobile devices, can be a costly activity. Much effort must be put in studying proper design and affordances to attract and motivate large number of people, and sometimes the latter can be the hardest part, as different studies reported [46]. Dergousoff and Mandryk have developed a framework emulating the “Freemium” model, in which players can play a game for free, but have access to additional contents by paying a fee [29]. Comparing non-gamified versions of tasks proposed in their game, they found a significant improvement in users’ engagement and found this model as a good approach to contain app development costs.

Data validity: findings of experiments conducted on a large number of users in natural settings can be generalized with higher confidence, i.e., the results have a higher *external validity*. On the flip side, the absence of control over the participants, their usage context, and other external factors entails that the findings are harder to attribute to the sole manipulation of independent variables. That is, the studied factors have less direct influence on the observed effects and thus the study has a lower *internal validity*. Clearly, studies based on large scale public experiments—as was already noted in a study by Henze et al. based on the same premises [46]—inherently trade off internal for external validity. According to Henze et al., because of the uncontrolled conditions of usage, this approach should be considered a quasi-experiment as it is difficult to determine which factors determine certain users’ behaviors or choices. Moreover, unpredictable usage of the app can produce

unforeseen usage, biasing the data: users could stop using the app because of external factors and leave it open, thus producing biased data.

Field trials vs. lab: even though it seems running experiments in uncontrolled environments could be the most reflective way of a natural usage of mobile devices, some studies still assert that it is often the wrong tool to be used. As argued by Kjeldskov et al. [62], field trials are labour-intensive and provide little added value to traditional experiments. Brown et al. also raise the objection that often field trials are used to prove points that would be better demonstrated theoretically or experimentally, because they search for technical aspects and no experiential concept has to be proven [17].

Still in the dawn: the application of gamification to several research fields, as already mentioned, is still fairly unexploited. The dearth of exhaustive theoretical frameworks to quantitatively evaluate gamification findings, cause a lack of comparable results for future studies, as shown in recent work by Hamari et al. [43]. Indeed, most of the studies conducted until now, have shown qualitative analysis of data, and poor standardized methods application in experiment, hence result comparison, and experiment replication is difficult, causing the lack of a strong basis for future studies. A good set of guidelines and standards should be developed, to be offered to future studies in order to extend existing findings.

Experiment duration: as noted by Henze et al., not only the quantity of the total collected data is important, but also the quantity of data collected by single users [46]. As observed in the study, apps are often used for short periods from each users and this makes them not suitable for researches that need long-term engagement. In this case, much more attention has to be paid to the App development and possibly controlled experiments should be considered instead. On the other hand, for apps producing immediately consistent data, it is important to start as soon as possible to collect data. When using this approach, it is also challenging to collect so large amount of data, and the experiment could possibly last longer than a lab study.

2.3 Reachability problem

As mentioned in previous chapter, there is still a lack of sets of structured mobile usability guidelines, though it does not mean guidelines are sparse.

In this study, issues concerning the different aspects of the reachability of the smartphones' screen has been chosen. The study of this problem involves guidelines pertaining:

- the best target position on screen.
- the best target size on screen.
- the influence of context of use—i.e. environmental conditions, user sitting, standing or walking, users encumbrances.
- the influence of the screen size.
- the influence of device grip—i.e. how many hands and which fingers are used to hold and perform gestures.
- the influence of different type of gestures—i.e. tapping, scrolling, circling and so forth.
- the implementation of different techniques to make the screen totally reachable with only one hand.

Generally all these guidelines have the aim of maximizing user performances in terms of speed of use, accuracy, and minimize the error rate, as usability metrics.

The touchscreen has become the dominant input technology for smartphones and other personal devices, such as tablets and, increasingly, laptop computers. Because of this pervasiveness, many efforts of research have been put on understanding users' behavior, touchscreens ergonomics, and interface design paradigms. If in earlier years it was generally accepted that smartphones could prevalently be operated one-handedly, using the preferred thumb, nowadays many other postures and operation modes must be taken into account when designing mobile interfaces, both because of the popularity of bigger screen sizes and the proliferation of touch gestures that require more than one finger (e.g., zooming in and out with the "pinch to zoom" gesture) [50].

In the last ten years the average screen size of touchscreen smartphones has almost doubled, passing from the average 2.59 inches of 2007, to 4.86 inches in 2014. Average screen size has increased by one inch (from an average of 3 to 4) in the five years between 2007 and 2012, but it took just two years

from 2012 to 2014 in order to reach an average of 5 inches. Moreover, over the last two years, only few newly released smartphone models have had a screen of less than 5 inches [7].

Because of this evolution, it is still important to get a deeper understanding of users' touch behaviour, but it is necessary to evaluate if existing findings and guidelines are still valid, or they need to be updated, and expanded.

Actually, several studies investigating and modeling user's touch behavior have been conducted, however many of them are still based on smaller devices and often concentrate solely on one-handed operation of the smartphone. To the best of our knowledge there is still a lack of studies that take into consideration the many possible device grips, both with preferred and non-preferred hands, on modern screen sizes.

Hence, the aim of this work is to use the gamification approach to investigate the reachability problem, in order to determine how promising the approach is, if to be applied to other usability problems. By means of a game application, the reachability problem has been studied, trying to understand if (1) existing guidelines are still valid, (2) new guidelines, specifically designed for different conditions of use should be considered, (3) the validity and generalizability of the used approach, as a means to answer the first two questions also for other usability issues.

Henceforth, after a brief overview on the evolution of smartphones' screen, a review of the literature investigating the reachability problem is presented.

2.3.1 Smartphones evolution

The first (handheld) mobile phone can be dated back to 1973, when Martin Cooper (considered the father of the cell phone), working at Motorola, demonstrated a call with a prototype of DynaTAC, made commercially available more than ten years later [27].

In the mid 80s, another milestone in mobile phone evolution has been put by the emergence of PDAs (Personal Digital Assistants), basically mobile phones with email and fax capabilities, mostly used by businessmen. These were precursors of modern smartphones: they were generally equipped with resistive touchscreens, which rely on resistance, hence only the applied pressure causes the phone to respond.

This touchscreen technology admits only a single point at a time to be sensitive, can be operated with any device different from a finger, and it is

generally more usable when using a stylus. Indeed, the first smartphones were not so strongly relying on finger touch and were provided also with a physical QWERTY keyboard [57].

A resistive touchscreen is composed of several layers, but basically there are two main layers separated by a gap, that are activated when the applied pressure causes the two layers to touch each other. Because of this composition, resistive touchscreens can be slower to react to the touch and cause the display to be less sharp, with lower contrast.

However, the dawn of modern smartphones has come in the early 90s, when, in 1993 IBM launched Simon. Simon is generally renowned to be the first smartphone in history: besides its ability of sending and receiving phone calls, email, faxes and pagers, it was equipped with some *applications* for different purposes—e.g. address book, calendar, appointment scheduler, calculator, world time clock, electronic note pad, handwritten annotations.

Nevertheless Simon was still a so-called *PDA-phone*, until 1997, when Ericsson used the term smartphone for the first time [75].



Figure 2.2: IBM Simon

In order to observe a great spread of smartphones, on the other hand, also mobile network evolution must be considered, as a fundamental support for the completeness of this technology to be mature.

Until the 1991, with the advent of 2G (digital) networks (GSM), mobile phones were working with analog technology. Ten years later, in 2001, the availability of 3G networks, which offered much improved capabilities of bandwidth, and hence speed, gave the smartphones' commoditization a great

push.

Nonetheless, time was not ripe until 2007, almost ten years later: until that time, smartphones still remain enclosed in work environments, mostly used for job tasks by business people.

In 2007, also ten years ago, twenty from the dawn of smartphones, Apple Inc. introduced the iPhone, transforming smartphones in a consumer desire: the primary task for smartphones' holder moved from communication/organization to entertainment.

One of the greatest advantages of the iPhone on its competitors has probably been its more performant, capacitive, multi - touchscreen: this—not so new—technology allows to have more than one sensitive point at a time and has a much lower response time when compared to the resistive ones.

Moreover, the image quality, in capacitive screens is much higher, because of the increased sharpness, speed and accuracy are much more improved, and, most of all, no clicking device is necessary beyond one or more fingers [57].

Henceforward, touchscreen technology has become the predominant one for smartphones, and it has been continuously evolving, in terms of screen dimensions, and touch capabilities. The pleasantness of use has been increasing over time, and it is used in the largest part of smartphones, making necessary to study its impact on usability of all devices in which it is used.

2.3.2 Literature review

Several studies have been conducted in the years to model human touch behavior on smartphone screens, depending on various factors, such as type of operations, target size, and location. Following these studies, algorithms to compensate touch offsets and techniques to facilitate operations across increasingly large screens have been developed.

Few works can be found that take into account the different device grips used to hold smartphones: most part of the studies concentrate their efforts on one-finger, and in particular one-thumb operations. However, given the aforementioned increase in average screen size found on modern smartphones, different device grips and postures must be taken into account. In

fact, no posture in particular can be said to be the prevalent mode of operation adopted when operating a smartphone, giving all different grips equal importance [50].

Moreover, it has been shown that field studies are relatively uncommon in HCI research. According to Kjeldskov et al., there is a reluctance to invest in long-term experiments that take a long time to evaluate and implement [62]. This clearly favors studies in artificial settings with laboratory based approaches, at the expense of studies conducted in a natural environment and focusing on real world usage, with data from a large population of uncontrolled test subjects.

2.3.2.1 Understanding one-handed touch behavior

A large strand of research has been dedicated to studying one-handed/one-thumb operations, originally thought to be the most common ones in modern smartphones.

Parhi et al. investigated the recommended target size for one-thumb operations on small touchscreen devices, distinguishing between discrete (e.g. single-target pointing tasks, such as radio buttons, check-boxes etc.) and serial (e.g. tasks involving a sequence of taps, such as text entry) operations. The collected results determined that targets of 9.2 mm and 9.6 mm respectively could be suitable measures on such devices, without implicating degradation of performance and preferences [102]. Previous works have been conducted on PDA, with different kind of hand postures and were not suitable for smartphones touchscreens, especially thought to be used without the older stylus for PDAs.

Similarly, Park et al. have explored the effects of target size and touch location on a PDA (HP iPAQ) with a screen resolution of 240×320 pixels, on one-handed thumb input. Three different target sizes and 25 locations have been tested on 30 right-handed subjects (25 of which never used a touch screen device), testing the task of clicking a target appearing randomly in one of the designated key location. Results confirm that tapping performance decreases as targets become smaller: targets below 4 mm in diameter showed the lowest performances in boundary regions. The authors observed the distribution of target hits (in terms of shift on the x and y axes) in relation with the screen area and derived a correction function to adjust the touch location and thus improve touch performance [103].

Another study by Perry et al. investigates the need of: (1) evaluating thumb tapping performance on touchscreen with both the preferred and the non preferred hand, (2) distinguishing between standing or walking and (3) how target position affects performance. An experiment has been conducted on 40 right-handed participants, half using their preferred hand, half non-preferred one; half walking and half simply standing. Twenty-five target positions and five different target size were tested on the same model of PDA used by Parhi et al.[102]. Data analysis shows that performance is significantly affected by whether the preferred or the non-preferred hand is used, whereas no significant difference has been found between standing or walking (even though the authors observed that laboratory conditions are generally fairly different from use in real world). As other following studies confirmed, targets positioned in the center of the screen were found to be easier to reach, but with less average accuracy [105].

Further studies conducted by Holz et al. have shown that target acquisition is made with the bottom of the users' fingers and not with the central contact area between finger and screen, as the general assumption is, in one hand operations [49]. Thus the authors present a projected center model to minimize error typing, showing performance improvement. In order to understand human touch behavior, four different studies have been conducted: first the user mental model of touch has been investigated; then models to design and evaluate the correct touch behavior have been developed. The proposed model is shown to represent a good approximation of how users proceed on acquiring targets on the screen's device, but it is argued that for better pointing accuracy, systems tracking finger movements using cameras, work better than the traditional systems based on capacitive sensing.

Xiong et al. also studied thumb muscle activity with one-hand posture and found that tapping smaller buttons cause rapid fatiguing of the thumb. The experiment has been conducted on twenty university students, while executing three different kind of tasks—tapping, moving and circling—using an iPhone4. Results show that small target should be avoided as are fatiguing for thumb, whereas certain gestures—which involve complicated movements with greater variations of muscle activity—have been shown to be less tiresome for the user than others, like circling direction, and that the effort strongly depends on the gesture. [125].

2.3.2.2 Interaction techniques

In order to overcome the difficulties of one-thumb operations on bigger screens, many studies have focused on implementing solutions to make device screens fully reachable with only one thumb.

Boring et al. proposed "FatThumb", a technique that makes use of the thumb's contact size as a form of simulated pressure measure. By interpreting the thumb's motion and pressure, this technique makes it possible to use only one hand to perform complex operations that generally require the use of both hands, like zooming in and out on a map [14]. A controlled experiment shows *FatThumb* to be more accurate (on small targets), to have the least number of strokes and to be the subjective preferred one, and has similar results to another famous adopted technique.

In the work by Kim et al. two proposed touch interface extensions were studied: one allowing the user to slide the interface around and thus to move distant targets closer to the thumb, the other providing an extensible cursor attached to the thumb. Two different triggering mechanisms were proposed and the four combinational configurations were studied in terms of performance and accuracy with a Samsung Galaxy S2 (4,3" with a 480x800 resolution [61].

Goel et al. developed "GripSense", a tool that makes use of the smartphone's sensors (accelerometer and gyroscope) and actuators (vibration motors) in order to infer the pressure and the posture used to handle the device: one or two-hands, thumb or index, and whether the device is laying on a table or not [38].

2.3.2.3 Context of use

Apart from understanding one-handed interaction, other studies have been conducted comparing different types of postures and considering the context of use.

In a recent work by Brewster et al., the effects of encumbrance were studied, combined with the three most used device grips (two-handed index fingers, one-handed preferred thumb and two-handed with both thumbs). As expected, they showed that using a one-handed posture entails the highest number of errors in the least reachable areas of the screen, and that the

effects of encumbrance must be taken into account when designing for mobile phones [83]. Experiment was run using a Samsung Galaxy S3, with a resolution of 720×1280 pixels, making users carrying one bag in each hand of 1.6 kg, and asking them to select a series of targets, one at a time on the touchscreen as quickly and accurately as possible, while walking around a pre-defined path.

Buschek et al. have tried to determine whether it is possible to identify a user through their touch behavior and derived some guidelines from this study. For instance they show that thumb usage is more distinct than index finger input, but it is less accurate on average. Also, touches close to the screen's borders are more descriptive of a user's individual patterns [20]. The authors also suggest to avoid unnecessarily large targets, which encourage less accurate touch behavior, and also very elongated target shapes, as such targets lead to less diverse offset angles, reducing pattern individuality.

Further studies on different device grips have been conducted by Azenkot and Zhai [3]. They compared performance of the three main hand postures in typing operations with soft-keyboards for smartphones. In order to choose the device grips to study, a survey was conducted on 75 Google employees, and showed that all the three postures are used and none of them is much more used than others, hence all of them should be taken into consideration. Results show that the two thumbs posture is the fastest one, whereas the one-thumb one is the slowest yet the most accurate in terms of error rate: performances have been evaluated measuring the Words Per Minute-average entry speed resulted in 41 WPM. Entry with two thumbs was faster (50 WPM) than one finger (36 WPM). In fact, measuring error rate (touches outside target), shows a clear tradeoff between speed and accuracy: the faster the posture is, the higher the error rate. Thus the thumb resulted to be the less error-prone than other postures, but no big difference between one thumb and one finger use, but quite surprisingly, no big difference has been found between one-thumb and one-index input performances, although the authors did not distinguish between right and left hand in one-hand postures [3].

A quite extensive study on hand postures has also been done by Music and Murray-Smith: they studied the relations existing between five hand postures and the use of a device while walking. The study has been conducted in a lab with a single device, highlighting that touch performance of

users are quite affected by the use in movement. Also, the index finger was found to be more performing than the thumb [82].

In a very recent study, Zhu and Li added the screen size as independent variable on the performances of one-handed operations, trying to determine to which extent it is possible to increase device size, without affecting the user's comfort. The study highlighted that size has an important impact on performance, measured qualitatively through a questionnaire, whereas the thickness of the device does not. It was found that devices should not be larger than 4.7 inches for best user comfort [127].

2.3.2.4 Crowd-produced data

The idea of collecting crowd-produced data using gamification, through the deployment of small games or applications for consumer devices, is still an emerging trend. A scarce few studies can be found using this approach in the usability field.

The largest—and one of the first—study adopting this technique has been conducted by Henze et al., who collected more than 120.000.000 tapping events through an *Android* game developed to study touch performance of users. The game consists in tapping circles appearing on the screen, with increasing levels of difficulty. Their results confirmed previous studies in user performance and error rate, also showing a systematic offset between touch and target position [47]. A compensation function has been derived in order to minimize the error, and tested by publishing on the Android Market an improved version of the application. One possible subsequent addition to the work, according to the authors, could have been the addition of the device grip as an independent variable, in order to see if and how offset could vary, depending on which hand posture is used.

In a subsequent study, Henze et al. developed another game to measure typing performance and inferred a dedicated compensation function for this type of operations on mobile soft-keyboards [48]. About 48.000.000 keystrokes from 73.000 installations have been recorded from publishing the game in the Android market. A systematic skew, to the bottom on the vertical axis and towards the center of the screen along the horizontal axis for taps on keys that have no adjacent free space next to them, has been observed in the collected data. In order to compensate this skew, the authors tried to influence users' touch behavior: first by automatically shifting

the touch position (using a different compensation function from the one already used in Android systems), then by shifting key labels and finally by showing touched positions using dots. Results were compared using the native shift, the adapted shift function and no shift used, and showed good performances with the adapted shift function.

Another game with the aim of improving typing performance has been developed by Rudchenko et al., however the main study was conducted through a controlled experiment even if the application had been publicly released and installed by several users [107]. Results show that users improved typing speed (words per minute) and accuracy (correct words) while playing. Moreover, showing a map with touch points where users tend to mistype keys, overlaid to the keyboard helped users, according to a post-hoc questionnaire, was shown to be another helpful tool for users. For the third goal, of generating personalized training data for key-target resizing, a simulation has been run using the first 10 rounds played, to predict users' performances on subsequent 10 levels and showed good results.

Chapter 3

Usability game

In Chapter 2 the proposed approach of using gamification, to collect large amounts of data, for the evaluation and development of usability guidelines has been introduced and motivated.

In this chapter, the implementation of *Usability game* is presented. *Usability game* is a smartphone application with the aim of collecting large amount of crowd-sourced data, coming from a heterogeneous population of users, using it in real world settings.



Figure 3.1: Usability game icon - freely inspired by Norman's book cover - The design of everyday things [99], taken from the "Catalogue d'objets introuvables", by Jacques Carelman. In 1969, Carelman published a funny book named "Catalogue of impossible objects", in which he collected a series of impossible, or at least debatable objects, as a parody of the popular and ubiquitous selling catalogues, sent via mail.

In the first release of the application, in order to deepen the knowledge about reachability problem, users are challenged in a simple task—such as tapping targets appearing in random positions of the screen as fast as possible— and information about recorded taps, device used, and user’s posture is collected in a local server, and analyzed.

The application has been published on the *Google Play* store for *Android* applications on 30 June 2016 and made use of the principal game-mechanics in order to attract users.

For the data collection, analysis and visualization, a web-server has been implemented in a local Department machine.

The application design, the used game mechanics, and data collection methods are explained in this chapter.

3.1 Application design

The application developed consists of a series of games focused on the user’s behavior in tapping the touchscreen. In order to find an acceptable trade-off between the user’s engagement and easiness of use, all games are very simple, require straightforward operations from the user, and can be performed in less than a minute. For each game session the user is given 30 seconds to hit the highest number of targets appearing in random locations of the screen. Targets are represented as simple blue circles with varying diameters, on a neutral white background as shown in Figure 3.2. As soon as a target is hit, it disappears and increases the user’s score, in a similar fashion to previous works [47, 103].

Effort was made to ensure that the application could be extended with various games, in order to collect a variety of different metrics regarding multiple aspects of usability and mobile application interface design. To make the application more challenging, in the first release, four variations of the base game were developed:

- *Standard*: the next target appears as soon as the previous one is hit by the user;
- *Delayed*: the next target appears after a random amount of milliseconds from when the previous one is hit;

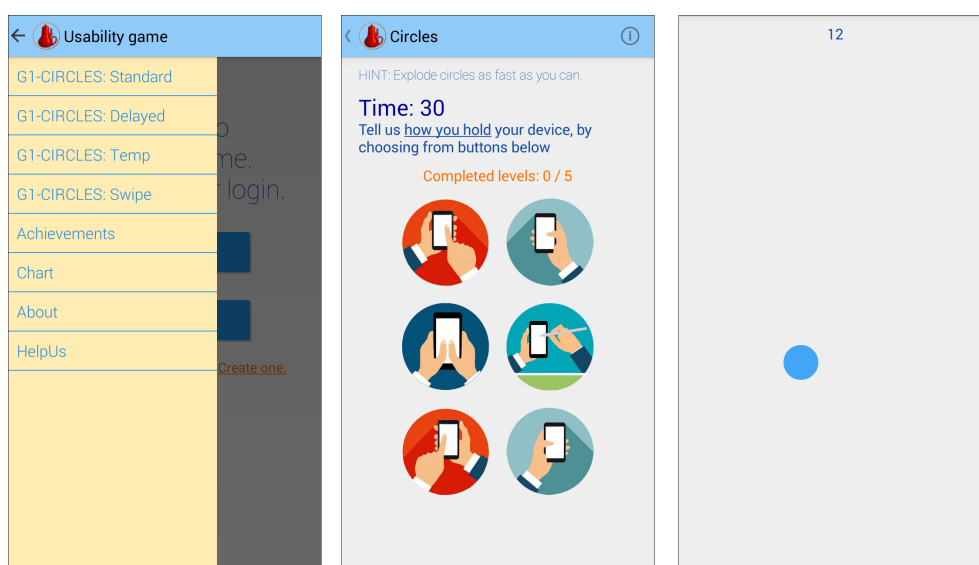


Figure 3.2: Application screenshots: game version selection, hand posture selection, actual game with circle to tap and count of remaining seconds.

- *Temp*: targets remain visible for a fixed amount of time (500 ms), after which they disappear and cannot be clicked anymore; a new target appears;
- *Swipe*: either a double click or a swipe gesture must be used to correctly "hit" a target.

When starting the game, the user is asked which variation of the game they wish to play.

Each time users complete a 30 seconds session of a game, they can move to the next level with increased difficulty. In the first session of each game variation, target circles have a radius of 120 device pixels. At each level, the radius of the target is decreased by 20 pixels, shrinking down to 40 pixels in the fifth and last level of the game. As the diameter decreases, the amount of points assigned for a successful hit increases.

In order to study the relations between screen reachability, screen size and device grip, the application also asks the user to select in which way they will be holding the smartphone during the game sessions. The possible device grips, internally called *game modes*, are shown in Figure 3.3:

- (a) *Right index*: smartphone held in left hand, user taps with right index.

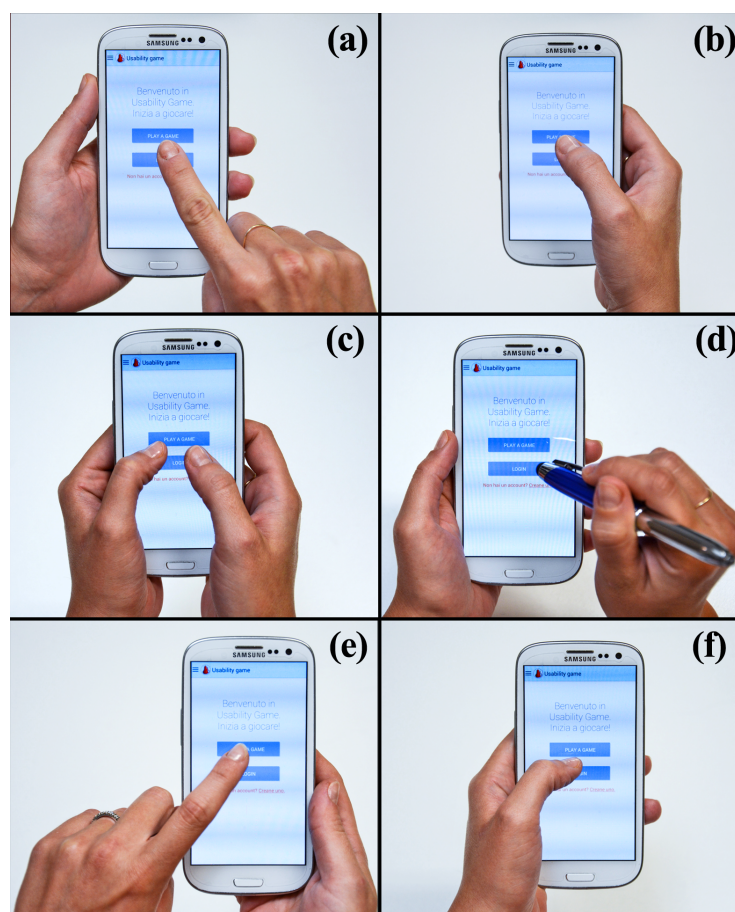


Figure 3.3: The six studied hand postures: (a) Right index, (b) Right thumb, (c) Both thumbs, (d) Stylus, (e) Left index, (f) Left thumb.

- (b) *Right thumb*: smartphone cradled in right hand, user taps with thumb of same hand.
- (c) *Both thumbs*: both hands used to hold the smartphone, both thumbs used to tap.
- (d) *Stylus*: smartphone held with either hand, a stylus is used to tap with the other hand.
- (e) *Left index*: smartphone held in right hand, user taps with left index.
- (f) *Left thumb*: smartphone cradled in left hand, user taps with thumb of same hand.

Even though this mode selection was originally intended to allow users to select their preferred way of handling their device, it was quickly discovered during initial testing that the different device grips were often interpreted as a further challenge. Many testers did report of trying out multiple different grips in order to compare their own scores in different game sessions. Hence, it can be assumed that users have been using both preferred and non-preferred hands when using the game. Like in previous studies, it has been observed that for some operations users choose not to use their preferred hand or grip. [105]

The white background of the game was chosen because we didn't want to affect the users' touch behavior with distracting optional elements on the screen. Unlike in the previous study by Henze et al., where colored and animated backgrounds were used to make the game more appealing, we wanted users to focus only on the targets and on their task [47].

The app is localized in English and Italian.

3.2 Game design elements

In designing the previously illustrated scoring system, levels, game versions and game modes, *gamification* principles—defined as “the use of game design elements in a non-game context” [30]—have been taken into account, both to better encourage users to try the app and to keep them engaged for a longer time.

3.2.1 Identity

The application gives players the possibility of optionally creating their own identity in the game, which collects the scores gained while playing.

User can choose a fictional username, a password and an avatar picture, and at every login, user's information regarding the game is loaded.

In order to keep a good trade-off between game incentives and privacy aspects, no personal data about users is collected, neither in the application nor on the server.

A random Universal Unique Identifier (*UUID*) is generated when the application is run for the first time. This Identifier (*ID*) is used by the server to distinguish between different devices running the game, but is never linked

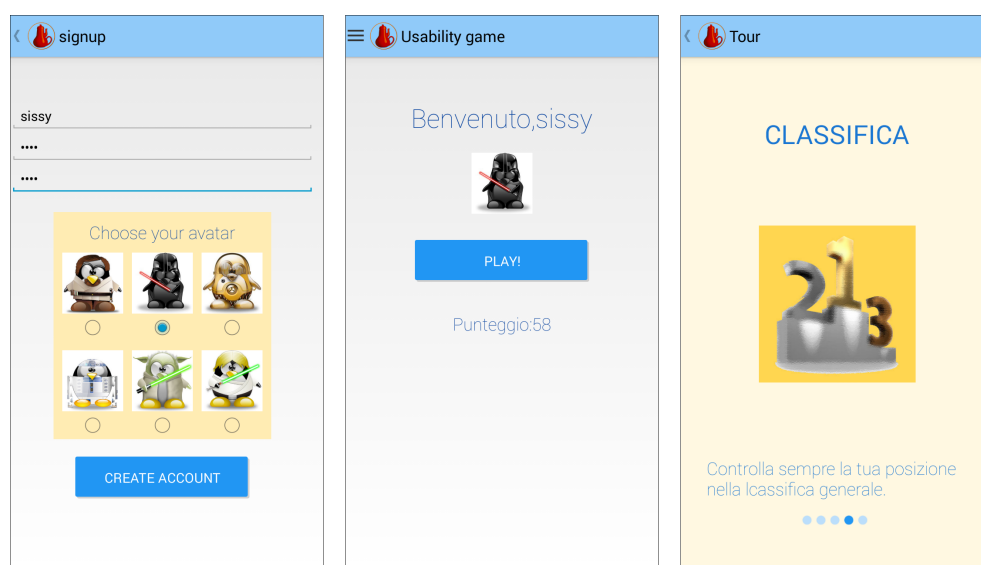


Figure 3.4: Various screenshots: the sign up page, welcome page for a registered user and one step of the initial explanation.

to user identities which are not stored on the server. Only registered users have UUID and username linked.

At the first login—and at the first launch of the application—a brief introduction to the game is given to the user, in order to guide them through the app, see Figure 3.4

3.2.2 Points, levels, achievements and scoreboard

A point system is implemented in the app: points are gained for each correct tap, depending on the level played. Levels are of increasing difficulty, as previously explained, and the higher the level, the more points are assigned for each tap. At the end of each game session of 30 seconds, the player can choose to advance to the following level or to play again the same level, to improve their results.

The score is saved in the app, and it is retrieved, and shown at every user access.

The application keeps track of the user's high score and assigns a *medal*, ranging from the "metal medal"—the lowest reward—to the "gold medal"—the highest reward—depending on the threshold reached (see Figure 3.5).

The score is also used to build a general scoreboard, that can be seen

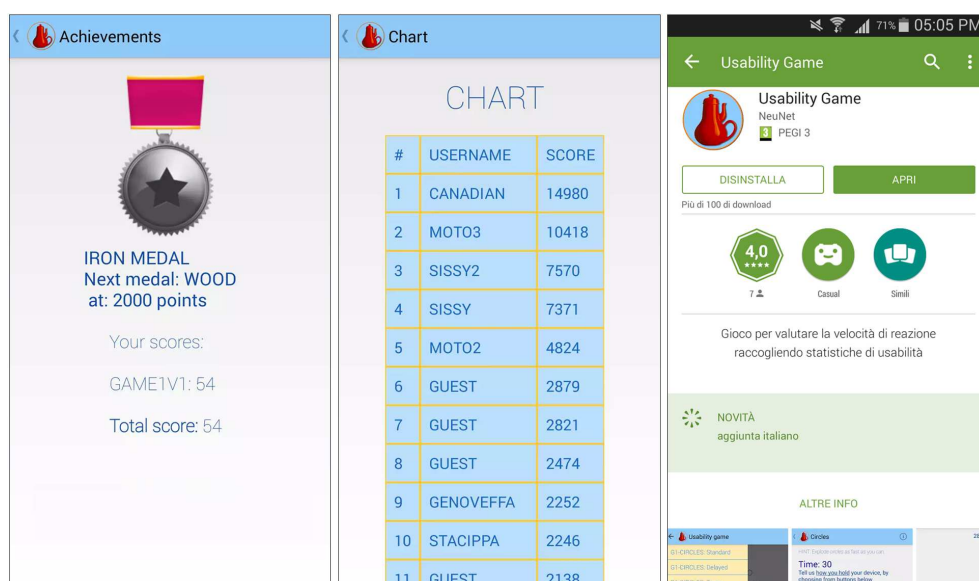


Figure 3.5: Various screenshots: the registration page, user profile

directly in the application, see Figure 3.5, or through the web site running on the server.

As it is possible to switch identity locally on the device, letting multiple users play on the same device and be recognized as different players by the server and the scoreboard. If the user chooses not to create an own identity instead, they are identified on the scoreboard as a generic "Guest" player.

3.2.3 Bonus

Extra points can be gained by claiming a *bonus*. At the end of each game session, users are asked whether they want to gamble for more points if they are confident of having performed better than average.

If they accept, the user's average time to hit a target for the current session is computed for each tile of the screen divided up into a matrix of 5×5 . Results are transmitted to the server and compared to the global average for each tile: if the user's timing is better, bonus points are gained.

Otherwise the score is decreased, possibly generating a negative bonus and decreasing the user's score.

In Figure 3.6 it is shown the user claiming their bonus: highlighted areas are the ones in which they performed better than the average.

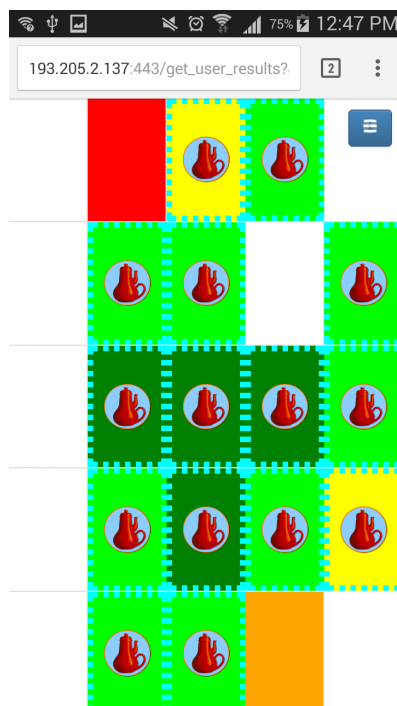


Figure 3.6: A screenshot of a user claiming bonus.

3.3 Data collection - The web-server

In order to collect, analyze and visualize data sent by apps, a web-server has been implemented and is running on a local-machine in the Department of Pure and Applied Sciences at the University of Urbino.

The server is implemented in Ruby-on-Rails [u17], a Model-View- Controller architectural pattern, that makes the development of web applications, easier and faster, and has good scalable capabilities.

The app and the server communicate through a shared folder in a public folder of the the local machine, and data is stored in a local database.

The server has two levels of accessibility: the frontend, a website publicly accessible, and the backend, a protected area reserved to administrators with proper credentials.

3.3.1 Architecture

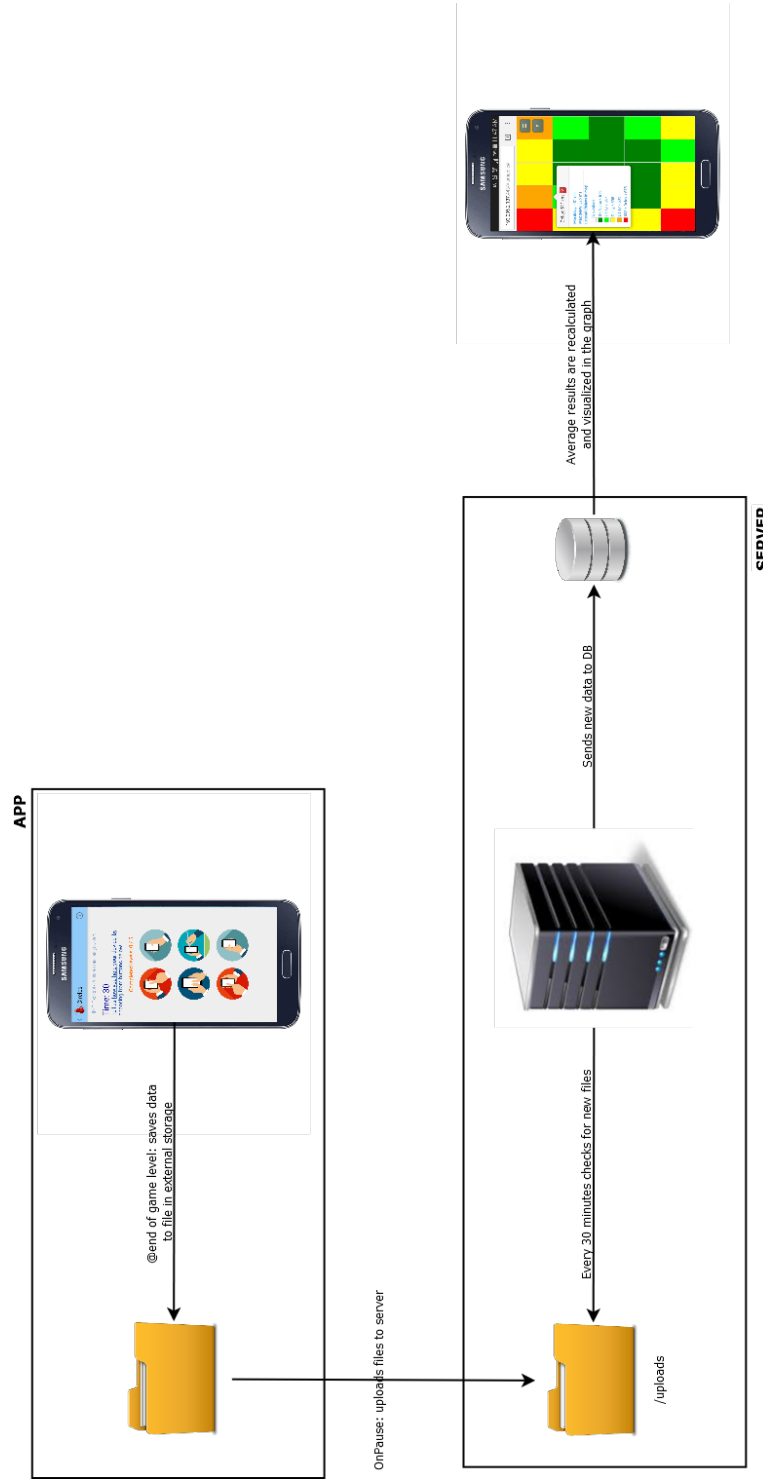


Figure 3.7: The communication scheme between app and server

In Figure 3.7 the way the app and the server communicates is shown.

During each game session of thirty seconds, each tap event by the user is logged in a CSV file saved on the user's smartphone, in the external storage Documents folder.

A different file is used for every different version of the game.

As soon as a wi-fi connection is available, when the app is in a paused state, all the stored files with log data are sent to the server.

In the same folder, a CSV file for the global chart is stored: whenever the log files relative to the game sessions are uploaded, the app downloads an updated chart file from a public location on the server.

Later on, every 30 minutes, the server checks the shared folder for new files, and imports the data in a database, updating the data visualizations.

3.3.2 Database

The structure of the database is fairly straightforward and composed by the minimum set of useful information for data analysis and website implementation:

- `admins`: a table for admins credentials
- `gameN_results`: storing data about different games (a single table is used for the different four versions of the *Circle game* is used)
- `general_scores`: storing data about users' score, keeping the newer ones each time a user uploads data of new game sessions. Users are uniquely and anonymously recognized by a UUID and a username, if they have created an account inside the app.
- `user_data`: for data collected with the optional form about some personal information about users.
- `avg_results`: a table for a fast visualization of the home page. Every time the server imports new data, data regarding the visualization of average metrics (delay, accuracy, and error rate, delay standard deviation, number of points per tile) for the basic 5×5 grid is recalculated.

For each tap event, data about the X and Y coordinates of the effective tap, X and Y coordinates of the target center, two timestamps (one marking the effective tap event, one marking the appearance of the target on screen), and the *target size* (in terms of pixels of radius) are stored.

The selected *game mode*, the *game version* and information about the device (*screen width* in pixels, *screen height* in pixels, *screen size* in inches, *device name*) are saved as well at the beginning of the game session.

For each *correct* hit (i.e., if the coordinates of the tap are included in the area of the target), the number of previous *misses* (i.e., the amount of taps outside of the area) is also recorded.

For the *Swipe* version of the game, also the *event type* (double click or swipe) is stored.

3.3.3 Web application - server side

The server is accessible on a website ¹ and has separated views for users (front-end) and for administrators (backend).

3.3.3.1 Front-end

The front-end website has the aim of displaying an overview of the collected results to users. Data can be freely accessed and downloaded as a CSV file.

Several sections are available, showing heatmaps with the average tapping time (in milliseconds), accuracy (distance in pixel between the center of the hit circle and the center of the tap), and error rate, separated by screen region. Sections with the global chart and with the explanation of the website are also available.

Data about tapping speed, accuracy, and error rate are visualized in heatmaps, divided into colored **quadrants**. For tapping speed and accuracy a classic color scale, ranging from red (highest delays) to dark green (smallest delays) has been used, whereas for error rate ten colors are used, ranging from dark to light red; for all parameters, white areas indicate the absence of taps in those areas. An example of the tapping speed heatmap is shown in Figure 3.8

Data visualizations can be filtered by the user through specific options, restricting the data to a specific session, game mode, screen size, or other parameters.

Colors applied vary dynamically, according to the applied filters: depending on the filters applied, the maximum and minimum values of the delays are different, so the delays interval relative to each color vary according to these 2 values.

¹ <http://193.205.2.137:443/>

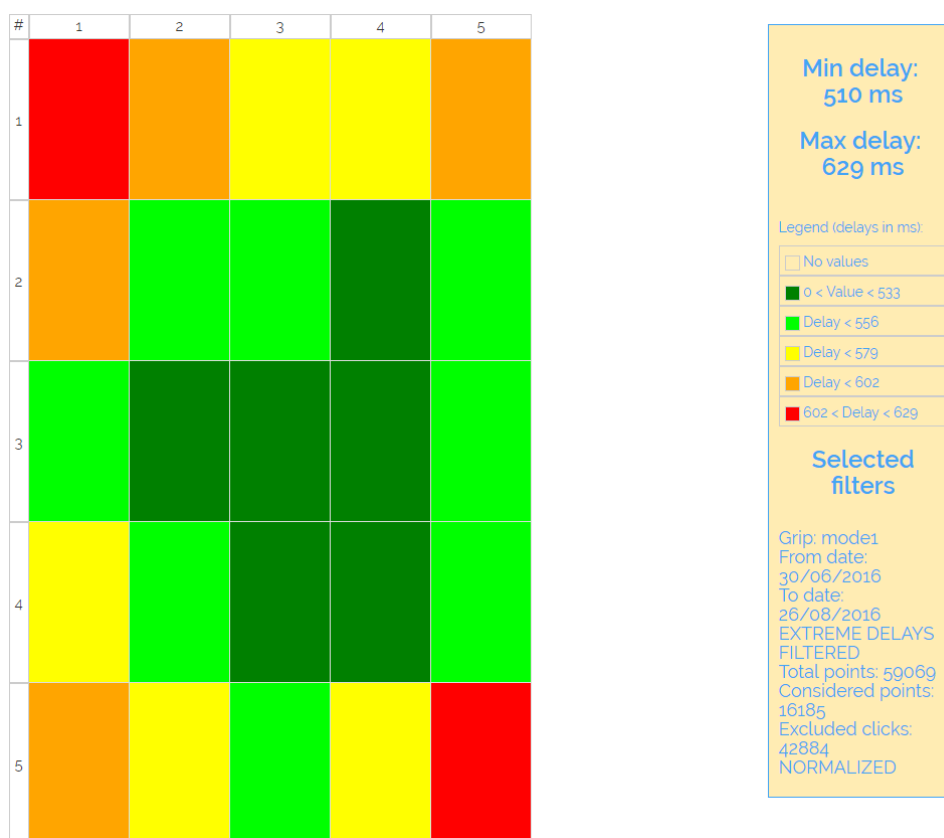


Figure 3.8: The heatmap representing tapping speed values with legend: exact values are displayed when the user clicks on grid cells.

Hide/show filters

Filters

Apply

Quadrants #

10 ▾

Normalize

Normalize

Screen size

All sizes

3.5 - 4.0 in

4.0 - 4.5 in

4.5 - 5.0 in

5.0 - 5.5 in

> 5.5 in

Version

All versions

V1

V2

V3

V4

Device grip

All modes

Mode 1

Mode 2

Mode 3

Mode 4

Mode 5

Mode 6

Level

All levels

Level 1

Level 2

Level 3

Level 4

Level 5

Release date

Date from:


Date to:

Session ids (comma)

Exclude extreme delays

Exclude

Device grips legend



MODE 1 MODE 2

MODE 3 MODE 4

MODE 5 MODE 6

Figure 3.9: The possible filters.

Several possible *filters* can be applied, as shown in Figure 3.9, hence varying the resulting heatmap:

- *quadrants number*: the number of rows (and columns) into which the grid is divided, from a minimum of 5 to a maximum of 50, in order to guarantee an acceptable time for the re-drawing.
- *normalized*: if the coordinates of the displayed clicks are normalized according to the maximum values of coordinates or they are visualized with their original value.
- *screen size*: the size (in inches) of the device that has produced the data.
- *version*: the version of game *Circles* used (*v1* → Standard, *v2* → Delayed, *v3* → Temp circles, *v4* → Swipe circles)
- *game mode*: the chosen device grip (*mode 1* → right index, *mode 2* → left thumb, *mode 3* → both thumbs, *mode 4* → stylus, *mode 5* → left index, *mode 6* → right thumb)
- *level*: for a particular level.
- *session id*: data relative to one or more game session
- *date from* → *date to*: to specify a time range

When the user chooses the preferred parameters, the heatmap is re-drawn according to the resulting average values, calculated with given data.

By clicking on a certain quadrant, and then moving the cursor on the heatmap area, additional information is available on the selected quadrant: the exact value of the parameter, the number of points considered in the quadrant, and the standard deviation are shown in a bubble.

A mobile version of the website is also available, shown in Figure 3.10

3.3.3.2 Backend

The backend is a protected area for admins, accessible with proper credentials: it is similar to the frontend, but it makes additional views and operations available.

It has been made protected in order to maintain a clean version for users.

Admins can access raw data about games and visualize them in a tabular version. Sections with user data, collected with the optional form in the app, and with statistics concerning the number of collected data are also available.

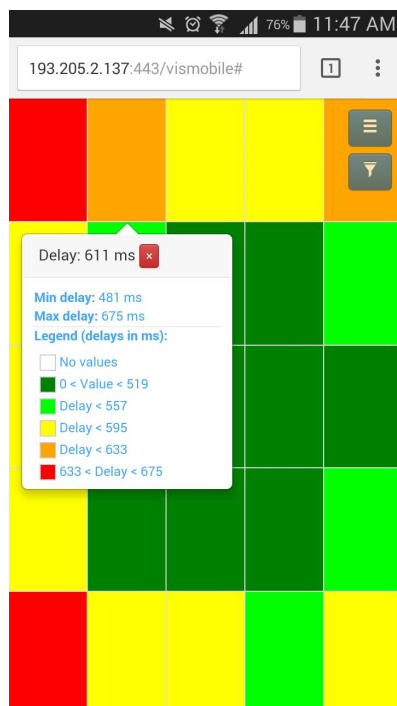


Figure 3.10: Mobile version of the user website

Moreover administrators can force the data import, bypassing the automatic import performed every 30 minutes, and export the chart, creating the relative CSV file downloaded by the app.

Chapter 4

Results and discussion

In this chapter, some of the preliminary data collected over two months through the app is shown and discussed.

Data analysis is focused on (1) comparing obtained results with existing ones, regarding target size and position, and how user speed and accuracy are affected; (2) understanding how screen size and device grip affect user's performances; (3) understanding the validity, generalizability, strengths, and weaknesses of the used approach.

4.1 Deployment and usage

The application was initially tested by a small number of beta testers, recruited through the student mailing list of the Department of Pure and Applied Sciences at the University of Urbino and then published in the Google Play Store on June 30th, 2016.

Data was collected at the end of August 2016, gathering the results of approximately two months of public usage.

A total of 59.530 user interaction events have been recorded, from 134 *Google Play* installations and 1.587 unique game sessions.

Of the total amount of data collected, interactions with abnormally high delays or error rates have been removed in order to filter outliers. In particular, touch events with a delay lower than 100 ms have been considered to be fortuitous correct taps, since the human visual reaction time alone should account for delays of at least one hundred milliseconds between single taps [116]. Events with a delay higher than 3 s have been considered

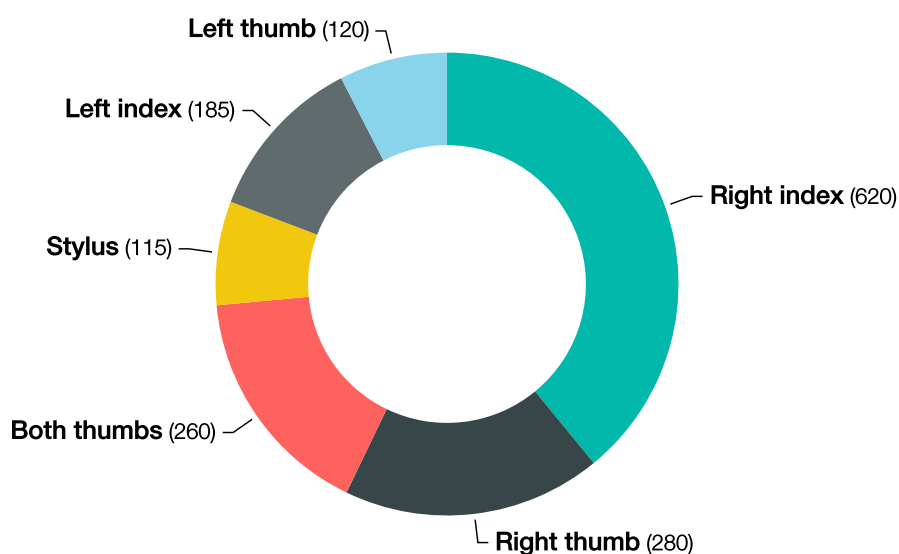


Figure 4.1: Distribution of game sessions by game mode.

as part of unintentionally interrupted game sessions and thus removed as well, as typical timings, according to the collected data, range from 600 ms to 1.2 s. This filtering step removed a total of 929 data points, amounting to approximately 1.5% of the original raw data collected.

The 1,587 game sessions are distributed among the 6 different game modes as shown in Figure 4.1: the large majority of users did use the first mode, i.e. left hand grip and right index tap. Both right hand modes correspond to more than 57% of sessions together, while left hand only modes cover slightly less than $\frac{1}{5}$ of usage. As could be foreseen, users rarely did play using a stylus. The preference for right hand grips matches the expected usage for predominantly right-handed players, however it is at least in part imputable to the order in which grip modes are presented: both right-hand modes were always presented as the first choices, thus possibly influencing users in using those modes before losing interest in the game. Presenting modes in random order could be an option to mitigate this issue, but it could possibly annoy returning users.

In Figure 4.2 the distribution of collected data points by device screen size in inches is shown. As can be seen, more than 93% of game sessions have been recorded on “modern smartphone”-class devices (i.e., devices with screens sizes ranging from 4.3 to 6 inches). A limited amount of users have used either devices with very small screens or tablets (screen larger than

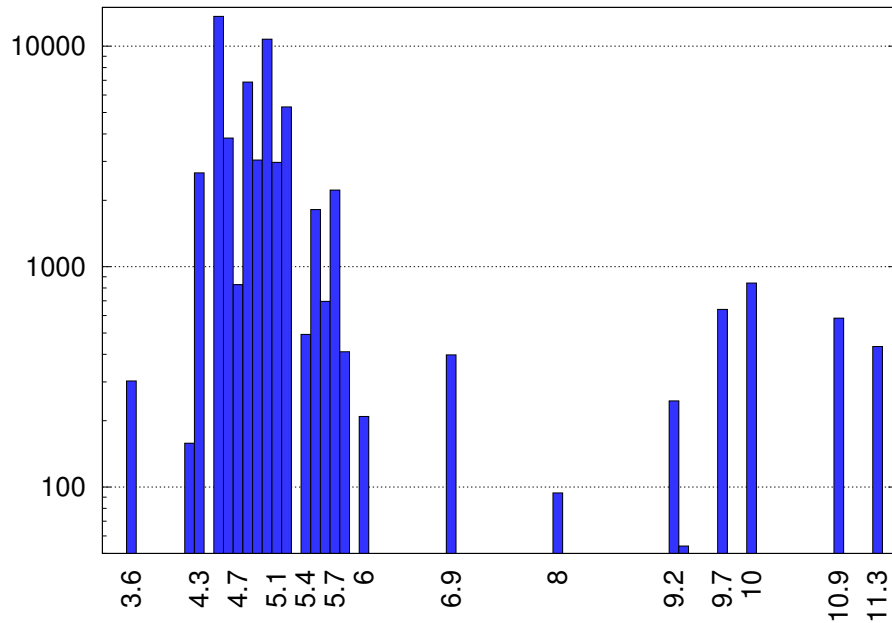


Figure 4.2: Total collected data points by device screen size.

8 inches).

Results shown in the next section have been accordingly separated between smartphone and tablet devices when deemed significant.

4.2 Evaluation

The aim of this work is to deepen the understanding of the effects of device grips and screen size on user touch performance, in terms of speed and accuracy.

Touch interactions collected through the application contain information about the game mode, the grip used by the user, the device, touch timing and coordinates. This work is focused on two main performance metrics: tapping delay and distance from target.

Tapping delay is the time interval in milliseconds between two consecutive target hits in the same user session. This value is used as the reciprocal of the user's *tapping speed*.

Distance from target is measured as the euclidean distance, in millimeters, between the user's actual touch location and the target center. This

value is also used as an inverse indicator of the user's *tapping accuracy*. Lower distances between these two points imply higher accuracy, and vice versa. Even if the target's center is not always the exact point that users are intentionally trying to hit, this accuracy metric is widely used in existing literature [3, 82].

4.2.1 Device grips

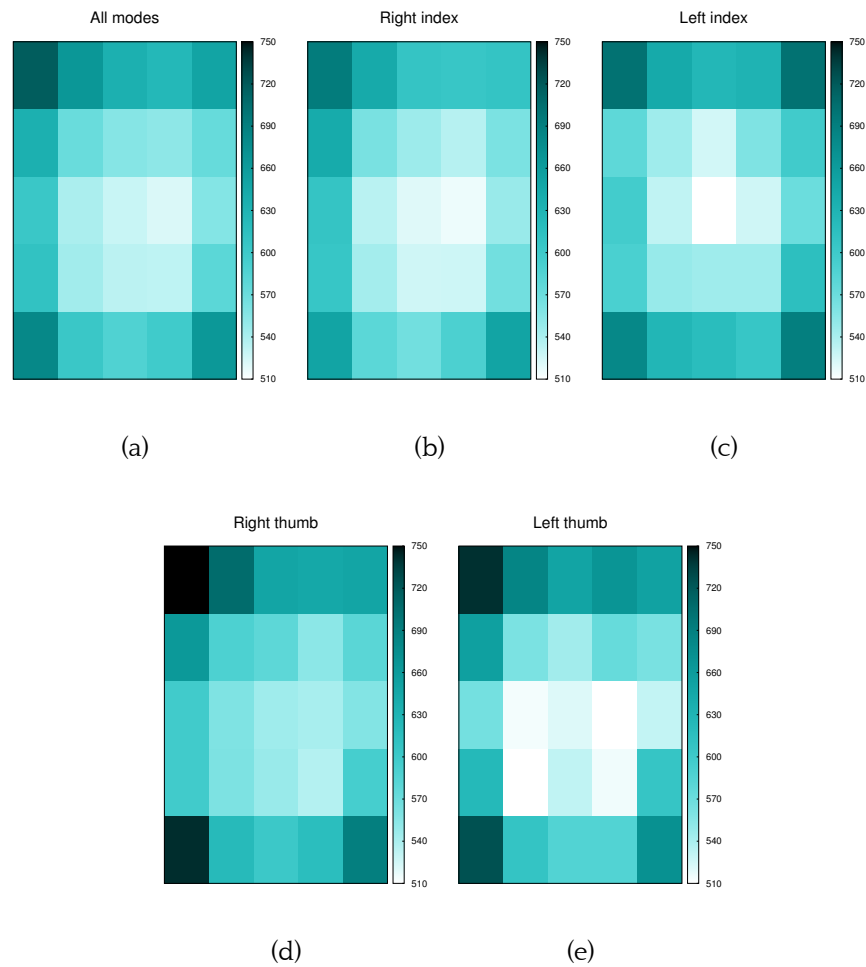


Figure 4.3: Delay heatmaps (milliseconds between subsequent correct taps). (a) Cumulative, all game modes; (b) Right index, mode a ; (c) Left index, mode e ; (d) Right thumb, mode b ; (e) Left thumb, mode f . See Figure 3.3 for game modes.

In Figure 4.3 tapping speed is shown as a set of heatmaps, representing the average delay between taps that have been registered in different screen regions. The screen has been divided in a 5×5 grid and, for each region, the average tap delay is represented using a color scale: lighter areas indicate lower delays, whereas darker ones represent higher average delays in that screen region. All heatmaps share the same color scale, which has been defined including the minimum and maximum delay values in all modes, for all quadrants: 499 ms and 771 ms respectively. Since the data collected stems from various devices with a wide range of screen resolutions, the effective tap coordinates were normalized based on screen size and are represented with coordinates ranging from 0 to 1 (where the point $\{0, 0\}$ is located in the top-left and $\{1, 1\}$ in the bottom-right of the screen).

In Figure 4.3a the *average delay* for all collected data in all game modes is shown. Validating the results by Perry et al. [105] and Henze et al. [47], users generally achieve higher tapping speeds towards the center of the screen, while targets close to the borders—especially the top and bottom margins of the screen—require more time to be hit. Heatmaps showing average delays for thumbs mode in Figures 4.3d and 4.3e indicate that the thumb grip enables far less reach towards the opposite borders. Moreover, it can be observed that thumb-modes (both left and right, mode *b* and *f*), are slower than other one-finger postures (mode *a* and *e*), as also outlined by Azenkot et al. [3]. As mentioned before, it should be noticed that heatmaps for left-handed grips in 4.3c and 4.3e are based on less data than right-handed ones. Data collected in mode *c* and *d* are also too sparse to give a significant representation.

Relationship between *screen coordinates* and *tap delay* is also shown in Figure 4.4, aggregating by *X* and *Y* coordinates respectively. Like in the previous figures, the average delay for all game modes is higher towards borders, both on the *X* and the *Y* axis. Left-handed grips have slightly higher delay values on the right side of the screen. Vice versa, right-handed grips have worse performance on the left side on average. Similarly, thumb modes report slightly higher delays on the whole screen surface, especially towards the top border ($Y = 0$) which is harder to reach because of the grip.

Overall delays by game mode, averaged on the whole screen surface, are shown in Table 4.1. Results for right-handed usage are characterized by

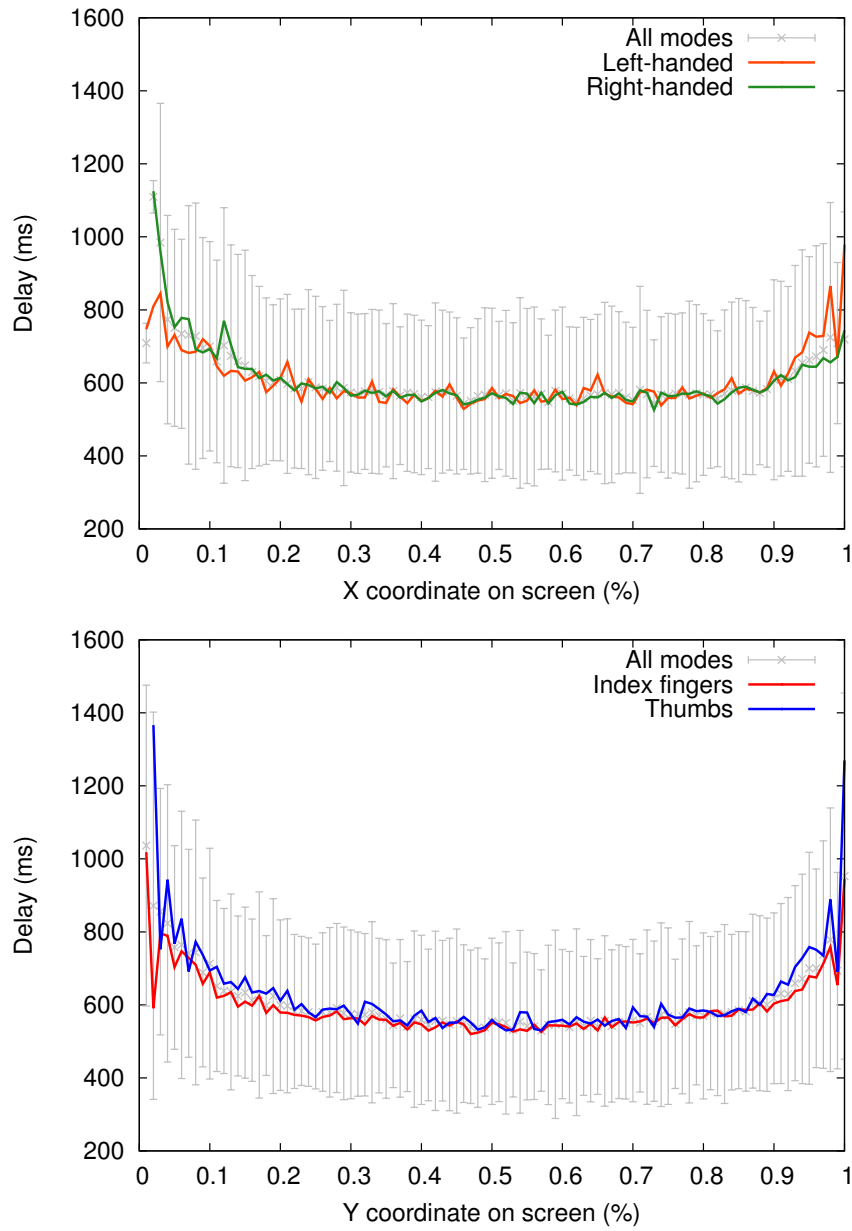


Figure 4.4: Average delay by position on screen.

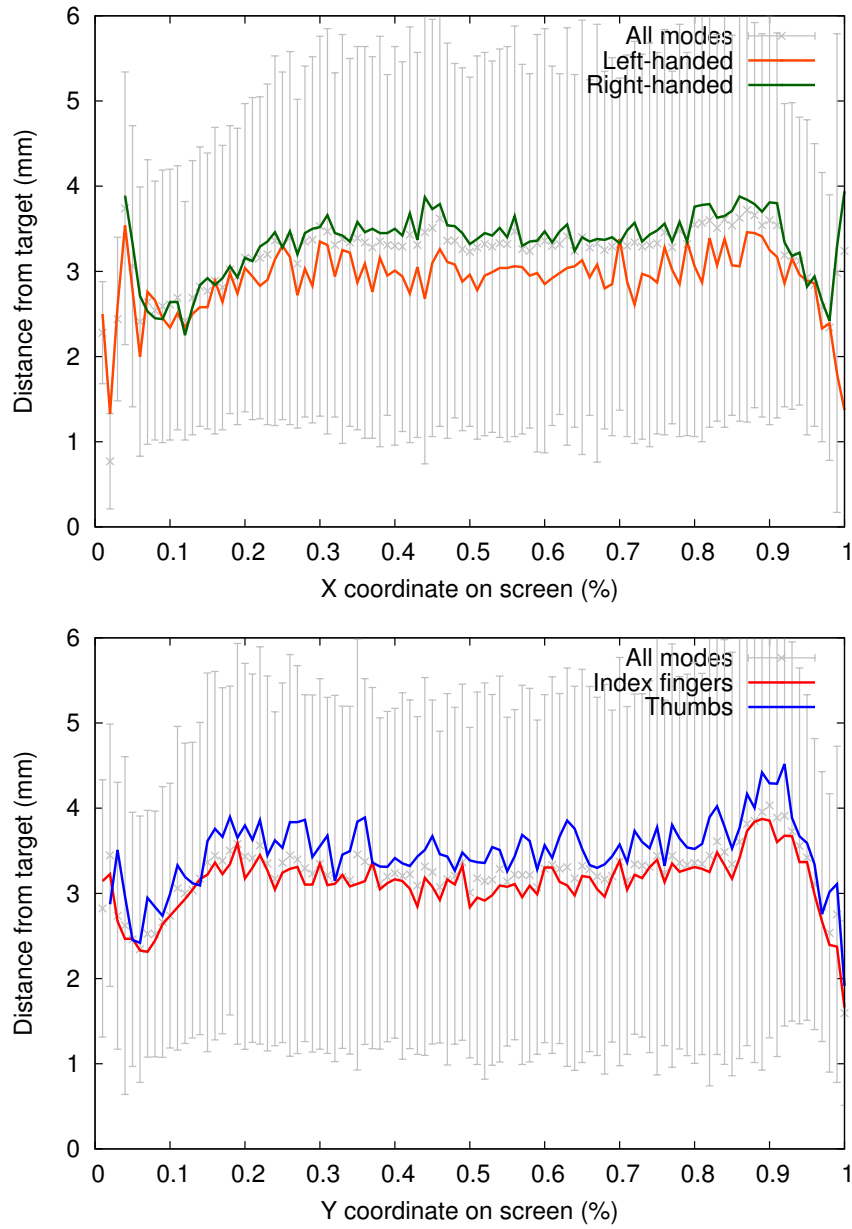


Figure 4.5: Average distance from target by position on screen.

a larger number of taps and show lower delay with the right index when compared to the thumb. The same result cannot be derived for left-handed usage.

In the work by Azenkot et al. it is observed that the two-thumbs posture is noticeably faster than all other considered grips when typing [3]. This observation is not reflected in our results, because of the difference in tasks that the users had to accomplish in the two experiments: when typing, the advantage of using two fingers almost simultaneously is clear, whereas in our game users cannot exploit any finger movement parallelism because of the intrinsically linear nature of the game. Both-thumbs usage has an average performance similar to that of single-thumb postures.

Average *distance from target* is shown in Figure 4.5, also aggregated by X and Y coordinate respectively. These results, when complemented with delay data from Figure 4.4, show that users tap more quickly but less precisely towards the screen center. On the contrary, when getting closer to the screen edges, taps become slower and show less average distance from the targets. Similarly, left-handed usage is generally more accurate, on the whole screen surface. It can be argued that—assuming users are prevalently right-handed—tapping with a non-preferred finger requires more attention by the user and thus yields higher precision, as is the case for tapping in the less comfortable zones close to the screen edges. Also, thumbs are generally less precise than index fingers, for both hands and on the whole screen surface.

These results confirm the conclusions by Buschek et al. in [20]: taps performed using thumbs consistently occur at a slightly larger distance from the intended target when compared to the average precision of index fingers. This is particularly interesting since, based on the average delay for right-handed usage in Table 4.1, thumb operations also show higher delays between taps.

4.2.2 Screen size and distance

In the collected data 25 unique screen sizes have been found, with varying counts of collected interactions as shown in Figure 4.2. Screen sizes can roughly be split into different categories: smaller phones (under 4 inches), smartphones (4–6 in), smaller tablets (7–8 in) and large tablets (larger than 9 in).

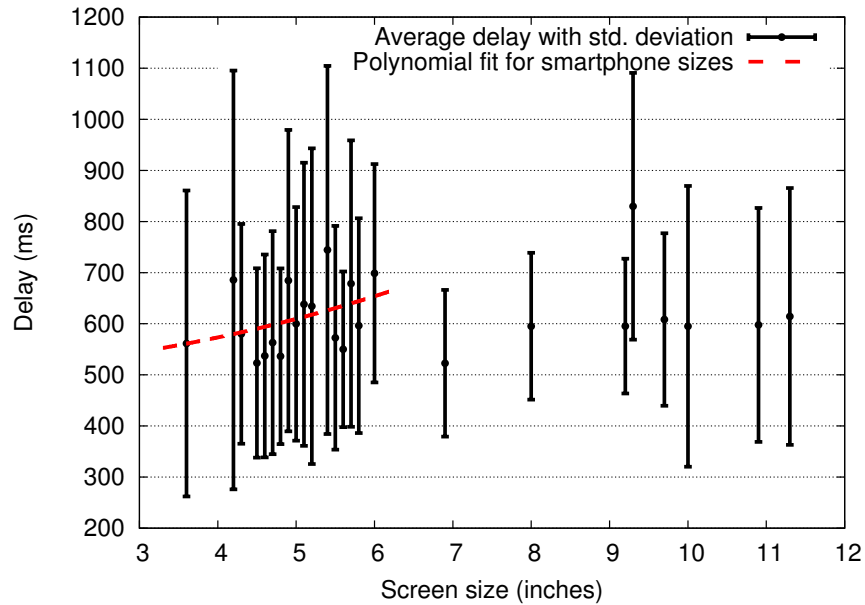


Figure 4.6: Delay by screen size.

Data collected from devices over 6 inches are too sparse to be analyzed, since data for each screen size class is contributed from at most two users. As mentioned before, the game did not take tablet usage in consideration in its design and it did not encourage tablet users specifically.

The impact of screen size on the user's average tap delay is shown in Figure 4.6. When excluding data for tablet-class devices, the data shows that the impact of screen size on the user's average delay between taps is pretty evident. The average tap delay increases by more than 100 ms for 6 in devices against smaller 3.6 in screens. Tap delay and screen size have

Table 4.1: Delay by game mode.

Mode	μ	σ	Taps
<i>Right thumb</i>	600.20 ms	240.79 ms	10550
<i>Right index</i>	569.56 ms	223.22 ms	23018
<i>Left thumb</i>	579.09 ms	247.58 ms	4441
<i>Left index</i>	588.84 ms	240.80 ms	7124
<i>Both thumbs</i>	587.56 ms	257.33 ms	9392
<i>Stylus</i>	605.64 ms	231.96 ms	4532

a Pearson correlation coefficient of 0.151, confirming results by Zhu et al. also showing the positive correlation between screen size and a qualitative performance evaluation by users [127].

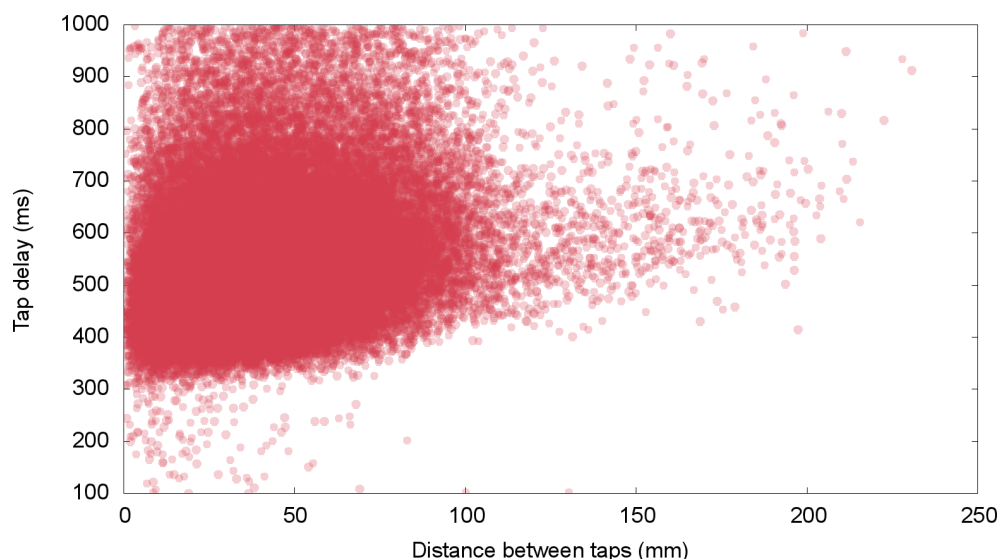


Figure 4.7: Delay by distance between subsequent taps.

It can also be considered that as the screen size increases, the average distance between subsequent targets increases too. As shown in Figure 4.7, this also has an impact on the average tap delay. A substantial linear trend can be seen between the increase of targets distance and average delay: considering target distance between 0 and 100 mm, where the highest part of taps is, the minimum tap delay ranges between 300 and 450 ms.

How target size affects tapping delay and distance from target is shown in Figure 4.8. As target size increases, the average distance from target also increases linearly. It is interesting to notice that the average distance from the target center appears to be constrained to $\frac{1}{5}$ of the target diameter.

Foreseeably, as target size decreases, tapping delay between successful hits increases, both because users need more focus and they incur in more misses. As recommended in several studies [102] and UI guidelines [u18], touchable targets should have a minimum size of 9 mm.

This is reflected in our results as shown by the average delay converging to little less than 600 ms for target diameters of 10 mm or greater.

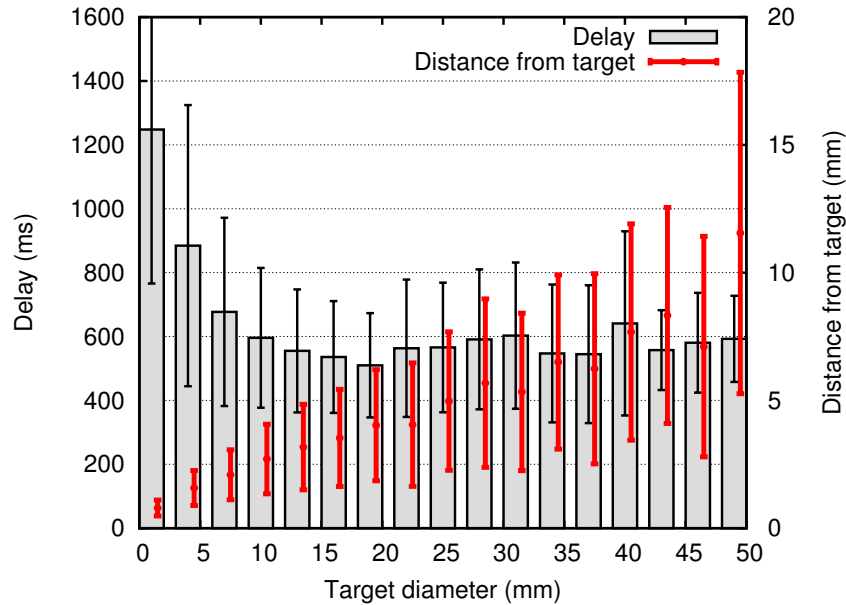


Figure 4.8: Average delay and average distance from target by target size.

4.3 Discussion

The type of data collected and the kinds of analyses performed have a generic scope, which however provide ample opportunities for future research on more specific aspects of mobile application interface design and usability.

Preliminary results collected during the first 2 months of public availability of the application have been presented, showing that, as screen size increases, tapping performance is indeed negatively affected. In fact, average tap delay for screen sizes over 5 inches is greater than 600 ms, the approximate average delay for all game modes and device sizes. Also, results show that the different device grips should be taken into account in user studies concerning touchscreen usage because they show a significant impact on tapping behavior and results based solely on single-handed operation may not reflect real world device usage. Further results that confirm previous studies concern the better performance of the index fingers compared to the thumbs (in terms of speed and accuracy) [20] and the recommended minimum size of touch targets [102].

In conclusion, results show that this data collection approach can be successfully applied to validate existing guidelines or to confirm small-scale

controlled experiments. Moreover, this methodology can be useful to renew and create new UI guidelines according to changes to common human-computer interactions due to the evolution of software and hardware, such as the growing average smartphone screen size. The application of *gamification* principles in designing apps for scientific field studies is promising and should be exploited to a larger extent.

4.3.1 Limitations and future work

In first instance, while this study focused in particular on metrics such as tapping speed and accuracy, different parameters such as error rate, drift, and fatigue could be taken into consideration, possibly adding specific game modes especially targeted for those metrics. Also, the presented application did focus on simple, single touch interactions with a single target, while more complex gestures which range larger parts of the screen or target multiple objects were not taken into consideration.

Also, a lacking aspect of the application's design is its relative lightness in user guidance. Even if users receive textual instructions on *how* to use the game through the game's interface itself, recommendations and examples could be stressed more in order to ensure more coherence in how users perform the tests and thus in the data collected.

Indeed, in open public studies like this, data validity must be taken into careful consideration. As observed by Henze et al., running experiments via apps published in app stores confers good external validity to the collected data, since results can easily be generalized to a wider population of users [46]. On the other hand, internal validity is poor because of the uncontrolled usage conditions of the app. For instance, even though it is possible to distinguish between device grips, there is no information about whether users are using their preferred hand or whether they are holding their phone in landscape or portrait mode, although the application's UI is locked into portrait mode by design. The lack of this information may also be mitigated by stronger usage guidance.

The application did present a section that allowed users to voluntarily disclose some personal information (age, occupation, education, location, etc.). However, since filling out this form was totally optional, very little data was collected.

Given the pressure of the short time limit of game sessions, users may

have privileged speed over accuracy. A future application without a visible time limit could explore if and how results are skewed by time pressure. Also, sessions could be extended to more than 30 seconds to study hand fatigue: while no degradation of touch performance over time has been detected in this study, other studies reported rapid decrease in speed when operating with the thumb over longer periods of time [127].

Finally, in collecting data for devices of different size, more care should be given to how tablets are handled in contrast to smartphones. Some of the postures taken into consideration in this study are not applicable to large screen phones and tablets (thumb-only operation in particular). Also, given that the targets on screen were drawn using device pixels, large-screen devices with low display density would present targets much larger (in terms of millimeters) than the ones presented on smaller devices. The collected data set contains less than 7% of entries from tablets, making the results less significant for that device category.

Chapter 5

Beyond mobile interfaces

In this chapter a comprehensive overview of the interfaces' evolution is given. Starting from a survey on the device hardware changes, following a parallel analysis of the development of the corresponding software interfaces is described.

In subsequent sections a more attentive gaze is reserved to conversational interfaces: these are considered the actual trend in mobile world. These last couple of years, have seen a "mobile engagement crisis": mobile apps stores have seen a drop of downloads and attracting new users, and keeping existing ones have become a main concern of app publishers.

This is one of the reasons that determined the big players of mobile software industry to search for an alternative to apps: last year has seen the proliferation of "bots", chatterbot-like agents with simple, textual interfaces that allow users to access information, make use of services, or provide entertainment through an online messaging platform. Conversational interfaces have drawn the attention of researchers for many years, creating a great amount of studies addressing the subject from its many facets, including natural language processing, artificial intelligence, human-computer interaction, and usability. Nonetheless, a conversational agent which works flawlessly and is perceived as human-like by users is yet to be designed. The new generation of bots seems to overcome the issues of natural language processing by making use of a communication framework that complements the flexibility of free language with structured conversational elements that guide users through the task they want to accomplish.

The second part of this chapter analyzes this trend, pointing out distinguishing features of messaging platforms that kindle this novel approach to conversational interactions. Then it investigates how the next generation of bots can provide an efficient interface to services and data. Lastly, a definition of these new chat-based agents and outlines their distinguishing features are proposed.

The question it tries to address is: will chatbots substitute mobile applications? A discussion about this issue is given and later a discussion about how usability guidelines are approaching these new trends is given.

5.1 Evolution of user interfaces

5.1.1 Physical Devices

Technology has become increasingly pervasive in everyday life, and devices have become progressively more individual and portable. In early 50s supercomputers started to be used in military research and industry (*ENIAC* is generally re-known as the first general purpose, Turing-complete computer in history, in 1946 [39]), until the 70s, when personal computers started to have success in the great public: the idea of having a “*personal, portable information manipulators*”, has taken twenty years to have tangible effects [59]. A strong stimulus, in 1982 has been given by the Commodore64, recognized as the greatest selling single computer of all time, with more than 22 million sold units [81].

At that point, probably the greatest revolution of the last century has started to occur: from a research and entertainment field, personal devices started to enter people’s life as an important support to everyday tasks, until becoming indispensable.

On the side of communications, mobile phones started their spread in the 80s, but it was not until the mid-1990s that it become a low cost, rich in features, and used world wide technology.

A first successful convergence, between personal computers and mobile phones happened in 1996: Nokia introduced the 9000 Communicator, one of the first PDA (Personal Digital Assistant or handheld computer) and best selling devices of this category. It was able to send and receive faxes, check e-mail and access the Internet in a limited way, but its effectiveness

was limited since cellular networks were optimized for voice, not data [32]. Another example of well-known PDAs were produced in the same period by Blackberry. PDAs firstly remained mostly popular among businessmen, using them as simple agenda, but as the data network started to improve, also started booming.



Figure 5.1: Nokia communicator

By the mid-1990s also mobile phones have become small and practical, telecommunications infrastructures, and the Internet were evolving with the same speed. In 2000, Sharp produced the first integrated camera phone, and it was estimated that by the end of 2004, 75% of mobile phones sold in Japan, were camera phones [32].

As mentioned in previous chapters 2.3.1, the true breaking point in smartphones' spread, has been in 2007 with the Apple iPhone, almost 10 years ago: a multi-touch 3.5 in screen, an ample set of dedicated apps for simple tasks, an advanced camera, everything in a single device with a "fascinating" design. In the last ten years a progressive commoditization of smartphones and personal devices, has been an unstoppable trend.

Another milestone in this process, has been put by tablets, after ten years of relative limited spread: from 2010, tablets similar to bigger smartphones on one hand, similar to smaller laptops have gained their market share. Some of them have also calling capabilities. More recently, hybrid devices, with dimensions similar to a tablet, and performances similar to a laptop, have also started to gain attention, as for example Microsoft Surface: it has a dedicated OS, a powerful processor (both in two different versions, according to the user's choice) and can be interchangeably as a tablet or as a laptop.

The evolution of mobile hardware devices is far to be completed, of



Figure 5.2: Microsoft surface

course. Recent trends are showing continuous steps towards the wearability of devices: from smartwatches, activity trackers, smart glasses, jackets, and clothing accessories of any kind. Hence, devices have become more than personal, as people's lives become totally immersed in technology, the so-called Internet of Things era: nearly every personal device is able to gather user's produced data making able devices and applications to analyze them. What is interesting to be noted, as stated in the beginning of this section, is the progressive acceptance of devices in any scope of everyday life from all users' categories. User acceptance and adoption feed the development of new technology: even if a product is technologically advanced, it has not always been successful, and sometimes it took a fair long time to see it succeeding [76]. On the other hand, it has also happened that companies have been too much hurried in releasing a new product: faulty devices, lack of infrastructure support have determined the abandon of some technologies considered as revolutionary, that have never seen the light (Google Glass for example).

5.1.2 Input/output devices

Considering the aforementioned evolution of hardware devices, a distinction must be made between the input and output user interactions when dealing with different devices. The evolution of I/O (input/output) devices has a parallel course to the one of physical devices.

Concerning input devices, virtual keyboards have almost fully substituted physical ones in mobile devices, as touchscreen has become the predominant hardware interfaces; touch is also the natural substitute of pointing devices (either mouse or stylus for PDAs) ([14], [3]). Another group of input interactions, is composed of "natural interactions": conversational interfaces speech activated, eye tracking (mostly still used for people with body impairments), and device sensors (gyroscopes, accelerometers, etc.) are other technologies in which research is more active.

Regarding output devices, the digital monitor has been the most used one in every physical device: CRT, LCD, LED monitors used with desktop and portable computers and mobile devices, to touchscreens that are input and output devices at the same time. The challenges for digital monitors is to improve the technology in order to present to users more and more impressive and realistic images, that resemble printed version of the images. For touchscreens, the challenge is also to enhance the accuracy of touch capture, in order to minimize the existing skew between the touched location and the center of the fingers [14].

5.1.3 Graphical interfaces

The biggest change in graphical interfaces has come with the introduction of graphics in the user's output. Older computers and mobile phones, allowed only text-based interfaces, without any visual embellishment. The introduction of graphical interfaces, with the first icon and windows systems has enlarged the user experience with physical devices.

The spread of Internet has created new kinds of graphical interfaces, based on hyperlinking: text links, buttons, images and in general multimedia content have greatly changed the users experience and interaction.

With the advent of advanced mobile devices (smartphones, tablets, smart-watches and so forth), from the merely texting interaction of older mobile phones, an app-centric model has been developed: every task can be accomplished with the aid of a dedicated application, a small program, often task-oriented and specifically designed for a determined OS or device.

The greatest drawback in the evolution of graphic interfaces, has been the considerable heterogeneity caused by the almost total freedom on the

development of graphics: if in desktop environments the program development is tightly bound to the operating system, hence it is with the visual elements, in web and apps this is not the same [114]. A lack of standards and control over the interfaces, has led to a wild spread of uncountable graphical styles, often confusing for users. Another crucial difference between desktop and mobile, is the limited difficulty of development and distribution of software: developing websites and web applications is often easier than developing desktop programs, generally thought for more complex tasks, hence equipped with very crowded sets of functionalities. This generally results in more limited cost and time efforts, hence light-weight applications, for which the distribution through Internet, for the web, and app stores for mobile, is a more suitable way, than classical selling channels.

A particular kind of interface, earlier in web with chats, actually with messaging application in mobile, is the conversational interface: firstly only text-based, enhanced graphical elements and objects are being introduced, bringing heterogeneity and learnability issues, not present before.

5.1.4 Outline

As seen in previous chapters, web usability issues have been studied for the last two and a half decades, whereas studies regarding smartphones and tablets are still confused and lacking in some facets. Still fairly unexplored are the usability issues deriving from the newest physical devices and conversational interfaces spreading in the last years.

As a conclusion of these brief excursions on the different kind of interaction, it must be noted that any new kind of physical device or software interaction, opens a series of usability issues that must be considered to make the success of new introduced technologies. In the next sections a deeper look is reserved to conversational interfaces, and at the end of this chapter a proposal for a systematic usability analysis will be given.

5.2 A deeper look at conversational interfaces

Despite the proliferation of new technologies, the market of portable devices is still dominated by smartphones [22].

Since the launch of the *App Store* by Apple in 2008, the number of third-party applications for mobile Operating Systems –usually simply called “apps”–has

dramatically grown, now reaching the order of millions. From games to e-commerce, from health-care to multimedia entertainment, nearly every aspect of a smartphone's owner everyday life can benefit from a number of dedicated apps, their spread having become almost unrestrainable.

Nonetheless, from the enthusiasm of the starting years, this trend has started to invert. Recent statistics have shown the users' propensity to only use a limited set of popular apps, mostly by a few big players. Moreover, the same data demonstrates how the largest part of smartphone owners are used to install nearly zero new apps on their devices on a monthly basis [67]. For app developers it has become more and more difficult to make their products visible in an already crowded app store, where competition is harsher than ever.

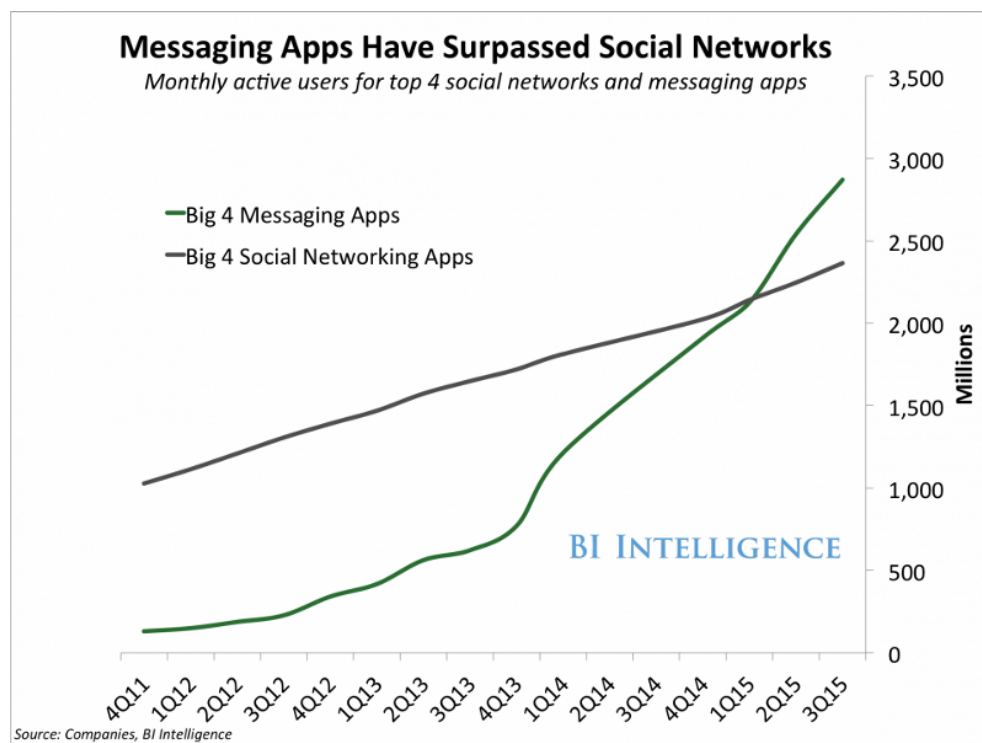


Figure 5.3: Users of online messaging applications overtaking the ones of social networks. Source: [u19]

Among the most used apps, instant messengers and social networks have always taken up the top spots in app stores. Instant messaging apps in particular have shown tremendous growth over the last years, recently taking over the lead in terms of number of users, growth, and user engagement [123].

Messaging platforms are getting immense attention and ferociously compete for user attention, introducing a growing set of features. Starting in 2014, many messaging systems have started opening up to “bots”: enhanced conversational agents that can chat with users, right inside the messaging app itself.

Bots live inside this familiar place, in a conversation thread right next to private conversations with friends and relatives, that is increasingly the most used feature of a user’s smartphone. Most users already use their messaging app several times a day and have a well-rounded understanding of the interface and its manner of working. Instead of trying to attract people to new apps, bots provide an incredibly convenient way for services and developers to engage with users where they already are, using the existing conversational paradigm, in a way that is easily comprehensible.

Even if the conversation follows the familiar conventions of the messaging system, the exchange does not need to be text-based. Thanks to the richness of the development frameworks made available by the most part of messaging platforms, bots can exchange information exploiting a set of alternative type of messages, interfaces, and UI primitives, that allow the conversation to be essential and efficient.

Bots are growing fast in number and many platforms have started offering *bot stores*, just like mobile OS platforms do for apps.

This and the following sections analyze this trend, addressing them as *modern bots* different from traditional ones: conversational agents with enlarged capabilities and purposes. Further, the distinguishing characteristics of such agents are identified and their main advantages over traditional applications are outlined.

Starting with a brief history of conversational interfaces, following the distinguishing traits of *modern bots* are presented. A detailed analysis of the advantages of these conversational agents is provided, followed by an overview and comparison of the most popular messaging platforms currently available. The differences between *modern bots* and apps are discussed, speculating about the future trends of these technologies. Finally an analysis and considerations about the usability of these kind of interfaces is given.

5.3 History of conversational interfaces

The idea of humans interacting with “intelligent” computers, endowed with their own conscience, often “devoted to evil energies” [77], has fulfilled the history of science fiction movies and literature: HAL 9000 in *2001: A Space Odyssey* by Stanley Kubrick, in 1968, was talking with astronauts while steering the ‘Discovery I’; other examples of computers with these capabilities have been made famous by other legendary movies, like *Star Trek* and *Star Wars*, in the 60s and 70s.

If a large strand of research in the last sixty years have followed the way of robots with humanoid appearance and resemblance, focusing on Human-Robot Interaction (HRI - a multidisciplinary field including also HCI), the other way has seen the proliferation of only conversational interfaces, from the ones with only texting capabilities to the more evolved ones, able to recognize and answer with spoken natural language.

This evolution in conversational interfaces, in present years seems to have found the leading directions between Virtual Private Assistants and the return of chatterbots with enlarged capabilities.

5.3.1 The dawn of “intelligent agents”

The idea of building a computer, or better still, a program, able to talk with humans, giving the illusion of a true human-to-human interaction, can be dated back to the '50s, when Alan Turing proposed his famous *imitation game* [117]. Better known as *Turing test*, it aims to determine if a machine can give the impression to other humans of being human itself. The game starts with an interrogator who should understand who, between two subjects, situated in another room, is the man and who the woman. One subject should help the interrogator, whereas the other should confuse him. If, after substituting the falsifier subject with a machine, the number of times in which the interrogator has the correct guess on who is the man and who is the woman, between the two subjects, remains the same as before the substitution, then the machine should be considered “intelligent” because it would not be distinguishable from a human. These have probably been the basis for artificial intelligence studies.

Today the Turing test is still used as a test for evaluating to which extent a program, a *bot*, is human-like: Loebner Prize is annually assigned to the

best computer system pretending to be human.

The first satisfactory example of such a program, trying to fool people during a conversation, was *ELIZA*, created by Weizenbaum [121] in 1966.

ELIZA did simply answer any of the user's utterances with other questions, "roughly as would certain psychotherapists (Rogerians)" do, on a typewriter. Weizenbaum's aim was to implement a program able to have a conversation with a human in natural language, fooling users on believing they were interacting with another human.

That is what has been called a *chatterbot* and what put the ground for building chatterbots (or chatbots, or simply bots) in the following 50 years, until today.

ELIZA is based on grammatical analysis for sentences, by the definition of input and output rules and keyword patterns, in order to build *intelligent* answers to provide to users. When a user starts the conversation, their sentence must be analyzed: text is decomposed into words and *keywords* are searched among them. If one or more keywords are found, these are sorted on a "keystack" by assigning a RANK to each of them, hence the initial sentence is manipulated according to the transformation rules associated with the keywords. After the decomposition, words are rearranged together with some additional words, according to the reassembly rule used, dependent from the decomposition rule used to split the sentence. This makes *ELIZA* independent from the context and from the language used, but rather complicated to extend.

Weizenbaum's observations on humans interacting with *ELIZA* had some important implications. Most of them knew *ELIZA* was not human, or were not fooled by it [60], but it seemed they were really enjoying the interaction with a computer program, as some were keeping the conversation for hours. This was one of the reasons that brought Weizenbaum to abandon the development of artificial intelligent systems. Observing this propensity of humans to interact with machines has also laid the foundation for the studies in *Human-Computer Interaction*.

Another example of a very famous and more recent chatbot is A.L.I.C.E. (Artificial Linguistic Internet Computer Entity), which won Loebner prize in 2000, 2001 and 2004. It has been developed by Wallace in 1995 and was inspired by *ELIZA*. *ALICE* relies on simple pattern-matching algorithm, it uses a simple pattern template to represent input and output and makes use of

recursive techniques for the application of rules [119].

The underlying intelligence is based on AIML [118] (Artificial Intelligence Markup Language, an XML - eXtensible Markup Language - based language), which makes it possible for developers (botmasters) to define the building blocks of the bot's knowledge [118]. The basic object of AIML is called *category* (an ontology), composed of an input question, called *pattern*, an output response, called *template* and an optional context, possibly of two types, called *that* and *topic*. ALICE's brain is based on Graphmaster, an object containing a pattern storage and a matching algorithm: whenever a user inputs a pattern, in order to find a suitable template, Graphmaster visits the tree of AIML categories to look for a suitable template. The tree is a collection of nodes called Nodemappers, which map the branches from each node. Each branch can be a single word or wildcard. The root of the Graphmaster, in ALICE's case, is made of 40,000 Nodemappers. The tree's leaves are represented by the categories and contain the template tag. Beautiful pictures have been made of ALICE's brain, like in figure 5.4.

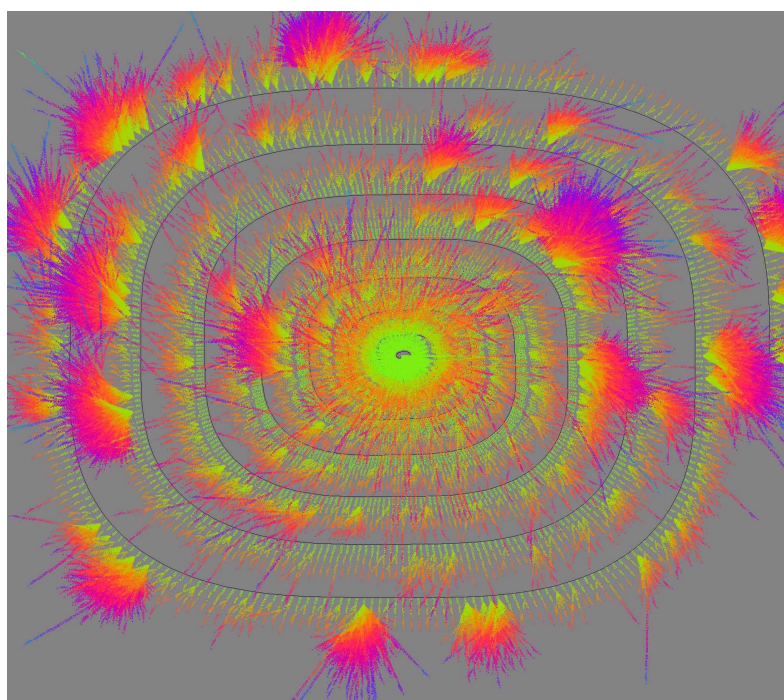


Figure 5.4: A representation of A.L.I.C.E. brain. Source: [u20]

Even though ALICE and ELIZA are similar in some facets, there are some

important differences between them. As mentioned before, ALICE uses recursion whereas some ELIZA's rules, can cause cycling or iteration [112]; another important point to consider is the ability of ALICE to combine two answers together in certain cases, whereas ELIZA cannot. Moreover, ALICE pattern-matching algorithm tries to find the longest pattern between the tree and the analyzed sentence, whereas ELIZA uses the first keyword found. On the other hand, the higher ELIZA's complexity is compensated by the fact that it can give different answers in case of same input during a conversation, trying to randomize them, and makes it able for users to build their own script files, helping the user to better understand how answers are generated. This has not probably been a great advantage, if a greater spread and use of AIML can still be observed.

From the merely demonstrative purpose of ELIZA and ALICE, to the possibility of developing "intelligent" agents, able to really converse with humans using natural language, bots have been constantly and increasingly employed in many fields with a discrete success, yet with limited spread.

They have been applied in e-commerce [60], like *Nicole* a virtual assistant with customer service tasks, or *Anna* by IKEA; in education, like *CHARLIE* [78], a bot that lets students communicate with the online learning platform *INES*, or *TQ-bot* [37], with purposes of students tutoring and evaluation; information retrieval, like [108], a chatbot acting as a virtual guide for people visiting historical sites, or [111], a different way of querying search engines and giving back answers, as an alternative to Google; in e-government services [74], to help citizens with public services. Most of those bots are AIML based ([109], [113]).

The growing pervasiveness of personal devices, has created the need to find a better way to let people interact with them [113], trying to create a more satisfactory experience of use, by making the interaction as natural as possible, letting users using their natural language.

5.3.2 Spoken interaction

Along with chatterbots, supported by the evolution of automatic speech recognition (ASR) systems, in the 80s, Spoken Dialog Systems (SDS) have started to draw the attention of academics and the industry [76]. The conversation was moving from a textual to a spoken interface, presumably easier and more natural to use for humans.

ATIS [45] was a telephone-based flight reservation system, funded by DARPA, developed in 1990, in the USA; in the same years, in Europe SUNDIAL was developed [104], with the same aim of giving information about flights via telephone. SUNDIAL was able to understand between 1,000 and 2,000 words in four different languages.

A similar, more recent system was Mercury [110]. Several issues had to be addressed with SDS, from the main speech recognition modules, to the correct understanding of user's requests, to giving the system the means of offering a satisfactory answer.

Different types of SDS can be recognized considering, for example, how active is the system in the conversation [129]: in *system-initiative* in which the computer is totally in charge of driving the conversation, asking questions to the user; on the other side, *user-initiative* systems leave the user totally free of speaking to the computer; in the middle there are *mixed-initiative* systems. This last type of systems, better simulates the natural interaction in human-to-human conversations, where the two subjects talk in turn and carry on the dialogue, but are more complex to implement.

A further step towards the "personification" of intelligent systems, has been achieved with the development of Embodied Conversational Agents [76], in the late 90s. Animated characters, with human features, able to simulate emotions with facial expressions and gestures have been employed in different fields, to interact with humans. They seem to be perceived as more trustworthy and agreeable, and with the hope that people would have accepted them better in everyday life than a simple textual or spoken interface. Cassell et al. [21] developed Rea (Real Estate Agent), a humanoid expert in real estates, that interacted with users and could sense them by means of cameras.

In more recent years, in the e-commerce field, Anna, the IKEA virtual assistant has gained quite big success, due to her good emotions expressiveness and polite reactions.

Kopp et al. give another example of an ECA that has been used as a museum guide, in order to engage visitors and to inform and interactively chat with them. In a real world study, not yet done before, the authors wanted to understand if Max was perceived and treated as human-like. It has been observed how users were inclined to use human-like communication strategies and perceived the agent as sociable [64].

5.3.3 Virtual private assistants

Besides these studies about the users acceptance and perception about virtual agents, either providing textual or spoken interfaces, only few years ago, smart assistants have drawn the attention of the greater public and have become a “personal” matter and to be perceived really helpful in everyday life.

According to McTear et al., different reasons have influenced this new success of conversational interfaces [76]: the progresses in artificial intelligence assistive technologies, like speech and image recognition; the emergence of semantic web; the increasing availability of connectivity and improvements in device hardware technologies; and the renewed interest of big technology players as a major impact factor.

Apple’s Siri [u21] (generally considered to be the the first public voice-enabled Virtual Private Assistant), Microsoft’s Cortana [u22], Google’s Google Now [u23] (substituted by Google Assistant in May 2016), Amazon Alexa [u24], and Samsung S Voice [u25] are the main actors of the last five years in the field of conversational interfaces.

Most notably, the reason of this growing success should also be searched in the many differences in respect to older personal assistants. First of all, smartphones have become pervasive in everyday life and are perceived as more personal than older devices.0 The same goes with the assistants, which are present 24/7 for the user (just think to the film *Her* of 2013 in which the main character falls in love with Samantha, his personal virtual assistant, even though, Spike Jonze, the director, told he was inspired by ALICE).

Assistants are perceived as more flexible [8]: they are not limited to a very narrow task, but are able to interact with a plethora of applications, internal and external to the device. The interaction is more “human-like”, using simple, yet impressive tricks: the capability of answering sassy questions (try to ask “Siri do you love me?” is probably the best example on how this unusual kind of interaction has brought the success of these assistants) whereas a chatbot was just answering “Try to change your question”; the effort made to provide direct answers, instead of a pool of possible answers; the improvements in the inference of the users’ intents and the correction of ambiguities; better interpretations of the semantic inputs; the fact that assistants have been provided with a synthetic and likable personality. All these factors have probably made the luck of modern VPA and the failure

of older ones, as e.g. *Wildfire* a VPA of the mid 90s, multi-modal and phone based, but with poor human-like interaction capabilities [58].



Figure 5.5: Logos of the main current VPAs, from left to right: Siri, Cortana, GoogleNow, Alexa, S Voice, M

Even though basic services of the aforementioned personal assistants are somewhat similar (web search capabilities, planning events, voice calls and messaging, music playing, shopping capabilities, personalized notifications, retrieve weather information), some peculiar aspects can differentiate them.

Siri and Cortana are the more similar ones, acting as personal assistants with a well defined “personality”: differently from all the others assistants, they are able to answer sassy questions. Both are proprietary and run to date only on Apple and Microsoft devices respectively. Google Assistant is also similar to these first two, but beside the facts that it can be installed either on Android and on Apple devices, it lacks of a well defined personality and it pulls information directly from the users’s online Google Account, possibly raising some privacy issues.

Samsung S Voice is probably the more “classical” one among the others and is not perceived as personal as the other ones. Indeed, at the beginning of October 2016, Samsung has acquired *Viv*, a new VPA developed by the same Siri’s authors and declared to be more advanced than Siri, in many aspects. Besides, it is not yet public and it is not known how it will be integrated in Samsung’s products.

Lastly, Amazon Alexa is somewhat going in a different direction: developed for Amazon Echo, a smart speaker, it brings closer the Internet of Things (IoT)¹ in everyday life and it can be integrated in different devices, is available for different operating systems and has mostly capabilities related to home automation and entertainment.

In October 2016, Google has unveiled Google Home, a voice-activated speaker powered by the Google Assistant, similar to Amazon Echo with Alexa.

¹ Defined as the set of pervasive things or objects – sensors, actuators, smartphones and so forth – which are able to interact and cooperate with each other in order to reach common goals [2].

Together with Google Home, Google Allo has been presented: a smart messaging app to interact with the Assistant.

5.3.4 Bot platforms

In the last couple of years a new approach to conversational interfaces has been gaining importance: an apparent return to the classic *chatterbots* texting interface has been observed, with the main difference that these bots have gained new capabilities than simply a conversational feature, and “live” in the cloud.

This new generation of bots resembles more and more the modern mobile applications and personal assistants than simply texting interface for simulating a conversation. Since 2014, many online messaging systems, like Kik, Telegram and WeChat have turned into development environments, opening up to third-party developers, offering the means to building bots and exchanging messages with users through the platform



Figure 5.6: The main bot platforms, from top left to bottom right: Kik [u26], Facebook Messenger [u27], Telegram [u28], Skype [u29], Line [u30], WeChat [u31], Slack [u32].

Application Programming Interfaces (API) for bots expose high level services (payments, bot directory, authentication etc.) and UI elements (buttons, locations, images, etc.) giving developers the possibility of implementing bots focusing more on offered services and on user experience than on programming issues.

Services offered by these new bots are of a higher level than the ones offered by older chatterbots and they often offer services that have greater utility in everyday life, such as ordering a pizza, managing an e-commerce purchase, booking restaurants, ordering a cab and so on.

Bots dedicated to health care, e.g. *Nombot* [40] a bot helping users to track their daily food consumption, on Telegram; educational purposes, like

the one by Chaniago et al., a bot that lets parent track student's presence at school [23]; education help, like *MOOCBuddy* that proposes MOOC courses over Facebook Messenger [55].

It is also noteworthy that the possibility of building more knowledgeable and useful bots is also due by the increased availability of *open data* and the increased proliferation of service APIs. Since data and services are increasingly accessible through programmatic interfaces, and given that bots often offer a simpler development platform than apps in terms of effort and maintenance required, the task of offering access to such services through a conversation interface is very approachable. In fact, given that most *open data* are made available by local governments and concern social services for citizens, a large number of *open government* bots have been developed recently.

Many of the major messaging platforms lately offer a service of *bot directory*, a repository of all available bots that can be accessed through the platform.

5.4 Modern bots

In the following sections, these *modern bots* will be more precisely identified, trying to define the distinguishing characteristics, advantages and disadvantages, and comparing them to traditional mobile apps. Among the plethora of new and old conversational interfaces, *modern bots* are a set of automatic conversational agents that, having been designed following principles of simplicity and effectiveness, can serve as a functional replacement of applications.

A *modern bot* is an agent that is endowed with a conversational interface accessible through a messaging platform, which provides access to data, services, or enables the user to perform a specific task. In particular, a *modern bot* is generally characterized by the following distinguishing features.

Thread as app *Modern bots* are stepping stones in the evolution from an *app-centric* mobile OS experience—where the whole user experience of the device is concentrated in mostly independent applications that serve as an enclave of unique services and functions—to a *thread-centric* experience—where services and information are provided as streams of messages and

notifications, presented using a coherent and consistent set of interface paradigms.

Messaging threads are personal conversations, which enclose relationships. Exchanging messages is how users of a mobile OS foster relations with other people, in a very natural and intimate way. These personal conversations can be naturally extended to services and businesses.

Everyone of these threads is an entity that can send updates and notifications to the user, even in multiple parallel conversations, while taking advantage of the built-in facilities of the messaging system. For instance, threads are entities on which users retain total control: they can choose to reply, mute the conversation, or even to permanently delete the thread. Also, threads have the capacity of keeping track of read/unread status and drafts on multiple devices or platforms, of any message to be searched, or of notifying users in a familiar way.

Most modern messaging apps are in fact presented as a threaded “inbox”, automatically grouping messages from the same sender and displaying recently updated conversations in a preeminent manner. Instead of having to hunt for the needed service in a specific and isolated app, with its own custom interaction modes, users can rely on the fact that frequently or recently used services are automatically promoted to a visible position. A user’s “inbox” acts as a replacement for the app-centric homescreen of a mobile OS, where the most recent threads represent a dynamic list of favorite apps.

Thus, conversation threads make it easy to provide integrated tools and services that make it easy to accomplish regular tasks, but in a recognizable and familiar place: a personal connection developed through an exchange of text messages.

History awareness Just like mobile apps, *modern bots* are designed to solve a specific and circumscribed issue. However, unlike most apps, *modern bots* inherently keep an exhaustive chronological log of past interactions with the user in their thread.

This ingrained feature of a thread of messages allows the user to explore past information in a familiar way, by scrolling through a timeline or by using built-in temporal search. Users can approach the service with more confidence, since past state appears frozen in past messages and data does not unexpectedly vanish. In most cases, it takes an explicit user action to delete past interactions.

History also serves as guidance to users, since past commands (and the results they generated) can be easily used as a template for future requests.

Also, past history provides the context in which all future interactions can be evaluated. Information collected by a *modern bot should* be maintained and used in order to streamline requests, skipping questions and automatically disambiguating between different choices if possible. A conversation is a natural and effective way to collect personal information, interests, purpose, and preferences of the user, all of which can be employed in order to improve the quality and accuracy of the service.

Enhanced UI Despite the fact that *modern bots* derive from chatterbot-like conversational interfaces, their UI does not *have* to consist of mere plain text messages.

Modern messaging platforms support a variety of messages, including pictures, “stickers” (preset or custom images that convey emotion), videos, and/or audio. Most platforms also allow the transmission of packaged data, as in the case of geo-locations or contact information, in addition to generic data files. While these platforms of course do not offer the graphical capabilities of apps, it is important to make use of the features available and to exploit the conventions of the messaging platform. For instance, instead of printing out a raw URL, some platforms may display links as nicely formatted cards with a preview.

Moreover, even plain text messages can be enriched on many messaging platforms to provide a limited subset of rich text formatting. Bold, italic, and embedded links are the most common formatting options, available through markup languages such as limited HTML or Markdown [u33]. Unicode-encoded *emoji* characters can also be used to efficiently give additional meaning or to convey emotion.

Typing should be thus reduced to a minimum: ideally it should be limited to very short and concise commands and replies, of few characters. A fitting comparison can be done with UNIX commands, which are an example in terseness, as they were designed to efficiently work over teletypeterminals—the primeval example of text-based interface—and reward users with powerful functionalities with very little typing.

Even if it can be argued that younger generations are getting more accustomed to typing on touchscreen soft keyboards, the practice of shortening common words or abbreviating expressions—in particular when using messaging apps—alone demonstrates why there is no need to enter a full-fledged message exchange with a conversational interface. Interactions with the service should follow linear conversation routes and avoid complicated branching or multiple passages of dialogue: anything above a couple of taps on the screen will become tiresome to most users. In a user study by Azenkot et al. the average text entry for all participants on a popular mobile OS keyboard was of 41.01 Words per Minute (WPM) [3]. Also, the maximum typing rate predicted by MacKenzie et al. was of 43.2 WPM for expert users on QWERTY keyboards [73], which is of course far lower to what can be achieved by expert typists on a physical keyboard. Text interactions should therefore be kept short and precise.

Many messaging platforms also feature new structured forms of messages, which can further enhance the flow of conversation. Structured messages may contain one or more buttons for “instant replies”, show different alternatives in a rich representation, or show a list of available commands. Examples of such messages are shown in the next section.

Their advantages are manifold. (1) They constrain the conversation into a limited number of expected outcomes, reducing the possibility of users feeling trapped in a dead end where they have to “guess” their way out. (2) They push the user to use the service, suggesting how the conversation can continue. They also reduce the need for the users to “explore” the interface, making it easier to learn and use. (3) Buttons and quick replies reduce any interaction to a single tap instead of requiring complex typing. (4) The service can be more easily implemented.

Modern bots try to fill the gap between plain conversational interfaces, which are inherently inefficient to use and offer little way in terms of customization and user experience, and the world of the full graphical interface.

Limited use of Natural Language Processing (NLP) Given the aforementioned rising interest in Virtual Private Assistants (VPA), voice appears to be a natural fit for conversational interfaces. However, to the best of our knowledge, existing bots on messaging platforms avoid voice processing and

choose digital text as the most direct and unambiguous form of communication, eventually adopting NLP systems in order to extract commands and intent from the user's messages. Even if natural language understanding is getting progressively more advanced [8], in many scenarios complex dialogues break down because of simple acoustic misunderstandings or because the user's context cannot be fully elicited from the conversation [76].

Modern bots should not attempt to provide the complex experience of a full VPA. Instead, in keeping true the principles of simplicity and effectiveness, *modern bots* should not necessarily use AI and support for NLP. At the scale of a single service, going after AI is mostly excessive and counterproductive, when the same results can be obtained with simple text commands using a limited language.

Modern bots should not pretend to be human: except in cases where this is desirable (e.g., for customer support or as a barrier before an actual conversations with a human operator is activated), it should be clear to users that they are talking to a machine. Even if artificial delays or "is typing" indicators can be used in order to make the conversation more recognizable to the user, faking human responses risk to increase the actual distance between service and user instead of decreasing it. Texting to a computer that doesn't understand what the users are saying can be a frustrating experience, in particular when the computer hides its failures inside a dialogue that is artificially kept "natural" and "human-like". This hides failure points in the conversation and makes the user feel less in control of the interaction.

However, this does not imply that *modern bots* cannot have a personality or take advantage of humor and emotional responses to provide a charming and likable interaction with the user.

Modern bots should rely on the limited but accurate interaction tools the messaging platform makes available, while NLP frameworks can optionally be employed to accommodate unforeseen user requests. AI and deep human-like dialogues are red herrings in the current development of conversational interfaces. *Modern bots* are about accessing services efficiently, a command-line-like interface to cloud-based APIs, not talkers.

Message self-consistency Each single message sent by a *modern bot* should contain the full context of the conversation and should represent what a single screen or UI dialogue represents for applications. Users should not

have to browse through their conversation history in order to figure out what they are attempting to do and what the service is expecting. Each message has an atomic meaning and stands on its own.

The scope of each message must be clear, its intent must be explicit, and what action must be taken by the user—if any—must be explicit and unequivocal. Indeed, a message delivered by a *modern bot* should be conceptually seen as a *micro application*, while the conversation is a timeline of past application screens.

Some messaging platforms have, in fact, the ability to alter messages after they have been delivered. In that case, messages can be changed based on the availability of new data or other changes in state, giving the impression of a living view on the service.

Guided conversation An important part of an application's UX design is focused on user guidance and, likewise, the same care should be applied when designing conversational interfaces through text messages. In fact, because of the free-form nature of the medium, it is easy for users to get lost and not to be certain of what commands or what exact syntax is required to perform the desired action.

A successful *modern bot* guides the user through a task in order to avoid this impasse. The service proactively suggests actions that are likely to follow up after the current interaction, offers alternative choices when needed, and generally offers a framework in which user interactions feel reliable. This can be achieved using the same UI enhancements mentioned before, that is through the use of buttons, formatted messages, or built-in menus that offer interface guard rails to the conversation.

Also, notice that when starting the first interaction with a bot, many messaging platforms offer a way to show a welcome message to the user. The design of the onboarding experience must take into account the initial user guidance and ensure that all functionalities are readily available.

5.5 Overview of bot platforms

In this section the most popular messaging platforms that support bots through their APIs will be taken into exam, describing distinguishing features of each one.

All of the following platforms allow third-party developers to register an identity for a virtual agent and to programmatically receive messages from any user of the platform, either accessing an API end-point (*pull* mode) or being called back by the platform itself using a “web-hook” (*push* mode). Both modes, for all platforms, make use of the HTTP protocol.

Kik : Launched in 2010, Kik’s user base has currently grown to over 200 million, including 40% of American youth, according to the developer’s website [101].

In September 2014 bot platform has been released and updated in 2015 with 80 bots, until 2016 when the “botshop” was also released. It offers an HTTP API or language specific libraries, more precisely for Python and Node.js. Requests are authenticated via HTTP authentication over SSL.

Facebook Messenger : released in 2011 as messaging application, it has been transformed into a platform in April 2016 and in September 2016 some important services have been added, like payments, webviews to give more custom graphic interfaces, share buttons, quick replies (the possibility of answering with predefined UI elements instead of text and simple buttons). This seems the more focused on user interface, more than on services and data. Even bot analytics are provided.

Telegram : founded in 2013 from the same creators of VK, the most famous Russian social networks, it has become mostly famous for its improved security in message exchange, compared to existing online messaging applications [120]. In 2015 the Bot API has been released, in January 2016 inline bots and lastly Gaming Platform has been announced by Telegram team and released at the beginning of October 2016.

Telegram Bot Platform is open and the communication with bots is over HTTPS.

The introduction of a gaming platform should be carefully observed and studied, as with previous observations about gamification and this also increasing trend, this can possibly make Telegram the most successful platform among these ones.

Skype : born as a VOIP desktop application, more used for video-calls among users, than for online messaging, it has been released in 2003. Acquired in 2011 by Microsoft, it substituted Windows Live Messenger online messaging application in 2013.

From April 2016 it is also a bot platform. It provides an SDK for development (in C# or Node.js), and a REST API for the interaction with bots. It gives the possibility to import bots from other platforms and to develop Video Bots. Microsoft also announced the future integration of HoloLens, Microsoft product for augmented reality.

WeChat : WeChat has always been more than an online messaging application. It integrated also features of social networking, payments and a heavy support for non-textual content (buttons, images, contacts and so on). Released in 2011 in China, according to recent statistics, of May 2016 (Business Insider) it has more than 700 billion monthly active users, making it one of the largest messaging apps. In April 2013 the bot platform have been launched and it started the bot explosion in the rest of the world.

A bot chat account is called “public account” and it distinguishes between two types of accounts: *subscription accounts*, for content publishers who need to send new content to their subscribers, and *service accounts*, dedicated to customer service provided by organizations.

It is less focused on Artificial Intelligence, but more on users’ needs.

Line : Line was not only an online messaging application, but also a social network and a game “platform”, dominating in Japan and South East Asia. It was born in 2011 to support damage to telecommunications infrastructure in Japan, after a devastating earthquake. The Bot API was one of the last launched, in 2015, a REST API, soon deprecated and substituted by the Messaging API in 2016, with more advanced features, including payment service, called Line Pay. Authentication is made via API key.

Slack : is more a team collaboration and communication tool, but it has served as a test bed for many types of bots, being one of the earliest to offer this facility. Created in 2013, initially used as an internal company tool by Tiny Speck, for the development of a no more existing game. Botkit has been launched in 2015, but “bot users” have a strict focus on workplace features

and this made Slack bots spread not so large. Authentication is made with OAuth.

In Table 5.1 the platforms considered in this work are listed, firstly reporting the amount of *Monthly Active Users* (MAU), which gives an approximation of the relative popularity of each system. Furthermore, the table shows whether the platform supports *group messaging* (i.e., if one or more bots can be added to a group conversation between real users) and *mentions* (i.e., if bots can be "called into" a conversation using a special combination of characters). Both of these features allow bots to be used in order to perform specific tasks for a group of users instead of only one. The table also reports the different *message types* supported by the platform (including pictures, voice, video, stickers, and structured messages, discussed in the next section) and other significant features.

Table 5.1: Table of platform features.

Platform	MAU†	Groups	Mentions	Message types	Buttons	Carousel	Quick reply	Payment
Kik	80 M	✓	✓	Picture, video, sticker, voice.			✓	
	Kik code identifiers, browser integration via Javascript.							
Messenger	800 M			Picture, video, file, voice.	✓	✓	✓‡	✓
	Persistent menus, several message templates (Airline trip, Buy, Receipt, Web link, etc.).							
Telegram	100 M	✓	✓	Picture, video, sticker, file, voice, location.	✓,		✓	
	Persistent commands, deep-links through https://t.me/telegram.me							
Skype	300 M			Picture, video.	✓	✓		
	Several message templates (Hero image, Thumbnail, Receipt, Sign-in, etc.), phone call support.							
Line	220 M	✓		Picture, video, sticker, voice, location.	✓	✓		
	Imagemap message template (picture with multiple hot-spots).							
WeChat	700 M		✓	Picture, video, sticker, voice, location.			✓‡	✓
	QR Code support, Rich media and Music messages.							
Slack	≈ 3 M	✓	✓	File.	✓			

† Monthly Active Users, based on most recent quarterly report published. Numbers for Slack are an approximation based on known Daily Active Users.

‡ In the form of custom defined menus shown in the conversation UI.

5.5.1 Interface features

Messaging platforms distinguish themselves not only for their underlying technical aspects, but also because of the different user interface elements they offer to bot developers and, ultimately, to the end-users. While sending and receiving text and basic multimedia messages is a common feature, more structured messages are available only on some platforms and often differ in key aspects.

Many platforms offer a way to suggest canned replies and to let users send them with a single tap. For instance, on Messenger bots may display a selection of *quick replies* that appear on the bottom of the conversation and remain valid for one interaction, as shown in Figure 5.7a. On Telegram and Kik instead, bots have the ability to replace the keyboard with suggested responses, as shown in Figure 5.7b.

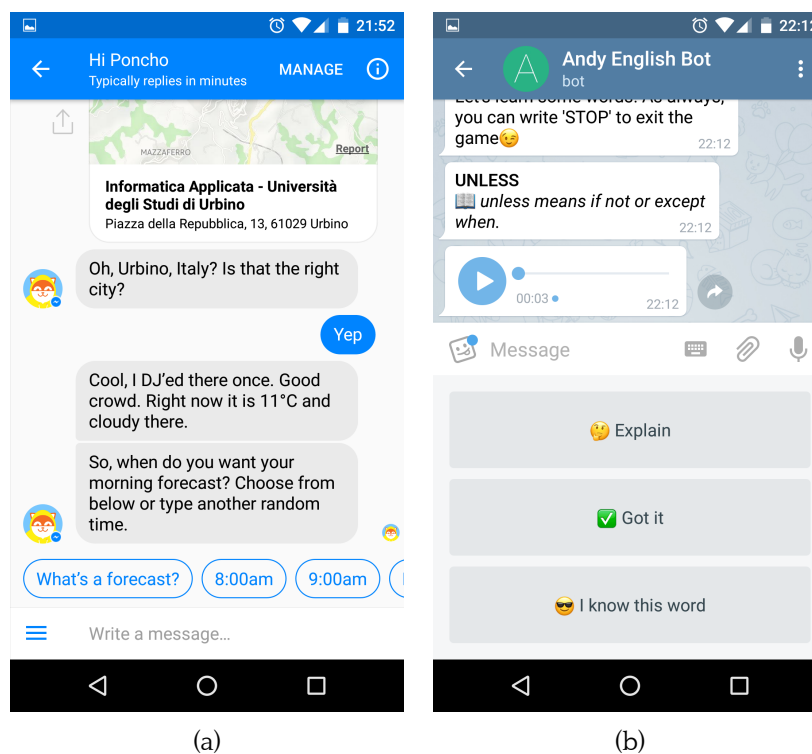


Figure 5.7: Features that allow users to pick suggested replies. (a) *Quick replies* on Messenger. (b) *Button keyboard* on Telegram.

While these preset replies can change in the course of the conversation, other UI features can immutably be added to the chat. In Figure 5.8a a

list of *commands* are shown. On Telegram, commands are characterized by starting with a `/` character and are always available to the user to perform basic tasks supported by the bot. A similar system, but with a hierarchical structure, is shown in Figure 5.8b: WeChat allows developers to add a fixed menu to the chat interface, showing a maximum of 3 first-level options and several second-level ones. Activating commands and menu elements alike, gives access to bot's functionalities.

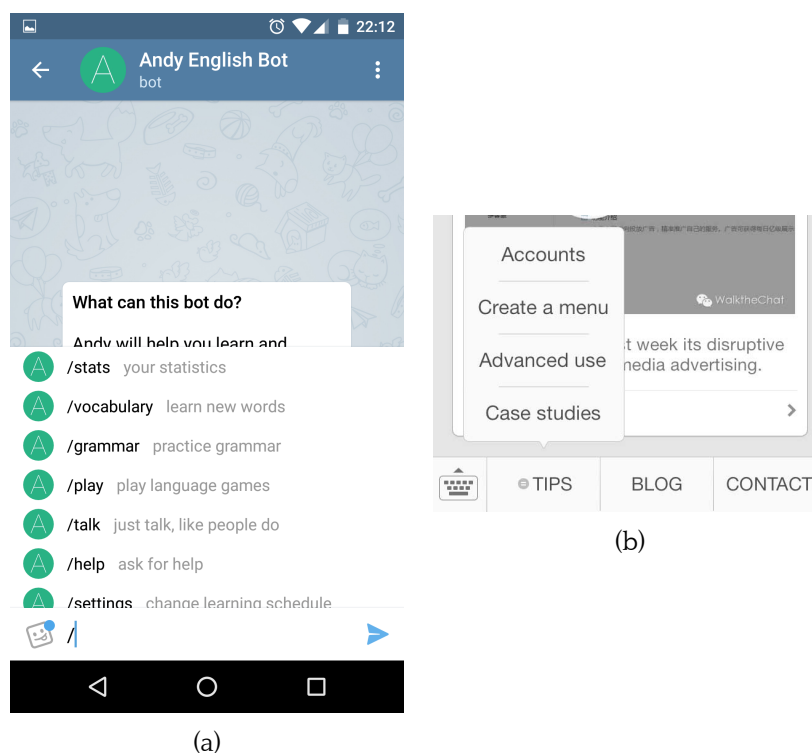


Figure 5.8: Structured commands available during the whole conversation. (a) *Commands* on Telegram. (b) *Custom defined menu* (with an open second-level menu) on WeChat.

Other UI features are not bound to the overall conversation, but are instead embedded in a specific message. For instance, *carousel* messages shown in Figure 5.9a make it possible to include multiple rich cards, provided with an image, a description, and a button, and make them horizontally scrollable to the user. In Figure 5.9b a message with *embedded buttons* is shown: in this case the actions provided to the user are not tied to the conversation, but to one single interaction. Both message formats allow bot

developers to show available alternatives to the user, providing a well defined path for the conversation.

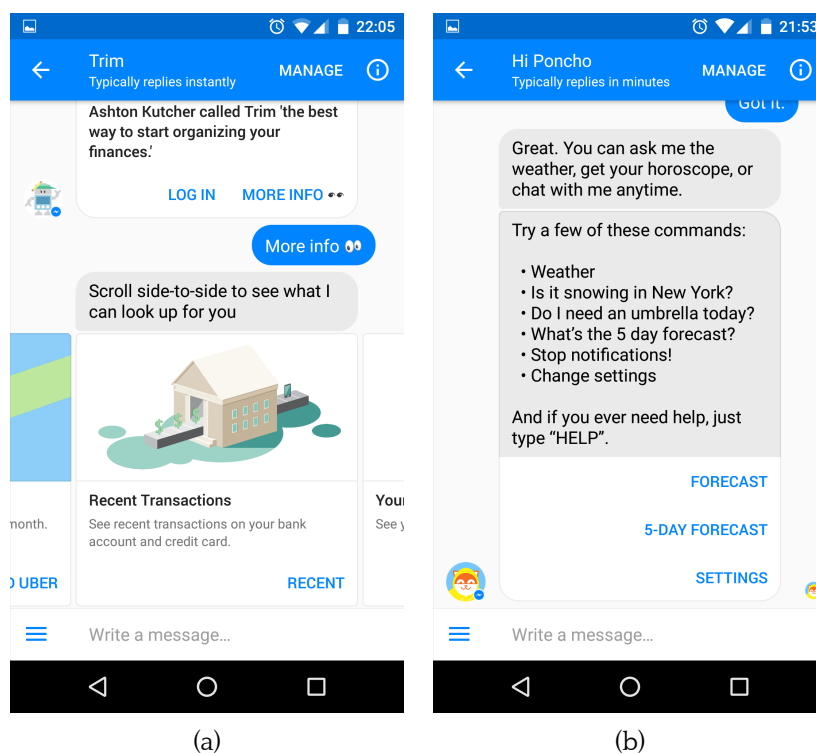


Figure 5.9: Structured message templates in Messenger. (a) *Carousel* presenting multiple choices through horizontal scrolling. (b) *Message buttons* embedded in a message.

Several other message formats are available to specific platforms, among which messages including a “Buy” link for shopping-oriented bots, messages with music support in WeChat and flight travel itinerary messages on Messenger.

In terms of making bots interoperate with external systems, it is noteworthy that several messaging platforms have included support for QR Codes or variations thereof. On WeChat, code scanning can be used internally to the system in order to send payments of a preset value to a given contact, which makes the platform compelling for sellers of physical goods as well. On the other hand, Skype and Telegram allow web sites and apps to launch conversations through the use of special URLs. In particular, Telegram also allows to send a custom data payload to a bot through an extension of this

mechanism, called “deep linking”, which can be used to embed bots in innovative and complex workflows.

5.5.2 Advantages of Bots for users

Instant availability One of the main advantages of bots is their instant availability, similar to Android *Instant Apps*: there is no need for installation and they’re immediately up and running as soon as they are added in the messaging app. Users only have to start a conversation with desired bots, as if they were normal “human” friends in their contact list. No additional device storage, nor waiting time for installation or configuration are required. This makes bots fast and light-weight, compared to traditional mobile apps.

Gentle learning curve Message texting services and applications have been used since the dawn of mobile phones and this makes fairly natural for users the interaction with such interfaces. Hence, learning how to interact with a bot results in a very easy task for all types of users, who can also be guided by the bot during usage. Providing hints, examples and graphical advanced elements, bots can easily teach users how to exploit all the available features.

Notably, bot platforms supply common UI building-blocks (such as buttons, carousels, quick-reply buttons and so forth) and a common vocabulary, that make bot interfaces fairly similar among them, hence quite easy to understand and learn for users.

Notifications Instant messaging applications already include efficient and functional push notification systems. There is no need to implement new ones and to overload users with several different disturbing notifications. Indeed, almost every new installed app has its own notification system, used as a mean to re-engage inactive users. However it should be considered that according to statistics [u34], more than the 50% of users find push notifications annoying. In case of bots, these come with the ones from the messaging apps, therefore they are perceived with a more well disposed attitude by users.

Social graph and contacts Users already store friend’s contacts in messaging apps, with bots there is no need of adding them again in their networks, as bots have access to the user’s contacts lists. Therefore, in case of a bot

able to simultaneously interact with multiple users, like group conversations participants are able to add others without inviting them to join the bot audience, as it would otherwise be done with a traditional app.

Platform independence Bots live inside instant messaging applications without have to worry about on which mobile platform they are being executed. This makes bots independent from the underlying hosting operating system. Each bot is inherently available on all the operating systems the messaging app support, without any graphical or functional adjustment. On the contrary, native mobile applications must be adapted and rewritten according to the main mobile operating systems they want to support. Users sometimes perceive the same application as different, if they are using it on different operating systems: this is obviously a very problematic issue for users' engagement.

Authentication User authentication is not needed in bots, as the hosting messaging application already provide the user's identity in a reliable way. Users are already authenticated and they do not have to create additional accounts and passwords in order to interact with the service.

Usually, for each new app the user install, they need to sign up with a brand new account, hence increasing the number of personal accounts to be remembered and protected, which is a known problematic issue for users.

Payment support Some of the analyzed platforms include payment services, already integrated in the messaging system. Often users have already connected their credit cards or bank accounts to such services, for purchasing purposes, hence bots do not have to require additional payment methods, directly using the existing ones. On the contrary, whenever installing a brand new app requiring payment capabilities, users have to re-connect their payment accounts, and most notably, have to trust the just installed app.

Discoverability As seen in previous sections, there exists a plethora of bot "stores", sometimes offered by the hosting platform, sometimes independent from them. Users can easily browse directories and find the most suitable ones for their needs. App stores also exist since the mobile apps spread,

but as notice in previous sections, it is difficult to emerge from these over-crowded systems.

Asynchronicity Exchange of instant messages is an asynchronous task: users send a message and do not have to wait for a reply, instead they are able to start other conversations in parallel. Furthermore, thread conversations store the context, letting user free of leaving a conversation and go back to it later, restarting the use of bots from the very last interaction. Multiple conversations can be carried forward at the same time, without losing any step.

Limited data size Downloading a new app can have a sizable footprint in a limited mobile data plan, whereas to start to use a new bot only requires to initiate a new thread in an existing messaging app, which has a negligible impact in terms of data traffic. Also data size of exchanged messages in an instant messaging app is very low, making bots very attractive for users with limited smartphone data plan. Even Facebook Messenger has developed its Lite version for emerging markets, with limited Internet connections.

5.5.3 Advantages of Bots for developers

OS independence As seen for bots users advantages, OS independence is a greater opportunity also for developers: a single bot is suitable for all operating systems, as it has to be compliant only with the hosting messaging app interface. When dealing with native apps, a very big effort has to be done by developers who want to deploy their product on different platforms.

Network reliability Instant messaging applications are naturally designed for fast and efficient message delivery, providing full support to all network issues: message retry in case of lost connection, fail-safety, security, and so forth. On the other hand, apps need to manage the mobile internet connection instability relying only on the mobile platform APIs.

Cheap to develop As previously seen in the advantages for users, several important and time consuming services are already implemented by messaging platforms and ready to be accessed by developers via API. Services

like user authentication and payments require great efforts of implementation by developers. Bots have these services encapsulated and ready to use, meaning they are cheaper and faster to be developed than traditional apps.

Fast iteration Similar to what happens for web pages, bot updates are almost costless. Developers do not need to tackle new deploy and update issues.

Limited design efforts Bots heavily rely on the instant messaging application UI and have (until now, see Messenger WebView) quite limited possibilities of layout customization. This reduces the interface design time, limiting it to minor customization graphical interface elements, like buttons, carousels and so forth.

5.6 Apps vs Bots

As shown previously, the behavior of users in respect to mobile OS applications has crystallized. Recent statistics have shown that users tend to rely almost exclusively on a very restricted set of consolidated, preferred apps, instead of trying out new ones, leading the number of new downloaded apps nearly to zero in the short-medium period [123]. This has triggered a fight inside app stores, where app publishers try to emerge from this crowd developing apps that can attract new users, keep existing ones, and convert them into customers. A lot of efforts is put in the design, development, UX and promotions of new and existing apps. Moreover, the aforementioned report, has also shown the surprising overtake of online messaging platform over social network apps [123].

From this point of view, bots are being seen as the way to solve the issues in the app distribution system: playing where users already are, inside messaging apps, seems to be the easy way to avoid the efforts of continuously building the audience for new and old apps.

Nevertheless, given the many advantages of bots discusses in the previous section, there are drawbacks that must be taken into consideration.

Not everything can be done through bots, and even if this were not the case, there are tasks that inherently are more convenient way by means of a dedicated app with access to local computational resources.

Even if bot development is more approachable by developers, because of the many advanced features offered by the hosting platforms, the tight dependence on the environment—the messaging app—can possibly lead to several limitations in a bot’s functionalities. Also, there are inherent limits to the customization of UI elements, the look and feel, and the whole user experience. The lack of full customization can be a considerable disadvantage for companies that want to rely on bots as a brand support. The brand risks being perceived as part of the hosting platform, and not as an independent entity, hence compromising the brand’s recognizability.

Moreover, besides the availability of “bot directories”, discoverability is still a crucial issue. All the development platforms are contriving different solutions to help bots to be easily found by users. Furthermore, the risk of bot overpopulation, as it has been for apps, seems reasonably plausible.

Another possible limitation to bots comes from the necessity of an active Internet connection, as it happens for all other apps. In spite of the pervasiveness of connectivity, lack of network coverage, depletion of available traffic of data plans, the lack of accessible free WiFi connections, are everyday common nuisances. Especially in growing markets, the availability of a reliable network connection cannot be granted. This can create discontinuities in the bot’s availability and possibly a drop of its perceived reliability.

Also, people is likely to not always find themselves in proper situations to chat with certain bots, as example when the bot is requesting a voice interaction: in a conference, in a noisy square, in a school lesson, a conversation is not the best way to interact in some cases. Another aspect concerning human behavior, is the lack of propensity of people to chat with other actors, being humans or computers: some people dislike conversation and are not likely to choose bots to accomplish their tasks. On the flip side, there exists also people who dislike chatting with humans and enjoy interacting with a computer.

Therefore, even though some of these issues can be addressed and mitigated in the near future, there are inherent limitations to what bots can offer, that limit how far they can serve as a replacement to apps. Potentially, every app could be implemented with a bot, but the result could elicit an awkward user experience. As an example, editing a picture with a bot could be achieved by sending it the picture and then sending commands to apply filters, or worse, to cut out certain areas of the picture. A users would have

to specify the exact coordinates to select the areas and it would obviously become a terrible experience. Generally, tasks heavily requiring graphical interactions are more recommended for apps, able to supply totally graphical interactions.

5.7 Usability in the third wave of HCI

At the beginning of this chapter, a brief overview of the evolution of digital physical devices and interfaces have been done. As mentioned several times around different chapters, technology is now pervasive in everyday life, and people live in a digital ecosystem [4]: as defined by Bødker [10], we are now in the third-wave of HCI.

The *first wave*, as discussed by Bannon, was focused on human factors and cognitive science: HCI aim was to optimise the interaction between humans and computers, searching for problems and measuring solutions and improvements in terms of performances metrics, using formal methods, and systematic testing [5]. Typical context of application was workplace, with users performing well defined tasks, it was users-centric.

The *second wave* was about *human actors* [5]: professional communities working with a collection of applications, using computers to improve the work quality and overall effectiveness. Methods of evaluation were more group-centric: participatory design, prototyping and contextual enquiries, were added as HCI evaluation methods.

The transition to the *third wave* has happened with technology spread from workplace to everyday life [9]: use context and application types are broadened and intermixed. Nowadays, interface is everywhere, it is fragmented in the environment, causing the role of HCI being not so well defined, and old analysis methods to become obsolete. Different approaches to the new interfaces must be taken into considerations: human body should be seen as an interaction means (as previously seen, the body can be the interface, as it can be sensors' activated and produce data by moving or walking), and interactions distributed in a digital ecosystem [4]. It is difficult to optimise such a digital ecosystem, as there are uncountable and not always so clear goals for the interactions: the almost endless technological possibilities available nowadays, should move the attention on a more societal point of view, trying to understand what goals are desirable to reach with

these technologies.

From this point of view, the role of HCI is to ask questions, and try to understand which future problems and issues will have to be faced with the emerging technologies, and new artifacts.

According to Schumpeter's view of *innovation*, not only the discovery of a new technology is necessary, but also its social acceptance and adoption are needed to make it valuable. In this context, Norman and Verganti [100] define two main types of innovation processes (among others):

- Incremental innovation - as a progressive improvement of what already exists, "*doing better what we already do*".
- Radical innovation - as a change to the overall framework, "*starting to do what we were not doing before*".

Disruptive technologies, as for radical innovations have become very rare, as noted by Norman and Verganti, and not so uncommonly they are addressed to failure: technology is not powerful enough to support them, customers are not ready to appreciate it, the supporting infrastructure is not mature enough, unfavourable market conditions, and so forth. A recent example can be done with Google glasses: virtual and augmented reality are technologies studied and implemented since long time, and now quite well supported by technology. Google is one of the biggest player in the pervasive technology, and rumors preceding the launch of glasses were enthusiastic, and as soon as they have been started to be sold, they become somewhat a status symbol for geeks. Unfortunately after 6 months the project have been dismissed, and moved to another lab. What happened? Technology was not really ready, they were expensive and not so useful.

Most radical innovations are not so disruptive in short terms, and need to be supported by incremental innovation processes to succeed.

From this point of view, actual conversational interfaces, *bots* are a perfect example of incremental innovation, with a very promising future: technology was ready, as online messaging applications are consolidated, accepted, and the most used among mobile devices' holders. As aforementioned, market is ready, as the mobile apps crisis is happening, AI progresses, and big players of technology attention, are factors contributing to the *Next Big Thing* of this year.

Definitely, interface usability could be taken into account in the bots design and development, but also social and ethical aspects are emerging important facets. As previously noted, scientific literature is full of studies concerning typing users performances on touchscreens, suitable target size and position, screen size and so forth, related to mobile's apps. Some of the same studies could and should be applied to bots, but further issues are arising.

5.7.1 Can traditional Usability metrics and guidelines be used for bots?

Given previous considerations on the different *waves* of HCI, it is reasonable to argue how to address the usability issues of *modern bots*.

Firstly it should be considered that *modern bots* are not a radical innovation, as mentioned before: they share conversational interfaces and mobile applications characteristics, mixing graphical and functional elements, but have far more additional components, from their two-faced nature. *Modern bots* bring inside questions concerning natural language processing and artificial intelligence, whereas applications additional concerns about discoverability and learnability. Furthermore, modern bots' concerns can not even be reduced to older chatbots' ones.

Studies about *modern bots*' usability are also different from the ones on radical innovative artifacts, that can be seen as independent HCI theories, which can be abstracted from the artifact itself [6]. *Modern bots* cannot be compared to wearables, or to any of the IoT (Internet Of Things) components: they are, basically, a thread in an online messaging application, thus not entailing issues from user's acceptance.

Besides the advent of the so-called *third wave* of HCI, started right before the boom of mobile apps, usual HCI methods and techniques for usability studies could still be applied. Until "traditional" mobile applications, as demonstrated in previous chapters, evaluation and formulation of guidelines is still an active strand of research.

Nonetheless, with *modern bots* it makes sense to ponder if the traditional metrics (efficiency, effectiveness, speed of use, target size etc.) and methods (guidelines, task-oriented testing etc.) can still be considered valid, or as Bødker argues, more focus should be given to more "personal" metrics, such as meaning-making, or experience, or emotions, also following

the Norman's emotional design approach ([12], [99]).

In the light of these different approaches, because of the inherent mixed nature of *modern bots*, related usability issues could be addressed from two different point of views:

- applying existing usability guidelines, plucking from web, mobile, and chatbots experiences, starting from a more broad point of view, and expand this set with more specific ones. As an example, Nielsen has applied traditional usability techniques to WeChat, including bots [25].
- face new *modern bots* issues, checking if possible solutions are already present, or could be suited for this.

5.7.2 Applying traditional metrics

Trying to measure usability of *modern bots* with traditional methods, speculations can be done starting from the broader definition of usability, then proceeding with the "9 heuristics of Nielsen and Molich", trying to derive more specific comprehension.

Even though design customization is limited by the inherent nature of a conversational interface, and by the design elements provided by the platforms, bots interaction can be heavily influenced by designers and developers.

5.7.2.1 Usability definition for modern bots

Giving possible examples and solutions to most frequent misleading *modern bot* implementations, a broad overview is given.

Learnability

One of the main problems of some *modern bots* is learnability: in a *modern bot*, no custom window layout is available, no navigation item is generally provided in the beginning of a conversation, no icons, but interaction is generally started by the bot, either with a sentence to introduce itself, or a question to the user, in order to start the conversation.

With a limited possibility of interface customization, it can be tricky to make it clear what are the purposes and capabilities of a *modern bot* at first

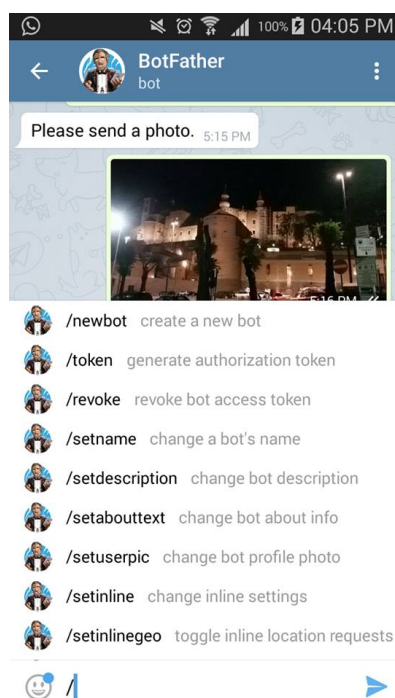


Figure 5.10: A list of commands accepted by BotFather in Telegram

glance. As for web, and for apps, the first 10 to 20 seconds are crucial for a user to decide whether to leave or to stay: same can be applied to *modern bots* [90]. The first impact, the comprehensibility with which a *modern bot* shows to users how to interact with it, is crucial to determine the *modern bots* success.

The way to provide good learnability depends on how the bot is thought to interact with users, according to developers' intent, and to which UI elements are offered by the hosting platform.

For example, if a bot is able to interact only through commands, after a first introduction, a list of available commands could be supplied, without letting the user guess which can produce a result, as shown in Figure 5.10. Alternatively a set of buttons, preceded by an explanation of what is the functionality intended for tapping any of them could be supplied.

Efficiency

Modern bots efficiency mainly depends on two factors:

- speed of bot's computation: meaning how long it takes to the bot in order to supply an answer to the user. There are bots with the aim of supplying complex solutions, or recommendations to the user, e.g. multiple, complete flight solutions, which require a not negligible amount of time.
- the number of interactions, question/answers, necessary to the user to reach the goal. E-commerce bots are particularly thorny: e.g. in order to buy a pair of shoes, a bot needs to gather quite a lot of information from the user, and supplying them one by one could be a cumbersome task, requiring many steps.

Modern bots requiring too much time to complete a task, are likely to result in an unpleasant interaction, and soon abandoned, most likely, by advanced users.

A compromise between answer complexity and completeness should be found, as well as a smart way of gathering numerous information in few steps.

Memorability

On one hand, memorability in *modern bots* can be helped by the conversation history: a user returning to a bot after some time, is still able to see past interactions, granting that the conversation has not been not deleted.

Furthermore, memorability depends on how much a user is guided in the conversation by the bot: a well designed bot, should never let the user feel lost, without knowing what is the next step to do; a bot helping the user in the interaction, does not even require the user to remember what steps to do, in order to reach their objective.

Thus, on the other hand, memorability is threatened for those *modern bots* leaving too much freedom to user, and for those which do not supply proper instructions.

It is very unlikely that users are able to remember exact understandable commands, and their features: e.g. a user going back to a bot supplying image search, after some time, do not probably wish to remember the different commands to ask for an animated gif, instead of an any format image. The bot should supply hints at any step of the interaction, and should encourage the conversation.

Errors

Catastrophic errors in *modern bots* are not likely to happen: the largest part of *modern bots* aims are dedicated to entertainment, customer service, and other light-weight tasks. More complex tasks, such as e-commerce purchase are maybe the most structured, and in general, bots entailing payments.

Failures should always be avoided, but where users behave differently with the bot from intended, a good error recognition and explanation should be provided. With the emotional side of bots, it is more likely users can get disappointed if not adequately advised and rescued from errors, and get negative feelings towards the bot.

Satisfaction

The main difference between an app and a *modern bot* is probably the “personality” of the latter one.

Besides the objective good performances of a *modern bot*, the increased “human” mimicking, can be a crucial factor: it has been shown for example, how the good feelings evoked by IKEA’s Anna, made “her” success.

Bots not able to answer to unexpected questions, could sometimes be perceived as dumb, and not trustworthy. Even bots not answering at all, or bots answering to wrong questions as if they were correct, are likely to be perceived as frustrating and annoying.

5.7.2.2 Nielsen’s 10 heuristics for modern bots

These heuristics have already been described in previous chapter, in Section 1.2.1 and have been applied, as a demonstration of use, to a famous mobile application. **still dunno which one, chapter has to be written**

Here the same will be done with two *modern bots*, living in two different platforms: *WeatherBot* [u35] a Telegram bot providing weather information; *Hipmunk* [u36] a Facebook Messenger bot, for a travel planning website [u37].

WeatherBot gives information about the current weather or the 3-days forecast, for a chosen location, letting the user chose among a series of recent locations, or of new ones. Location can be provided either by selecting

it from a map, or by geolocated city name.

Hipmunk is a travel planner, that lets users' choose among different types of solutions: users can look for a flight, or a hotel with specific dates and destination, or ask Hipmunk for solutions based on less specific options (trip type, flexible destination, best time to fly to a specific destination).

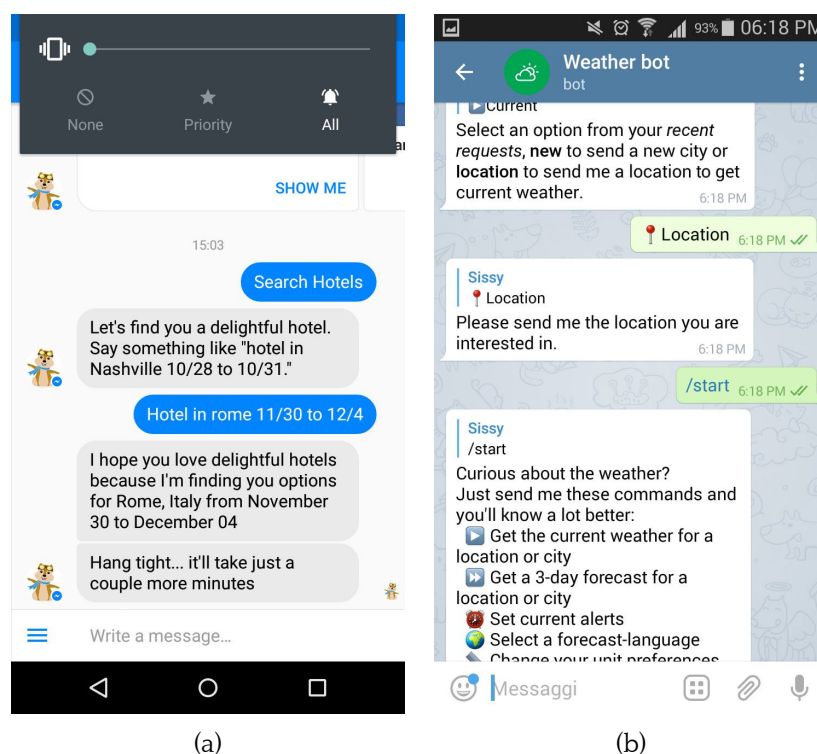


Figure 5.11: Different ways of showing to the user what the bot is doing or waiting for. In Figure 5.11a Hipmunk informs the user that it is working to find a solution, as the operation could last several seconds. In this case, the user can choose to wait or to move to another conversation; in any case, they will not feel lost in the interaction.

In Figure 5.11b, WeatherBot, in the case a location is not provided and the menu is closed, forces the user to restart the bot to obtain again the initial buttons with multiple choices. Users can get confused as the bot does not react, nor gives an information about what it is expecting from the user, which can result frustrating, and annoying.

Visibility of system status

The main activity with a *modern bot* is represented by a conversation, as in an online messaging app. The only information about the system status,

thus, concerns the conversation status, meaning a clear indication on who is supposed to speak, what the other subject is expecting from the partner, or what it is doing.

Modern bots should make it clear to the user if they are working on some request, or if they are unable to answer to some utterance: letting the user without an answer for long time, or not informing them on what they are expecting can make users feeling lost, or disappointed.

See Figure 5.11 for an example.

Match between system and the real world

Modern bots language is generally fairly clear, as it is inspired by users' language.

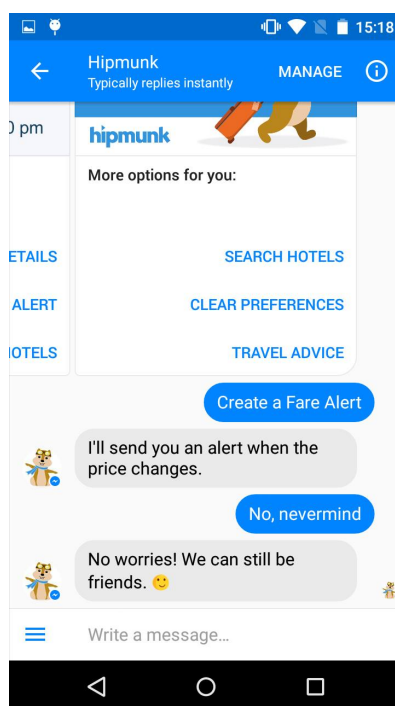


Figure 5.12: Hipmunk talks in a very friendly way, increasing empathy with the user.

Target users should carefully be studied, in order to provide appropriate phrases and words, customizing them according to some users' characteristics: bots whose audience is mostly constituted by teens should perhaps

use a different language from the ones dedicated to elderly. Moreover, style of conversation should be kept consistent all over the interaction.

Problems could be generated from the understanding of users' language, if these can interact with the bot by texting in natural language. Natural language processing, as previously seen, is a known source of problems of interaction, but in *modern bots* it is possible to bypass problems with the aim of predefined UI elements, preventing users from writing free utterances, thus avoiding misunderstanding.

As an example, see Figure 5.12.

User control and freedom

Another great difference between *modern bots* and traditional mobile apps is the lack of navigation, and the possibility of undoing and redoing. Interaction with *modern bots* follows a continuous flow, as a conversation does, even though some platforms (e.g. Messenger) allow the user to send a "delayed" command, meaning it is possible for the user to choose a proposed option from older ones, even if between there have been some more interactions.

For instance, with Hipmunks, if scrolling up the conversation, any voice from previous structured sent messages can be selected and the bot will still answer to that command. See Figure 5.13

Especially in long and fairly complex actions, like purchasing some items, it would be useful for the user to have granted the possibility of changing some provided options, as in a traditional web e-commerce, when changing an order options is feasible until before payment options.

As an example, see Figure 5.14.

Consistency and standards

The limited customization of *modern bots* design clearly favours consistency in their interface. Botmasters are limited to use pre-defined UI elements, with low possibility of tailoring them: this clearly reduces the possibility of creating confusing e.g. buttons and carousels without eliminating this risk. This cannot be said for commands: as these are totally free, it could happen that commands are not clearly explained or differentiated, commands with similar features but different names.

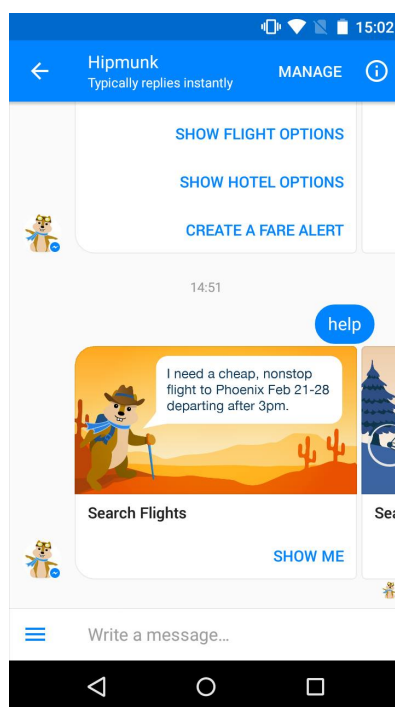


Figure 5.13: Even after some more requests and response, user is able to select one of the voices of a precedent structured message.

As an example of inconsistency, see Figure 5.15.

Error prevention

Considering the possible (and probable) misunderstanding generated by the interaction with natural language, as well as typos, or wrong tapping, users' errors should be thought as highly possible.

In order to prevent critical steps to be protected, one way could be to ask for confirmation before proceeding with an action.

Recognition rather than recall

As interaction with bots is fairly fragmented, as a conversation is, botmasters should remember users on their possibilities. For instance, during a fairly long conversation, a user could forget which possible commands can be sent to the bots: providing hints or examples through the conversation, could be a good way of helping users.

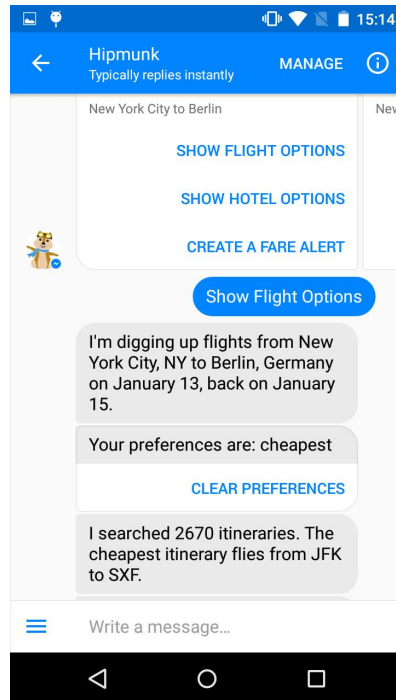


Figure 5.14: Hipmunk allows the user to clear previous preferences, giving a perception of control over the conversation going on.

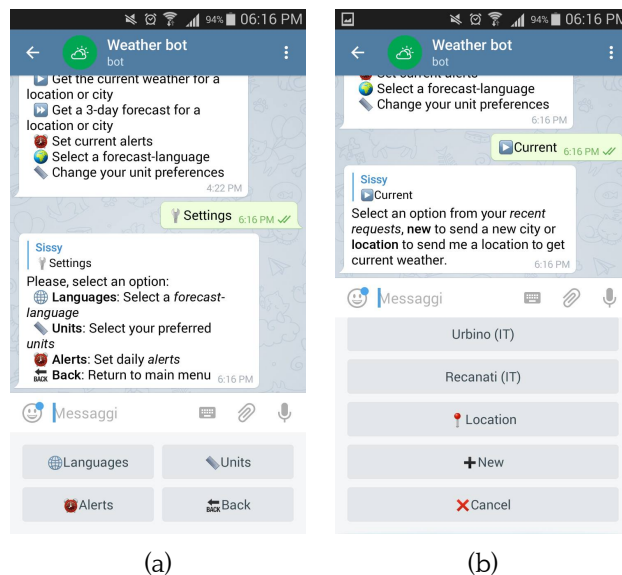


Figure 5.15: In WeatherBot there are two buttons with the same functionality, once called Back and once Cancel, different names and different icons: the user possibly thinks they have different aims, or different consequences, but they are the same.

WeatherBot proposes to the user a list of already asked cities, in order to avoid asking all the times the same information, or asking the user to remember. See figure 5.16

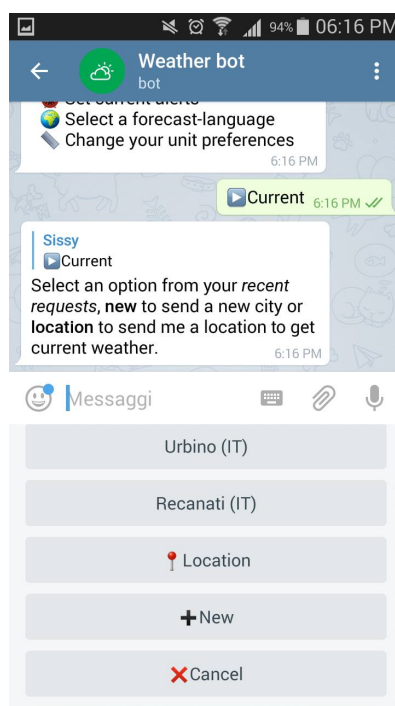


Figure 5.16: WeatherBot - list of recent locations

Flexibility and efficiency of use

In order to differentiate the interaction between experienced and new users, bots can easily offer different ways to achieve the same goal, for instance counting on the aid of predefined UI elements. Where an experienced user could directly insert a command to make a request, for a new user, more guided steps, via UI elements could be provided.

As an example of flexibility, see Figure 5.17.

Aesthetic and minimalist design

Dealing with *modern bots*, it should be kept in mind that they should be also endowed with some personality and often people has a propensity for conversation. Thus, even though a restricted set of information should be given when answering to users' request, it also should be considered the

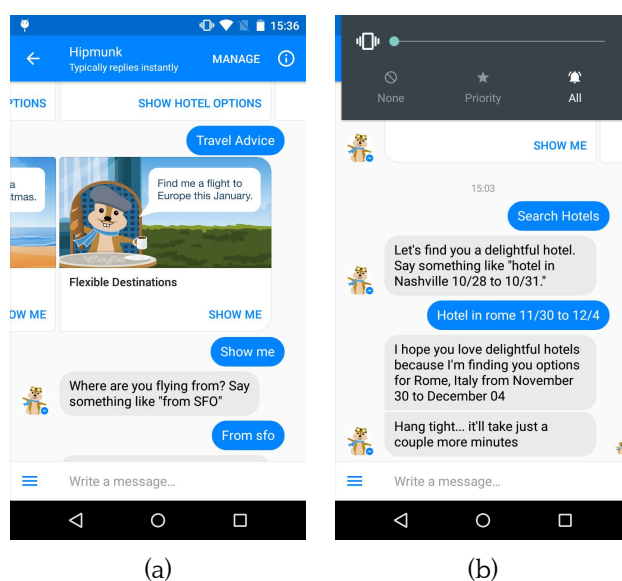


Figure 5.17: Hipmunk offers different types of interactions, depending on the level of the user: a new user can be guided to reach the goal, as in Figure 5.17a, whereas an experienced user can ask for multiple things in only one utterance Figure 5.17b

leisure aspect of the interaction: some people could enjoy a conversation out of the main goal of the bots, and bots should be able to contemplate also this type of requests.

See Figure 5.18, as an example.

Help users recognize, diagnose, and recover from errors

As for granting user control over the conversation, providing means for correcting wrong options of commands would be the good way to prevent the user from having a frustrating experience of use.

Moreover, answering to wrong requests in a “fancy” way, helps the user to keep the sense of control and not to consider a lack of bot’s intelligence.

See Figure 5.19 as an example of not-diagnosed errors.

Help and documentation

Several bots provide users with help command; others give hints and suggestions to users during the ongoing conversation. Even though it is also helpful to always have a designated command to receive some inline help.

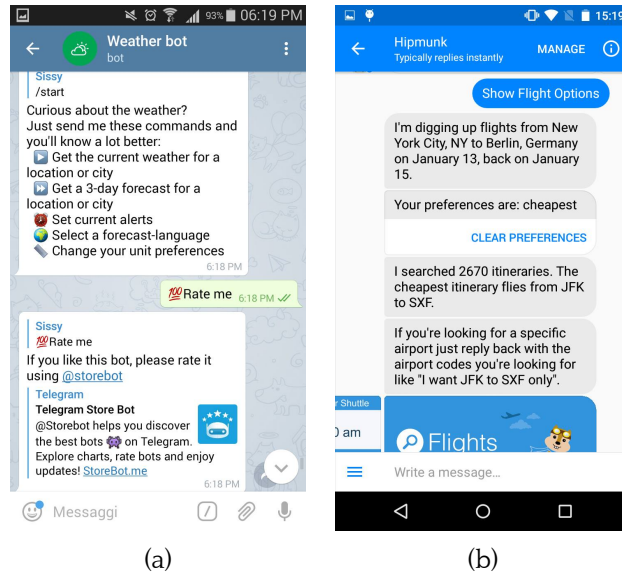


Figure 5.18: Both bots sometimes show too much propensity for conversation and have long explanation, maybe a bit redundant.

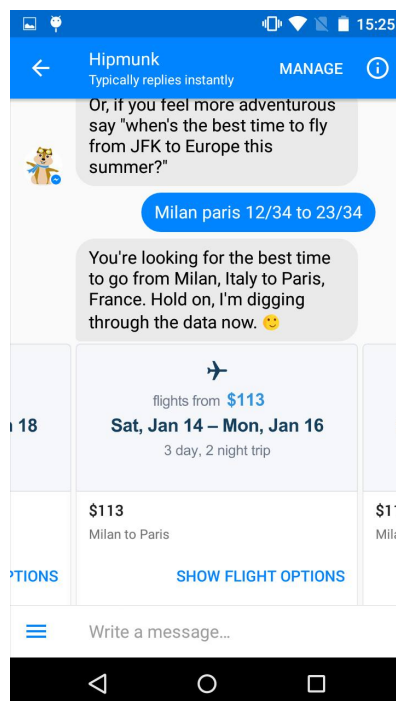


Figure 5.19: Even if user inserted non-existing dates, Hipmunk replies ignoring the misunderstood part of the utterance.

See Figure 5.20, as an example.

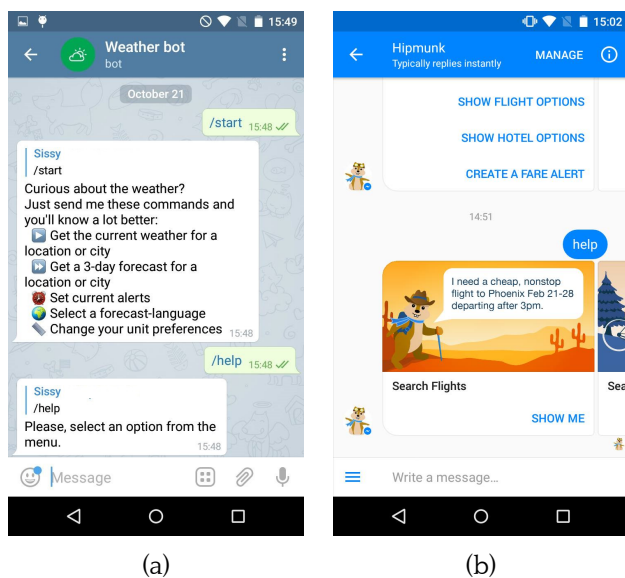


Figure 5.20: Both bots have easy access to some kind of documentation.

5.7.3 New issues beyond usability

Because of the inherent complex nature of *modern bots*, a larger spectrum of issues should be considered affecting user perception and experience when dealing with such agents.

Reputation

as for mobile applications, malicious *modern bots* and bots are real. With the rise of social bots, which produce content and interacts with humans on social media, many malicious behaviours and attacks have been registered in the last couple of years [34]. Lastly, this year, Microsoft's Tay has been revoked, after only 16 hours from its release, as it started posting racist and sexually-explicit tweets in response to other users [u38].

Other examples of harmful bots have been found spreading unverified information, or rumors. A notorious case has been the one concerning false accusations on Boston marathon [34]. Ferrara et al. have tried to develop a system to recognize bots from humans, with a discrete success. But they were talking about older bots, requiring more AI and possibly more inclined to be fooled by humans.

Something similar could happen with mobile apps, and the far more adopted solutions are the ones of authorized stores, like Apple Store and Windows store. Microsoft bot platform is doing the same. This gives perhaps more trustworthy bots, and perhaps it doesn't limit their spread, if there are enough developers who want to try.

As an alternative, a system for building *modern bot* reputation, feeded by the community reviews could be used, as already done in several bot stores.

Emotional issues

The ambition of developing a human-like bot which is able of fooling people during a conversation in natural language, is now more constrained to research: *modern bots* are known to not be humans from people, and users know they are not interacting with another human, but with a program.

Nevertheless, it is normal to confer some kind of consciousness to a device, even more if dealing with a program mimicking human behaviour: how many times people use to blame their smartphone for not connecting to the network, as if it was intentionally acting? The situation can be more emphasized if dealing with a program with human facets.

People dealing with a faulty *modern bot*, answering with wrong utterances, or worse, not answering at all, get sooner frustrated, or angry. On the other side, a bot unable to answer, can be considered to be dumb, and hence not influential.

From this point of view, of such ethical and emotional issues, particular attention must be put in designing pleasant bot reactions to unexpected situations.

Natural language processing

Troubles with natural language processing, either written or spoken, are well-known since the spread of older chatbots.

In *modern bots*, this issue can be strongly mitigated by the use of UI elements available from platforms, but the use of text to interact is still fairly used.

As previously seen, problems can arise from users asking multiple questions, or difficult ones, or questions bounded to particular situations for which bots are not prepared. All these possible situations result in frustrating user experiences, worsened by the emotional issues mentioned above.

In order to overcome this problem, developers should strongly rely on UI pre-defined elements, if not even get rid of texting.

Another issue comes from the inherent inefficiency of texting over speaking or simply tapping on the screen: number of characters should be limited as much as possible in order to increase the efficiency in interaction.

Discoverability

Since the spread of bot platforms, *modern bots* are booming, as it has been for apps almost ten years ago. Their number is rapidly increasing, and it is becoming usual to find a *modern bot* in any category, for any kind of task.

This obviously raises an issues on the discoverability of boplications, as it is for apps.

Almost all bot platforms make available *bot directories*, like app stores, but the risk of getting lost and of overcrowding is the same.

Thus, having learnt a lesson from apps, the sprawling development, release and distribution of *modern bots* should be avoided by companies and developers, before a point of saturation is reached, as it is happening now with apps.

5.7.4 Discussion

From previous analysis of *modern bots* usability some speculations can be done.

It is fairly clear that the ongoing debate, on how to approach usability studies of emerging technologies, is far from receiving a unique answer, and to be solved. However this seems fairly reasonable, as hardware and software interfaces have been rapidly and dramatically evolving in the last ten years.

At the beginning of usability studies, there was a single device, the personal computer, mostly used for working tasks, and only later for entertainment, that needed usability studies. With the spread of smartphones, apps, tablets, *modern bots*, wearables, and so forth, the combinations of types of interaction and interfaces have exponentially grown, making the quantity of possible contexts of use, and variables to be considered in usability studies, incredibly wide.

Moreover, above considering emotional design, more than task-oriented studies, also ethnographic and cultural factors of use have to be examined,

and taken into account as an independent variable.

Hence the different approaches, the more traditional one, still based on the analysis of usability meant as the efficiency, effectiveness and satisfaction of use, versus a broader one, more oriented to the context and motivations of use, could be considered as complementary.

As shown above, traditional usability studies are still applicable to emerging artifacts, but it is not probably enough to understand how to increase their possible success, nor to interpret possible failures.

Thus, it is not conceivable to reinvent usability metrics and research methods for any new emerging technologies, but a more farsighted point of view should be found, in order to be able to apply it as soon as a new artifact is proposed to the great public, or even before proposing it. This could guarantee to launch in the market fairly mature and desirable technologies, instead of trying too late to adjust a possibly failing technology.

Conclusions and future works

In this work an ample overview on the usability and serviceability of data and services through mobile interfaces has been presented. Notably, the necessity of new usability studies techniques, to address the continuous and rapid changes and evolution of hardware and software interfaces, has been investigated.

As initially seen, usability studies for software systems and desktop web sites have been heavily stressed over the past three decades, letting the web to reach an acceptable degree of usability. Expert studies, for the development of usability guidelines, and user testing, for the validation of existing interfaces, have produced plenty of tools to help developers building better interfaces.

Different techniques for the usability evaluation have been discussed, together with the principal areas covered by structured guidelines sets. It has been argued that there exists a large disparity between available tools for web interfaces and those for mobile ones, particularly a lack of structured sets of guidelines: in ten years of mobile apps dissemination, guidelines are still lacking.

Moreover, observing the ever and rapid emergence of new ways of supplying information to users, this issue becomes more constraining. Thus, another necessity that has been identified is to better pinpoint what are the more promising emerging interfaces, and if traditional methods can be adapted to those ones.

For the first problem, an approach based on gamification has been suggested. A dedicated mobile game app has been designed and implemented, in order to study a specific usability problem: the screen reachability, in relation to screen size and device grip. The aim was to collect a large amount of crowd-produced data in a cheap and fast way, and use them to evaluate

existing guidelines on the same matter, and help producing new ones.

More notably, the aim was to investigate whether this approach could be a good one to support existing traditional methods. The reachability problem has been addressed to study the existing relation between screen size and device grip. Results have shown, as expected, that the increasing screen size degrades user performance, and that different device grips have an observable impact too, on user speed and accuracy.

Results have also confirmed previous studies regarding target size and position, and the higher performance of the index finger, when compared to the thumb. These results show that this data collection approach can be a valuable method to validate existing similar guidelines, and possibly produce new ones, more specific for different type of devices.

In order to address the problem of understanding which emerging interfaces can be more promising and how to address new challenges in usability, a deeper look has been given to the evolution of hardware and software interfaces, with special attention to conversational interfaces. It has been seen how the transformation of online messaging platforms into environments for the easy and fast development of bots has given a pulse to a renewed interest in this type of conversational interfaces. Due also to the mobile apps crisis and to the overtaking of messaging apps on social networking ones, these old-new conversational agents have drawn large attention in the last year, leading to ask whether they will be able to substitute apps.

A detailed study on the principal characteristics, advantages, and disadvantages of modern bots has been given, together with a comparison with mobile apps. The resulting discussion led to the consideration that modern bots are not likely to fully substitute apps, but that it is important to soon understand emerging artifacts, in order to make them fully valuable, and to discern their relations with existing technologies.

Furthermore, it has been attempted to apply a traditional usability evaluation method, such as the heuristic evaluation, in order to understand whether these methods can valuably be adapted to emerging artifacts.

The preeminent conclusion of this study is the necessity of developing new tools to address the dawning challenges raised from new hardware and software interfaces. Even though existing methods are still valuable to develop new guidelines, they need to be supported by new methods allowing to (1) collect usability data in faster and cheaper way, in order to adapt to

ever evolving existing technologies; (2) soon understand what the future of emerging technologies can be, in order to make them successful. The latter point is probably even more crucial, considering the fast evolution of hardware devices, ubiquitous computing, and the increasing pervasiveness of technology in everyday life. On the software side, each hardware device has specific purposes and interactions, thus also more usability issues arise.

Hence, further studies are necessary, to better understand the actual panorama and envisage the future one. Concerning this study, further works should be conducted to improve the mobile app game, extending it with further small games, able to investigate other usability problems. Collecting more data and developing specific metrics to evaluate them could possibly be a good method to sustain existing usability techniques, as shown throughout this thesis. Future works, could try to also address different types of devices, such as tablets or smartwatches, with the same approach based on gamification.

Further studies should also be conducted on emerging conversational interfaces, as a promising alternative or support to mobile apps. A more comprehensive study on the existing modern bots, with the development of a taxonomy and a better scope definition, should help to understand their future and conceivably possible future evolutions in mobile interfaces, and the different ways in which users will have access to data and services.

Online references

This section serves as a bibliography for online references, including different web resources.

It has been added in order to distinguish URLs of common web sites, from bibliography with scientific relevance. All the URLs have been accessed in the month of November 2016, before the thesis submission.

[u1] Eyequant (Sec: 1.2) - <http://www.eyequant.com/>

[u2] FLUD (Sec: 1.2) - <http://zing.ncsl.nist.gov/cifter/TheCD/WebTools/Flud/Readme.html>

[u3] WAI (Sec: 1.2.2) - <https://www.w3.org/WAI>

[u4] Available apps in Play Store (Sec: 1.4) - <https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/>

[u5] Google Material design (Sec: 1.4) - <https://material.google.com/>

[u6] Apple guidelines (Sec: 1.4) - <https://developer.apple.com/ios/human-interface-guidelines/>

[u7] Adobe Flash Player (Sec: 1.4) - <http://www.adobe.com/products/flashplayer.html>

[u8] Threadless (Sec: 2.1) - www.threadless.com

[u9] iStockPhoto (Sec: 2.1) - www.istockphoto.com

[u10] Innocentive (Sec: 2.1) - <https://www.innocentive.com/>

- [u11] KickStarter (Sec: 2.1) - <https://www.kickstarter.com/>
- [u12] Amazon Mechanical Turk (Sec: 2.1) - <https://www.mturk.com/mturk/>
- [u13] oDesk (Sec: 2.1) - <https://www.upwork.com/>
- [u14] CrowdFlower (Sec: 2.1) - <https://www.crowdfunder.com/>
- [u15] FourSquare (Sec: 2.2.3) - <https://www.foursquare.com/>
- [u16] ShareTribe (Sec: 2.2.3) - <https://www.sharetribe.com/>
- [u17] Ruby On Rails (Sec: 3.3) - <http://rubyonrails.org/>
- [u18] Microsoft's "Guideline for targeting" (Sec: 4.2.1) - <https://msdn.microsoft.com/windows/uwp/input-and-devices/guidelines-for-targeting>
- [u19] Source for online messaging apps image (Fig: 5.3) - <http://static3.uk.businessinsider.com/image/57238507dd08958f388b46aa-960/mavssn.png>
- [u20] Source for ALICE brain image (Fig: 5.4) - <http://www.alicebot.org/documentation/gallery/bigpat.jpg>
- [u21] Apple Siri (Sec: 5.3.3) - <http://www.apple.com/it/ios/siri/>
- [u22] Microsoft Cortana (Sec: 5.3.3) - <https://support.microsoft.com/en-us/help/17214/windows-10-what-is>
- [u23] Google Assistant (Sec: 5.3.3) - <https://assistant.google.com/>
- [u24] Amazon Alexa (Sec: 5.3.3) - <http://alexa.amazon.com/spa/index.html>
- [u25] Samsung S Voice (Sec: 5.3.3) - <http://www.samsung.com/>
- [u26] Kik (Sec: 5.3.4) - <https://dev.kik.com/>
- [u27] Facebook Messenger (Sec: 5.3.4) - <https://developers.facebook.com/docs/messenger-platform>
- [u28] Telegram (Sec: 5.3.4) - <https://core.telegram.org/bots>
- [u29] Skype (Sec: 5.3.4) - <https://www.skype.com/en/developer/>
- [u30] Line (Sec: 5.3.4) - <https://developers.line.me/messaging-api/overview>
- [u31] WeChat (Sec: 5.3.4) - <http://dev.wechat.com/wechatapi>

[u32] Slack (Sec: 5.3.4) - <https://api.slack.com/bot-users>

[u33] Markdown (Sec: 5.4) - <http://daringfireball.net/projects/markdown/>

[u34] How users feel about push notifications (Sec: 5.5.2) -

<http://info.localytics.com/blog/the-inside-view-how-consumers-really-feel-about-push-notifications>

[u35] Weatherbot (Sec: 5.7.2.2) - <http://botsfortelegram.com/project/weather-bot/>

[u36] Hipmunk Bot (Sec: 5.7.2.2) - <https://www.messenger.com/t/hipmunk/>

[u37] Hipmunk website (Sec: 5.7.2.2) - <https://www.hipmunk.com/>

[u38] On bot reputation (Sec: 5.7.3) - <http://www.bbc.com/news/technology-35890188>

Bibliography

- [1] Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg, and Jure Leskovec. Steering user behavior with badges. In *Proceedings of the 22nd international conference on World Wide Web*, pages 95–106. ACM, 2013. 49
- [2] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer networks*, 54(15):2787–2805, 2010. 105
- [3] Shiri Azenkot and Shumin Zhai. Touch behavior with different postures on soft smartphone keyboards. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services*, pages 251–260. ACM, 2012. 60, 81, 82, 85, 95, 110
- [4] Sebastiano Bagnara and Simone Pozzi. The third wave of human computer interaction: From interfaces to digital ecologies. *DigitCult*, 1(1), 2016. 125
- [5] Liam Bannon. From human factors to human actors: The role of psychology and human-computer interaction studies in system design. *Design at work: Cooperative design of computer systems*, 25:44, 1991. 125
- [6] Liam J Bannon and Susanne Bødker. Beyond the interface: Encountering artifacts in use. *DAIMI Report Series*, 18(288), 1989. 127
- [7] A Barredo. A comprehensive look at smartphone screen size statistics and trends, 2014. 54
- [8] Jerome R Bellegarda. Spoken language understanding for natural interaction: The siri experience. In *Natural Interaction with Robots, Knowbots and Smartphones*, pages 3–14. Springer, 2014. 104, 111

- [9] Susanne Bødker. When second wave hci meets third wave challenges. In *Proceedings of the 4th Nordic Conference on Human-computer Interaction: Changing Roles*, NordiCHI '06, pages 1–8, New York, NY, USA, 2006. ACM. ISBN 1-59593-325-5. doi: 10.1145/1182475.1182476. URL <http://doi.acm.org/10.1145/1182475.1182476>. 125
- [10] Susanne Bødker. Third-wave hci, 10 years later—participation and sharing. *interactions*, 22(5):24–31, 2015. 3, 125
- [11] Susanne Bødker and Ellen Christiansen. Poetry in motion: appropriation of the world of apps. In *Proceedings of the 30th European conference on cognitive ergonomics*, pages 78–84. ACM, 2012. 4, 36
- [12] Susanne Bødker and Clemens Nylandsted Klokmoose. Dynamics in artifact ecologies. In *Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design*, pages 448–457. ACM, 2012. 128
- [13] Matthias Böhmer, Brent Hecht, Johannes Schöning, Antonio Krüger, and Gernot Bauer. Falling asleep with angry birds, facebook and kindle: a large scale study on mobile application usage. In *Proceedings of the 13th international conference on Human computer interaction with mobile devices and services*, pages 47–56. ACM, 2011. 33
- [14] Sebastian Boring, David Ledo, Xiang'Anthony' Chen, Nicolai Marquardt, Anthony Tang, and Saul Greenberg. The fat thumb: using the thumb's contact size for single-handed mobile interaction. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services*, pages 39–48. ACM, 2012. 59, 95
- [15] Nis Borneo and Jan Stage. Usability engineering in the wild: How do practitioners integrate usability engineering in software development? In *International Conference on Human-Centred Software Engineering*, pages 199–216. Springer, 2014. 18
- [16] Daren C Brabham. Crowdsourcing as a model for problem solving an introduction and cases. *Convergence: the international journal of research into new media technologies*, 14(1):75–90, 2008. 37, 38

- [17] Barry Brown, Stuart Reeves, and Scott Sherwood. Into the wild: challenges and opportunities for field trial methods. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1657–1666. ACM, 2011. 5, 37, 50, 52
- [18] Raluca Budiu. Mobile user experience: limitations and strengths. *Nielsen Norman Group*, 19, 2015. 5, 31
- [19] Raluca Budiu and Jakob Nielsen. Usability of ipad apps and websites. *Useit.com*, 2011. 30, 34
- [20] Daniel Buschek, Alexander De Luca, and Florian Alt. Evaluating the influence of targets and hand postures on touch-based behavioural biometrics. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 1349–1361, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-3362-7. doi: 10.1145/2858036.2858165. URL <http://doi.acm.org/10.1145/2858036.2858165>. 60, 85, 88
- [21] Justine Cassell, Timothy Bickmore, Mark Billinghurst, Lee Campbell, Kenny Chang, Hannes Vilhjálmsón, and Hao Yan. Embodiment in conversational interfaces: Rea. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 520–527. ACM, 1999. 103
- [22] Monica Anderson PEW Research Center Survey. Cellphones, computers are the most commonly owned devices. 29th October 2015, available at: <http://www.pewinternet.org/2015/10/29/technology-device-ownership-2015/> (Accessed January, 2017), 2015. 96
- [23] Muhammad Benny Chaniago and Apri Junaidi. Student presence using rfid and telegram messenger application. In *8th Widyatama International Seminar on Sustainability (WISS 2016)*. Widyatama University and IEEE, 2016. 107
- [24] Jenova Chen. Flow in games (and everything else). *Communications of the ACM*, 50(4):31–34, 2007. 40, 42
- [25] Yunnuo Cheng and Jakob Nielsen. Wechat: China's integrated internet user experience. *useit.com: Jakob Nielsen's Website*, 2011. 128

- [26] World Wide Web Consortium et al. Web content accessibility guidelines (wcag) 2.0. *World Wide Web Consortium*, 2008. 24
- [27] Martin Cooper, Richard W Dronsuth, Albert J Leitich, Jr Charles N Lynk, James J Mikulski, John F Mitchell, Roy A Richardson, and John H Sangster. Radio telephone system, September 16 1975. US Patent 3,906,166. 54
- [28] Mihaly Csikszentmihalyi. Flow: The psychology of optimal performance. *NY: Cambridge University Press*, 1990. 40
- [29] Kristen Dergousoff and Regan L Mandryk. Mobile gamification for crowdsourcing data collection: Leveraging the freemium model. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 1065–1074. ACM, 2015. 38, 51
- [30] Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. From game design elements to gamefulness: defining gamification. In *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*, pages 9–15. ACM, 2011. 5, 39, 67
- [31] Trinh Minh Tri Do, Jan Blom, and Daniel Gatica-Perez. Smartphone usage in the wild: a large-scale analysis of applications and context. In *Proceedings of the 13th international conference on multimodal interfaces*, pages 353–360. ACM, 2011. 33
- [32] Tom Farley. Mobile telephone history. *Teletronikk*, 101(3/4):22, 2005. 93
- [33] Rosta Farzan and Peter Brusilovsky. Encouraging user participation in a course recommender system: An impact on user behavior. *Computers in Human Behavior*, 27(1):276–284, 2011. 48
- [34] Emilio Ferrara, Onur Varol, Clayton Davis, Filippo Menczer, and Alessandro Flammini. The rise of social bots. *arXiv preprint arXiv:1407.5225*, 2014. 140
- [35] Zachary Fitz-Walter, Dian Tjondronegoro, and Peta Wyeth. Orientation passport: using gamification to engage university students. In *Proceedings of the 23rd Australian Computer-Human Interaction Conference*, pages 122–125. ACM, 2011. 48

- [36] Carlos Flavián, Miguel Guinalíu, and Raquel Gurrea. The role played by perceived usability, satisfaction and consumer trust on website loyalty. *Information & Management*, 43(1):1–14, 2006. 20
- [37] FA Fonte, Juan Carlos, Burguillo Rial, and Martín Llamas Nistal. Tq-bot: an aiml-based tutor and evaluator bot. *Journal of Universal Computer Science*, 15(7):1486–1495, 2009. 102
- [38] Mayank Goel, Jacob Wobbrock, and Shwetak Patel. Gripsense: using built-in sensors to detect hand posture and pressure on commodity mobile phones. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pages 545–554. ACM, 2012. 59
- [39] Herman H Goldstine and Adele Goldstine. The electronic numerical integrator and computer (eniac). *Mathematical Tables and Other Aids to Computation*, 2(15):97–110, 1946. 92
- [40] Bettina Graf, Maike Krüger, Felix Müller, Alexander Ruhland, and Andrea Zech. Nombot: simplify food tracking. In *Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia*, pages 360–363. ACM, 2015. 106
- [41] Anton Gustafsson, Cecilia Katzeff, and Magnus Bang. Evaluation of a pervasive game for domestic energy engagement among teenagers. *Computers in Entertainment (CIE)*, 7(4):54, 2009. 41, 49
- [42] Juho Hamari. Transforming homo economicus into homo ludens: A field experiment on gamification in a utilitarian peer-to-peer trading service. *Electronic commerce research and applications*, 12(4):236–245, 2013. 48
- [43] Juho Hamari, Jonna Koivisto, and Harri Sarsa. Does gamification work?—a literature review of empirical studies on gamification. In *2014 47th Hawaii International Conference on System Sciences*, pages 3025–3034. IEEE, 2014. 36, 37, 43, 47, 52
- [44] Rachel Harrison, Derek Flood, and David Duce. Usability of mobile applications: literature review and rationale for a new usability model. *Journal of Interaction Science*, 1(1):1, 2013. 35

- [45] Charles T Hemphill, John J Godfrey, and George R Doddington. The atis spoken language systems pilot corpus. In *Proceedings of the DARPA speech and natural language workshop*, pages 96–101, 1990. 103
- [46] Niels Henze, Martin Pielot, Benjamin Poppinga, Torben Schinke, and Susanne Boll. My app is an experiment: Experience from user studies in mobile app stores. *International Journal of Mobile Human Computer Interaction (IJMHCI)*, 3(4):71–91, 2011. 5, 14, 39, 51, 52, 89
- [47] Niels Henze, Enrico Rukzio, and Susanne Boll. 100,000,000 taps: Analysis and improvement of touch performance in the large. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services, MobileHCI '11*, pages 133–142, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0541-9. doi: 10.1145/2037373.2037395. URL <http://doi.acm.org/10.1145/2037373.2037395>. 5, 39, 51, 61, 64, 67, 82
- [48] Niels Henze, Enrico Rukzio, and Susanne Boll. Observational and experimental investigation of typing behaviour using virtual keyboards for mobile devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '12*, pages 2659–2668, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1015-4. doi: 10.1145/2207676.2208658. URL <http://doi.acm.org/10.1145/2207676.2208658>. 39, 61
- [49] Christian Holz and Patrick Baudisch. Understanding touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '11*, pages 2501–2510, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0228-9. doi: 10.1145/1978942.1979308. URL <http://doi.acm.org/10.1145/1978942.1979308>. 58
- [50] Steven Hooper. How do users really hold mobile devices? 2013. <http://www.uxmatters.com/mt/archives/2013/02/how-do-users-really-hold-mobile-devices.php>, 2013. 53, 57
- [51] Jeff Howe. The rise of crowdsourcing. *Wired magazine*, 2006. 37
- [52] John A Hoxmeier and Chris DiCesare. System response time and user satisfaction: An experimental study of browser-based applications. *AMCIS 2000 Proceedings*, page 347, 2000. 32

- [53] Paul Huber. Inaccurate input on touch devices relating to the fingertip. *Media Informatics Proseminar on "Interactive Surfaces"*, 2015. 31
- [54] Kai Huotari and Juho Hamari. Defining gamification: a service marketing perspective. In *Proceeding of the 16th International Academic MindTrek Conference*, pages 17–22. ACM, 2012. 39
- [55] Adrian Iftene and Jean Vanderdonckt. Moocbuddy: a chatbot for personalized learning with moocs. In *ROCHI-INTERNATIONAL CONFERENCE ON HUMAN-COMPUTER INTERACTION*, page 91, 2016. 107
- [56] W ISO. 9241-11. ergonomic requirements for office work with visual display terminals (vdts). *The international organization for standardization*, 45, 1998. 9, 35
- [57] Anjul Jain, Diksha Bhargava Bhargava, and Anjani Rajput. Touch-screen technology. *International Journal of Advanced Research in Computer Science and Electronics Engineering (IJARCSEE)*, 2(1):pp-074, 2013. 4, 55, 56
- [58] Philippe Jeanrenaud, Greg Cockroft, and Allard VanderHeidjen. A multimodal, multilingual telephone application: the wildfire electronic assistant. In *EUROSPEECH*, 1999. 105
- [59] Alan C. Kay. A personal computer for children of all ages. In *Proceedings of the ACM Annual Conference - Volume 1*, ACM '72, New York, NY, USA, 1972. ACM. doi: 10.1145/800193.1971922. URL <http://doi.acm.org/10.1145/800193.1971922>. 92
- [60] Ian R Kerr. Bots, babes and the californication of commerce. *University of Ottawa Law and Technology Journal*, 1(1-2):20004, 2004. 100, 102
- [61] Sunjun Kim, Jihyun Yu, and Geehyuk Lee. Interaction techniques for unreachable objects on the touchscreen. In *Proceedings of the 24th Australian Computer-Human Interaction Conference, OzCHI '12*, pages 295–298, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1438-1. doi: 10.1145/2414536.2414585. URL <http://doi.acm.org/10.1145/2414536.2414585>. 59

- [62] Jesper Kjeldskov and Connor Graham. A review of mobile hci research methods. In *International Conference on Mobile Human-Computer Interaction*, pages 317–335. Springer, 2003. 5, 50, 52, 57
- [63] Wolfgang Kluth, Karl-Heinz Krempels, and Christian Samsel. Automated usability testing for mobile applications. In *WEBIST (2)*, pages 149–156, 2014. 19
- [64] Stefan Kopp, Lars Gesellensetter, Nicole C Krämer, and Ipke Wachsmuth. A conversational agent as museum guide—design and evaluation of a real-world application. In *International Workshop on Intelligent Virtual Agents*, pages 329–343. Springer, 2005. 103
- [65] Steve Krug. *Don't make me think: Web Usability: Das intuitive Web*. MITP-Verlags GmbH & Co. KG, 2014. 10
- [66] Michael O Leavitt and Ben Shneiderman. Research-based web design & usability guidelines. *US Department of Health and Human Services*, 2006. 4, 15, 16, 18, 21, 24, 25, 28
- [67] Adam Lella and Andrew Lipsman. The us mobile app report. *21st August, available at: <http://www.comscore.com/Insights/Presentationsand-Whitepapers/2014/The-US-Mobile-App-Report> (Accessed November, 2016)*, 2014. 8, 13, 33, 34, 97
- [68] Adam Lella and Andrew Lipsman. The us mobile app report. *21st August, available at: <http://www.comscore.com/Insights/Presentationsand-Whitepapers/2016/The-US-Mobile-App-Report> (Accessed January, 2017)*, 2016. 11, 12
- [69] Florian Lettner and Clemens Holzmann. Automated and unsupervised user interaction logging as basis for usability evaluation of mobile applications. In *Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia*, pages 118–127. ACM, 2012. 19
- [70] Janne Lindqvist, Justin Cranshaw, Jason Wiese, Jason Hong, and John Zimmerman. I'm the mayor of my house: examining why people use foursquare—a social-driven location sharing application. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 2409–2418. ACM, 2011. 47

- [71] Di Liu, Randolph G Bias, Matthew Lease, and Rebecca Kuipers. Crowdsourcing for usability testing. *Proceedings of the American Society for Information Science and Technology*, 49(1):1–10, 2012. 38
- [72] Desmond Lobo, Kerem Kaskaloglu, ChaYoung Kim, and Sandra Herbert. Web usability guidelines for smartphones: a synergic approach. *International journal of information and electronics engineering*, 1(1):33, 2011. 10, 32
- [73] I Scott MacKenzie, Shawn X Zhang, and R William Soukoreff. Text entry using soft keyboards. *Behaviour & Information technology*, 18(4): 235–244, 1999. 110
- [74] RP Mahapatra, Naresh Sharma, Aakash Trivedi, and Chitransh Aman. Adding interactive interface to e-government systems using aiml based chatterbots. In *Software Engineering (CONSEG), 2012 CSI Sixth International Conference on*, pages 1–6. IEEE, 2012. 102
- [75] Brad McCarty. The history of the smartphone. *TNW Network All Stories RSS*. Np, 6, 2011. 55
- [76] Michael McTear, Zoraida Callejas, and David Griol. *The Conversational Interface*. Springer, 2016. 6, 94, 102, 103, 104, 111
- [77] Katina Michael. Science fiction is full of bots that hurt people:... but these bots are here now. *IEEE Consumer Electronics Magazine*, 5(4): 112–117, 2016. 99
- [78] Fernando A Mikic, Juan C Burguillo, Martín Llamas, Daniel A Rodríguez, and Eduardo Rodríguez. Charlie: An aiml-based chatterbot which works as an interface among ines and humans. In *EAAEIE Annual Conference, 2009*, pages 1–6. IEEE, 2009. 102
- [79] Markus Montola, Timo Nummenmaa, Andrés Lucero, Marion Boberg, and Hannu Korhonen. Applying game achievement systems to enhance user experience in a photo sharing service. In *Proceedings of the 13th International MindTrek Conference: Everyday Life in the Ubiquitous Era*, pages 94–97. ACM, 2009. 49

- [80] Benedikt Morschheuser, Juho Hamari, and Jonna Koivisto. Gamification in crowdsourcing: a review. In *2016 49th Hawaii International Conference on System Sciences (HICSS)*, pages 4375–4384. IEEE, 2016. 36, 38
- [81] Computer History Museum. Timeline of computer history: memory and storage. <http://www.computerhistory.org/timeline/computers/>, 2016. Accessed: 2016-11-21. 92
- [82] Josip Musi and Roderick Murray-Smith. Nomadic input on mobile devices: the influence of touch input technique and walking speed on performance and offset modeling. *Human-Computer Interaction*, pages 1–52, 2015. 51, 61, 81
- [83] Alexander Ng, Stephen A. Brewster, and John H. Williamson. Investigating the effects of encumbrance on one- and two- handed interactions with mobile devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 1981–1990, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2473-1. doi: 10.1145/2556288.2557312. URL <http://doi.acm.org/10.1145/2556288.2557312>. 17, 60
- [84] J Nielsen. Mobile sites vs. apps: The coming strategy shift. *Fremont, CA: Nielsen Norman Group: UX Training, Consulting, & Research*. Retrieved November 2016., 27:2012, 2012. 30
- [85] Jakob Nielsen. Enhancing the explanatory power of usability heuristics. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 152–158. ACM, 1994. 21
- [86] Jakob Nielsen. Guerrilla hci: Using discount usability engineering to penetrate the intimidation barrier. *Cost-justifying usability*, pages 245–272, 1994. 20
- [87] Jakob Nielsen. Heuristic evaluation. *Usability inspection methods*, 17(1): 25–62, 1994. 21
- [88] Jakob Nielsen. *Usability engineering*. Elsevier, 1994. 10, 11, 13, 17, 19, 24
- [89] Jakob Nielsen. *Designing web usability: The practice of simplicity*. New Riders Publishing, 1999. 8

- [90] Jakob Nielsen. How long do users stay on web pages. *useit.com: Jakob Nielsen's Website*, 2011. 34, 129
- [91] Jakob Nielsen. Usability 101: Introduction to usability (2012). URL: <http://www.nngroup.com/articles/usability-101-introduction-to-usability/>[Accessed November 2016], 2012. 9, 35
- [92] Jakob Nielsen. Nielsen norman group. Internet: <https://www.nngroup.com/>, November 2016. 24
- [93] Jakob Nielsen and Raluca Budiu. *Mobile usability*. MITP-Verlags GmbH & Co. KG, 2013. 5, 34
- [94] Jakob Nielsen and Hoa Loranger. *Prioritizing web usability*. Pearson Education, 2006. 3, 8, 15, 20, 24
- [95] Jakob Nielsen and Rolf Molich. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 249–256. ACM, 1990. 8, 18, 21
- [96] Mie Nørgaard and Kasper Hornbæk. What do usability evaluators do in practice?: an explorative study of think-aloud testing. In *Proceedings of the 6th conference on Designing Interactive systems*, pages 209–218. ACM, 2006. 18
- [97] Donald Norman. The design of everyday things (originally published: The psychology of everyday things). *The Design of Everyday Things (Originally published: The psychology of everyday things)*, 1988. 20
- [98] Donald A Norman. *Emotional design: Why we love (or hate) everyday things*. Basic books, 2005. 13, 14
- [99] Donald A Norman. *The design of everyday things: Revised and expanded edition*. Basic books, 2013. 10, 63, 128
- [100] Donald A Norman and Roberto Verganti. Incremental and radical innovation: Design research vs. technology and meaning change. *Design Issues*, 30(1):78–96, 2014. 126
- [101] Kenneth M Ovens and Gordon Morison. Forensic analysis of kik messenger on ios devices. *Digital Investigation*, 17:40–52, 2016. 113

- [102] Pekka Parhi, Amy K. Karlson, and Benjamin B. Bederson. Target size study for one-handed thumb use on small touchscreen devices. In *Proceedings of the 8th Conference on Human-computer Interaction with Mobile Devices and Services*, MobileHCI '06, pages 203–210, New York, NY, USA, 2006. ACM. ISBN 1-59593-390-5. doi: 10.1145/1152215.1152260. URL <http://doi.acm.org/10.1145/1152215.1152260>. 57, 58, 87, 88
- [103] Yong S. Park and Sung H. Han. One-handed thumb interaction of mobile devices from the input accuracy perspective. *International Journal of Industrial Ergonomics*, 40(6):746 – 756, 2010. ISSN 0169-8141. doi: <http://dx.doi.org/10.1016/j.ergon.2010.08.001>. URL <http://www.sciencedirect.com/science/article/pii/S0169814110000806>. 57, 64
- [104] Jeremy Peckham. Speech understanding and dialogue over the telephone: an overview of the esprit sundial project. In *HLT*, 1991. 103
- [105] Keith B. Perry and Juan Pablo Hourcade. Evaluating one handed thumb tapping on mobile touchscreen devices. In *Proceedings of Graphics Interface 2008*, GI '08, pages 57–64, Toronto, Ont., Canada, Canada, 2008. Canadian Information Processing Society. ISBN 978-1-56881-423-0. URL <http://dl.acm.org/citation.cfm?id=1375714.1375725>. 58, 67, 82
- [106] Steven Reiss. Multifaceted nature of intrinsic motivation: The theory of 16 basic desires. *Review of General Psychology*, 8(3):179, 2004. 41
- [107] Dmitry Rudchenko, Tim Paek, and Eric Badger. Text text revolution: A game that improves text entry on mobile touchscreen keyboards. In *Proceedings of the 9th International Conference on Pervasive Computing*, Pervasive'11, pages 206–213, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 978-3-642-21725-8. URL <http://dl.acm.org/citation.cfm?id=2021975.2021993>. 62
- [108] Antonella Santangelo, Agnese Augello, Antonio Gentile, Giovanni Pilato, and Salvatore Gaglio. A chat-bot based multimodal virtual guide for cultural heritage tours. In *PSC*, pages 114–120, 2006. 102

- [109] Md Shahriare Satu, Md Hasnat Parvez, et al. Review of integrated applications with aiml based chatbot. In *2015 International Conference on Computer and Information Engineering (ICCIIE)*, pages 87–90. IEEE, 2015. 102
- [110] Stephanie Seneff and Joseph Polifroni. Dialogue management in the mercury flight reservation system. In *Proceedings of the 2000 ANLP/NAACL Workshop on Conversational Systems - Volume 3, ANLP/NAACL-ConvSyst '00*, pages 11–16, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics. doi: 10.3115/1117562.1117565. URL <http://dx.doi.org/10.3115/1117562.1117565>. 103
- [111] Abu Shawar, Eric Atwell, and Andrew Roberts. Faqchat as in information retrieval system. In *Human Language Technologies as a Challenge for Computer Science and Linguistics: Proceedings of the 2nd Language and Technology Conference*, pages 274–278. Poznan: Wydawnictwo Poznanskie: with co-operation of Fundacja Uniwersytetu im. A. Mickiewicza, 2005. 102
- [112] BA Shawar and E Atwell. *A comparison between Alice and Elizabeth chatbot systems*. University of Leeds, School of Computing research report 2002.19, December 2002. 102
- [113] Bayan Abu Shawar and Eric Atwell. Chatbots: are they really useful? In *LDV Forum*, volume 22, pages 29–49, 2007. 102
- [114] Maria Shitkova, Justus Holler, Tobias Heide, Nico Clever, and Jörg Becker. Towards usability guidelines for mobile websites and applications. In *Wirtschaftsinformatik*, pages 1603–1617, 2015. 34, 35, 96
- [115] Andreas Sonderegger and Juergen Sauer. The influence of laboratory set-up in usability tests: effects on user performance, subjective ratings and physiological measures. *Ergonomics*, 52(11):1350–1361, 2009. 16, 50
- [116] Simon Thorpe, Denis Fize, Catherine Marlot, et al. Speed of processing in the human visual system. *nature*, 381(6582):520–522, 1996. 78
- [117] Alan M Turing. Computing machinery and intelligence. *Mind*, 59(236): 433–460, 1950. 99

- [118] Richard Wallace. The elements of aiml style. *Alice AI Foundation*, 2003. 101
- [119] Richard S Wallace. The anatomy of alice. In *Parsing the Turing Test*, pages 181–210. Springer, 2009. 101
- [120] Daniel Walnycky, Ibrahim Baggili, Andrew Marrington, Jason Moore, and Frank Breitingner. Network and device forensic analysis of android social-messaging applications. *Digital Investigation*, 14:S77–S84, 2015. 113
- [121] Joseph Weizenbaum. Eliza - a computer program for the study of natural language communication between man and machine. *Commun. ACM*, 9(1):36–45, January 1966. ISSN 0001-0782. doi: 10.1145/365153.365168. URL <http://doi.acm.org/10.1145/365153.365168>. 100
- [122] Joel West and Michael Mace. Browsing as the killer app: Explaining the rapid success of apple’s iphone. *Telecommunications Policy*, 34(5): 270–286, 2010. 4
- [123] Will McKittrick. The messaging app report: How instant messaging can be monetized. Technical report, BusinessInsider, 2015. 5, 6, 33, 97, 123
- [124] Maximilian Witt, Christian Scheiner, and Susanne Robra-Bissantz. Gamification of online idea competitions: Insights from an explorative case. *Informatik schafft Communities*, 192, 2011. 49
- [125] Jinghong Xiong and Satoshi Muraki. An ergonomics study of thumb movements on smartphone touch screen. *Ergonomics*, 57(6):943–955, 2014. 58
- [126] Rosa Yáñez Gómez, Daniel Cascado Caballero, and José-Luis Sevillano. Heuristic evaluation on mobile interfaces: A new checklist. *The Scientific World Journal*, 2014, 2014. 35
- [127] Rui Zhu and Zhongzhe Li. An ergonomic study on influence of touch-screen phone size on single-hand operation performance. In *MATEC Web of Conferences*, volume 40. EDP Sciences, 2016. 61, 87, 90

-
- [128] Gabe Zichermann and Christopher Cunningham. *Gamification by design: Implementing game mechanics in web and mobile apps.* " O'Reilly Media, Inc.", 2011. 40, 41, 42, 43, 47
- [129] Victor W Zue and James R Glass. Conversational interfaces: Advances and challenges. *Proceedings of the IEEE*, 88(8):1166–1180, 2000. 103

Acknowledgments

I would like to express my special appreciation and thanks to my advisor Professor Alessandro Bogliolo for giving me the opportunity of running research on topics to which I am really attached and always wanted to deepen, and hopefully will continue to work on them in the future.

For being always there, present whenever I needed support outside my studies, for never letting me feeling alone, nor abandoned, one of my fundamental points of my life, on which I know I can count in any moment: my beloved family. Even though he's not here anymore, he will always be alive in my heart, thanks to my grandfather too: it is also because of him if I can be here.

For his strength and patience in standing these last difficult months beside me, for helping me finding true happiness since we met, and also in the worst moments ever, but simply for who you are, and hopefully will ever be, next to me, for the rest of our lives, Simone.

Beyond the family in which we were born, there are friends, the true ones, that after some time are no more friends, but become a part of your life, and memories. Molly, for still being here after more than 15 years, for our nightly calls and delirious messages in our insane moments, because friendship, as we know and demonstrate, is not a matter of being always next to each other. Italia, Giorgia e Enrico, aka 'the family', for all the laughs, the crazy nights, and all the times they have listened to me and my paranoia, and for being still here, in one way or another. Because friendship does not need to wait for long to become special, as they are.

A big thanks goes to the other people "living" in the same place in which I've spent the largest part of these last two years: my fellow labmates. Lorenz for his great support in correcting this thesis and for being always ready to help me and give me the right advice. Saverio, Andrea, Brendan and Matteo for their technical support and for tolerating all my eccentricities.

Lastly I cannot forget my Ph.D. fellows, Maria, Caterina, and Laura for sharing hilarious and frustrating moments, that only a Ph.D. student can fully understand: I am happy to have shared tears and pleasures of a Ph.D. with you.