INSTITUT FÜR INFORMATIK der Ludwig-Maximilians-Universität München

DESIGNING AN **ASYNCHRONOUS IMAGE** LABELING GAME FOR MOBILE DEVICES

Stefanie Kloß

Bachelorarbeit

Betreuer

Aufgabensteller Prof. Dr. François Bry Prof. Dr. François Bry, Martin Bogner

Abgabe am

26.02.2020

Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig verfasst habe und keine anderen als die angegebenen Hilfsmittel verwendet habe.

München, den 26.02.2020

Stefanie Kloß

ii

Abstract

In this thesis, an existing image labeling game aimed at desktop users is adapted to suit the technical characteristics of mobile devices and the behavioural specificities of their users. The games fall within the category of games with a purpose which aim to solve computational problems while providing players with a fun experience. When creating a game with a purpose, the enjoyability of the game and the quality of the gathered data constitute two main pillars to its success. Information on how to design enjoyable games is gained from the gaming industry. The asynchronous nature of the game concepts of popular multiplayer games for mobile devices as well as their graphical user interfaces form guidelines for the design process. To implement a certain data quality and, in the case of image labeling games, filter the labels collected by the game, verification techniques are taken a look upon. Based on all this information and design constraints for mobile applications, the desktop version of the game, that is played in one session and simultaneously by two players, is transformed into a game concept which breaks the game session apart into several sections and, thus, enables asynchronous turns. That is how a player can take his turn time-independent of his teammate. The resulting game, which is given the name DICO, is reachable through the browser and implemented as a single-page application that communicates with the back-end of the Artigo platform. The final game concept can be extended in many ways. This work ends with suggestions on how to improve the game in the future.

iv

Zusammenfassung

In dieser Arbeit wird ein schon existierendes Image-Labeling-Spiel, das sich an Desktop-Nutzer richtet, an die technischen Eigenschaften mobiler Endgeräte und die Verhaltensbesonderheiten ihrer Nutzer angepasst. Die Spiele fallen in die Kategorie der Games with a Purpose, die darauf abzielen, rechnerische Probleme zu lösen und den Spielern gleichzeitig eine unterhaltsame Beschäftigung zu bieten. Bei der Entwicklung eines Game with a Purpose bilden der Unterhaltungsfaktor und die Qualität der gesammelten Daten zwei wesentliche Säulen für den Erfolg des Spiels. Informationen über die Gestaltung unterhaltsamer Spiele werden aus der Spieleindustrie gewonnen. Die asynchronen Spielkonzepte beliebter Multiplayer-Spiele für mobile Endgeräte sowie deren grafische Benutzeroberflächen bilden ebenfalls Richtlinien für den Designprozess. Um eine gewisse Datengualität zu erreichen und im Falle von Image-Labeling-Games die vom Spiel gesammelten Labels zu filtern, werden Verifikationstechniken betrachtet. Basierend auf all diesen Informationen und den gestalterischen Anforderungen, Einschränkungen und Konventionen mobiler Anwendungen wird die Desktop-Version des Spiels, die in einer Sitzung und parallel von zwei Spielern gespielt wird, in ein Spielkonzept transformiert, welches die Spielsitzung in mehrere Abschnitte aufteilt und so asynchrone Runden ermöglicht, um einen Spieler zeitunabhängig von seinem Mitspieler seinen Zug ausführen zu lassen. Das so entstandene Spiel mit dem Namen DICO ist über den Browser erreichbar und als singlepage Applikation implementiert, die mit dem zur Artigo-Plattform gehörenden Backend kommuniziert. Das finale Spielkonzept kann in vielerlei Hinsicht erweitert werden. Die Arbeit endet mit Vorschlägen, wie das Spiel in Zukunft verbessert werden kann.

vi

Acknowledgments

Most of all, I thank Martin Bogner for the incredibly supportive mentoring of my thesis. Thank you for always taking the time to answer my questions, for leading me in the right direction and for continuously contributing many helpful ideas and thoughts.

Finally, I want to say thank you to my family and, especially, Nico for the patience, the mental support and the proof-reading.

First, I would like to thank Prof. Dr. François Bry for all his suggestions on how to improve my project. His enthusiastic outlook on the possibilities resulting from the combination of art and modern computation is truly contagious. Prof. Dr. Bry has the ability to think out of the box and transfer concepts to other contexts and constantly encouraged me to do so as well.

viii

Contents

1	Introduction 1											
2	Related Work											
	2.1	2.1 The Artigo Platform										
	2.2	Games with a Purpose and Related Concepts	4									
		2.2.1 Human Computation	4									
		2.2.2 Gamification	5									
		2.2.3 Motivators Behind GWAPs	5									
	2.3	Designing a GWAP	6									
		2.3.1 Enjoyability	6									
		2.3.2 Data Verification	8									
	2.4	Design Constraints for Mobile Applications	9									
		2.4.1 Mobile versus Desktop – a Comparison	9									
		2.4.2 Pictures on Small Screens	10									
		2.4.3 Learning from Existing Mobile Apps	12									
3	Karido											
	3.1	Game Concept	15									
	3.2	Input Similarity	16									
	3.3	Data Quality	17									
		3.3.1 Scoring	17									
		3.3.2 Tag Verification	18									
		3.3.3 Security Gaps	18									
	3.4	Simulated Player	18									
	3.5	Rationale for a Mobile Version	19									
4 Dico - Design Choices and Their Justification												
-	4.1	Game Concept	21									
	4.2	Rules	24									
	4.3	Data Quality	25									
	1.0	4.3.1 Scoring	25									
		4.3.2 Tag Verification	26									
		4.3.3 Security Gaps	27									
	4.4	Comparison to Karido	28									

CONTENTS

5	Imp 5.1 5.2	Plementation of Dico Front-End Back-End 5.2.1 Important Classes 5.2.2 Challenge: Multiple Game Sessions per User	29 31 32 35					
6	Con 6.1 6.2 6.3 6.4	Analyzing the EnjoyabilityAnalyzing the Data QualityPossible ExtensionsOutlook	37 37 38 39 40					
Appendix								
Bibliography								

x

CHAPTER 1

Introduction

Play has always been a part of human culture [20]. Nowadays, digital games are a popular source of play. People like to be entertained by facing and trying to overcome challenges with no real risk attached. Their motivation is most often intrinsic: Playing is rewarded by the enjoyable experience granted when noticing the own improvement, getting affirmative feedback from the game and reaching goals set either by the game or oneself. Gamers appreciate the mental stimulation as well as the relaxation and stress relief triggered by playing [5]. Luis von Ahn, a pioneer in the field of human computation, suggests using the concept of games to let people solve computational problems. His idea is to profit from the many hours spent gaming with the help of so-called *games with a purpose*. Games with a purpose are designed around such a problem and produce side effects that take place when playing the game. These side effects (i.e. the saving of data) are used for the solution-finding. Therefore, a person plays a game with a purpose to benefit from the enjoyable experience while unknowingly contributing to problem-solving. Providing the games on the Internet makes them accessible to millions of potential players. Von Ahn states the following:

"For the first time in human history, hundreds of millions of people can, via the Internet, easily collaborate on the same problem." [2, p.98]

A traditional field of games with a purpose is the labeling of images. It constitutes a pillar of the cooperation between the Institutes of Art History and Computer Science of the Ludwig-Maximilians-Universität München that led to the development of an online platform called Artigo. The platform provides a number of games that collect data describing artefacts. The descriptions are stored in a database in the form of labels and serve as a basis for a search engine over the artefacts. This search engine is hosted by the Artigo platform, too. It is getting improved constantly due to the continuous operation of the games and, thus, growing number of labels.

Artigo was conceived for desktop devices in the late 2000s. However, mobile devices have gained in importance in the last few years. Smartphones have become a staple in many people's day to day life. Leaving the house without one's smartphone is comparable with forgetting the keys or the wallet. Mobile phones have become far more than just a tool for communication. Through mobile devices, the Internet is accessible anywhere.

With the possibility to smoothly access the mobile web and an increasing screen size, today's smartphones make browsing a decent experience. In 2019, 47.96% of the global web traffic originates from smartphones [38]. Mobile devices are used for social networking, productivity and entertainment. During idle periods, people tend to reach for their phones and games are a welcome and popular source for amusement and distraction. Recently, a turning point regarding the most used playing device could be observed: In 2019, the smartphone has passed the personal computer as most commonly used device for gamers in the USA [4, 5]. This development shows just how popular mobile games are.

The website of Artigo is currently designed for desktop users only. The implementation of the various available games as well as their user interfaces are unsuitable for the use on mobile devices. However, considering the trend of mobile web traffic, almost every other website call is made from a mobile device. Thus, more potential players could be reached by adapting Artigo's gaming ecosystem to mobile visitors. This would increase the number of games played and, in turn, positively influence the quality of the search engine.

In this thesis, the image labeling game KARIDO, that is hosted on http://artigo.org, is being redesigned to create a game specifically aimed at mobile users. Chapter 2 covers the related work and presents the above mentioned concept of games with a purpose in more detail. The aspects of enjoyability and data verification, which are of particular importance for games with a purpose, are incorporated along the way. Behavioural characteristics of mobile users and technical specificities of mobile devices are taken a look at to adapt the game concept in a way that meets the needs of mobile users and to create an user interface tailored for small touch screens. The process of a redesign is untwisted and put into steps, so that this chapter acts as a manual for the design of mobile games with a purpose. Chapter 3 describes the basic idea and concept of the desktop version of KARIDO in detail. Core mechanisms such as the ones implemented to achieve a certain data quality are presented. In the next chapter, the game concept for a mobile version of KARIDO is developed. The game is named DICO. All design choices are made building on the previous chapters. The chapter contains the current rules of the game and ends with a comparison of the desktop and mobile version. The implementation using the client-side framework React.js and the server-side framework Express.js within the runtime environment of Node.js is introduced in Chapter 5. Finally, Chapter 6 draws a conclusion of the new game concept and fields that require – or can benefit from – future work are revealed. The thesis ends with an outlook on possible extensions to the game and on alternative contexts to which DICO can be applied to and profited from.

CHAPTER 2

Related Work

To start off, the hosting platform of KARIDO, Artigo, is introduced in more detail. It makes use of a concept central to this thesis – *games with a purpose*. Having a concrete realization in mind, an explanation of the idea of GWAPs, which brings together the underlying concepts of *human computation* and *gamification*, follows. Afterwards, a closer look is taken at the procedure of developing a game concept for a GWAP. Same as a classical game, GWAPs aspire to create an enjoyable user experience. However, the game concept of a GWAP additionally sets focus on data generation and verification. In the last section, design constraints that are specific for mobile devices are presented. To do so, the evolution of mobile internet traffic is regarded and mobile users are compared to their desktop counterparts. Popular gallery as well as multiplayer apps are introduced to give an idea about best practices considering the design of mobile game concepts and their graphical user interfaces.

2.1 The Artigo Platform

The Artigo platform¹ was developed as part of *Play4Science*, a research project by humanists, computer scientists and computational linguists of the Ludwig-Maximilians-Universität München started in 2007 [25]. Artigo consists of two main parts: A search engine for artefacts² and a gaming ecosystem. The available games are based on the exact artworks that make up the foundation of the search engine, as both interact with the same database. All game concepts were designed to continuously improve the Artigo search engine. To do so, each game collects annotations³ describing the artefacts as a by-product of playing. The goal is to not only characterize the pictures' main visible contents but also unique details and sentiments they evoke. To optimize the diversification of the tags, the games cover complementary areas: Simple *description games* collect basic tags of the works of art, whilst *diversification games* provide the search engine with more specific annotations and such of a deeper semantic. *Integration games* grant the possibility to cluster labels in order to create even preciser descriptions – with *Tag a Tag*, for example, it is possible to combine a label with related annotations. Detailed information about the individual games

¹http://artigo.org

²In the following work, the terms *artefact, artwork* and *work of art* are used as synonyms.

³The terms *tag*, *label* and *annotation* are used as synonyms in this thesis. The term *description* is used to refer to a number of tags.

of Artigo and how they work can be found in [7].

The game KARIDO is classified as a diversification game. It is designed for two people that are shown the same nine images of artworks. While Player A (the Describer) describes one of these pictures, Player B (the Guesser) has to guess which artwork is being described to him. The nine displayed images are selected with an algorithm choosing images with similar annotations. Therefore, the Describer needs to come up with tags that differentiate the selected artwork from the others in order to ease the decision process of the Guesser [42]. As of right now, all of Artigo's games are targeted at desktop users only. Mobile users are confronted with unadjusted user interfaces and game concepts which require a longer period of time to focus, since all games are intended to be played in one session and without breaks. The Artigo platform is, therefore, currently unsuitable for mobile devices.

2.2 Games with a Purpose and Related Concepts

The idea behind games with a purpose (GWAP) is simple yet brilliant: Enjoyable games use the players' inputs to solve computational problems. The work aspect is hidden, making solution finding a pure side effect of the game. Users get rewarded with a fun experience whilst contributing to solving a given issue. GWAPs combine the two concepts of *human computation* and *gamification* that are described below.

2.2.1 Human Computation

Even though artifical intelligence and machine learning algorithms are constantly improving, there are still tasks extremly complex to solve by a processor that a human brain can tackle within seconds [45, 14]. That is due to the different approaches of man and machine: Humans act judgement-focused opposed to the prediction-focused artifical intelligence [1]. Describing images is a popular example: Whilst a person can differantiate between millions of objects in a matter of milliseconds, machine learning algorithms still get fooled by blueberry muffins resembling Chihuahuas [8]. Until today, letting people label images manually is the best method for creating meaningful descriptions [45]. When dealing with a large amount of pictures, only a big community of people working together can master the task of labeling them in a reasonable time span. This approach is called *human computation*, as the role of the participants is similar to that of processors [45] in "traditional" computation. Jain and Parkes summarize the concept as follows.

"The paradigm of "human computation" seeks to harness human abilities to solve computational problems or otherwise perform distributed work that is beyond the scope of current AI technologies." [23, p.58]

The Artigo search engine was developed with this concept in mind in order to solve the problem of artwork labeling. A major difference between human and machine solution finding is the aspect of motivation. For humans, problem solving is considered work. Either extrinsic or intrinsic incentives are needed to get people commited to a task. A typical extrinsic motivator is money. However, mobilizing the amount of people needed to solve a large-scale problem by paying them constitutes a significant monetary factor [45, 7]. In some cases, the funds for such an investment are missing – the sector of art history, where the labeling of images is of interest [7], is just one example. More creativity is needed to get people to work for intrinsic reasons. This is where gamification comes in handy.

2.2.2 Gamification

By adding game design elements, all kinds of non-game services can be turned into gameful versions of themselves [9]. Unlike traditional games, the game design elements are not used for the pure enjoyment of players, but for positively influencing the intrinsic motivation and attitude of the service's customers [9, 16]. As users are ultimately the creators of value, highly engaged customers lead to an increased value creation [21]. The gamification provider profits from the value – may it be money or data as in the case of Artigo. Known by the term gamification, this concept is a trending topic in both industry and academia [21, 16]. In their paper dedicated to defining gamification, Huotari and Hamari conclude:

"Gamification refers to: a process of enhancing a service with affordances for gameful experiences in order to support user's overall value creation" [21, p.19]

The above definition focuses on the goal of gamification as opposed to giving instructions on how to implement a gameful experience. However, the inclusion of games in service systems does not automatically result in gamification as it is required that the gamified service enhances value creation. Huotari and Hamari [21] also emphasise the differentiation between a *gamification provider*, the *core service* benefiting from the value creation, an *enhancing service* supporting the core service and a *gamified service*. In the context of this paper, the platform Artigo acts as gamification provider. Its core service is to provide a search engine for artworks and the continuous labeling of these artworks represents the enhancing services. Finally, the implemented GWAPs such as KARIDO constitute the gamified service.

In conclusion, Artigo makes use of both concepts, human computation and gamification. Therefore, the games provided by Artigo are games with a purpose.

2.2.3 Motivators Behind GWAPs

Luis von Ahn, a pioneer in the field of human computation and developer of the *ESP Game*⁴, describes the following factors as main motivators of the GWAP approach [2]:

Internet Access

A constantly rising number of people have access to the internet. In three years time, the count of internet users in Europe increased by over 100 million people, reaching a total of 704 million people in 2018 [39]. Portable devices enable their users to enter the world wide web on the go. In addition, they are a more affordale alternative to desktop computers and laptops, making the internet accessible to a wider part of society [14].

Limits of Computational Classification

As mentioned above, some tasks are easy to solve for humans yet highly complex for computers [45]. Translating words depending on the context they are used in or describing sounds and images as well as the feelings they evoke are relatively simple tasks for a human brain. Yet, these problems remain troublesome for computers.

Game Consumption

Humans wish to be entertained [2]. Playing video or computer games is widely seen as an enjoyable activity and people willingly spend hours of their free time doing so [5]: the average German citizen spents almost eight hours per week gaming [22].

⁴The ESP Game is a popular GWAP based on which Google developed the Google Image Labeler. Read more about the ESP Game here [45].

Combining these factors unveils the vast potential of GWAPs. If all the time and energy people commit to gaming was channeled, large-scale problems like the labeling of all images in the internet could be solved within weeks [45, 13].

To collectively solve an extensive problem, it needs to be deconstructed into a large number of facile tasks. Participants should be able to work on these with a minimal amount of previous knowledge and training [13]. To create data in a game, an input-output behaviour is implemented. The input is the data a game session is centred around. It is passed to a game session often randomly and can be seen as the assets of the game session. The output is the data newly created by the player that depends on the respective input. The two major challenges of a GWAP are to provide a fun experience whilst making sure the output is correct.

2.3 Designing a GWAP

Once the decision is set, that a GWAP is the appropriate method to solve a problem, it is time to approach the design and development in a structured manner. Several factors are associated with the quality and popularity of a GWAP. Being aware of their impact at an early state helps to create a smooth developing process.

Grace and Jamieson [14] describe three ways on how to take the first step in development. The **problem determinant approach** starts out with a problem to be solved. Then, a game is designed around it. In contrast, the **game determinant approach** applies a problem to an earlier agreed upon game. The third pattern is a mixture of both: Beginning with a given problem, one trys to find suitable gaming mechanisms from already existing games. This **design by analogy** offers the possibility to rely on already established, well-proven mechanisms. The last two procedures mentioned can reduce the complexity of development significantly, as the video game industry has accumulated knowledge about exciting games over the last 35 years [14]. Working with this knowledge can be of great advantage as it provides information about what users want from games.

When creating a game concept, it is useful to know the difference between a play and a game. According to Grace and Jamieson [14], the five elements required to call a play system a game are the following:

- *competition* The aim of a game is competition and to have competition, a goal is needed. An opponent is not necessary to realize said goal.
- *implements* With the help of implements, the player overcomes obstacles in order to win.
- *territory* Physical and theoretical parameters setting boundaries and limits belong to the attribute of territory.
- *inventory* Items collected during the play are referred to as inventory.
- *rules* Rules define the usage of the four other attributes and establish a relationship between them.

Deconstructing an existing game/play concept in the five mentioned categories can reveal design flaws [14].

2.3.1 Enjoyability

The worth and success of a GWAP is highly correlated with the amount of people playing it [2]. Consumers only stay as long as they find the experience satisfying [14]. It doesn't matter how thoughtful the logic behind a game nor how accurate the gathered data is when no

2.3. DESIGNING A GWAP

one is playing. Thinking of Artigo, the challenge lies in designing a game that transforms the monotonous job of describing images into an entertaining and fun game. Hence, creating an enjoyable and satisfying experience is of top priority. Only then it is to focus on the quality of the computed outputs.

In his study about the design of highly motivating instructional environments [28], Malone identifies three main categories of features implying entertaining concepts, that are applicable to any gameful environment:

Challenge

Setting a well-defined, challenging goal and giving the player useful performance feeback in real-time are key charecteristics of any game [28], as goal setting is highly related to the performance of the player. Goals work in many ways: They focus the person's energy on achieving an objective, have an energizing function and improve his persistence [26]. After 35 years of research on goal-setting theory, Locke and Latham [26] conclude that the highest levels of effort and performance are produced when having set the highest/most difficult goals. Even if people don't think they are able to reach a goal, they will still perform better the higher it is set. However, this requires a high commitment to the goal as the outcome is uncertain. The goal-setting practices found in the context of gaming differ from the just mentioned theory. According to works of Malone [28] as well as von Ahn [2], the outcome of a game should be uncertain to prevent players from either getting bored or frustrated. If the goal is set too low, players are not challenged enough and, therefore, get bored soon. On the other hand, a goal seemingly impossible to reach discourages players. In both cases, the customers are left unsatisfied. Locke and Latham agree in this point: They, too, found people with the highest goals (and consequentially highest performance) to be the least satisfied [26]. An explanation for the differences of goal-setting in a gaming environment could be the goal commitment. People act because they strive for the satisfaction anticipated from their actions [46]. To make someone commited to a goal, the outcome has to seem like an appropriate reward. At a workplace, for example, there are many ways to enhance the goal commitment. Support of leaders, adequate training and monetary incentives are just some of them [26]. Games are more restricted when it comes to reinforcing a player's engagement. The measures taken to reward players for their actions are mostly performance feedbacks like scores, sound cues and banners. In conclusion, it is advisable to go for Malone's and von Ahn's recommendation of a balanced goal-setting when it comes to games, but to set the goal rather too high than too low.

Fantasy

In this context, fantasies describe the mental images and feelings conjured by a video game. A specific fantasy can be highly motivating for some people whilst being demotivating for others. Therefore, fantasies are to be chosen very carefully and have to be adapted to the target audience. Besides being emotionally appealing, a fantasy should include metaphors already known through similar environments. This helps constructing a system that is easier to learn and use due to its familiarity [28].

Curiosity

The level of information complexity is important to keep the attention and interest of a player. Activities that are either too well-known or too complicated bore or overstrain the player. When adding audio and visual effects or including humour in the game, it is favourable to keep novelty and familiarity of the features at balance. To create curiosity, a player should have an idea what happens after a certain achievement but not be sure about it [28].

The above section has clarified many factors that simplify the creation of a fun game. In von Ahn's definition of a GWAP, the motivating factor fully consists of the players' enjoyment, "People will play [...] to be entertained, not to solve a problem–no matter how laudable the objective." [2, p.98]. However, people behave selfless in peer production systems like Wikipedia and Question and Answer sites [23]. They contribute for the pleasure of working and participating itself [23]. Instead of completely separating the game and its purpose by hiding all aspects of contribution, letting the player know how he contributed to solve the underlying problem can act as another source of motivation. Especially for long-term, repeatedly playing participants this feedback is an interesting extension to a GWAP. This extension goes hand in hand with McGonigal's statement that gamers nowadays want to have a bigger mission than just winning a game [30].

2.3.2 Data Verification

The relation between the provider of a GWAP and the players can be mapped to the principal-agent problem that is widespread in business economics. The gaming provider can be seen as a principal delegating responsibilities to its agents, the players. Therefore, he is relying on the players to make the right decisions. A conflict of interests between the two parties arises if the agent does not consistently pursue the interests of his principal. Due to the existence of asymmetric information (i.e. principal doesn't know about the selfinterest of its agents), this can result in an suboptimal outcome for the principal [31]. When it comes to GWAPs, a player can act against the interest of the provider and manipulate the game by entering false data. The motivation for such behaviour can be a malicious intent as well as sheer boredom. That is why a GWAP has to verify the players' inputs. With the help of rules and winning conditions, players are encouraged to perform correctly and with the best intentions [2]. Thus, entering accurate input should always be rewarded and enforced with positive feedback (i.e. high scores). Consequently, false inputs or inputs basing on uncertainty have to be a bad strategy. However, rules can't ensure the validity of the entered data. To make the game resistant to sabotage, a form of verification has to be integrated. There are several possibilities to implement input verification. The following three templates for two-player GWAPs proposed by von Ahn [2] integrate data verification straight into the game concept. An **output-agreement game** relies on two players unable to communicate who receive the same input from the game and have to produce the same output in order to win. The best known example in this category is the ESP Game, a highly successful image labeling game. Two players are matched randomly and given the same images. Their goal is to both enter an identical tag. A lexicographic utility model of *match*early preferences shows the player's preference to match rather than not to and, secondly, to match earlier rather than later [23]. That is why the ESP Game and similar games (i.e. Artigo Game) collect very general annotations. Inversion-problem games build up on the principal of inversion: Here, only Player A is given an input. Player B has to reproduce this input based on the data he is receiving from A. The game concept of KARIDO (see Chapter 3, p.15 to read about the concept in detail) belongs to this group of games. Changing the roles of A and B after each round makes the game more diversified. Done the right way, inversion-problem games provide a fun social interaction. The last design template is based on the verification mechanism of **input-agreement**. Both players receive inputs. By exchanging data, they try to figure out whether their given inputs are the same or different. The user inputs are considered verified only if each player has decided correctly. Because of the fifty-fifty chance to succeed with random guessing, it is important to implement special rules discouraging such behaviour.

2.4 Design Constraints for Mobile Applications

This section focuses on adjusting game concepts to a mobile audience and creating graphical user interfaces for mobile devices. There are several factors to be taken into account: Differences compared to the usual environment of a desktop user, small screens as well as expectations of mobile users.

2.4.1 Mobile versus Desktop – a Comparison

Since the release of the first iPhone in 2007, smartphones have drastically grown in popularity. Nowadays, about three quarters of the German population is using a smartphone and the upward trend has yet to come to a stop (compare Figure 2.1) [40]. Even elderly people catch up on the phenomenon: In 2013, the share of Germans at the age of 55 or above owning smartphones was set at 14%, rising to 55% in 2017. For comparison, 96% of Germans between the age of 14 to 49 did so [40]. Today's mobile web browsers provide a smooth access to the World Wide Web leading to an increased mobile internet traffic. As a consequence, many websites were adapted to give a more pleasant experience to their mobile visitors. In turn, those mobile friendly websites make browsing on portable devices even more attractive explaining the high share of mobile internet traffic in web traffic in early 2019, the percentage of internet traffic produced from smartphones (excluding tablets) was 38.6% in Europe and 47.96% globally [38]. This development also affects the gaming industry. The Entertainment Software Association⁵ claims that by now, the most commonly used playing device of American adult gamers is the smartphone [5]. Another interesting aspect of mobile internet traffic concerns minority groups and people with a low income. They are relying more on portable devices than desktop computers for internet access as they constitute a smaller cost factor. Therefore, mobile applications have a wider sociodemographic coverage [24, 34].

Technical Specificities of Mobile Devices

There are a various things to take into account during the creation of mobile content as desktops and mobile devices differ in multiple aspects. Regarding the design of a graphical user interface, the screen size and aspect ratio constitute the biggest difference. Mobile users have to deal with a much smaller screen in portrait format. It is the main reason why desktop versions of websites and mobile devices don't go well together. To adapt to this facor and not overwhelm the user, mobile applications and websites should divide their content into small blocks [15]. Concerning text inputs, one has to consider that the physical keyboard is replaced by an on-screen one. Not only does this increase the error rate and make users about 80% slower in writing [48], it also takes away even more of the screen size when opened. Additionally, buttons, input fields and fonts are to be adjusted to the small devices [15] because a finger on the touchscreen can be placed far less precise than a mouse on the desktop [12]. Further alteration is needed when displaying images. A section is dedicated to this topic as pictures constitute a big part in the design of the mobile version called DICO (see 2.4.2, p. 10). Other technical factors that impact the development of mobile content are the limited memory capacity [19] and the shorter battery life of mobile devices. Because there is a wide variety of hardware and software features, it is hard to develop content suitable for all devices. Another restricting aspect is the limited bandwidth of wireless internet [17]. In addition, developers of mobile applications need to consider temporary interrupts in the network connection that can occur at any time. Hills and valleys as well as man-made obstacles like tunnels or buildings can interfere with radio transmission

⁵The Entertainment Software Association is an American organization that releases yearly reports containing essential facts of the gaming sector.



Figure 2.1: Forecast of the smartphone user penetration rate in Germany from 2014 to 2022 [40]

signals. Dead zones are formed, when a reliable connection cannot be established due to weak signals [33]. Thus, entering a dead zone results in the loss of internet connection. An app should provide a smooth way of reestablishing the session.

Multiple studies confirm that users of mobile devices take longer to complete tasks like participating in a survey [24, 34]. However, this is not only due to the just mentioned technical differences: The nature of portable devices brings differences in the way users deal with them: the behavioural differences.

Behavioural Specificities of Mobile Users

Users of mobile devices often are – as the name indicates – on the go, whilst desktop users usually stay in a familiar environment with few distractions. Whether on the subway, in a restaurant or surrounded by friends, distractions are the norm rather than the exception when using smartphones. The studies of Sommer et al. [34] and Keusch and Yan [24], which compare the survey participation of mobile and desktop users, illustrate the impact of said distractions. In both studies, mobile respondents were significantly (2.2 and 2.8 times) more likely to break off the questionnaire than desktop users. Therefore, Kesch and Yan propose to divide a long survey into several small sections which can be answered over the course of multiple days. Regarding the completion time, the studies yield similar results, too: Mobile users needed 118% and 125% of the time desktop users needed to get through the survey, despite leaving more items unanswered. These results lead to believe that the quality of input data collected from mobile respondents could be inferior to data gathered from desktop users. However, the analysis of data quality did not indicate any inferiority of the data collected by mobile devices. Sommer et al. conclude in their detailed validity investigation that "the responses of mobile-device users were just as valid as the responses of desktop users" [34, p.384].

2.4.2 Pictures on Small Screens

The game concepts of KARIDO and DICO are based on the comparison of images. In the desktop version, the amount of pictures getting compared is set to nine. As pointed out, mobile devices are highly limited in screen size. When it comes to deciding how many pictures can be handled by the average player, having a basic understanding of the human visual memory is beneficial. A short excursion into neuroscience follows.

2.4. DESIGN CONSTRAINTS FOR MOBILE APPLICATIONS

Visual Short-Term Memory

The visual short-term memory (VSTM) is part of the human working memory, which the fundamental process of "thinking" is based upon [47]. Object-based abstractions of precise sensory information, that are gathered through eye-fixation, are stored here [18]. Various studies [27, 35, 44] reveal that the average VSTM system has a highly limited capacity of objects stored at a time. This capacity varies among individuals from 1.5 to five objects [43]. According to Luck and Vogel [27], the memorized objects shouldn't be understood as individual features but rather as integrated objects. In their study, 16 features distributed across four objects were retained just as accurately as four features distributed across four objects, meaning that complex images can be remembered just as well as simple ones. By contrast, other works [3] show a relation between the visual information load of an item and the number of retainable objects. According to those, both information load and number of items are limited on their own. Combining those two interferes with their capacity bounds. This leads to a trade-off between the complexity and number of memorized items.

Visual Long-Term Memory

To build up long-term memory, it is necessary to study images repeatedly or for various seconds [29]. Just like the VSTM, the visual long-term memory (VLTM) retains visual information in an object-based manner. However, the memorized items are stored with a significant loss of precision compared to the VSTM. Various studies [32, 37, 36, 6] suggest that the VLTM can store a massive amount of objects. Hollingworth's [18] results imply that information is remembered relatively stable and robust for at least a day.



Figure 2.2: Three stages of visual memory for natural scenes [18]

Figure 2.2 illustrates the three-divided processing of visual stimuli over a period of time as well as the preciseness of their storage in the aforementioned memories. Particularly remarkable is the difference in capacity between VSTM and VLTM. Everything that a person remembers for more than a few seconds is stored in the long-term memory, which has no known capacity limit. However, the precision of the stored objects is greatly reduced. Comparing images can require a rough outline or a detailed vision of said images depending on how similar they are. Therefore, the number of pictures humans are able to compare

depend on the capacity of their VSTM, the information load of the images, the amount of time spent memorizing them and the degree of similarity of the pictures.

2.4.3 Learning from Existing Mobile Apps

The interaction of a user with his portable device is based on experiences gained from earlier interactions. Confronting users with an entirely new environment can be overwhelming and frustrating. Studying popular existing apps, detecting and adopting best practices improve the user experience.

Handling Images

Everyone has expectations when it comes to handling pictures on mobile devices. Apps like Instagram⁶, Photos for iOS⁷ or Samsung Gallery⁸ are used daily by millions of people. They set standards on how to organize and structure images and as to what is expected of performing certain gestures. These apps have a lot in common: To navigate through the pictures, it is possible to get an overview of a group of images that consists of miniature previews of them. Clicking on one of these results in the opening of said image in full screen mode. In full screen mode, swiping left and right (or up and down in the case of Instagram) opens one of the neighbouring images. By using two fingers, it is possible to zoom in and out on images and a swipe down (respectively a swipe to the right for Instagram) leads back to the overview window. In the full screen mode of the app Photos, a bar at the bottom of the screen shows miniatures of the sixteen neighbouring pictures to simplify the navigation even more. Shopping apps like *Amazon*⁹ or *Asos*¹⁰ have similar bars: After opening the page of an item, little dots appear at the bottom of the cover picture. The count of dots matches the amount of images of the item. One of the dots is highlighted indicating which picture is currently being shown. Asos' implementation of this technique can be seen in Figure 2.3. These kinds of "dot bars"



Figure 2.3: Clothing item in the Asos App: Example for navigating through a set of images with a "dot bar"

are a simple yet effective method of navigating through a rather small number of images. The above features help creating an intuitive way for users to keep an overview over a set of pictures and feel comfortable dealing with them on the small screen of a smartphone or tablet.

Asynchronous Multiplayer Games

As DICO is not the first mobile game played in pairs of two, looking at already existing multiplayer games helps to meet the users' expectations and eases the design process. The Swedish app *Quizkampen*¹¹(German version: Quizduell) developed by FEO Media is a popular example of a multiplayer game – according to its website, the app has over 100 million users worldwide[11]. Quizkampen matches two players and lets them contest in six rounds. The game is not played in one session but parted into multiple short ones and the

⁶https://apps.apple.com/de/app/instagram/id389801252

⁷https://www.apple.com/de/ios/photos/

⁸https://play.google.com/store/apps/details?id=com.sec.android.gallery3d&hl=de

⁹https://www.amazon.com/gp/feature.html?ie=UTF8&docId=1000625601

¹⁰https://www.asos.de/entdecken/unsere-apps/

¹¹http://www.quizkampen.se



Figure 2.4: Screenshots of the menus of the multiplayer games "Quizduell" and "Stop".

opponents play asynchronously one after the other. Therefore, a player has to wait for his opponent to finish his turn before playing himself. However, during this period of waiting the user is free to do whatever he wants. Each player can continue playing whenever he has the time to do so without being dependent on his opponent. The lack of simultaneous actions and the short duration of each turn make the game well-suited for mobile users, picking up their device in short periods of inactivity (compare 2.4.1 Behavioural Specificities of Mobile Users, p.10). It is possible to be part of various game sessions at the same time. Thus, if an opponent takes long to finish his turn, another game session can be continued or a new one can be started.

Fanatee's app *Stop*¹², *SongPop* 2¹³ by FreshPlanet and *Draw Something*¹⁴ by OMGPop/Zynga offer completely different games whilst being based on the same concept of splitting a game session into various short rounds and playing them asynchronous. In spite of being multiplayer games, players can act almost completely independent making it possible to integrate the games into the everyday life of their users.

The main screens of all mentioned multiplayer apps resemble each other: All game sessions of the user – mostly displayed as rectangular buttons – are listed one below another, creating an overview. Each game session can either be blocked when it is the opponent's turn or ready for the player to start the next turn. Depending on this state, the corresponding button is either disabled or not. In case of the latter, clicking on the button results in continuing the game session. The applications with all of their features are built around the overview screen. Designing a graphical user interface in the same pattern as popular existing apps helps to create an environment providing intuitive user interaction, which in turn improves the overall user experience.

¹²https://stop-fanatee.com

¹³https://www.freshplanet.com/songpop-2

¹⁴https://www.zynga.com/games/draw-something/

CHAPTER 2. RELATED WORK

CHAPTER 3

Karido

KARIDO was added to the Artigo gaming ecosystem in early 2011. By introducing this diversification game, the quality of image retrieval queries including particularly explicit labels was to be improved. Thus, the GWAP is designed to collect specific, deep semantic and comprehensive tags that differentiate similar artworks from each other to make telling them apart easier. This chapter starts off by explaining the basic concept of the inversion-problem game. Afterwards, the mechanism of input-similarity is introduced, followed by principles considering the data verification of KARIDO and the functionality of simulated players. Finally it is discussed, whether it is reasonable to develop a mobile version of KARIDO.

The following sections are based on Steinmayr's diploma thesis "Designing Image Labeling Games For More Informative Tags" [41] in which the game concept is developed and justified. It provides more detailed information on all mentioned aspects of KARIDO.

3.1 Game Concept

A game session is started as soon as two players are matched. One of them takes on the role of the Describer whilst the other one acts as Guesser. During a game round, the roles of Describer and Guesser stay unchanged. In the beginning of a game round, nine artworks are selected and displayed in a grid (see Figure 3.1). Both players view the same input images. However, they are arranged in different orders. The Describer selects one of the artworks, the goal image, and sends tags describing it to the Guesser. The labels must not be longer than three words and are sent without punctuation. With the help of the received labels, the Guesser has to choose the artwork his teammate is referring to (the goal image). When being uncertain, the Guesser has the ability to ask questions by forming and sending annotations himself. The Describer can answer those with "Yes" or "No". By double clicking on the artwork matching the tags the most, the Guesser performs a guess. If the goal image has been chosen correctly, the score of both players is increased. A wrong choice accordingly entails the deduction of score points. Rather than being opponents, the players have to act as a team to collectively gain points. This enhances the cooperative nature of KARIDO. A round ends after eight images have been guessed. For the second game round, Describer and Guesser switch roles to keep the gaming experience more exciting. Each game session consists out of two game rounds that are build around their own set of nine





similar artworks.

To create a goal for the game, its duration has to be limited. Before starting the game, the client is asked to choose between the two game modes "time limit" and "turn limit". The exact way a game round is conducted is dependent of the selected game mode.

Time Limit

When playing in this game mode, both rounds last 90 seconds. During this time, Describer and Guesser can send tags and questions at all times. After one and a half minutes the round is stopped, no matter how many images have been guessed.

Turn Limit

There are three atomic actions in KARIDO: sending annotations, asking questions and submitting guesses. In this mode, the amount of atomic actions per game round is limited to 30. The duration of the game varies in accordance with the time the players take to perform actions. To ensure an equal participation in the game and keep the cooperative nature of KARIDO, the players have to take turns. The action of submitting a guess, however, can be performed at any time.

3.2 Input Similarity

When having totally different artefacts, basic surface annotations would be enough to tell them apart. KARIDO, however, aims to collect highly characteristic labels and such of a deeper semantic. To do so, every game round is played with nine similar artworks. Unlike the other Artigo games, the works of art in a round of KARIDO are, therefore, not selected completely randomly. At the beginning of a new game round, one art resource – the *base*

3.3. DATA QUALITY

image – is picked at random. With the help of labels, that have been tagged already, an algorithm recognizing similar artworks can be implemented: All the other artworks in the database are compared to the base image by counting the shared annotations. The eight images with the highest amount of equal labels as well as the base image act as input for the new round. This approach is referred to as *input-similarity*. Two works of art that share all annotations are going to be added to the same game round sooner or later. The Describer of the round has to find a tag that applies to one of them only, fulfilling exactly what KARIDO is about: The artworks can be told apart as they now do not share all annotations any more.

Practical Application

In practice, the computation and selection of similar artworks on the whole dataset creates a performance bottleneck as Artigo's database contains more than 65,000 artefacts [7]. Therefore, 1000 images are preselected randomly as a first step. Out of this data pool, the 100 least tagged resources are identified and one is picked randomly as base image. Now, the 36 images which share the most labels with the base image are selected from the data pool to minimize the probability of equal game rounds. At last, the base image and eight of the 36 pictures (again randomly chosen) are added to the game round.

3.3 Data Quality

The game with a purpose applies several techniques to improve the quality of the entered data. By adapting the score after each guess, players are directly influenced by learning that rule-consistent behaviour is rewarded with the best scores. Because malicious players may not care about their score, each annotation still has to be verified itself.

3.3.1 Scoring

KARIDO's scoring mechanism is laid out to make random guessing a bad strategy. When playing as Guesser, the chance to select the right artwork on the first try by taking a random guess varies between 11% and 50% depending on the amount of artworks that are left to choose from. It is likely to make the right guess after a few random picks, making it necessary to discourage this approach. Therefore, false guesses are punished by the deduction of score points. As honest clients are likely to take false guesses every now and then, the penalties should not be set too high as this would frustrate players. A successful game can be defined as the achievement of a positive score. Thus, KARIDO uses a scoring mechanism that sets the expected score of a randomly guessing player equal to zero or lower. Therefore, the penalty for deciding on a wrong artwork exceeds the bonus one gains from submitting the right one. The values of bonuses and penalties change for each goal image. Because KARIDO gets more difficult the less images are left in the grid (artefacts, which are the hardest to describe, are more likely to be selected last as goal images), bonuses increase as the game proceeds. To keep the expected score of the strategy of random guessing negative, the penalties decrease symmetrically. The bonus factor is set to 10, the penalty factor to -11. The *bonuses B* and *penalties P* are calculated depending on *S*, the one-based index of the current goal image ($1 \le S \le 8$). As an example, S = 1 when the first and S = 8 when the last goal image is played.

$$B_S = 10 \cdot S$$

 $P_S = -11 \cdot S$

3.3.2 Tag Verification

To prevent incorrect data from getting integrated in the database, all tags need to be verified by at least one other player. Because KARIDO is an inversion-problem game, the verification occurs with the identification of the goal image by the Guesser. However, a Guesser will eventually pick the right artwork no matter how bad the quality of the sent tags is which falsifies the verification mechanism. Therefore, all labels are discarded if 30% of the left artefacts are selected incorrectly before the goal image is chosen. To furthermore ensure the validity of the entered labels, each tag - paired with the associated goal image - is given a real-valued relevance score. After first assigning a tag, its relevance score is set equal to zero. As soon as the same combination of art resource and tag occurs, the relevance score of the pair is increased by a calculated relevance weight. The relevance weight is a value between zero and one. The amount of labels that have been necessary to sufficiently describe the goal image as well as the count of guesses the Guesser took to figure out the goal image influence the relevance weight. The highest possible weight implies that one single label was given and the goal image was selected on the first try. With the help of the accumulated relevance weights - the relevance score - it is possible to order the tagged annotations for each artwork in the database after relevance.

3.3.3 Security Gaps

Despite the scoring mechanism and tag verification implemented to improve the data quality, KARIDO is still not safe from many kinds of misuse. Players can cheat by using other channels to communicate the goal image besides the input channel of KARIDO. The sent labels, which could have nothing to do with the chosen art resource, will be mistakenly verified. A measurement taken to prevent players to simply send the web address of the goal image as label is address obfuscation. Thus, the images of the teammates have different addresses. Finally, the unfiltered communication allows the exchange of inappropriate messages. Therefore, the communication is monitored and further measures will be taken if necessary.

3.4 Simulated Player

To make it possible for single players to play KARIDO, the second player can be substituted by a simulated one. This bot is capable of acting both in the role of Describer and in the role of Guesser.

To be deployed as Describer, previous rounds are replayed by the bot. The goal images are selected in the same order and the same sequence of tags is being sent with delays of equal length as in the modelling round. To answer questions, the simulated player checks the tags already assigned to the associated goal image. If the tags of the prototyping round do not suffice for the Guesser to find the goal image, the bot sends random labels out of the pool of assigned annotations.

When replacing the Guesser, the bot compares the sent annotations with the already assigned tags of each artwork in the grid and calculates their *match percentages*. With the help of these match percentages, the bot decides when and with which artwork to make the guess. To prevent the game from coming to a stop when the Describer fails to send accurate annotations of the goal image, the simulated player starts to guess randomly at some point. A more precise description of this process can be found in [41].

3.5 Rationale for a Mobile Version

This section tries to answer the question whether there is the need for a mobile version of KARIDO. To do so, the role of mobile devices in the internet and the benefit of the Artigo platform from such an extension are taken a look upon.

The importance of mobile technology is increasing steadily (compare Subsection 2.4.1, p.9). In Europe, about 39% of the web traffic is of mobile origin [38]. Especially when it comes to gaming, a mobile device is a popular – if not the most popular – choice [5]. However, the gaming ecosystem of Artigo is still oriented towards desktop clients only. When coming across the website of Artigo on a portable device, the client is confronted with a website not suited for mobile access. It is possible yet highly unsatisfying to play one of the implemented GWAPs on a smartphone. Figure 3.2 shows the way KARIDO is displayed when opened on an iPhone.

Artigo relies on the amount of users playing the available games: The more players, the more tags, the better the search engine. Thus, Artigo profits from a higher reachability. Looking at the numbers above, introducing a mobile friendly game on Artigo seems thoroughly reasonable. In 2011, when KARIDO was first introduced, the future development of a mobile version was proposed already [42]. Compared to the other games ARTIGO GAME, TAG A TAG and COMBINO, KARIDO provides the most gameful experience as the two different roles make it more diversified. Therefore, the implementation of a mobile-friendly gaming ecosystem is started off by adapting KARIDO to mobile clients.



Figure 3.2: KARIDO opened on a smartphone.

CHAPTER 3. KARIDO

CHAPTER 4

Dico – Design Choices and Their Justification

The mobile version of KARIDO was designed with the enjoyability of users and the data quality of the gathered tags kept in mind. Despite being adapted to the benefit of mobile clients, the spirit of the game should stay the same. The new game is given the name DICO, which is Latin for "I say" and refers to the action of finding labels for the images and communicating them to the teammate. In addition, the sound of DICO fits to the other games available on Artigo.

In this chapter, the game concept of DICO is introduced whilst justifying the taken design choices. The presentation of the rules of DICO is followed by the implementation of a certain standard of data quality with the help of a motivating scoring mechanism and tag verification. The chapter concludes with a comparison of DICO to the desktop version KARIDO.

4.1 Game Concept

Following Grace's and Jamieson's [14] design by analogy, DICO tries to solve the problem of improving Artigo's search engine by continuously adding distinctive annotations to the artworks. The game concept is being developed based on the desktop version – KARIDO. To provide an enjoyable experience when playing KARIDO on a mobile device, the game concept needs to be adjusted. To do so, technical differences of mobile devices and behavioural specificities of mobile clients to their desktop counterparts (compare Subsection 2.4.1, p.9) are taken into account.

Mobile users are easily distracted and have a higher tendancy to break off activities. A KARIDO game session consists of two game rounds. When playing in the time-limited mode, the game session lasts 3 minutes. As players, who are competing in the turn-limited mode, can take all the time they want, the game sessions tend to be longer: The average length of this mode amounts to 5 minutes 31 seconds [41]. In both modes, it is not possible to take breaks because the opponents play at the same time. Based on the same approach taken on by the multiplayer games mentioned in Subsection 2.4.3 (Quizkampen, Draw Something, etc.), DICO is not played in one continuous session but broken up into several shorter ones, that can be played over the course of multiple days. This complies with Kesch and Yan's proposal to divide long surveys into small sections, when they are aimed at mobile users. Also, the players do not play in parallel but asynchronously. Thereby, Player



Figure 4.1: All actions of Player A and Player B taken during a whole game session.

A can execute his turn¹ whenever he wants to – the action is performed independently of Player B. The concept for DICO does not implement a time limit for the following four reasons: First of all, the time consumed to gain an overview over all artefacts depends on the screen size of the used device. A time limit creates unequal circumstances by favouring the users of tablets. Secondly, typing on the on-screen keyboard is error-prone [48]. Limiting the time to correct their typing errors. Thirdly, when typing, the on-screen keyboard makes up a lot of the screen and partially covers the artworks. Going back and forth between typing and looking at the pictures takes time and rushed players are reflected in less precise annotations. Lastly, dead zones or a slow internet connection, which occur regularly with wireless networks, pose a problem for time limits. To sum up the above, both user experience and tag quality suffer from a time limit. Therefore, DICO can only be played in one mode that is similar to the turn-limited mode.

Turn Limit

In the turn-limited mode of the desktop version, the players' actions are added up and a round ends whenever all goal images are guessed correctly or when the limit of 30 actions is reached. The graphical user interface of DICO only allows a certain amount of operations, making counting superfluous. The amount of tags is strictly restricted: When describing the current goal image, the Describer needs to enter exactly four different annotations. There is no possibility to add more tags – even if the fellow player cannot figure out, which image has been described to him. In this case, the goal image can be discarded by the Guesser and the game continues without him having to find the goal image. This measure emphasizes the diverting character of DICO by preventing being stuck with one picture. A study on how many tags and questions are sent on average per image in KARIDO has yet to be done. The count of tags has been set to four to grant a relatively exact characterization whilst not letting the client run out of ideas for tags as he currently cannot choose to leave tags empty. The opportunity to pose questions is not given, as this option would shorten the turns of the players whilst prolonging the waiting periods and the game session itself – the Describer would merely have to answer the question and immediately enter another waiting period.

Changing Roles

Another measure is taken to minimize the waiting periods: The roles of Describer and Guesser change after each game section as opposed to each game round. A game section is defined as the period between selecting a new goal image and correctly guessing (respec-

¹A turn is defined as the time between two waiting periods.

4.1. GAME CONCEPT

tively giving up on finding) the said goal image. A game round consisting of N artworks contains N-1 game sections. By constantly changing the roles, a standard turn² includes a player acting as Guesser and Describer without a waiting period in between. The exact course of the game is shown in Figure 4.1.

Input Similarity

The factor of input similarity is just as important in DICO as in the desktop version because it drives the creation of more specific labels. The mechanism of choosing similar images has been adopted completely from KARIDO only differing in the number of artworks selected per game round.

Amount of Art Resources per Game Round

A game round of KARIDO consists of a set of nine similar art resources. The images of those artefacts are displayed in a 3x3-grid. The earlier mentioned biggest restriction of a mobile device is its comparatively small screen. Fitting three pictures next to each other on the screen of a smartphone results in a small image size, which makes the needed comparison of their contents troublesome. This, in turn, leads to an unsatisfactory user experience. Giving the player the option of viewing the pictures in a slideshow solves the problem of the image size. Details of the individual artworks can be detected by scrolling through the slides. Then again, a slideshow makes the direct comparison of artefacts more difficult. If a picture is studied repeatedly or for various seconds, it enters the visual long-term memory [29]. Thus, an ambitious player has all artworks in his long-term memory. However, images are stored there rather imprecisely. Since details are important in both the role as Describer and the role as Guesser, the player needs visual support. Clicking through a slideshow of nine images is overwhelming for a player describing or looking for a specific image. The direct visual comparison, that is possible when looking at a grid, is still required.

In summary, the graphical user interface of DICO needs

- ... a grid granting an overview over all artworks to make comparison easier.
- ... a *slideshow* allowing a closer look at the individual artefacts.

Deciding on the amount of artefacts per game round is not exclusively based on a reasonable image size in the grid. The purpose of the game – collecting diverse, unique tags – and the importance of data quality is included in the decision. The fewer pictures available, the better the chances of success through random guessing. Therefore, choosing the number of art resources per round is a trade-off between user satisfaction and data quality. For a DICO game round, a set of six art resources has been selected in order to balance both requirements. Visually represented in a 2x3-grid, the images have a decent size. Adding another row to the table results in problematically small dimensions. The same applies to an additional column.

Just like in the desktop version, the arrangement of the pictures in grid and slideshow differs for each player to avoid the success of annotations describing the spatial position of the goal image like "top left" and similar.

Limitations

A player can be part of multiple game sessions at a time. However, the amount of parallel game sessions is limited. This restriction hinders one single client from being able to block the entire DICO platform by sending countless requests to start a new game, being matched to new game sessions and not playing. Currently, the maximum count of game sessions of

²All turns except the first and last one of a game round are standard turns.

a player is set to five. In the future, this number can easily be adjusted, should it become apparent over time that more or less parallel game sessions are more reasonable.

Other limits are set concerning the duration of a game session. To prevent prolonged game sessions, the count of game rounds has been lowered to one. As a reminder, a game round is defined by its art resources: As soon as the artworks change, a new round has been started. Due to the divided session, the duration of a game round can vary significantly. It is attached to the frequency in which the players make their moves. A game round may last fifteen minutes or be stretched out over several days. As a player can only take part in five game sessions at a time, the length of a game session has to be limited. Otherwise, a client is fully reliant on the time his opponents need to take their turns. Therefore, a game session ends 48 hours after it was started by the first player. Again, the limit of 48 hours can be easily raised or lowered in the future if more beneficial.

4.2 Rules

In the following, the rules for carrying out the actions are explained in more detail. The sequence, in which the players have to take the actions, is displayed in Figure 4.1.

Join-Action

The client starts a new game or joins an existing one by pressing the corresponding button. This action is only possible, if he is currently taking part in less than five game sessions.

Wait-Action

The player has to wait as it is the turn of his opponent. The duration of a waiting period is limited by the maximum length of a game session: A game session ends prematurely if it lasts longer than 48 hours.

Describer-Action

1. Selection of the Goal Image

The player chooses the artwork he wants to describe in the next step. To do so, he has to decide between the artefacts of the round that have not been chosen as goal image yet.

2. Description of the Goal Image

The player enters four tags in the input fields provided for this purpose. Neither more nor less than four labels can be submitted. As in the desktop version, a label must not contain more than three words. Additionally, a label can only hold a maximum of 20 characters. The input fields are enumerated. Strategically, it is best to insert the most specific annotation into the first input field and the least important one into the fourth field.

Guesser-Action

1. Revealing the Tags

The Guesser reveals the tags one by one. Thereby, it is not possible to open the tags in arbitrary order. They can only be opened in the sequence the Describer entered them. Besides, not all labels have to be revealed by the end of the turn. The labels are received in all capital letters and without punctuation. 2. Taking a Guess

As soon as one tag has been revealed, it is possible to take a guess. If the player chooses the right artwork, the score increases and the Guesser-Action ends. Otherwise, the score decreases and the player is free to reveal the next tag, perform another guess or give up – assuming the conditions required to carry out these operations are met.

3. Giving Up

As soon as the player has opened all four tags, he is given the option to capitulate. If it is not possible to detect the goal image with the sent description, giving up is the better strategy than guessing as a wrong choice results in the deduction of points. By executing this option, the Guesser-Action ends immediately. The goal image is marked as played and, thus, cannot be selected as goal image in this game round again.

4.3 Data Quality

Just like KARIDO, DICO saves data entered by clients in a database. To prevent the injection of false data, it uses the same two mechanisms to achieve a higher level of data quality: Scoring and Tag Verification. However, the implementations of both differ in a few aspects.

4.3.1 Scoring

The importance of the scoring is not to be underestimated. It plays a major role in the goal setting of the clients by quantifying their success: Scores provide an incentive to improve the own performance. By giving the player a feedback to his actions in real-time, he can figure out a strategy on how to gain the most points. This means that the scoring impacts the players behaviour. With the help of scoring, clients can be directed to act in favour of the game provider.

In the case of DICO, the assignment of points encourages a way of playing that leads to the collection of high-quality data. Same as in the desktop version, random guessing has to always be a bad choice. DICO implements an extended version of the scoring mechanism of KARIDO. Each guess is immediately evaluated with a change in the score – a right guess results in the addition of a bonus, a wrong guess leads to a penalty being applied. The height of the bonus/penalty is not only determined by the number of the current game section. Another contributing factor is the count of already revealed tags. The less labels revealed, the higher the bonus in case of a right guess. This measure is taken to motivate the players even more to find labels unique for one artefact. However, this makes the opening of tags less appealing and encourages the unwanted speculative guesses. To counteract this, the penalty for wrong guesses is higher the fewer tags are unveiled.

The action of giving up is rewarded with a bonus, too. While this seems unreasonable at first, it actually helps to improve data quality. Suppose a Guesser has opened all four labels and is torn between two artworks. If he chooses the right picture, he will be rewarded with a bonus. If he chooses the wrong one, he loses points. Even if the expected value of the guessing is negative, the 50% chance to win points can be more attractive than giving up and not getting any points. Especially if the player wants to achieve a new high score. Attaching a bonus to the option of giving up, makes it much more attractive and benefits an honest player. Just like in KARIDO, both players get the same score. There is the possibility of treating the bonus for giving up differently by only adding the points to the score of the Guesser. This would prevent rewarding a Describer for sending insufficient tags. However, assigning points to only one player goes against the cooperative character of KARIDO. Thus, it was decided against different scores for the teammates.

	1 Tag Revealed		2 Tags Revealed		3 Tags Revealed		4 Tags Revealed			
Section S	Bonus	Penalty	Bonus	Penalty	Bonus	Penalty	Bonus	Penalty	Giving Up	E(S)
1	130	-143	120	-132	110	-121	100	-110	50	-86,25
2	260	-286	240	-264	220	-242	200	-220	100	-156,4
3	390	-429	360	-396	330	-363	300	-330	150	-198,375
4	520	-572	480	-528	440	-484	400	-440	200	-184
5	650	-715	600	-660	550	-605	500	-550	250	-28,75
	•	•	•	•	•	•		•		-653,775

Figure 4.2: The scores assigned for taking guesses and the expected value for random guessing.

The following are the formulas for calculating *bonus B*, *penalty P* and the *bonus for giving up G* dependent on the *section number S* and *count of revealed tags T*. It holds $1 \le S \le 5$ and $1 \le T \le 4$.

$$B_{S,T} = 100 \cdot S \cdot \left(1 + \frac{4 - T}{10}\right)$$
$$P_{S,T} = -110 \cdot S \cdot \left(1 + \frac{4 - T}{10}\right)$$
$$G_S = \frac{1}{2} \cdot B_{S,4}$$

The bonus factor of 10 and penalty factor of -11 of KARIDO have been multiplied with ten to prevent the occurrence of decimal numbers. Figure 4.2 provides an overview over all resulting bonuses and penalties. The *expected value for random guessing in each section* $\mathbb{E}(S)$ and the sum of all $\mathbb{E}(S)$ have been added to the table. For simplification, the calculated expected values only take into account one random guess per section. In addition, the bonus for giving up does not occur in the calculation because this option is not part of the randomized selection of artworks. The values for $\mathbb{E}(S)$ are obtained with the formula below.

$$\mathbb{E}(S) = \frac{1}{4} \cdot \left(\frac{1}{7-S} \cdot \sum_{T=1}^{4} B_{S,T} + \frac{6-S}{7-S} \cdot \sum_{T=1}^{4} P_{S,T} \right)$$

4.3.2 Tag Verification

Besides giving the player internal motivation to enter valid annotations through the scoring mechanism, the tags of a game section are verified before persisting them in the database. DICO is an inversion-problem game. Therefore, the main verification takes place when the Guesser is choosing the goal image with the help of the annotations sent by the Describer. The process of tag verification is divided into four steps:

1. Action of the Guesser

Only the labels revealed by the Guesser are added to the set of tags to be verified. All annotations that remain covered are discarded. If the Guesser gives up, the process of verification ends right away and no tags are persisted.

2. Calculating the Percentage of Wrong Guesses

As soon as the Guesser took the right guess, the second step of the verification process starts. The percentage of wrong guesses in relation to the number of artefacts to choose from is calculated.

 $percentGuessedWrong = rac{countOfFalseGuesses}{countOfLeftImages}$

4.3. DATA QUALITY

The count of left images is equal to the amount of artworks per game round minus the number of previous sections of the game round. All tags are considered invalid if the result is bigger than 0.3, meaning that over 30% of the artworks were guessed falsely before finding the goal image. Otherwise, the verification process proceeds with the next step.

3. Calculating the Rating

The set of tags that reaches this stage in the verification process is getting persisted in the database together with a rating. The rating, also referred to as relevance score, represents the significance and informative value of the label. The relevance score is computed in a way that is similar to the procedure of KARIDO. In KARIDO, later entered annotations get a higher relevance score than the ones submitted earlier. This is justified by the argument that the last labels are decisive for the right guess. However, it can be argued the other way around. It is possible that the first term excludes most images as the goal image, but more terms are needed to finalize the decision on one artwork. It was decided that in DICO, each tag in the set is considered as equally relevant.

$$rating = 0.5 + \left(\frac{0.5}{countOfRevealedTags} \cdot (1 - percentGuessedWrong)\right)$$

The highest rating possible is 1, when the Guesser had to reveal only one tag and selected the goal image on first try. The lowest rating of a verified tag (0.5875) is yielded when all four tags are revealed and the share of wrong guesses equals 0.3.

4. Persisting the Tags in the Database

The set of tags is persisted in the database together with the respective art resource, game round and person. The rating is accumulative, meaning that it is added to the relevance score of a tuple of tag and art resource in the database.

4.3.3 Security Gaps

The security gaps of the desktop version mentioned in Subsection 3.3.3 also apply to DICO. Additionally, a client can easily play a game session with a friend, making it possible to communicate over another channel. By using two different browsers or multiple devices, a client can also play against himself. That is because even though players get matched randomly, there is a high chance of being added to the same game session when requesting a new game session in quick succession. However, that risk diminishes as the client base of the game grows.

4.4 Comparison to Karido

	Kai	ido	Dico		
Game Rounds per Game Session	2		1		
Amount of Artworks per Game Round	9		6		
Game Session	Played in one continuous session		Played in 6 sessions (3 for each player)		
Modes	Time limit	Turn limit	Implicit turn limit		
Duration of a Game Session	2 x 90 seconds	A few minutes	A few minutes up to 48 hours		
Presentation of Artworks	3x3-	grid	2x3-grid and slideshow		
Tags per Goal Image	s per Goal Image Varying		4		
Role of Describer and Guesser	Changes after each game round		Changes after each game section		
Option of Giving Up/Discarding a Goal Image	Not given		Given		
Revelation of Tags controlled by Guesser	No		Yes		
Factors influencing the height of bonuses and penalties	Number of current game section		Number of current game section and amount of unveiled tags		

28

CHAPTER 5

Implementation of Dico

DICO follows the classic client-server architecture. As the game is part of the Artigo platform, the back-end can make use of the already existing server of Artigo and extend it with the rules of DICO. The front-end is exchangeable and only responsible for the visual representation of the game. It was decided to implement DICO as a single-page application that is accessible through the internet browser of any mobile device independent of its operating system. Still, the used framework *React.js* eases a transformation into a native app if desired in the future.

5.1 Front-End

Artigo is reachable through the browser. By making DICO accessible as a mobile web application through a browser as well, several advantages arise: Whoever visits the website http://artigo.org has immediate access to DICO. The obstacle of downloading and installing an app before even playing it is passed. Still, the implementation of the web app as a *single-page application* gives a feeling similar to that of a native app. The react framework specifies the term as follows:

"A single-page application is an application that loads a single HTML page and all the necessary assets (such as JavaScript and CSS) required for the application to run. Any interactions with the page or subsequent pages do not require a round trip to the server which means the page is not reloaded." [10]

Mobile devices nowadays are equipped with highly functional browsers, providing access to the world wide web independent of the respective operating system. Their performance is improving constantly – the average webpage loading time was lowered from 40 seconds in 2009 to less than five seconds in 2014 [49]. A mobile web application is cheaper in the development and maintenance as one version is executable on all operating systems. Another upside is the unnecessity of updates – every client is using the latest version automatically. However, when visiting the website the first time, a bundle including all required assets is loaded at once.

To implement DICO as a single-page application, the web framework React.js1 is used

¹https://reactjs.org



Figure 5.1: Photos of the paper prototypes.

within the runtime environment *Node.js*².

The main function of the front-end is interpreting and displaying data received from the server and sending inputs from the client back to the server. It barely implements any game logic. The main focus lies on the graphical user interface: the visual presentation of the game and the interaction with the user. By taking common practices into account which are set by the mentioned gallery and multiplayer apps (compare Subsection 2.4.3, p.12), paper prototypes of the mobile web application's user interface were developed. The paper prototypes give a first idea of what the application could look like and make is easy to adjust design errors. View the final paper prototypes in Figure 5.1. The screen is divided into three areas: header, main screen and footer. There are several different kinds of main screens and accompanying footers while the appearance of the header remains the same. The header's functionality is the navigation between the various main screens. It is the main screen where the user specific data is displayed. The footers change depending on the currently opened main screen. The main user interactions are made through these footers. There are screenshots of all the mentioned screens in the appendix. The three different types of main screens are the following:

Menu Screen

In the menu screen, an overview over all game sessions that the user is currently participating in is given. Each game session is displayed as a button, containing a miniature image of an artwork of the respective game round as identifier, a short text stating the current status of the session and a progress bar, displaying how much of the session is left to play. The button is disabled whenever the user is in a waiting period in the corresponding game session. Otherwise, a click on it leads to the game screen and the player can take his turn. The footer of the menu screen contains a button to request a new game session. See Figure 6.2.

Info Screen

The info screen contains the basic rules of the DICO. It is accessible through the right button in the header, so that a player can open it at all times. This screen does not have a footer. See Figure 6.1.

²https://nodejs.org

Game Screen

A game screen is bound to a specific game session. It varies in its appearance depending on the game state of this session. The four versions of the game screen are the *select* and *describe screen* (Figure 6.3), the *guess screen* (Figure 6.4) and the *rewind screen* (Figure 6.5).

A game section starts with the select screen, in which the player is asked to pick one of the shown pictures as goal image for this section. After doing so, the screen switches to the describe screen. The player now has to enter four tags that distinguish the goal image from the other artworks before entering a waiting phase and automatically returning to the menu. The guess screen is displayed when a waiting phase ends and the player reenters the game session. This screen enables the player to view the labels received from his teammate and to perform a guess.

The rewind screen constitutes an exception as the player does not perform actions in it. It is accessible as soon as a game session has ended and it is the only place where the player is shown the key data of the artworks. Every artwork of the completed game session is displayed in full view together with its title, the artist and the institution it is located in. The score gained in the respective game section is displayed, too. The artefacts are ordered vertically and can be viewed by scrolling up and down. This screen can only be opened once. After returning to the menu screen, the game session is removed from the overview and has finished completely.

In the select, describe and guess versions of the game screen, the artworks of the current game round are displayed in a grid or a slideshow. The default view of the images is a 2x3-grid. By pressing a button on the bottom right of the grid, the slideshow mode is entered, allowing a more detailed view of the individual images. In the slideshow, the client can swipe right or left to get to the previous or next picture. To facilitate navigation, a bar at the bottom shows the images in miniature. The miniature version of the artwork currently presented in full view is highlighted. The slideshow contains a button as well, letting the user switch back to the grid. The corresponding footers make it possible to...

a) confirm the selection of a new goal image,	(footer of select screen)
b) enter four tags describing it,	(footer of describe screen)
c) reveal the sent tags one by one and confirm a guess.	(footer of guess screen)

The rewind screen does not have a footer.

In the select and guess screens, the user needs a possibility to select an artefact. In the grid, clicking on an artwork results in the preselection of it. This is made visible to the client by blurring all other pictures. In the slideshow, the artefact that is in full view is automatically preselected. To confirm the preselected artwork, the player presses the confirm-button in the respective footer.

5.2 Back-End

The server side of DICO is embedded into an already existing environment of Artigo, implemented by Nicola Greth in 2018/19 as part of a master thesis at the LMU. The code is written in JavaScript, using Node.js as runtime environment and the server framework Express.js³. The connection to Artigo's PostgreSQL⁴ database is realized with the SQL query builder Knex.js⁵ and its extension Objection.js⁶ that enables object-relational mapping. An activity diagram (see Figure 5.2) illustrating the active and passive phases of each player

³https://expressjs.com

⁴https://postresql.org

⁵https://knexjs.org

⁶https://vincit.github.io/objection.js/



Figure 5.2: The activity diagram of a DICO game round.

was set up to ease the integration of the new game into the existing code. The most relevant difference between DICO and Artigo's other GWAPs is its asynchronous nature. The fact that a user can interrupt a game session to continue playing later and be in multiple game sessions at a time required an adaptation of parts of the existing code dealing with the client-server communication. The problems arising from this change are explained later in this chapter.

5.2.1 Important Classes

The class diagram in Figure 5.3 displays the simplified versions of the most important classes of the back-end. The attributes as well as the operations of the classes are reduced to the most significant ones and some names are adjusted. The mentioned "handleAndPassOn"-functions are summing up the multiple methods implemented in the respective class, which serve the purpose of processing some of the data sent from the client and transmitting parts of it to another instance, forming some sort of chain of responsibility.

GameSocketCommunicationHandler

The class GameSocketCommunicationHandler is instantiated once throughout the entire program execution. This instance establishes all socket connections to the clients. It is responsible for passing on data received from the clients and all data sent to the clients are emitted from here. The object contains the only instance of GamesManager and transfers the received data on to this manager to deal with it further.



Figure 5.3: A simplified class diagram containing DICO's main classes.

GamesManager

Same as the GameSocketCommunicationHandler, the program holds a single instance of the class Games Manager only. It contains a list of all current game sessions and a map matching the socket id of each user to his game sessions. Whenever a user reloads or reconnects to the website, a new socket connection is established and the user's socket id changes. Thus, this map is updated whenever a new socket connection is established. When receiving data from the GameSocketCommunicationHandler, the socket id and gameSessionId are checked and the data is passed on to the respective game session, if the matching one is found. Additionally, it handles the request of a player to join a new game session. New game sessions are added and old ones are removed within the Games Manager.

DicoGameSession

Whenever a new DICO game session is started, DicoGameSession is instantiated and the object is given a unique id. DicoGameSession extends the class GameSession and overwrites some of its methods to make them suitable for the asynchronous nature of DICO. For instance, a game session of DICO starts with only one player connected to it while all simultaneously played game sessions wait for all players to join before the first round starts. A game session contains an instance of the class GameMetaInformation that holds the specificities of the game type, the players of the game session and the language it is played in among other information of the game session.

The class attribute artResourceManager of the type ArtResourceManager deals with the selection of the art resources for the game rounds. Also, the game session passes on received data to the game round after checking whether it was sent by one of the session's players and whether the session is not over yet.

DicoGameRound

The class DicoGameRound inherits from the general GameRound, adapts some of its methods and extends it by introducing game sections. A game round contains an array of the six art resources that the round is played with. Before passing on data to the current game section, the game round makes sure the sender is allowed to perform the requested action. When a new goal image is selected, the game round starts the next game section.

The initial game concept of DICO consists of one game round only. Putting the game round in an own class makes it easy to change the count of game rounds per game session when needed.

DicoGameSection

Other than the two latter classes introduced, game sections are unique for DICO. Each art resource of a game round belongs to one of the six instances of DicoGameSection. As soon as the art resource is chosen as goal image, the respective game section sets its class attribute artResourceState equal to "current goal image" and the section defines the current game state until the next goal image has been selected. The instance containing the are resource that never gets set as goal image remain unchanged throughout the whole game round. In here, most of the information of the game is stored – the current states of the art resources and players, the tags, the score of the section and more. Most data sent from the players is passed from object to object until it reaches the instance of game section where the action is, finally, performed. The data reaching the game section does not equal the data package sent from the player, as the instances of game manager, game session and game round only pass on the data relevant for the succeeding stations. Thus, only the data required to carry out the action in question is reaching it.

5.2.2 Challenge: Multiple Game Sessions per User

The back-end - more precisely the GameSocketCommunicationHandler - establishes a bidirectional and real-time connection to each user with the help of sockets provided by the framework socket.io⁷. Therefore, every instance of Player owns an attribute socketId to make it possible to emit data to a specific user. Because all of the already implemented games of Artigo are played synchronously, every user could be part of only one game session at a time. Thus, each data package received from a specific socket could be matched to a maximum of one game session. Analogically, a client receiving data from the server over its socket had only one game session to match it to. With DICO, this changes: The asynchronous nature of it allows a user to be part of several game sessions at once. A socket id does not stand in an 1:1-relation to a game session any more, but rather in a 1:nrelation, with $0 \le n \le 5$. To adjust the server in a way that data can be passed on to the right game session, a data object received from a socket and aimed at a game session needs to include the id of the game session. In addition, the socket id of a player can change throughout the course of a game session as a user can interrupt a connection at any time and resume playing the game later on. When reloading or reconnecting to the website, a new socket connection is established. To be able to assign the new socket id to the user and his game sessions, the front-end sends its ladder socket id after every reconnection and saves the new one immediately in the browser's local storage. In turn, the data packages associated with a specific game session sent from the server to a client include the game session id as well. Therefore, most of the functions in charge of processing the received data as well as those responsible for sending back the responding data were adjusted and extended with the attribute gameSessionId.

⁷https://socket.io

CHAPTER 5. IMPLEMENTATION OF DICO

CHAPTER 6

Conclusion and Future Work

The enjoyability of a game with a purpose and the quality of the produced data make up two central aspects when evaluating its success. Therefore, this chapter reviews the implementation of both in the developed game concept of DICO. The aim of the analysis is the recognition of strengths and weaknesses of the game and the unveiling of areas requiring future work. A collection of features whose realization improve and extend the game is given. Finally, the last section presents an outlook on alternative application areas that can benefit from DICO's game concept and on the future of the Artigo platform.

6.1 Analyzing the Enjoyability

To analyze the realization of enjoyability in the game concept, the implementation of Malone's [28] three main categories of entertaining concepts introduced in Subsection 2.3.1 are taken a look upon. Checking on the design constraints mentioned can reveal design flaws.

Challenge

The users of DICO are challenged to score as high as possible. To maximize their outcome, players develop strategies depending on the performance feedback that is also given through the scores. In the role of the Describer, the player's best strategy is to think of the four most characteristic, unique annotations, which differentiate the goal image from the other artworks, and sort them by descending relevance. As Guesser, the player is challenged to figure out whether the already revealed tags are distinctive enough to decide on an artwork and perform a guess or if opening the next annotation is the smarter choice. There is still a lot of room to develop further challenges and goals in the future. Creating a leader board to have an overview over the best scores can create an incentive to defeat high scores. Also, future challenges could put users to the task of going a certain amount of games without performing a false guess and reward a success with a rise to the status of a "superior player". Taking a wrong guess would result in the loss of this status. The labels of users that have this status could undergo a less strict verification method. Playing against friends is another well known option in multiplayer games. Gaming with friends enhances fun and motivation. However, this feature increases the risk of cheating as friends might be close in space or communicating over other channels. Before implementing this feature, risks and benefits need to be evaluated. The practice of challenging players by letting them compete in levels of increasing difficulty is widely known. A gradual limitation of the number of tags per game section is a possibility of how levels can be implemented in DICO. In the hardest level, the Describer could only send one single annotation to his co-player. This way, the game would collect highly distinctive labels only.

Fantasy

The same fantasy can be highly motivating for one person and just as demotivating for another one [28]. As DICO is aimed at a wide sociodemographic group, fantasies were intentionally avoided in the user interface. The graphical design is kept clean and basic, the colour scheme is build around a subtle light blue.

To ensure a user interaction with the app that is as easy and intuitive as possible, the graphical user interface consists of only a few elements. To create familiarity, best practices, that are implemented by popular multiplayer or gallery apps, are adapted in DICO's graphical user interface.

The only fantasy in DICO is created by the artworks and, thus, differs from round to round. Whether it is to find the most characteristic annotations of the goal image or the artefacts that suits the given description the most – studying the artworks is a central task of each action. The examination of art evokes emotions in the users, especially in the art lovers among them. To make use of the motivating factor a fantasy can have, a player could be given the option to influence the fantasy by deciding on an epoch, artist or style of the artworks of his game session.

Curiosity

Curiosity is rooted directly in the game concept of DICO as each game round is different: The art resources are chosen randomly, each tag revelation is a surprise and most guesses are taken with some uncertainty left. Whilst artworks and labels vary for each game, the other parts of the game remain the same. That is how novelty and familiarity stay balanced. To enhance the curiosity, sounds could be added to actions to give auditive feedback. The previously mentioned possibility of letting the user choose a certain class of artworks he is interested in has a positive impact on the curiosity, too.

6.2 Analyzing the Data Quality

The rationale of the game is to obtain a description of the artworks' unique features. To measure just how well DICO serves its purpose, a study evaluating the collected data has to be done in the future. However, it is possible to have a look at the game concept as well as the user interface to get an idea of the data quality. As presented in Section 4.3, DICO implements a scoring mechanism, that encourages the players to enter correct and precise labels, and a tag verification to curb the storage of false data in the database. The effect of the scoring mechanism is highly correlated to the player's engagement: An engaged player is interested in his score and, thus, is sensitive to the feedback he gets from the score. He is more likely to enter tags of a high quality. A user who is not interested in the outcome of a game session due to a low engagement, however, is not influenced by the scoring technique. The engagement level of each player is the result of his intrinsic motivation and hard to determine: It varies not only from user to user but also among the same user according to many influencing factors like stress, time pressure and mood. Thus, the scoring only promotes but does not ensure a high data quality.

The tag verification does not depend on external factors and its effect on the data quality is, therefore, more stable. By discarding tags, which did not lead to a successful recognition

of the goal image, a part of the entered data is filtered out. Clearly, this is done to prevent invalid tags from being stored. But the technique is not free of flaws: On the one hand, an unfocused or frivolous Guesser can take wrong guesses despite having received valid labels. In these cases, the tag verification mechanism mistakenly gets rid of accurate data. On the other hand, false data is not guaranteed to be filtered out, as the Guesser can coincidentally make the right guess based on invalid labels. However, when thinking of an scenario in which at least one of the players in a game session acts rule-compliant, the implemented tag verification works fine: If a "malicious" Describer enters a false annotation of the goal image, a "fair", not randomly speculating Guesser will take the option of giving up and the false tags get discarded. If a "malicious" Guesser is paired with a "fair" Describer, the sent tags are valid and, trivially, no false data can be stored. In short, the tag verification does a decent job filtering the user input but there is still the possibility that inaccurate data is saved in the database. To check the actual data quality in normal operation, it is reasonable to conduct a study in the future.

6.3 **Possible Extensions**

In the analysis of the enjoyability, a few ideas on how to improve the game concept were mentioned. This section describes them further and presents other suggestions for possible extensions.

Customized Game Sessions

In the future, creating a very engaging and motivational environment can be made possible by giving the users the option to compose their preferred version of DICO. Before starting a new game, the user could for instance decide between the following options to create a customized game session to his liking:

• Amount of Labels per Artwork

The game can be customized by giving the player the option to define the amount of descriptive tags that are exchanged for every work of art. A possible range is a maximum of four and a minimum of one tag. The player's choice influences the difficulty of the game session as guessing is harder when exchanging only one annotation. People looking for a more demanding or shorter game can profit from less labels.

• Amount of Artworks

The player can choose between various numbers of artworks (for example nine, six or four artworks). Choosing a high number of artefacts results in longer game sessions and, again, in a higher difficulty as more images have to be compared to each other.

• Classified Artworks

Users, especially art lovers, might have a preferred epoch, artist, drawing style or artistic medium. With the help of already tagged labels, it is possible to classify the artworks of the database and generate game session containing artefacts of the desired category only. This would not only provide a more enjoyable experience to the player but also result in tags that do not focus on the chosen category as all artefacts share that equal characteristic.

• Time Limit

Despite the many factors speaking against a time limit, that are mentioned in Section 4.1, there is a reason in favour of it: The time pressure leads to more spontaneous annotations. Thus, when looking for labels that express first impressions a time limit is a good choice.

• Sentiments Only

The users are not allowed to write labels describing the visual content of the artefacts but only the emotions they evoke in them. This creates a unique version of the game that challenges the players to come up with tags of a deeper semantic. However, it is necessary to develop a verification method that recognizes whether the entered annotation really describes an emotion.

Pleasure of Working

The current implementation of DICO hides the serious aspect of the game, which is the improvement of the search engine, completely from the players. However, making the contribution to the search engine visible can act as an additional motivating factor. One possibility to visualize the user's contribution are counters. For example, the game could provide a general *contribution counter* that is incremented each time one of the player's actions makes a change to the database. A more specific *unique label counter* could represent all the labels the player came up with first, meaning they were not associated to the artwork in the database before. However, these counters are not to be included in the challenges mentioned above. Reaching a threshold should not influence the game by unlocking new features or in a similar way. Of course the players can challenge themselves to get a high count, but the counters' main purpose is the visualization of the contribution to the search engine, to motivate the player further by granting him the pleasure of working.

Art Gallery

Another possible extension to the game is the implementation of an *art gallery* consisting of a set of artworks. The idea behind the feature is the following: Whenever a game session contains an artefact that the user likes in particular, he can save it and add it to his art gallery. The art gallery provides the opportunity to view all saved artworks. Without this feature, artefacts of already terminated game session cannot be viewed again after the rewind screen has been closed. In the art gallery, each work of art should be captioned with some basic facts including artist, title and year of creation. An art gallery enhances the focus on the art aspect and is an extension that appeals to art lovers. By providing additional facts about each artwork, the art gallery can establish an informative environment for its users, adding a learning aspect to the app.

Simulated Player

Another desirable addition is the implementation of a simulated player to make it possible to play the game alone. Another advantage of a computer opponent is the absence of waiting periods. The bot of the desktop version can be used as orientation and the replaying of previously recorded game rounds can serve as basis of a simulated opponent.

6.4 Outlook

In the preceding section, many additional features extending the current version of the DICO application have been presented. Until now, all extensions of DICO mentioned are built around Artigo and the idea of labeling artworks to feed a search engine. However, the game concept can be applied to other areas as well. Obviously, it can be used to describe all kinds of pictures – it is not restricted to works of art. A user himself could upload the images he wants to play with and create tags for his own pictures. A possible scenario is the labeling of photos of trips with friends: By playing together, the photos get enriched with tags of memories and emotions connected to the pictures. From this it is evident that the descriptions generated through the collected tags can be used for more than just a search

40

6.4. OUTLOOK

engine. For instance, it is possible to utilize the descriptions to illustrate images to those who are visually impaired by reading out the associated tags ordered by their relevance score. Another idea is the transformation of implicit knowledge to explicit knowledge. For example, successors of long-standing employees can profit from the implicit knowledge their predecessor has about the clients and procedures in a company. Confronting an employee with pictures of clients with the help of games like DICO induce him to think about and write down the specificities of each client. In this version, the game would only contain the role of Describer.

Aside from the generation of data, the game concept can be applied in a pedagogical context as an educational game. By showing six preselected images to the players, the skill of differentiating similar images and putting the differences into words can be improved. A possible application area for this are Egyptology classes in which students have to recognize the epochal affiliation of Egyptian buildings. The students, who are shown six buildings of different Egyptian periods, would have to verbalize the characteristics they connect to specific periods and make their decision not intuitive but explicit. The created data could be used and interpreted in learning analytics. Clearly, this can be done in alternative settings to Egyptian buildings such as in the context of churches, castles or palaces of different epochs, styles or religions.

The development of a mobile-friendly version of KARIDO has only been the first step. A consequent next step is to tailor Artigo's search engine to mobile users as well. In the future, further games of the gaming ecosystem can be adapted to the benefit of mobile visitors to enlarge the range of visitors and continue fulfilling the ecosystem's purpose – the labeling of artworks.

Appendix

Screenshots of a game session of DICO on an iPhone can be found on the following pages. The titles and artists of the shown artworks are: *position in the grid*

1. "Theatre Drawing" by Sergei Iur'evich Sudeikin	top left
2. "Der Brand des Houses of Parliament" by William Turner	top right
3. "Schotel met Landschap" by Frederik van Frytom	middle left
4. "Bauernmahlzeit" by Louis Le Nain	middle left
5. "Jüdische Hochzeit (nach Delacroix)" by Pierre Auguste Re	enoir bottom left
6. "Still Life" by Aleksandr Iakovlev	bottom right

CHAPTER 6. CONCLUSION AND FUTURE WORK







Figure 6.2: The menu screen with one game session.

6.4. OUTLOOK



Figure 6.3: The select screen (*a*) and the describe screen (*b*).



Figure 6.4: The guess screen.

CHAPTER 6. CONCLUSION AND FUTURE WORK







Figure 6.6: The image grid (*a*) and the slideshow (*b*).

Bibliography

- [1] AGRAWAL, A., GANS, J., AND GOLDFARB, A. What to expect from artificial intelligence. MIT Sloan Management Review, 2017.
- [2] AHN, L. V., AND DABBISH, L. Designing games with a purpose. Communications of the ACM 51, 8 (aug 2008), 58–67.
- [3] ALVAREZ, G. A., AND CAVANAGH, P. The capacity of visual short-term memory is set both by visual information load and by number of objects. *Psychological science* 15, 2 (2004), 106–111.
- [4] ASSOCIATION, E. S. 2018 essential facts about the computer and video game industry. https://www.theesa.com/wp-content/uploads/2019/03/ESA_ EssentialFacts_2018.pdf, 2018. Accessed on: 2019-12-22.
- [5] ASSOCIATION, E. S. 2019 essential facts about the computer and video game industry. https://www.theesa.com/wp-content/uploads/2019/05/ESA_ Essential_facts_2019_final.pdf, May 2019. Accessed on: 2019-07-02.
- [6] BRADY, T. F., KONKLE, T., ALVAREZ, G. A., AND OLIVA, A. Visual long-term memory has a massive storage capacity for object details. *Proceedings of the National Academy* of Sciences 105, 38 (2008), 14325–14329.
- [7] BRY, F., AND SCHEFELS, C. An analysis of the artigo gaming ecosystem with a purpose. Research Report PMS-FB-2016-6, Institute for Informatics, Ludwig-Maximilian University of Munich, 2016. https://www.en.pms.ifi.lmu.de/ publications/PMS-FB/PMS-FB-2016-6/PMS-FB-2016-6-paper.pdf.
- [8] BUXMANN, P., AND SCHMIDT, H. *Grundlagen der Künstlichen Intelligenz und des Maschinellen Lernens*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2019, pp. 3–19.
- [9] DETERDING, S., DIXON, D., KHALED, R., AND NACKE, L. From game design elements to gamefulness: Defining "gamification". In Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments (New York, NY, USA, 2011), MindTrek '11, ACM, pp. 9–15.
- [10] FACEBOOK INC. Glossary of react terms. https://reactjs.org/docs/glossary.html#singlepage-application/, 2019. Accessed: 2019-10-03.
- [11] FEO MEDIA. Quizduell Fordere deine Freunde heraus! http://www. quizduell-game.de, 2019. Accessed: 2019-9-29.

- [12] FORLINES, C., WIGDOR, D., SHEN, C., AND BALAKRISHNAN, R. Direct-touch vs. mouse input for tabletop displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2007), CHI '07, ACM, pp. 647–656.
- [13] GOOD, B. M., AND SU, A. I. Games with a scientific purpose. *Genome biology* 12, 135 (2011), 1–3.
- [14] GRACE, L. D., AND JAMIESON, P. Gaming with purpose: Heuristic understanding of ubiquitous game development and design for human computation. In *Handbook of Digital Games* (2014), pp. 645–666.
- [15] GÜLER, Ç., KILIÇ, E., AND ÇAVUŞ, H. A comparison of difficulties in instructional design processes: Mobile vs. desktop. *Computers in Human Behavior 39* (oct 2014), 128– 135.
- [16] HAMARI, J., KOIVISTO, J., AND SARSA, H. Does gamification work?-a literature review of empirical studies on gamification. In *HICSS* (2014), vol. 14, pp. 3025–3034.
- [17] HASHEMI, M., AZIZINEZHAD, M., NAJAFI, V., AND NESARI, A. J. What is mobile learning? challenges and capabilities. *Procedia-Social and Behavioral Sciences* 30 (2011), 2477–2481.
- [18] HOLLINGWORTH, A. The relationship between online visual representation of a scene and long-term scene memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition 31*, 3 (2005), 396–411.
- [19] HUANG, Y.-M., KUO, Y.-H., LIN, Y.-T., AND CHENG, S.-C. Toward interactive mobile synchronous learning environment with context-awareness service. *Computers & Education* 51, 3 (2008), 1205–1226.
- [20] HUIZINGA, J. Homo Ludens. Routledge, 1949.
- [21] HUOTARI, K., AND HAMARI, J. Defining gamification: a service marketing perspective. In Proceeding of the 16th international academic MindTrek conference (2012), ACM, pp. 17–22.
- [22] ISFE. Time spent gaming weekly in european countries in Q1 2018. https://www.statista.com/statistics/448427/ time-spent-in-european-countries-per-week/, Aug. 2018. Accessed: 2019-07-13.
- [23] JAIN, S., AND PARKES, D. C. The role of game theory in human computation systems. In *Proceedings of the ACM SIGKDD Workshop on Human Computation* (2009), ACM, pp. 58–61.
- [24] KEUSCH, F., AND YAN, T. Web versus mobile web. Social Science Computer Review 35, 6 (nov 2017), 751–769.
- [25] LMU MUNICH. Play4science projekt. https://www.play4science.uni-muenchen.de/projekt/index.html. Accessed: 2019-08-13.
- [26] LOCKE, E. A., AND LATHAM, G. P. Building a practically useful theory of goal setting and task motivation: A 35-year odyssey. *American psychologist* 57, 9 (2002), 705–717.
- [27] LUCK, S. J., AND VOGEL, E. K. The capacity of visual working memory for features and conjunctions. *Nature* 390, 6657 (1997), 279–281.

- [28] MALONE, T. W. Heuristics for designing enjoyable user interfaces: Lessons from computer games. In *Proceedings of the 1982 Conference on Human Factors in Computing Systems* (New York, NY, USA, 1982), CHI '82, ACM, pp. 63–68.
- [29] MARTINI, P., AND MALJKOVIC, V. Short-term memory for pictures seen once or twice. *Vision research* 49, 13 (2009), 1657–1667.
- [30] MCGONIGAL, J. Reality is broken: Why games make us better and how they can change the world. Penguin, 2011.
- [31] SCHIERENBECK, H., AND WÖHLE, C. B. *Grundzüge der Betriebswirtschaftslehre*, 19 ed. De Gruyter Oldenbourg, 2016.
- [32] SHEPARD, R. N. Recognition memory for words, sentences, and pictures. Journal of verbal Learning and verbal Behavior 6, 1 (1967), 156–163.
- [33] SMITH, G. J., AND VAN LEEUWEN, G. W. Mobile communication optimization near wireless dead zone regions, Apr. 13 2004. US Patent 6,721,572.
- [34] SOMMER, J., DIEDENHOFEN, B., AND MUSCH, J. Not to be considered harmful: Mobile-device users do not spoil data quality in web surveys. *Social Science Computer Review* 35, 3 (2017), 378–387.
- [35] SPERLING, G. The information available in brief visual presentations. *Psychological monographs: General and applied* 74, 11 (1960), 1–29.
- [36] STANDING, L. Learning 10000 pictures. *The Quarterly journal of experimental psychology* 25, 2 (1973), 207–222.
- [37] STANDING, L., CONEZIO, J., AND HABER, R. N. Perception and memory for pictures: Single-trial learning of 2500 visual stimuli. *Psychonomic science* 19, 2 (1970), 73–74.
- [38] STATCOUNTER. Mobile internet traffic as percentage of total web traffic in february 2019, by region. https://www.statista.com/statistics/306528/ share-of-mobile-internet-traffic-in-global-regions/, Mar. 2019. Accessed: 2019-06-11.
- [39] STATISTA. Number of internet users worldwide from 2009 to 2018, by region. https://www.statista.com/statistics/265147/ number-of-worldwide-internet-users-by-region/, Sept. 2018. Accessed: 2019-07-02.
- [40] STATISTA. Smartphones in Germany. https://www.statista.com/study/ 39558/smartphones-in-germany-statista-dossier/, 2019. Accessed: 2019-06-11.
- [41] STEINMAYR, B. Designing image labeling games for more informative tags. diploma thesis, Institute for Informatics, Ludwig-Maximilian University of Munich, 2011. https://www.en.pms.ifi.lmu.de/publications/diplomarbeiten/ Bartholomaeus.Steinmayr/DA_Bartholomaeus.Steinmayr.pdf.
- [42] STEINMAYR, B., WIESER, C., KNEISSL, F., AND BRY, F. Karido: A GWAP for telling artworks apart. In *Proceedings of the 16th International Conference on Computer Games* (Louisville, KY, USA, July 2011), pp. 193–200.
- [43] VOGEL, E. K., AND MACHIZAWA, M. G. Neural activity predicts individual differences in visual working memory capacity. *Nature* 428, 6984 (2004), 748–751.

- [44] VOGEL, E. K., WOODMAN, G. F., AND LUCK, S. J. Storage of features, conjunctions, and objects in visual working memory. *Journal of Experimental Psychology: Human Perception and Performance* 27, 1 (2001), 92–114.
- [45] VON AHN, L. Games with a purpose. Computer 39, 6 (2006), 92–94.
- [46] VROOM, V. H. Work and motivation, vol. 54. Wiley New York, 1964.
- [47] WHEELER, M. E., AND TREISMAN, A. M. Binding in short-term visual memory. *Journal of Experimental Psychology: General 131*, 1 (2002), 48–64.
- [48] ZEREGA BRAVO, R., AND LAZAROV, B. To touch or not to touch: A comparison between traditional and touchscreen interface within personal computers. bachelor thesis, Södertörn University, School of Communication, Media and IT, 2011. http: //www.diva-portal.org/smash/get/diva2:497191/FULLTEXT01.pdf.
- [49] ZHU, Y., HALPERN, M., AND REDDI, V. J. The role of the cpu in energy-efficient mobile web browsing. *IEEE Micro* 35, 1 (2015), 26–33.