DigiTaps

Vaspol Ruamviboonsuk
Computer Science and Engineering
University of Washington

June 10, 2014

Abstract

Visually impaired people use touchscreens on smartphones with a screen reader. The screen reader provides audio feedback to the user for navigating the phone. However, screen readers are difficult to use in a noisy setting because the users must pay careful attention to the audio feedback. We introduce *DigiTaps*, an eyes-free number entry method with minimal audio feedback. The digits are represented by a combination of simple primitive gestures such as tap and swipe. Users can perform the gestures anywhere on the screen. We conducted a preliminary lab study to compare the DigiTaps methods to the screen reader method. The participants achieved a significantly lower error rate using the DigiTaps method than using the screen reader method, while still achieving a higher entry rate. To evalutate the gestures further in a real-world setting, we develop DigiTaps game; a number entry game designed for conducting user studies in the wild. We made it available on the Apple App Store and use it as a platform to collect the players data such as touch events and game playing statistics. With the data collected, we analyze the data to evalutate the two DigiTaps methods.

Contents

1 Introduction

Visually impaired people use touchscreen on smartphones with a screen reader. Screen readers, such as Apple's VoiceOver [1], provides audio feedback for the user to navigate the phone. The user uses his finger to find the target item on the screen. The screen reader gives continuous voice feedback to the user. Once the target is reached, the user performs a selection gesture. A selection gesture can be either a split-tap or a double-tap anywhere on the screen. Since the user has to pay careful attention to the voice feedback, using a screen reader in a noisy setting is difficult. Wearing headphones is not practical because blind users feel unsafe since they use the sounds around them to navigate and understand their surroundings [3].

We introduce *DigiTaps*; an eyes-free number entry method that uses minimal voice feedback. The digits are represented by a combination of simple gestures such as tap and swipe. Users can perform the gestures to enter the digits anywhere on the screen. For example, the number 20 is entered by performing a two-finger tap and a swipe anywhere on the screen. We conducted a preliminary study to compare the DigiTaps method to a numeric keyboard with a screen reader. The participants achieved a significantly lower error rate by using DigiTaps method than using the screen reader method, while still achieving a high entry rate.

In addition to the preliminary evaluation, a longitudinal lab study was conducted to compare the two DigiTaps codes when the codes are used with a voice and haptic feedback together and when the codes are used with only haptic feedback. The lab study shows that the participants can enter numbers fast and accurate using the DigiTaps number entry method and the DigiTaps codes are good alternatives to using a numeric keyboard with screen reader. [2]

Even though a longitudinal lab study was conducted, we would like to evaluate the gestures in a real-world setting. To emulate a real-world use of the gesture, we developed DigiTaps game for conducting number entry method user studies in the wild. We distribute the DigiTaps game via Apple App Store to expose it to a large number of players. The player has to enter the numbers shown by the game using one of the DigiTaps code. The game consists of 5 levels and the difficulty increases as the player progresses through the levels. We collect data at every touch event that the player has performed along with the metadata information such as the indicator of the beginning of the level, the number presented to the player and what number was shown to the players. We use the data for evaluating the DigiTaps codes.

2 Related Work

2.1 Eyes-free Text Entry Methods

Blind users use smartphones by using a screen reader that pre-installed on smartphones such as iOS's Voice Over [1] and Android's TalkBack [6]. Both screen readers use the interaction techniques similar to what Kane et al. presented in [10]. To interact with the screen, a user explores the screen with his finger and gets audio feedback from the phone when an element is touched. The user has to carefully listen to the audio feedback. When the target is reached, the user performs a selection gesture which can either be a double-tap anywhere on the screen or a split-tap (tap another finger on the screen while the first finger is still on the selection target). Even though the screen readers are widely adopted in the blind community, using screen readers to enter text achieves a low text entry rate. Studies have shown that blind users can enter text at only 4.5 words-per-minute using VoiceOver [4]. Furthermore, using screen reader method is error-prone [13].

Many eyes-free text entry methods have been introduced recently. Azenkot et al. presented PerkInput, a chorded input for touchscreens where Braille cells are input one column at a time [4]. Southern et al. presented BrailleTouch, an application for typing braille based on the six-key braille keyboard [15]. These text entry methods require finger calibrations or allow the user to touch in only certain locations on the screen. In contrast with DigiTaps, users are not required to calibrate their fingers with the phone and can perform the gesture anywhere on the screen. Furthermore, knowledge of Braille is a requirement for using PerkInput and BrailleTouch while DigiTaps does not require the users to know Braille.

MacKenzie et al. introduced an eyes-free text entry method using a joystick called H4-writer [12]. This method achieved a 20WPM text entry rate. H4-writer uses an optimal prefix-free code, Huffman Code, for encoding the words. Unlike the DigiTaps code, the Huffman Code does not relate to the semantics of the symbols. Thus, it is not as intuitive as DigiTaps' code which all the codes are related to the actual meanings of the symbols.

Graffiti and Unistroke [5] uses print symbols for representing each character. Unlike Graffiti and Unistroke, DigiTaps uses simple gestures like tap and swipe to represent its symbols. Blind users do not have the knowledge of how the print symbols look like. While using the print symbols in the text entry method is efficient for sighted users, blind users can have trouble performing them because their drawing skill would be less accurate than sighted users [11].

2.2 Games for User Study in the Wild

Henze et al. conducted a user study in the wild on analyzing touch performance [7] using a text entry Android game distributed via Google Play Store. The game collected important data from the players such as touch positions to analyze the accuracy and the performance of the touch events. We adopted the application design presented in [8] and developed DigiTaps game as a platform for conducting number entry method user studies in the wild.

3 DigiTaps Codes

Numbers are represented in different ways using different numeral representation system such as base-2 and base-10. Numbers that we use on a daily basis are mostly represented in base-10 numeric system. If we represent the first 10 digits, 0 through 9, in base-5, the digits are represented as 0, 1, 2, 3, 4, 10, 11, 12, 13, 14. We call these representations *codes*. There are two types of codes fixed length and variable length. All the codes represented by fixed length have the same length. For example, if 0 through 9 are represented with fixed-length code in base-5, we write them as 00, 01, 02, 03, 04, 10, 11, 12, 13, 14 and all the codes are 2 digits long.

In contrary, variable length codes can have variable length. For example, to write the first nine digits, 0 through 9, in base-5 system using variable length code, we write them as 0, 1, 2, 3, 4, 10, 11, 12, 13, 14. The first five codes are 1-digit long and the rest of the codes are 2 digits long. Variable-length codes are shorter on average. Since the variable length codes are potentially shorter than fixed length codes on average, we decided to use variable-length codes for representing the numbers.

Although variable length codes can represent numbers with shorter codes than fixed length codes. It introduces an ambiguity to the codes. For example, suppose we have a program to decode base-5 variable length codes to base-10 digits. We enter the number 1 into the program. The program cannot determine whether the number 1 suppose to be the code 1 or the first digit of 11. The problem does not arise in fixed length codes because all the code have the same length. Thus, the digit 1 is represented by 01. To resolve the ambiguity, we use prefix free code, a variant of variable length code.

Prefix free codes do not allow a code to be a prefix of another code [9]. In our example above, if we use prefix free code to represent 0 through 9 in base-5, there cannot be 10, 11, 12, 13, 14 in the codes because 1 is already a code by itself and it cannot be a prefix of another code. Since using prefix free code removes the ambiguity among the codes and it is a variable length code, we use prefix free code to represent our codes in DigiTaps.

3.1 Espresso

We develop an easy to learn prefix free code called Espresso. With Espresso, the digits can be derived by adding the numbers 0, 1, 2, and 3. The numbers 0, 1 and 2 are represented by one finger swipe, one-finger tap, and two-finger tap, respectively. 3 is represented by three-finger tap and a swipe, $(3+0)$. Similarly, 4 is represented by three-finger tap and a one-finger tap, $(3+1)$. 6 is represented with 2 two-finger taps and one-finger swipe, $(3+3+0)$. The final swipe is required to ensure that the code is prefix-free. Finally, 9 is represented by 3 three-finger taps, $(3+3+3)$. A swipe is not necessary for 9 because 3 three-finger taps is not a prefix to another input (see table 3.1).

The numbers 0 through 2 in Espresso are 1 gesture long. The next 3 digits, 3 through 5, are 2 gestures long and the rest of the numbers are 3 gestures long. Assuming all digits are equally likely, this gives us an average gesture per digit of 2.1.

$$AVG(Espresso) = \frac{1 \times 3 + 2 \times 3 + 3 \times 4}{10} = 2.1$$

3.2 Cappuccino

We developed an optimal prefix-free code on 4 symbols that requires lower number of gestures per digit on average called Cappuccino. In Cappuccino, one-finger swipe represents either 0 or 10. A two-gesture digit is derived by subtraction, with the one-finger swipe representing 10. For example, 9 is represented by a swipe followed by a one-finger tap $(10-1)$. Similarly, 8 is

Table 3.1: Espresso Codes

Digit	Code
0	1-finger swipe
1	1-finger tap
2	2-finger tap
3	3-finger tap + 1-finger swipe
4	3-finger tap + 1-finger tap
5	3-finger tap + 2-finger tap
6	3-finger tap + 3-finger tap + 1-finger swipe
7	3-finger tap + 3-finger tap + 1-finger tap
8	3-finger tap + 3-finger tap + 2-finger tap
9	3-finger tap + 3-finger tap + 3-finger tap

represented by a swipe and a two-finger tap $(10-2)$ and 7 is represented by a swipe and a three-finger tap. In all other cases, one-finger swipe represents 0. For instance, the digit 3 is represented by three-finger tap and a swipe $(3+0)$. However, the digit 0 is represent by 2 one-finger swipes. The second one-finger swipe is added to make the code prefix free. Furthermore, the number 6 is represented by two three-finger taps. Espresso and Cappuccino have the same code for the digits 1 through 5 (see table 3.2).

The numbers 1 and 2 in Espresso are 1 gesture long. The next 3 digits, 3 through 9, are 2 gestures long. Assuming that all the digits are equally likely, this gives us an average gesture per digit of 1.8.

$$AVG(Cappuccino) = \frac{1 \times 2 + 2 \times 8}{10} = 1.8$$

4 Preliminary Study

To evaluate the potential of the gestures, we compare the Espresso method to a standard accessible numeric keyboard entry method, VoiceOver, using (1) a theoretical analysis of the methods and (2) an empirical comparison with five users. We choose the Espresso method over the Cappuccinno method because it is more intuitive to learn.

Table 3.2: Cappuccino Codes

Digit	Code
0	1-finger swipe + 1-finger swipe
1	1-finger tap
2	2-finger tap
3	3-finger tap + 1-finger swipe
4	3-finger tap + 1-finger tap
5	3-finger tap + 2-finger tap
6	3-finger tap + 3-finger tap
7	1-finger swipe + 3-finger tap
8	1-finger swipe + 2-finger tap
9	1-finger swipe + 1-finger tap

4.1 Theoretical Analysis

In the standard method, there are two steps involved in entering a digit. First, the user has to explore the screen to locate the button where the digit resides. The seek time is hard to quantify, but it requires listening to the buttons touched until the correct one is found. Once the button is located, the user performs a selection gesture. One selection gesture is a split tap, holding a finger down on the target button and using another finger to tap the screen. Another selection gesture is one-finger double tap anywhere on the screen. At a first attempt, this method appears to be more difficult than the 2.1 taps per digit.

4.2 Empirical Evaluation

Our empirical evaluation consists of a study with five sighted participants. In each of the study, the participants entered 10 six-digit numbers using the standard and the Espresso methods. Participants held the smartphone beneath the desk so that they were not able to see the screen. After a brief practice session, the participants achieved an average number entry rate of 1.99 seconds per digit ($SD = 1.25$) and 2.77 seconds per digit ($SD = 1.24$) using the Espresso method and the standard VoiceOver method, respectively. The error rates were far lower in the Espresso method than in the VoiceOver method. The participants averaged a Mean String Distance (MSD)[1] of only 1% using Espresso method. Whereas, using the standard method, the participants achieved a MSD of 14.2% on average. The Espresso method outperformed the standard method in both speed and accuracy. Unsurprisingly, all five participants perferred the Espresso method to the standard method [14].

While the results are from a preliminary study, they show that the Espresso method has potential to out-perform the standard numeric entry method. We decided to conduct an extensive study not only on the Espresso method, but on both the Espresso and the Cappuccino methods.

5 DigiTaps Game

Since the preliminary evaluation of the DigiTaps code shows promising results, a more rigorous study is needed to evaluate the Espresso and the Cappuccino methods. The Espresso and the Cappuccino methods have been evaluated in a lab study as presented in [2]. The study compared the performance of the Espresso method to the performance of the Cappuccino method when using the methods with different kinds of feedback, only haptic feedback and both haptic and audio feedback. With both the audio and haptic feedback on, the participants entered the numbers at a rate of 0.82 digits per second and 0.78 digits per second using the Cappuccino method and the Espresso method, respectively. The participants achieved a low mean uncorrected error rate of 2.17% and 2.22% using the Cappuccino method and the Espresso method, respectively. The longitudial lab study's results agreed with the preliminary evaluation that the Espresso method is fast and accurate.

Even though the lab study and the preliminary evaluation showed that the participants entered numbers with a fast speed and accurately using the DigiTaps codes, it does not reflect the real use of the gestures in a real-world setting. We want to design a user study that simulate real world use of the gestures. Hence, we adopted the user study model presented in [7] to conduct our user study in the wild. We developed DigiTaps Game for conducting number entry method user studies in the wild. The game is designed to emphasize the number entry speed and the accuracy of inputting

[1] Defined as $\sum$ distance from original string $\div$ total digits

numbers, which we considered as the prominent performance factors in real world use of the gestures.

The DigiTaps game collects touch events that the player performs while they are playing the game. We collect the touch events for evaluating the Espresso and the Cappuccino methods. We expose the DigiTaps game to a large number of players by distributing the game via the Apple App Store.

5.1 Flow of DigiTaps

The first screen shown to the player after launching DigiTaps is the main menu of the game as (see Figure 5.1). This screen consists of three choices for the player to choose (1) Tutorial, (2) Start Game, (3) Leaderboard. Selecting different button leads to different modes of the DigiTaps game and leads to different screens that the player visits.

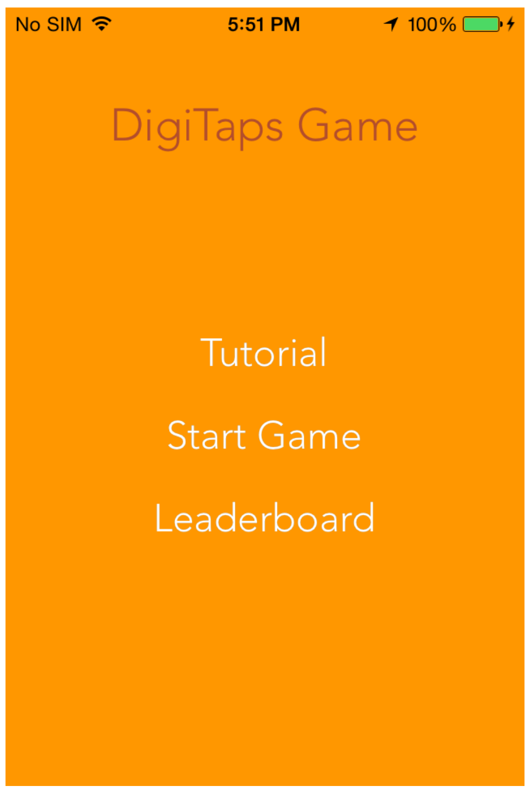

Figure 5.1: Main menu and the first screen of DigiTaps

5.1.1 Tutorial

We provide tutorials for the players to get started with the DigiTaps codes. Each tutorial consists of the description of the code and the practice mode for the code. There are three tutorials total. The first tutorial is the overall description of the game. It describes the main gestures of the game such as long-press to submit the number. The two other tutorials describe the DigiTaps codes. In each DigiTaps code description screen, a table describing how the gestures map to the DigiTaps codes is provided. The practice screen allows the player to try out the gestures they learned in the description screen. The practice screen gives the user feedback on the action the player just perform. For example, if the player tap the screen with three fingers, the screen shows that it is a three-finger tap (see the rightmost state in figure 5.2).

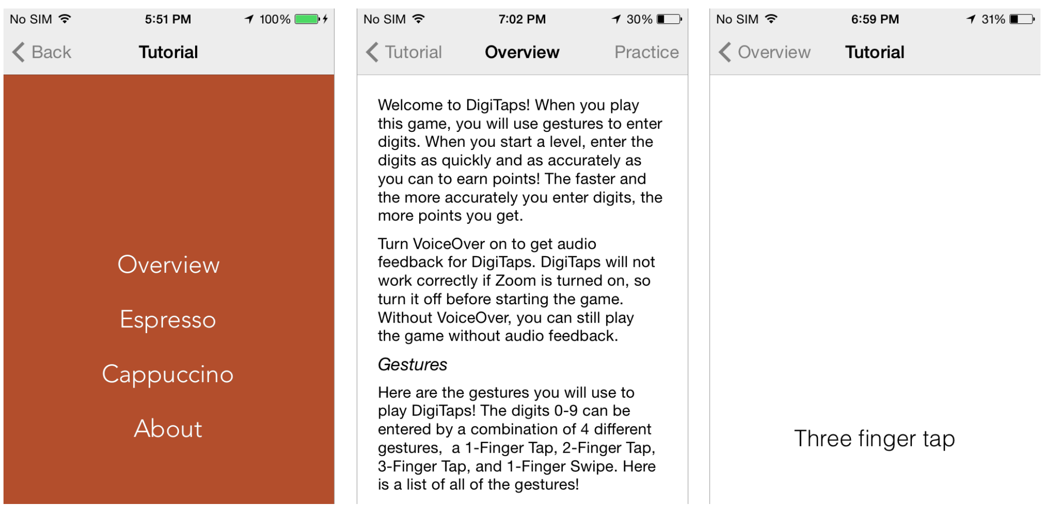

Figure 5.2: The state on the left shows the options the player can choose to learn about. The state in the middle is an example of a tutorial page where descriptions are provided. The state on the right is an example of a practice screen.

5.1.2 Game Play

There are 4 main screens (1) a DigiTaps method selection screen, (2) a level selection screen, (3) a game play screen, and (4) a summary screen. At the method selection screen, the DigiTaps game lets the player chooses either the Espresso method or the Cappuccino method. The player will use the method chosen to enter numbers through out the game. After the method has been chosen, the player has to choose a level to start playing the game, indicated by the red arrow in figure 5.3 from the top row to the leftmost state on the second row.

At this point, the game play screen slides in and show the first number to the player (see the leftmost state of the second row in figure 5.3). The player performs the gestures to enter the number shown on the screen. To submit the number, the player has to tap and hold that finger on screen with one finger until a ringing or a buzz sound occurs. The sound means that the number has been submitted. A ringing sound indicates that the number input was correct, and the buzz sound indicates an incorrect attempt. The player, then, advances to the next number. The player has to enter 10 numbers per level and the number of digits per number varies based on the level. The numbers starts with 3 digits per number at level 1 and the number of digits increases by one digit as the player progresses through the levels. At the last level, level 5, each number is 7 digits long.

After entering all the numbers required, the summary screen appears. The summary screens provides the player's performance on that level. The information includes the points earned, the accuracy and the average time per digit. On this screen, the player has the option to advance to the next level, replay the same level or quit to the start menu as (see the last state of figure 5.3).

5.1.3 Leaderboard and Scoring

To provide the players with competition, we include Apple's Leaderboard into DigiTaps. The Leaderboard ranks the player's score with other players around the world who plays the DigiTaps game and uses Apple's Game Center. Players are being ranked by the scores they achieved in the game (see figure 5.4). Players can compare their scores with other players' scores and see how well they are doing.

We compute the points based on the player's speed and accuracy. Each number in a level contributes to the overall points in a level. If the player incorrectly enters a number, the player will get zero points for that number. If the player enters the number correctly, the score for that number will be

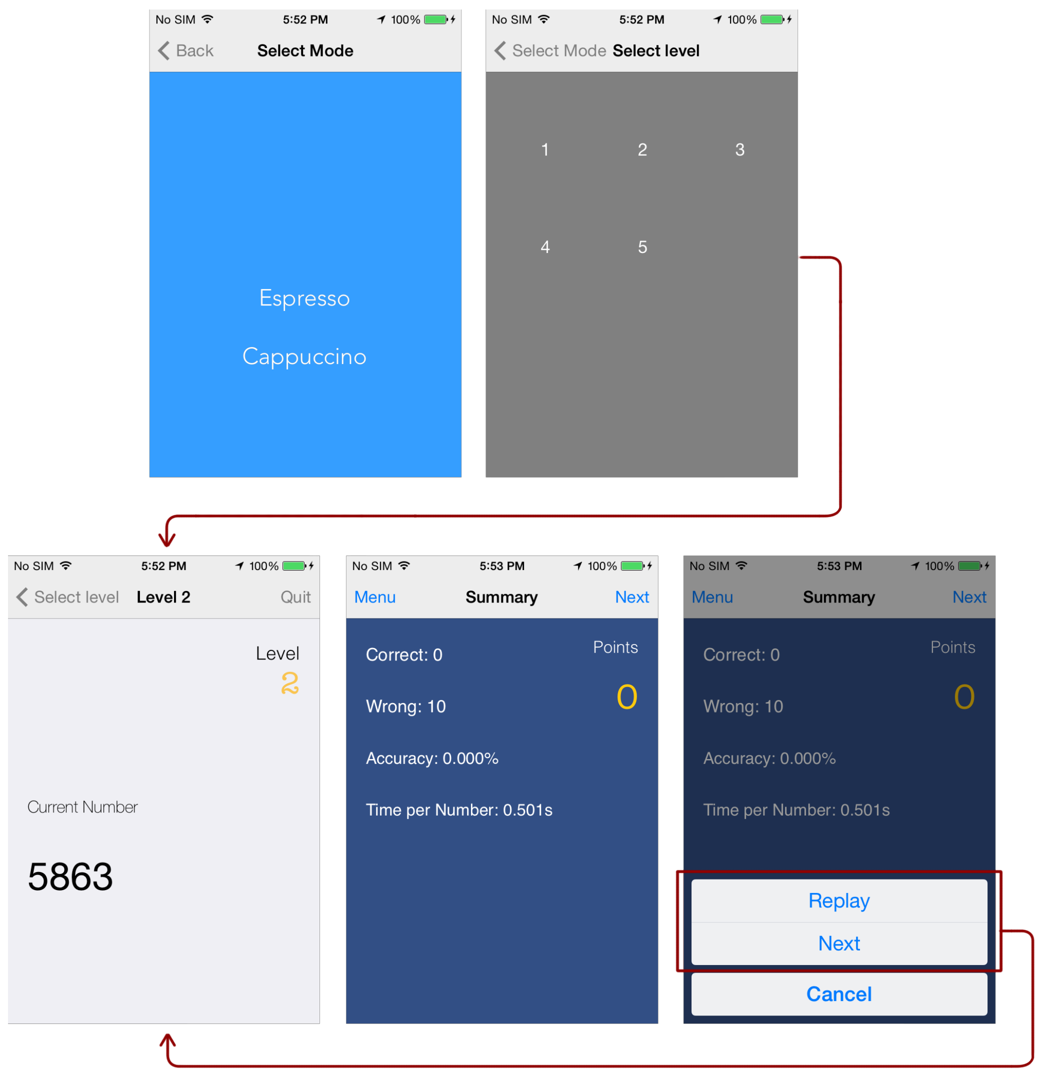

Figure 5.3: Shows the state diagram of DigiTaps gameplay mode.

computed. The score is computed using the following formula.

$$score = \frac{\text{expected time used}}{\text{actual time used}} \times \text{baseline}$$

In other words, if the player enters the number faster than what the game has expected, the game will reward the player with a higher score than the baseline. In contrary, if the player enters the number slower than what the game has expected, the game will penalize the player with a lower score. The baseline is an arbitrary number.

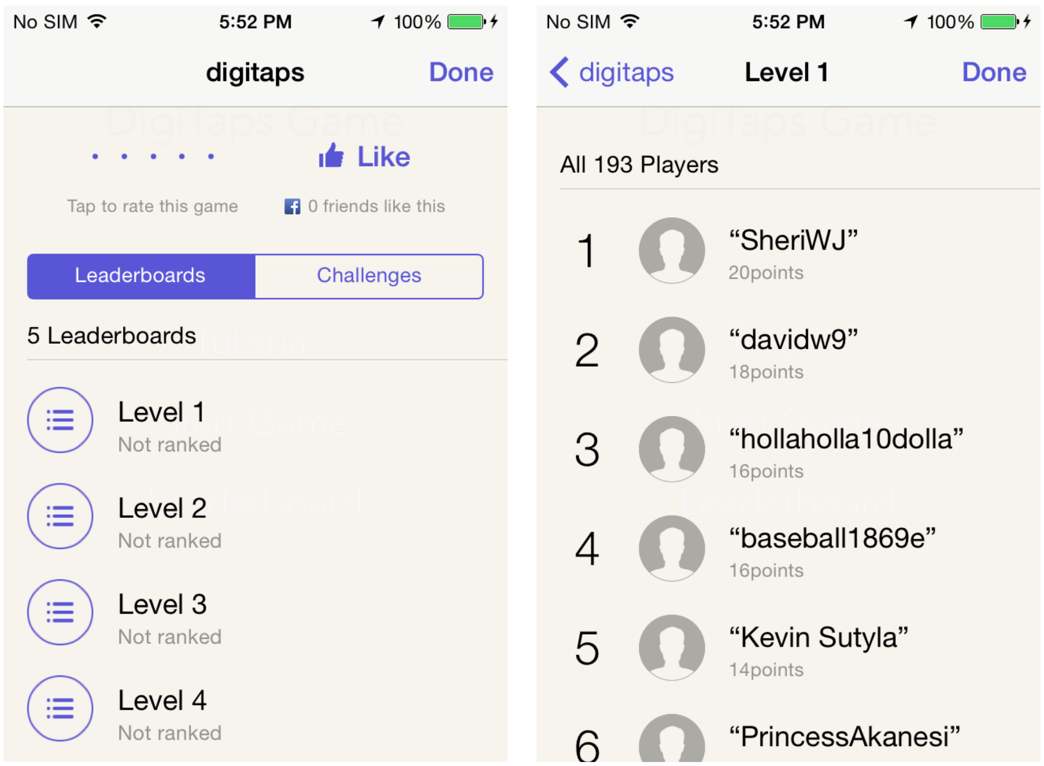

Figure 5.4: The state on the left shows the leaderboard of each level. The state on the right shows the rankings of the first six places for level 1 with their respective points.

5.2 Interaction Gestures

In addition to the DigiTaps gestures described in the DigiTaps code section, the DigiTaps game uses two other gestures for players to interact with the game. The first gesture is two-finger swipe in any direction. This gesture does two actions in the game. If the player already input numbers into the game, two-finger swipe in any direction activates the backspace action. Since the number initially shown on the screen disappears when the player starts entering the numbers, we use two-finger swipe in any direction to repeat the number. DigiTaps game repeats the number by reading the number to the player again and shows the numbers on the screen again.

The second special gesture is the long-press gesture. This gesture is used for advancing to the next number in the level or if there is no more number on that level, this gesture takes the player to the summary screen. When the player finishes inputting the given number, the player holds one-finger down on the screen until there is a bell ring or a buzz sound from DigiTaps. The sound indicates that the game has advanced to either the next number or the summary screen. The bell sound indicates a correct attempt and the buzz sound indicates a failed attempt.

5.3 Data Collection

In order to conduct the user study, we collect two different kinds of information from the players, the players' demographics and the touch events that the players performed. The demographics information includes the player's age, gender, experience with accessibility in iOS. Each player is uniquely identify with an identification number. However, we cannot trace back to who the actual player was. This identifier is used to match the player to the other information of this player that collected.

The touch events that the players performed in the game is also recorded. In each of these touch events, we record the number entered and the timestamp when the event occurred. Furthermore, we recorded the state of the game such as when the game starts, the level finishes, and the number is inputted. We recorded the numbers presented to the players and the numbers that the players entered. We used this information to measure the accuracy of the players.

Using the information collected, we can gain some insight on how the players behave in the game. More importantly, we can evaluate the speed and the accuracy of the players performing the gestures by analyzing the players' data.

6 Evaluation

As of writing this paper, there are 654 registered players. In addition, we collected 129,906 events which contains 96,892 gesture events.

6.1 Method

We distributed the DigiTaps game to the Apple App Store. We would like the game to be exposed to a large group of users both sighted and unsighted. We also publicized the game through mailing lists that are actively used in the braille community.

6.2 Demographics

The following are the result of the 6 demographics questions we asked.

1. **How old are you?**
   47.71% of the 654 players are under the age of 25 (see figure 6.1) with players aged between 18 and 25 years old is the largest population in this group. Since DigiTaps is a game, this result is not surprising. People at a younger age may want to try out the game more than the older people.

2. **What is your gender?**
   The genders are split almost evenly between male and female. Male accounts for 48.01% and female accounts for 51.99% of all the DigiTaps players.

3. **How would you identify yourself?**
   We advertised DigiTaps through several blind organizations' mailing lists. It is not surprising that 48.5% of the 654 players indentified themselves as blind (see figure 6.2). Sighted players contributes 37.9% of all the players.

4. **Do you use any accessibility tool on a daily basis?**
   Since 48.5% of the players identified themselves as blind, it is not surprising that the majority of the players uses VoiceOver on a daily basis. However, there are 61.3% players who use VoiceOver regularly, which is more than 10% greater than the number of blind players (see figure 6.3. One possible explanation is 13.6% of the players identified themselves as having low-vision. Low-vision and blind players combined are 62.1% of all the players.

5. **How long have you owned an iOS device?**
   We also asked how long that they owned the device, so we can measure

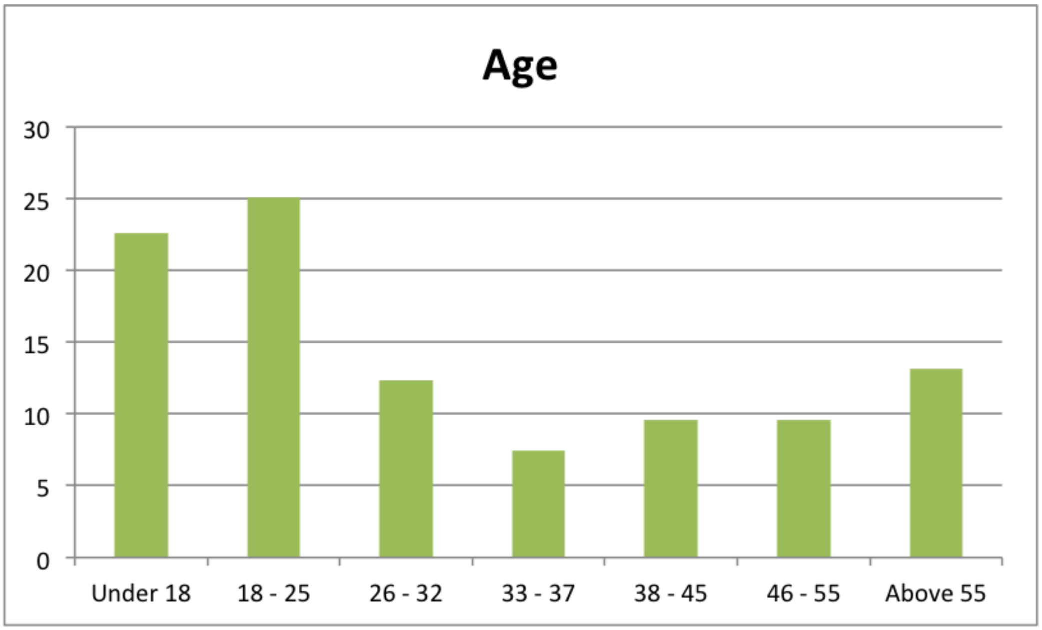

Figure 6.1: Age among all the players.

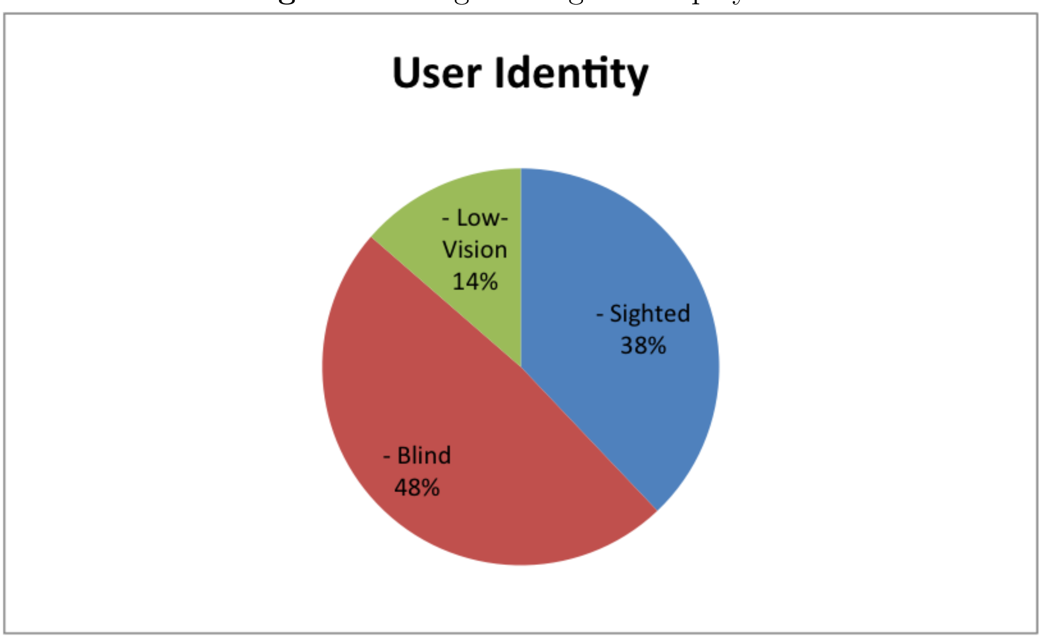

Figure 6.2: Players' identification among all the players

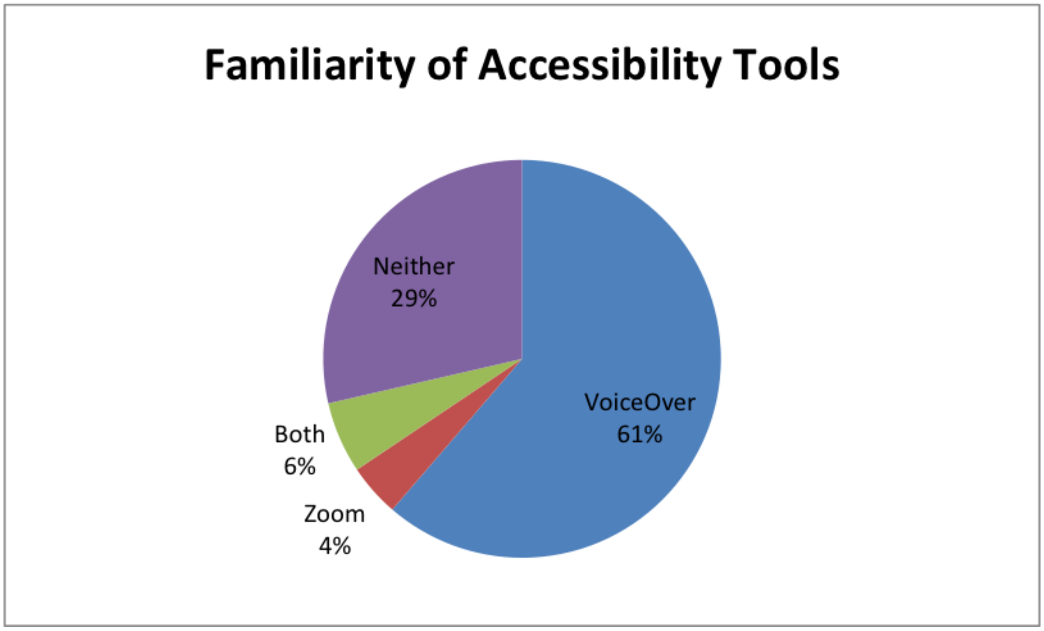

Figure 6.3: Accessibility tool usage among all the players.

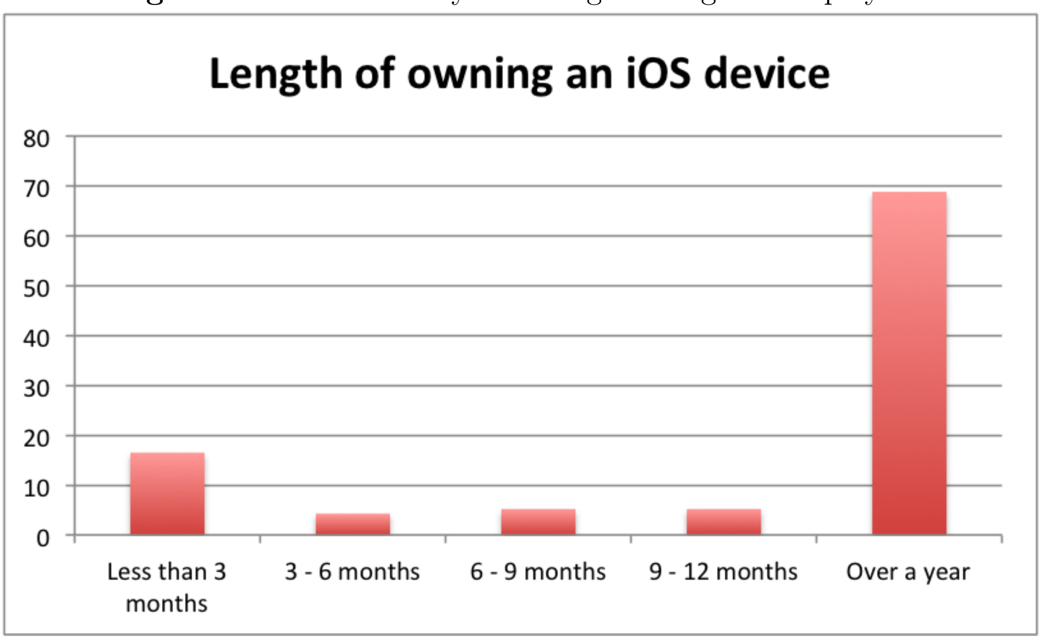

Figure 6.4: How long the player possessed his device among all the players

their fluency with the platform. The majority, 68.8%, of the players own their device for over a year (see figure 6.4). The second largest population is the players that owns the phone less than 3 months.

From all the players who played the DigiTaps game, we select the top 6 players ranked by the amount of numbers the entered. We evaluate the performance of each player on different type of gestures by measuring the accuracy and the speed when the player enter the numbers. We will identify each of the player as P1 to P6, respectively.

6.3 Results

Even though there is some game play that can do some data analysis on the two gestures, the number of taps of each method varies significantly. For example, P3 entered only 7 numbers using Cappuccino while entered 392 number using Espresso. Likewise, P6 did not enter any number using the Cappuccino method. As a result, some results of some players is significantly different from other players and strongly influenced our results.

6.3.1 Overall Accuracy

By using the Espresso method, P1 entered 87% of the numbers correctly and was the most accurate player among the six players. By contrast, the least accurate player, P4, entered 75% of the numbers correctly. On the other hand, P1 was also the player who entered the number most accurately using the Cappuccino method. P1 achieved an accuracy rate of 90% over 962 digits.

6.3.2 Overall Speed

P4 entered the numbers the fastest at the rate of 0.48 digits per second ($SD = 0.16$) with an accuracy of 75% using the Espresso method. P2 was slowest among all the six players with 0.32 digits per second ($SD = 0.13$). In comparison, P5 achieved the highest entry rate of 0.52 digits per second ($SD = 0.09$) using the Cappuccino method while the player is still 87% accurate. P2 was also the slowest using the Cappuccino method. P2 entered the numbers at the rate of 0.31 digits per second ($SD = 0.13$).

6.3.3 Accuracy Development

We use the same subset of players, P1 through P6, to examine how the players improve their accuracy and entry rate over time. To mimic one session in the lab study presented in [2], the numbers that a player entered are divided into six equal groups. In addition to the six groups, there is one more group that contains the rest of the numbers that cannot fit into the first six groups.

In Espresso, players started with accuracies ranging from 70% to 89%. As the players progressed, they did not necessary improve their accuracies between sessions. For example, P4's accuracy dropped by 45% as the player progressed from session 2 to session 3. However, as they progressed to further sessions, their accuracies improved. At the end of the sixth session, all of the players improved their accuracies (figure 6.5). We also observed the same trend in the Cappuccino method (figure 6.6).

6.3.4 Speed Development

In Espresso, Players started with entry rates ranging from 0.20 to 0.55 digits per second. As the players progressed through the sessions, most players gradually improved their entry rates and achieved at least a 40% increase. However, we observed a different result for P4. Surprisingly, the number entry rate of P4 declined at the end of the sixth session (figure 6.7). The players' number entry rates improvement is not as clear in the Cappuccino method. There are marginal improvements to the number entry rates of the players. However, all the players improved their number entry rates at the end of the sixth session. The seventh session may contain few numbers and that can strongly influence the entry rate for the session (figure 6.8).

6.3.5 Game Trends

Players played the first few levels more often than the later levels. On level 1, The 6 players entered more than 200 numbers using the Cappuccino method and almost 150 numbers using the Espresso method. However, the amount of numbers entered decreased by 62.5% and 66.7% in the Cappuccino method and the Espresso method, respectively. The amount of numbers entered continued to decline in the subsequent levels (figure 6.9).

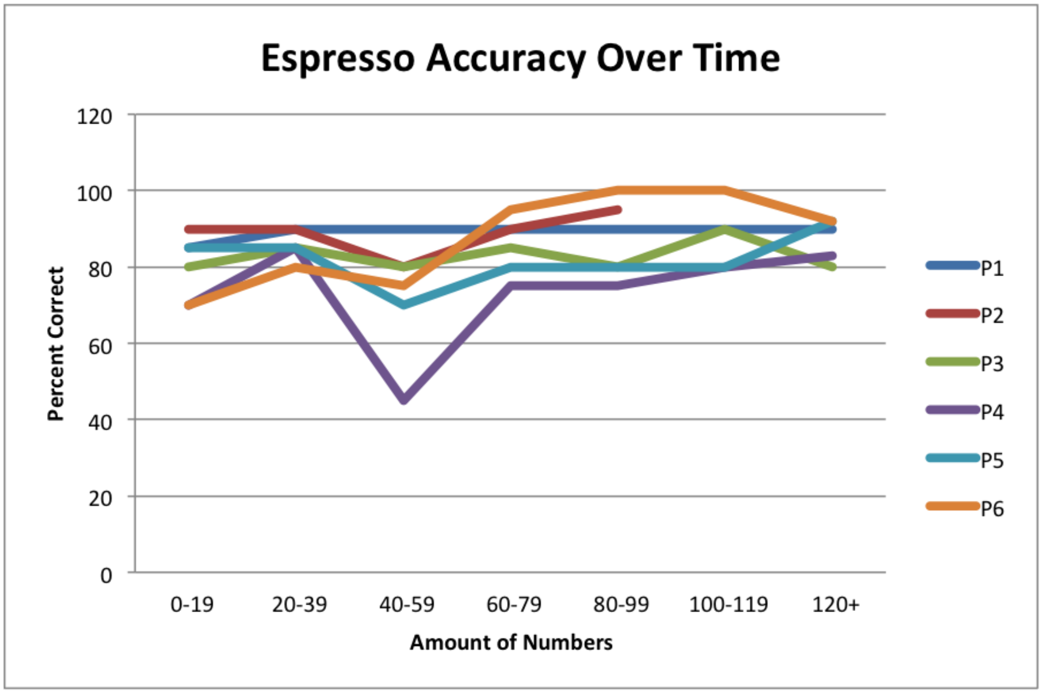

Figure 6.5: Breakdown of the accuracy per every 20 numbers for players using the Espresso method

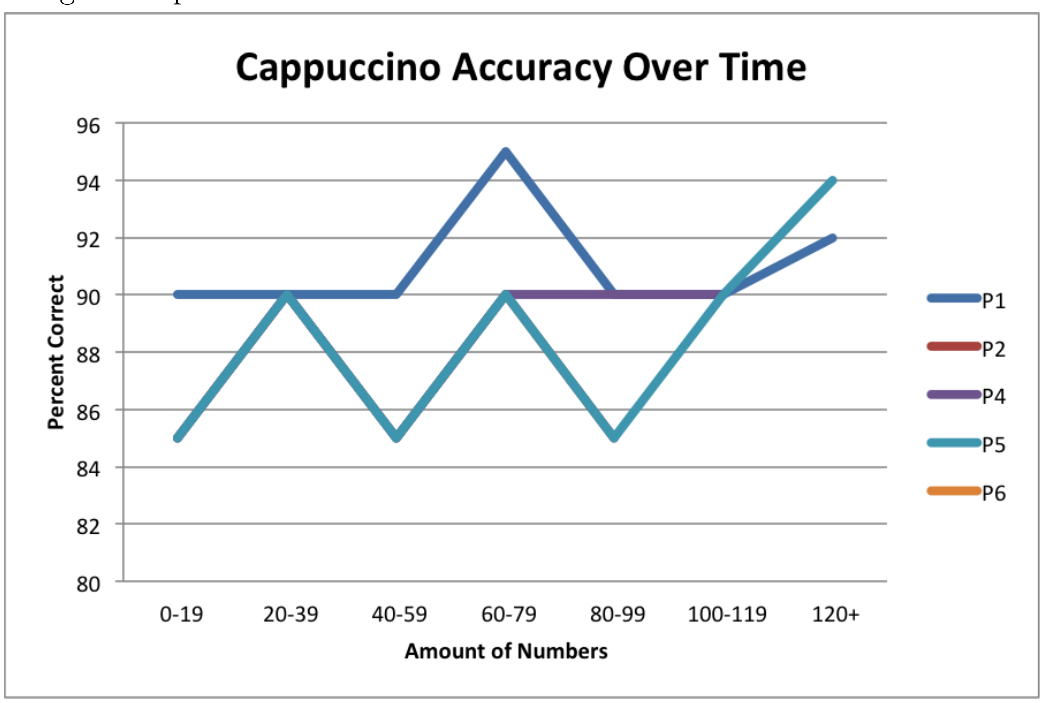

Figure 6.6: Accuracy per every 20 numbers for players using the Espresso method

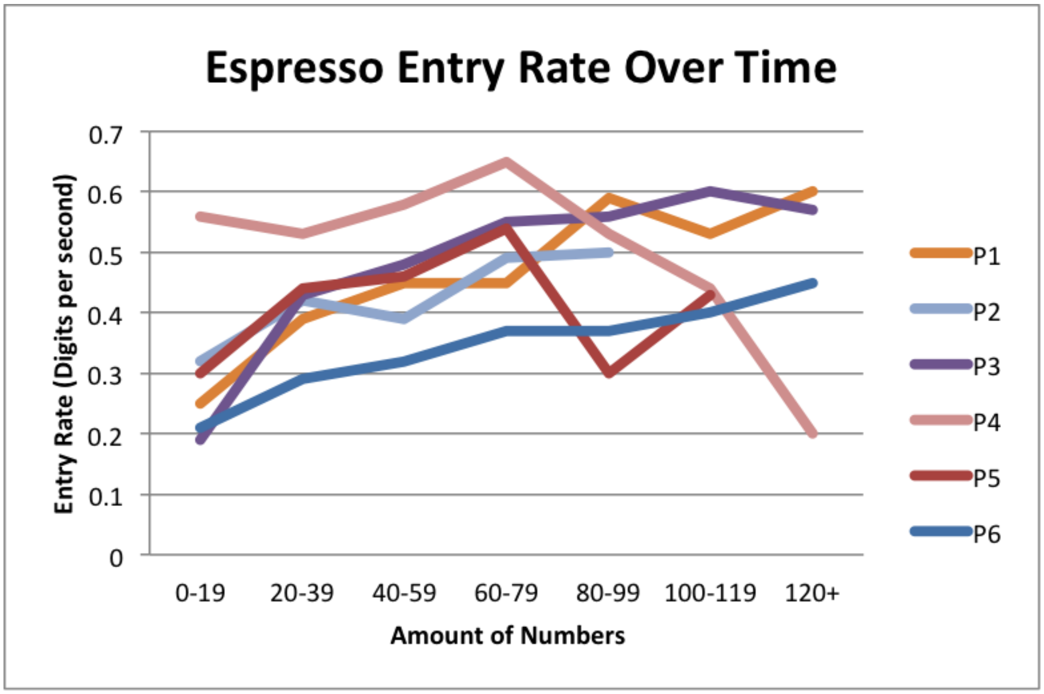

Figure 6.7: Speed per every 20 numbers for players using the Espresso method

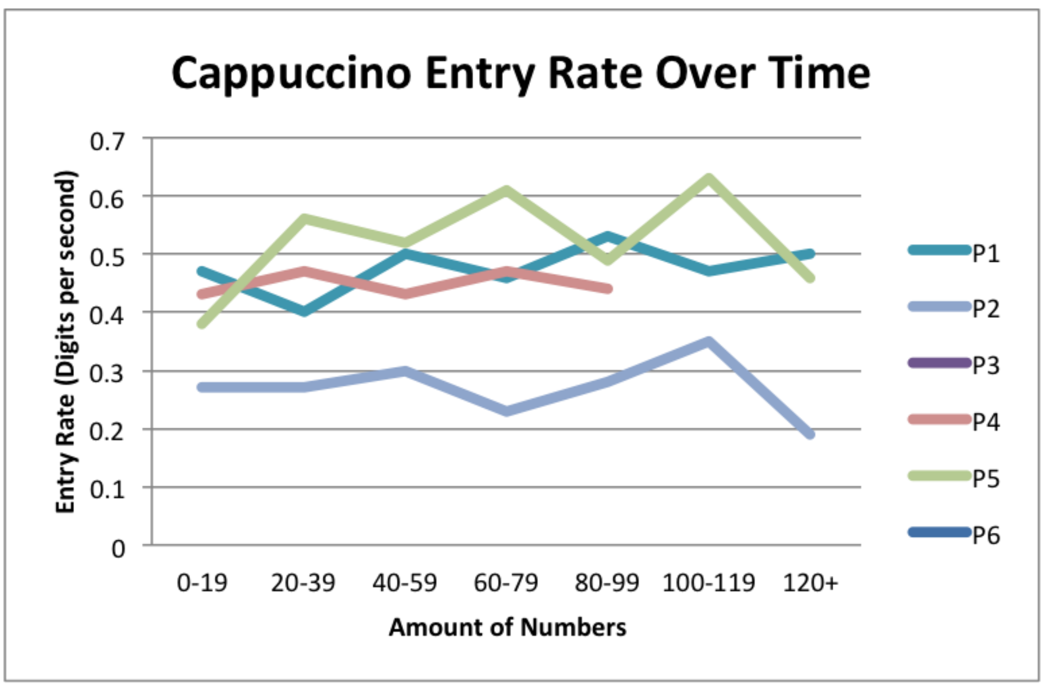

Figure 6.8: Speed per every 20 numbers for players using the Espresso method

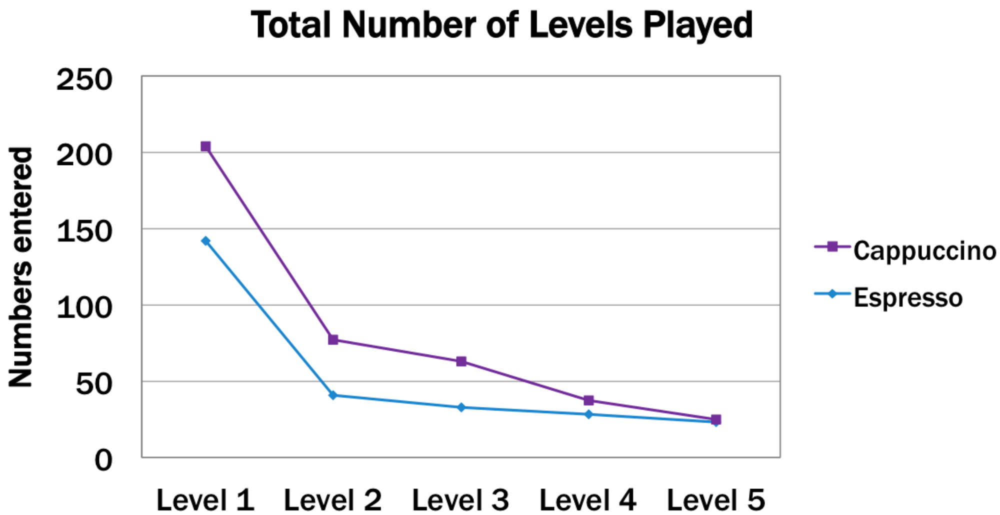

Figure 6.9: Amount of numbers entered in each level.

7 Discussion

One interesting finding is that most of the players enter the numbers more accurately using the Cappuccino method, despite being less intuitive than the Espresso method. Furthermore, P1, P2 and P5 played more DigiTaps games using the Cappuccino method. Since the Cappuccino method requires 1.8 taps per digit on average, players transitioning from the Espresso method to the Cappuccino method is expected. However, P3 briefly tried the Cappuccino method, but eventually switched back to playing the Espresso method and P6 did not enter any number using the Cappuccino method. The accuracy of the two methods do not differ greatly and the Cappuccino method is marginally more accurate than the Espresso method.

In terms of speed, we get a mixed result which we cannot justify that Cappuccino is evidently faster than Espresso or Espresso is faster than Cappuccino. P3 entered numbers using the Espresso method almost two times faster than using the Cappuccino method. On contrary, P5 can enter the numbers using the Cappuccino method 0.1 digits per second faster than using the Espresso method. There is not a clear that Espresso is better than Cappuccino and vice-versa.

The levels in the DigiTaps game are not played equally. Most the players played the first few levels more often than the later levels because the numbers presented in the early levels has fewer number of digits. Numbers in level 1 consists of only 3 digits whereas numbers in level 5 consists of 7 digits. As the player progresses through the levels, DigiTaps game becomes number memorization game instead of a number entry game.

8 Conclusion and Future Work

We introduced DigiTaps code, a prefix-free code that uses minimal voice feedback. The DigiTaps codes require at most 2.1 taps per digit on average and uses only primitive gestures in its codes such as swipe and tap. In prior work, DigiTaps shows promising results in the preliminary evaluation with the entry rate of 1.99 seconds per digit on average ($SD = 1.25$) using the Espresso method whereas using VoiceOver achieves only 2.77 seconds per digit on average ($SD = 1.24$) [14]. A lab study also showed that the participants can enter the numbers fast and accurate using the DigiTaps codes.

To evaluate the DigiTaps gestures in a real-world context, we developed DigiTaps game as a platform for conducting user study in the wild. DigiTaps is a number entry game. It gives a number to the player and the player uses one of the DigiTaps gestures to enter the number. DigiTaps is distributed through the Apple App Store. We collected information on how the gestures were performed and other metadata such as the number given to the player in the game. At the time of writing this paper, there are 654 players registered and 129,906 events collected. Using a subset of the data collected through the DigiTaps game, we did a preliminary evaluation of the two gestures.

In terms of accuracy, the players achieved a high accuracy rate, they entered the numbers more than 80% correctly on average with both DigiTaps codes and each player achieved similar number entering speed. Furthermore, there is no clear evidence on whether which of the code is better. Some players can enter numbers faster using the Cappuccino method while some players can enter numbers faster using the Espresso method. Even though we cannot justify which of the method is better, both DigiTaps codes are still great alternatives for an eyes-free number entering method.

Even though DigiTaps code and the DigiTaps game has been developed and studied to some extent, there are several aspects of the project that can be improved.

1. **Get recurring player:**

   Even though some results were presented, the number of participants is still low. If we can recruit more players to play the DigiTaps game on a regular basis, we can gather more data and get more useful data from the players. Making the game more engaging is one of the possible ways to get players to play the game more frequently. In addition, modifying the tutorial screens to be more interactive will also help new players to get started with the gestures and the game better.

2. **Implement DigiTaps codes in a real-world application:**

   DigiTaps code shows potential to be fast and works well in noisy settings. We can incorporate DigiTaps code to any number entering application such as a calculator, personal identification number (PIN) entry and making phone calls.

9 Acknowledgments

I would like to thank Professor Richard Ladner, Shiri Azenkot for their invaluable advise during my involvement in DigiTaps. I also would like to thank everyone in the MobileAccessibility research group for the guidance and help. I also thank the Mary Gates Research Endowment for the Mary Gates Research Scholarship.

References

[1] Apple Inc. iPhone Accessibility. http://www.apple.com/accessibility/iphone/vision.html. Accessed on June 10, 2014.

[2] S. Azenkot, C. L. Bennett, and R. E. Ladner. Digitaps: Eyes-free number entry on touchscreens with minimal audio feedback. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, pages 85–90, New York, NY, USA, 2013. ACM.

[3] S. Azenkot, S. Prasain, A. Borning, E. Fortuna, R. E. Ladner, and J. O. Wobbrock. Enhancing independence and safety for blind and deaf-blind public transit riders. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 3247–3256, New York, NY, USA, 2011. ACM.

[4] S. Azenkot, J. O. Wobbrock, S. Prasain, and R. E. Ladner. Input finger detection for nonvisual touch screen text entry in perkinput. In *Proceedings of Graphics Interface 2012*, GI '12, pages 121–129, Toronto, Ont., Canada, Canada, 2012. Canadian Information Processing Society.

[5] D. Goldberg and C. Richardson. Touch-typing with a stylus. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, CHI '93, pages 80–87, New York, NY, USA, 1993. ACM.

[6] Google. Android Accessibility. http://developer.android.com/design/patterns/accessibility.html. Accessed on June 10, 2014.

[7] N. Henze, E. Rukzio, and S. Boll. 100,000,000 taps: Analysis and improvement of touch performance in the large. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '11, pages 133–142, New York, NY, USA, 2011. ACM.

[8] N. Henze, E. Rukzio, and S. Boll. Observational and experimental investigation of typing behaviour using virtual keyboards for mobile devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2659–2668, New York, NY, USA, 2012. ACM.

[9] D. A. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the Institute of Radio Engineers*, 40(9):1098–1101, September 1952.

[10] S. K. Kane, J. P. Bigham, and J. O. Wobbrock. Slide rule: Making mobile touch screens accessible to blind people using multi-touch interaction techniques. In *Proceedings of the 10th International ACM SIGACCESS Conference on Computers and Accessibility*, Assets '08, pages 73–80, New York, NY, USA, 2008. ACM.

[11] S. K. Kane, J. O. Wobbrock, and R. E. Ladner. Usable gestures for blind people: Understanding preference and performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 413–422, New York, NY, USA, 2011. ACM.

[12] I. S. MacKenzie, R. W. Soukoreff, and J. Helga. 1 thumb, 4 buttons, 20 words per minute: Design and evaluation of h4-writer. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 471–480, New York, NY, USA, 2011. ACM.

[13] J. a. Oliveira, T. Guerreiro, H. Nicolau, J. Jorge, and D. Gonçalves. Blind people and mobile touch-based text-entry: Acknowledging the need for different flavors. In *The Proceedings of the 13th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '11, pages 179–186, New York, NY, USA, 2011. ACM.

[14] V. Ruamviboonsuk, S. Azenkot, and R. E. Ladner. Tapulator: A non-visual calculator using natural prefix-free codes. In *Proceedings of the 14th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '12, pages 221–222, New York, NY, USA, 2012. ACM.

[15] C. Southern, J. Clawson, B. Frey, G. Abowd, and M. Romero. An evaluation of brailletouch: Mobile touchscreen text entry for the visually impaired. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services*, MobileHCI '12, pages 317–326, New York, NY, USA, 2012. ACM.