

UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA



Understanding Gesture Demands of Touchscreen Games to Accommodate Unconventional Gamers

Anabela Araújo Rodrigues

Mestrado em Informática

Dissertação orientada por:
Prof. Doutor Tiago João Vieira Guerreiro
Dr. Kyle Montague

2016

Acknowledgements

I would like to thank my thesis adviser Professor Tiago Guerreiro for his full support and expert guidance throughout my study and research, as well as for his positive response and great support when I approached him a year ago about the possibility of an Erasmus+ internship; if not for his help I would not have lived as a researcher abroad and had had such a unique experience. I would like to thank my coadviser Kyle Montague, who has been a great sport about unexpectedly getting someone under his tutelage and has helped me stay on track while also being an infinite source of ideas. I would also like to thank André Rodrigues for always being available to help and acknowledge him for also being a big part of this project. To my study participants, thank you all so much for your time and for your contribution to this project.

To the amazing people at Newcastle University's Open Lab, thank you for taking me in with open arms and accepting me as one of your own in so little time. A lot can happen in 4 months and I carry fond memories of my time there and of the people I met. There are too many to list one by one, but I'll miss the discussions while we got our caffeine in the 4th floor kitchen, the banter in our 'corner' of the lab, and the great diversity of people that I got to meet inside and out of the lab. I've grown and learned so much with you all.

To the Navigators in LaSIGE, thank you for the laughs and for suffering with me while we all finished writing our Masters dissertation together. To Desmarques, thanks for getting fat with me when we pulled all-nighters at the University, writing our dissertation while eating donuts. To Pina, despite having left us to learn to fly planes, thanks for being a big part of my undergrad years and still nagging me today about dumb things. To everyone I've met during my time at the University, thank you for the great times and still appearing for Friday drinks despite us all getting old and wrinkly.

To my boyfriend, Cristiano, thank you for your constant support and encouragement over these almost-five years, and for dragging me out of the house every once in a while when I was being too much of a nerd.

To my parents, Carlos and Fernanda, thank you for always leading by example and teaching me to work hard to accomplish my goals. Thank you for your love and support, as well as for the calls every few days asking if I've finished my dissertation

yet. To my sisters, Lisette and Rita, thank you for being the funniest, most annoying and incredible sisters anyone could ask for.

Resumo

Os dispositivos móveis tornaram-se uma parte importante das nossas interações diárias, e são usados para permanecer ligado com a família, os amigos e os colegas de trabalho. De acordo com um estudo realizado pela *GSM Arena* [3], os principais motivos que levam pessoas a comprar telemóveis são para navegar na internet em qualquer lugar e em qualquer momento, usar redes sociais, desfrutar de jogos móveis, e para continuar a trabalhar depois de sair do escritório. Com a crescente popularidade de aplicações e jogos para dispositivos móveis como o *Facebook*, *WhatsApp*, *Candy Crush* e *Angry Birds*, existe um interesse global em fazer parte deste fenómeno social. Portanto, há uma necessidade de tornar esta tecnologia acessível a todos, tanto para aumentar lucros como responder à procura. Isto inclui tornar os telemóveis acessíveis a pessoas com vários tipos de deficiência: motora, visual, auditiva e / ou cognitiva. Nesta dissertação, vamos focar em deficiências motoras.

Atualmente, as ferramentas de acessibilidade móveis mais utilizados são ferramentas ao nível de sistema, tais como a capacidade de ativar e desativar o *Auto Rotate*, o ajustamento da sensibilidade de atraso de toque e a disponibilidade de vários teclados, como o teclado de toque e teclado de reconhecimento de voz, por exemplo, proporcionando ao usuário mais opções de entrada de dados [15]. Estas ferramentas apoiam utilizadores com dificuldades motoras, todavia, continuam bastante limitados no que podem fazer com dispositivos móveis.

Existe um setor em particular que tem sido demasiado complexo para endereçar: jogos móveis. De acordo com a *Big Fish Games* [45], 59% dos norte-americanos jogaram videojogos em 2015. Esta vasta percentagem de jogadores demonstra a atual importância dos jogos na nossa sociedade.

Os jogos enriquecem as nossas mentes e desenvolvem a nossa capacidade de resolução de problemas complexos, a nossa criatividade, a nossa coordenação óculo-manual e exercitam habilidades como foco, velocidade e pensamento flexível [4]. Eles são um dos meios mais eficazes de ensino, bem como uma fonte de estímulo mental e emocional. O principal conceito dos jogos, que é superar obstáculos para alcançar um objetivo, ensina lições importantes e faz entender o valor de trabalhar arduamente para superar desafios. Um dos aspetos mais importantes dos jogos é a interação social;

permitem-nos socializar com outras pessoas, seja jogando com elas virtualmente ou no mesmo espaço físico. Permitem conhecer pessoas novas e fazer amigos, e fazem os seus jogadores sentir-se parte de uma comunidade. Estes benefícios não devem ser negadas a ninguém, o que torna a inclusão de todos de uma extrema importância.

A acessibilidade de jogos móveis tem sido abordada de várias maneiras. No entanto, a maioria das soluções existentes são aplicadas diretamente no jogo, na fase de desenvolvimento. No entanto, a maioria dos desenvolvedores de jogos não levam a acessibilidade em conta, e vêem-no como um desperdício de dinheiro e recursos. Devido a isso, é difícil para pessoas com dificuldades motoras encontrar jogos que os incluam, apesar do facto de que eles constituem uma grande parte da população de jogadores. Em geral, eles jogam mais e por períodos mais longos de tempo. O investimento na criação de jogos acessíveis é pequeno em comparação com o grande aumento de lucros em que resultaria [6].

As abordagens para acessibilidade de jogos móveis são escassas, e os métodos existentes raramente são implementados. Existe ainda um grande espaço entre o que existe e o que é necessário. Este é um problema que deve ser resolvido de uma vez, devido ao facto de que milhões de pessoas estão a ser excluídos das comunidades de jogos e dos benefícios que eles proporcionam.

Para resolver este problema, primeiro precisamos de ter uma visão clara dos requisitos de *input* dos jogos atuais e dos problemas que estes acarretam para pessoas com dificuldades motoras. Como os desenvolvedores de jogos têm liberdade para definir os seus próprios reconhecedores de gestos, existem inúmeras possibilidades de condições de jogo, o que leva à necessidade de uma análise aprofundada da questão.

Para alcançar este objetivo, colecionamos dados de jogo dos 25 melhores jogos do Google Play, e analisamos os seus requisitos de *input*. Com esta análise, criamos um catálogo de gestos, que lista os gestos mais usados nestes jogos e parametriza os gestos de jogo, proporcionando-nos com detalhes como duração, velocidade e distância percorrida dos gestos. Caracterizamos e relacionamos os jogos de acordo com os seus gestos de jogo predominantes e com os requisitos de *input*, dando-nos assim uma visão geral dos jogos de hoje.

No segundo estudo, analisamos a viabilidade destes gestos para pessoas de diferentes capacidades motoras. Realizamos este estudo com participantes sem dificuldades motoras, crianças, idosos e pessoas com deficiências motoras. Com isto, foi possível concluir quais os gestos que proporcionaram as maiores dificuldades, bem como recolher parâmetros de desempenho detalhados dos diferentes grupos.

De seguida, comparamos os requisitos dos jogos com o desempenho de gestos de cada grupo, de modo a determinar quais os jogos jogáveis para pessoas de diferentes graus de mobilidade, e de modo a descobrir em detalhe quais os requisitos de gesto que eram demasiado altos, e porquê.

Estes resultados permitiram-nos criar uma proposta para uma solução a acessibilidade impulsionado pela doação de dados por pessoas, na qual jogadores experientes doam os seus dados de jogo a uma comunidade *online*. Estes dados são analisados manualmente e são anotadas os gestos de jogo mais importantes. O sistema, em seguida, usa os *sets* de dados resultantes para adaptar o *input* de jogadores com dificuldades motoras, de acordo com um *user model* particular de cada pessoa, proporcionando assim uma adaptação de *input* personalizado. Assim como a adaptação de *input*, o sistema também proporciona adaptação de interface de jogo.

Como prova de conceito, implementamos um protótipo da ferramenta de anotação de gestos, o que nos permite gravar uma sessão de jogo e anotar os gestos mais importantes da sessão, criando assim um *set* de dados para o jogo.

Esta dissertação tem como principais contribuições:

- Conjunto abrangente de heurísticas da literatura, que avalia usabilidade, mobilidade, jogabilidade e interação dos jogos. Estas heurísticas foram utilizados para avaliar os jogos escolhidos para o primeiro estudo, no qual foram analisados os requisitos de *input* de jogos atuais. As três primeiras heurísticas foram retirados das heurísticas de jogabilidade dos jogos de Ponnada e Kannan [37]. As últimas heurísticas, que avaliam a interação do jogo, foram adicionados por nós para cobrir aspetos relacionados com gestos e *input* de jogos.
- Catálogo de Requisitos de jogo, com base nos resultados do nosso primeiro estudo. Este estudo analisou os dados de jogo dos 25 melhores jogos da loja do Google Play, jogadas por 25 participantes sem dificuldades motoras. Foram identificados os gestos mais usados em jogos atuais, e listamos os requisitos de *input* detalhados de cada jogo.
- Análise das capacidades de toque de participantes de diferentes graus de mobilidade, baseada no nosso primeiro estudo, e comparando o desempenho de gestos de jogadores sem dificuldades motoras com jogadores de vários graus de mobilidade: crianças, idosos e pessoas com deficiências motoras.
- Um conjunto abrangente de implicações de desenho para a criação de jogos acessíveis para jogadores com dificuldades motoras, mostrando como os desenvolvedores de jogos podem incluir mais jogadores no seu jogo. Em

suma, estas implicações incluem evitar anúncios *pop-up* em jogos, não assumir que todos os jogadores têm as mesmas capacidades, permitir a personalização de *input*, oferecer flexibilidade de velocidade de jogo, tornar a área de rabisco menor, e oferecer alternativas para gestos de múltiplo toque.

- Uma proposta de solução de acessibilidade, que é criado com base em nossos resultados de estudos e pesquisa sobre temas relacionados. Esta solução é construída sobre a adaptação de jogos impulsionada pela doação de dados de jogo por pessoas, e explora a anotação de gestos de jogo e a adaptação de *input* e de interfaces de jogos.

Com este trabalho, demonstramos os requisitos de *input* dos jogos atuais, as capacidades de input de pessoas de vários graus de mobilidade, avaliamos a jogabilidade de jogos atuais por pessoas de vários graus de mobilidade, e propusemos uma solução de acessibilidade, criando um protótipo como prova de conceito.

Palavras-chave: dispositivos móveis, acessibilidade, ecrã de toque, deficiências motoras, jogos

Abstract

Games enrich our minds and develop skills such as problem solving, creativity, focus and hand-eye coordination. They are one of the most effective teaching means, as well as a source of mental and emotional stimulus. One of the most important aspects of games is social interaction: they allow us to engage with each other socially, either by playing with others virtually or in the same room. They help us relate with others and feel like part of a community. These benefits should not be denied to anyone, making the inclusion of all of utmost importance. However, currently, players with less motor dexterity are not considered in the design process of mobile games, excluding them by being fast paced and requiring high touch precision and multi touch gestures. Mobile game accessibility approaches are still scarce, and the existing methods are rarely implemented into games; developers are not aware of the importance of accessibility, or do not know how to implement it. To understand the complete scope of this issue, we explore current games and their input demands, as well as the input abilities of unconventional gamers. In our first study, we create a catalogue of the most commonly used gestures and the specific demands of each game. We then use these results to perform a second study, in which we analyse gesture performance of people of varying abilities. Finally, we compare the results of both studies to determine the accessibility of current mobile games. We conclude that 48% of our game sample is not playable for motor impaired players. As a result of our research, we provide design implications and propose a human-powered, system-wide accessibility solution, which depends on crowdsourced gameplay data to adapt games to individual needs. As a proof of concept, we implement a prototype of a gameplay annotation tool.

Keywords: mobile, accessibility, touchscreen, motor impaired, games

Contents

Chapter 1 Introduction	1
1.1 Motivation	1
1.2 Building a catalogue for accessible gaming	2
1.3 Contributions	4
1.4 Dissertation Roadmap	4
Chapter 2 Related Work.....	6
2.1 Accessible Gaming	6
2.2 Input Challenges and Adaptation	13
2.3 Discussion.....	20
Chapter 3 Catalogue of Input Demands of Touchscreen Games	22
3.1 Data Collection	22
3.1.1 Participants	22
3.1.2 Apparatus	23
Hardware Technology.....	23
Game Selection Criteria	23
3.1.3 Tools.....	27
3.1.4 Procedure.....	29
3.1.5 Design and Analysis.....	29
3.2 Results	30
3.2.1 Taps and Long Presses	30
3.2.2 Swipes	32
3.2.3 Drags: Regular, Scribbling, Rotation and Shapes	32
3.2.4 Pinch and Spread.....	35
3.3 Discussion.....	35
3.4 Summary.....	36
Chapter 4 Understanding the Abilities of Unconventional Gamers	37
4.1 Background.....	37

4.2	Data Collection	38
4.2.1	Participants	38
4.2.2	Apparatus	39
4.2.3	Gesture Prompt Application.....	39
4.2.4	Procedure.....	41
4.2.5	Design & Analysis	42
4.3	Results	43
4.3.1	Measuring user abilities	43
4.3.2	Comparing abilities and demands	50
4.4	Discussion.....	53
4.5	Summary.....	53
Chapter 5 Game Design Implications		54
5.1	Context	54
5.2	Implications for Accessible Gaming	55
5.3	Human-powered Adaptation of Games	56
5.4	Annotation Tool Prototype	57
Chapter 6 Conclusions and Future Work.....		59
6.1	Limitations.....	60
6.2	Future Work.....	60
Bibliography		61
Appendix.....		69
Appendix A – Game Input Demands Catalogue.....		70
Appendix B – First Study Script.....		83
Appendix C – Second Study Script		84
Appendix D – Newcastle University Ethics Form.....		85

List of Figures

Figure 1 Extended TBB Data Model	28
Figure 2 Median Tap Intervals chart.....	31
Figure 3 Canvas drawing of a scribble performed in Talking Tom.....	33
Figure 4 Canvas drawing of one finger rotation performed in 8 Ball Pool	34
Figure 5 Various shapes performed in Words Crush.....	34
Figure 6 Gesture Prompt shapes	40
Figure 7 Taps outside small target chart	44
Figure 8 Durations of every gesture performed by every group.....	44
Figure 9 Travelled Distance of every gesture performed by every group	45
Figure 10 Speed of every gesture performed by every group.....	46
Figure 12 Travelled Distance vs Path Size of rotate.....	47
Figure 11 Travelled Distance vs Path Size of swipe, drag and shape.....	47
Figure 13 Scribble area covered.....	48
Figure 14 Pinch diagonal onDOWN and onUP.....	48
Figure 15 Spread diagonal onDOWN and onUP.....	49
Figure 16 Spread performed by motor impaired participant	49
Figure 17 Sample boxplot comparing game demands and the four ability groups	50
Figure 18 Crossy Road boxplot comparing tap duration game demands with tap durations of ability groups.....	51
Figure 19 Marvel boxplot comparing swipe duration game demands with swipe durations of ability groups.....	52
Figure 20 Initial Annotation Tool interface designs	57
Figure 21 Annotation Tool Prototype screencap	58

List of Tables

Table 1 Game Usability Heuristics	23
Table 2 Mobility Heuristics	24
Table 3 Gameplay Heuristics	24
Table 4 Game Input Requirements	25
Table 5 Game Usability Results	26
Table 6 Mobility Results.....	26
Table 7 Gameplay results.....	27
Table 8 Game Input Requirements results.....	27
Table 9 Shape frequency.....	35

Chapter 1

Introduction

1.1 Motivation

Mobile phones have become ubiquitous in our society. The mobile phone market is in constant growth: Google's Android phones reached an annual revenue of \$74.5 billion in 2015 [1] and, according to the International Data Corporation [2], Android leads the OS market with an 82% share of sales in 2015, followed by iOS with 14%.

Smartphones have become an important part of our daily interactions and are used to remain connected with family, friends and co-workers. According to a survey by GSM Arena [3], the main reasons people buy smartphones are to browse the internet anywhere and at any time, use social networks, enjoy mobile games, and continue working after leaving the office. Smartphones are also the most economical and agile way to access games. With the rising popularity of smartphone applications and games such as Facebook, WhatsApp, Candy Crush and Angry Birds, there is a global interest in being a part of this social phenomenon. Therefore, there is a need to make this technology accessible to everyone, both to increase profits and to respond to public demand. This includes making mobile phones accessible to people with variable abilities: motor, visual, hearing and/or cognitive. In this dissertation, we focus on motor impairments.

In particular, mobile game accessibility is an area that has been too complex to fully address. According to Big Fish Games [45], 59% of Americans played games in 2015. This vast percentage of gamers is telling on how important games are in our current society. Games develop our problem-solving skills, creativity, hand-eye coordination, and exercise skills like focus, speed and flexible thinking [4]. They are one of the most effective teaching means, as well as being a source of mental and emotional stimulus. The main concept of games, which is overcoming obstacles to achieve a goal, teaches us valuable lessons and gives us the understanding of working hard to overcome challenges. One of the most important aspects of games is the social interaction that comes along with it; they allow us to engage with each other socially,

either by playing with others virtually or by playing in the same room. They allow us to make friends, relate with others and feel like a part of a community. These numerous benefits should not be denied to anyone, making the inclusion of all of utmost importance.

Mobile game accessibility has been approached in various ways. For example, the NOMON interaction modality [7] is a switch style interaction which associates a clock face to each selectable element. Each of these clocks have a clock hand that is constantly rotating; when its passing noon, the element becomes selectable by the switch. Once the user presses the switch, NOMON calculates which element was selected based on all of the clock hand positions. Other approaches include supporting various input methods, allowing interaction or interface customisation, providing auto aim, and using user models to adapt games. System-level accessibility exists as well, such as the ability to activate/deactivate Auto Rotate, setting the touch delay sensitivity, and the availability of multiple keyboards, such as touch keyboard and voice recognition keyboard, providing the user with more input options[15]. These tools aid motor impaired users greatly, however, they are still very limited in what they can do with mobile phones [12].

However, most of these solutions need to be implemented directly into the game, and when they are, only a few adaptation approaches are usually chosen to be implemented. Most game developers do not take accessibility into account at all, and see it as a waste of money and resources. Due to this, it is difficult for motor impaired people to find games that will include them, despite the fact that they constitute a large population and play games more often and for longer periods of time. The investment in making games accessible is small compared to the great increase of game sells it ensues [6].

Mobile game accessibility approaches are still scarce, and the existing methods are rarely implemented into games. There is still a large gap between what exists and what is needed. This is a problem that needs to be addressed at once due to the fact that millions are being excluded from gaming communities and the benefits that they provide.

1.2 Building a catalogue for accessible gaming

To address the problem described above, we first need to have a clear view of the demands of current games and the issues that these entail for motor impaired people. As game developers have freedom in defining their own gesture recognizers, there are endless possibilities in game gesture demands, which leads to the need for an in-depth analysis of these demands.

To achieve this, we collected data from the top 25 Google Play games and analysed their input requirements. From this analysis, we created a gesture catalogue, which tells us which are the most used gestures in these games and parametrises the gameplay gestures, providing us with details such as duration, speed and travelled distance of the gestures. We characterized and related the games according to their predominant gameplay gestures and gesture demands, giving us a general overview of today's games.

The second study we conducted analysed the feasibility of these gestures to people of varying abilities. We conducted this study with able bodied participants, children, elders and motor impaired participants. From this we were able to conclude which gestures provided the most difficulties, as well as collect detailed gesture performance parameters of the different groups. We then compared the game gesture demands to the gesture performance of each ability group so as to determine which games were playable to people with different degrees of mobility, and discover in detail which gameplay gesture demands were too high for them and why.

The largest issues we found were that multi touch gestures were unfeasible for motor impaired players, and that motor impaired participant's gesture duration was generally larger than the other ability groups, consequently making them play at a slower pace, making faster-paced games unplayable for them. These two aspects were the main factors of game exclusion of motor impaired people in our comparison of game gesture demands with each ability group's performance.

As was mentioned in 1.1, the existing accessibility methods are rarely implemented into games, either due to game developers seeing accessibility as a waste of money and resources or to them not knowing how to implement it. The results from our studies and research allowed us to create a proposal for a human-powered system-wide accessibility solution, in which able bodied expert players donate their gameplay data to an online community. This gameplay data is manually analysed and important gameplay gestures are annotated. The system then transforms the resulting data sets into input adaptation shortcuts for motor impaired users, according to each user's particular user model, thus providing personalised input adaptation.

As a proof of concept, we implemented a prototype of the Annotation Tool, which allows us to record a gameplay session and annotate the most important gestures of that session, creating a data set for the game.

1.3 Contributions

This dissertation studies touchscreen game demands and how these create barriers to players with different abilities. The results from our research provided the following contributions:

- *Comprehensive set of Heuristics from Literature and Game Input Requirements*, which evaluates game usability, mobility, playability and interaction. These heuristics were used to evaluate the games chosen for the first study, in which we analysed game gesture demands. The first three heuristics were taken from Ponnada and Kannan's playability heuristics [37]. As well as these heuristics, we created a list to classify game input requisites to cover input-related aspects of games.
- *Game Demands Catalogue*, based on the results of our first study. This study analysed gameplay of the top 25 games from the Google Play store, by 25 able bodied participants. We identified the most used gestures in current games, and listed each of the games' specific gesture demands in this catalogue.
- *Analysis of touch capabilities of participants of varying abilities*, based on our second study, in which we contrasted the gesture performance of able bodied gamers with unconventional gamers: motor impaired, children and elders. These results show their main difficulties and differences, using able bodied performance as a baseline.
- *A comprehensive set of implications for the design of games accessible to motor impaired players*, showing how game designers can include more players into their game. In sum, they include avoiding pop up advertisements in games, not assuming that every player is at same baseline, allowing input customization, offering flexible game speed, making scribble area thresholds smaller, and providing alternatives to multi touch gestures.
- *Accessibility solution proposal*, which is created based on our study results and research into related topics. This solution is built on human-powered adaptation of games, and explores input and interface adaptation of games using manual game classification and annotation.

1.4 Dissertation Roadmap

At the beginning of the dissertation, we set out to design and develop an accessibility framework, intending to adapt games to everyone's abilities. However,

after a few months of research on the topic, in which we analysed mobile game accessibility, input adaptation and interface adaptation, we found that work related to the topic was scarce. We did not know the full scope of the problem and we did not have the bases to create this framework. We decided that, to be able to build an accessibility tool, we first needed to explore the issue in depth.

Our first step in this direction was to run a study in which we would collect gameplay data of touchscreen games, with the intent of defining current gesture demands in games. The games analysed were selected from Google Play's top games (as of 1st March 2016), so as to have the most up-to-date sample. We ran the study with 25 games and recruited 25 able bodied participants from Newcastle University's Open Lab, due to the researcher running the study being there on an Erasmus+ internship.

After cataloguing the game gesture demands from the results of this study, we realised that, to fully understand what we needed to adapt and how to make these adaptations, we would need to collect gesture data from unconventional players – players with less dexterity or different motor abilities - and contrast their performance with the expected game performance.

We ran this study with 2 motor impaired participants in Dundee University, and 4 able bodied, 4 elders and 4 children in Portugal, due to the researcher running the study ending her Erasmus+ internship. We analysed the results from this study and compared them with the game gesture demands catalogued in the first study. We discovered which gestures provided the most difficulties for players with different abilities and learned which games excluded players based on their motor abilities.

With the full scope of the problem, we were now able to propose a sensible solution. As a proof of concept, we designed and implemented a first prototype which explores one of the aspects of our proposal – gesture annotation from expert gameplay.

Chapter 2

Related Work

In terms of related work, we identified two main sections: accessible gaming and input challenges and adaptation. We will detail these sections by introducing their current state, and by discussing the existing issues associated to each. We will describe frameworks associated to these sections, using them as examples for possible solutions to these issues.

2.1 Accessible Gaming

We will begin by introducing the benefits of increasing game accessibility, followed by explaining the different methods that people with motor difficulties use to adapt to smartphones. We then analyse present mobile game accessibility and the various approaches toward it, providing examples of frameworks using these approaches. Finally, we identify and understand the current issues of gaming, and present solutions and guidelines that can be followed to make games more accessible.

Garber [6] explains the various benefits of increasing game accessibility: firstly, it is good to include this demographic in the social phenomenon of gaming communities. Games provide physical and mental health benefits as well, such as stress relief and improvements in manual dexterity. For the gaming industry, there are financial benefits: disabled players constitute a large population and they play games more often and for longer periods of time; the investment in making games accessible is small compared to the great increase of game sells. There is a need to adapt these games so that everyone can play, regardless of age or disability.

People currently use and adapt their smartphones for interaction in various ways. Kane et al [5] conducted a study on how motor-impaired people adapt and use smartphones in their daily lives, conducted via interviews and a diary study. The most common adaptive strategies found in this study were device modification (both via the device settings and hardware modification) and the installation of accessibility software.

Also, simply holding their phones in an unconventional way helped the users to use their phones successfully.

Android currently offers a large variety of system-level accessibility software to aid users in adapting to their smartphones. Rahman [15], from Bridging Apps, describes the current Android features targeting physical and fine motor abilities. These include the ability to activate/deactivate Auto Rotate and set the touch delay sensitivity, which tells the OS how long between touch down and touch up is considered a touch or a hold. The availability of multiple keyboards, such as touch keyboard, handwriting recognition keyboard, voice recognition keyboard, etc. gives the user more input options as well.

Despite not having a standard OS level switch interface, Android developers have created various applications which allow the use of switch technology. But to benefit from this, the applications being used also have to be switch enabled.

Currently, most mobile games do not take accessibility into account. Kim et al [17] analysed the most popular iPad games in terms of their accessibility. The most used gestures in these games were short tap, drag, and swipe. 24% of the games required multi touch and the top game genres were simulation, puzzle, action, strategy, and adventure. The first two are slow-paced, which is better for accessibility while, for example, action games are mostly fast-paced, which is very challenging for motor-impaired users. 47% of the games had no speed requirements, 29% were fast-paced, and 24% had minimal speed requirements. This shows that slowing down gameplay could make most games playable for motor-impaired users, with the exception of fast-paced multiplayer, unless lag is introduced for all parties. 24% of the games allowed customization, and 50% required complex gestures.

Current approaches to game accessibility mostly involve creating games which target one or more categories of disabilities: visual, hearing, cognitive or motor. Gnomon [7] is a one-switch framework that uses NOMON interaction, which associates a clock face to each selectable element. Each of these clocks have a clock hand that is constantly rotating; when its passing noon, the element becomes selectable by the switch. Once the user presses the switch, NOMON calculates which element was selected based on all of the clock hand positions. Two games were developed with this framework: “One-Switch Lady Bugs”, which allows the user to select differently coloured ladybugs on the screen which emit unique sounds, and “One-Switch Invaders”, which allows the user to select the dynamically moving elements before three of them hit the ground. The first game has no score or time constraints, as it was merely designed to explain NOMON’s functionality, while in the second game the user scores points when they kill the randomly generated aliens before they hit the ground.

VBGhost [8] is another accessible game, designed for visually-impaired players as an adaptation of “Ghost”, a word game in which each player adds a letter to an existing word fragment, without completing a word or writing an invalid fragment.

This game offers a high contrast menu and reads options and word fragments aloud on touch, choosing them on double tap. To add letters to the fragment, the game offers the player a VBraille interface, which presents a blank braille character to the user. When the user clicks on the braille dots, its respective number is read aloud (one to six), and the dot becomes black on double tap. This way, the user can “write” the braille letter they want to play.

These games limit who can play, also making developers cultivate a preference for addressing certain disability sectors. Bierre et al [13] analysed a random sample of accessible games, of which most focused more on users with visual impairments than any other type of disability. Motor impairment was the sector that was the least addressed by these games.

Although this approach to game accessibility achieves the goal of allowing players with disabilities to play, there is the aspect of social exclusion to consider. Rather than include these players in the existing gaming community, a stigma is inadvertently created with these games, as they’re designed specifically for disabled players, isolating them from being a part of the social phenomenon that is mass multiplayer online (MMO) communities.

There have been efforts to improve this, instead adding accessibility features to existing popular games that were originally designed for able-bodied users, making them accessible to everyone. Universally Accessible Games are games that adapt to the needs of the broadest user population possible. They target various disability groups simultaneously, allowing the use of assistive devices and altering the user interface to tailor to each individual’s needs.

Grammenos et al [9] designed a universally accessible version of Space Invaders, called Access Invaders. Access Invaders supports various input methods, is highly customizable, uses profiles to adapt the game to each person, and supports non-visual gameplay as well.

The main method of adaptation is the use of user profiles: in this project, seven different profiles are offered to the players, which make the game seem like a collection of games. These profiles can adapt the speed of gameplay, the visual complexity, quantity, position and size of game elements, the speed and strength of enemy firepower, the contrast and the sound. Sound can become 3D to allow spatial feedback to visually-impaired players.

Access Invaders allows people with diverse disabilities to play cooperatively, but this was hard to accomplish as players with different abilities perceive the gameplay and game content differently due to the previous adaptations. Here we define game universe as each game instance with adapted gameplay. The solution offered by the Gramennos et al is, despite playing in different game universes, to find a way to reflect the state of the universes on each other.

Trewin et al [10] made an existing 3D multi-player virtual world, Power Up, accessible. They approached this in various ways, starting first by conducting a survey to understand how people with disabilities play virtual world games. The results of these surveys revealed that nearly all the surveyed players wanted to participate in online multiplayer virtual worlds, despite them lacking the necessary accessibility features. The main accessibility features that were added to the game's HUD were the use of contrast, allowing the user to customize GUI font, and keyboard and mouse interaction with speech and visual feedback. The 3D virtual world accessibility features include the ability to zoom in, the use of sound effects to translate visuals into audio, enabling continuous movement of the avatar without the need to sustain mouse button presses, and the use of captions and images to translate sounds into visuals. They also include some useful gameplay functionalities such as the ability to teleport the avatar within the virtual world. The 'find' command allows the users to scan the current view for objects that they can lock on. The 'controlled walk' function allows the user to lock onto a target and make the avatar automatically walk toward it. Once it reaches the target or an obstacle blocks the way, the avatar automatically stops walking. The audio-only version of Power Up adds speech to these functionalities; for 'find', the system reads aloud the name, distance and relative position of the object. For 'controlled walk', the system warns the user once the avatar reaches the target and when the avatar encounters obstacles. There is an added 'look' command which describes the virtual world scene.

Carvalho et al [47] created an audio-based puzzle game for blind people which allows users to solve musical puzzles. Based on traditional cardboard puzzles, the main goal of the game is to reconstruct a fragmented song by putting each individual segment in the correct order. The game extends an existing puzzle game in which square pieces of images are randomly placed on the screen, and the user can drag and drop them into the correct position in order to solve the puzzle. This extension includes the removal of visual feedback, changing interaction into directional slides to move pieces and taps to require help, as well as audio feedback. By double tapping in the middle area, the user can listen to the current piece. By sliding left and right, the next piece is selected. By sliding up, the user can place the piece in the target strip. Different sounds and vibration patterns are associated with success and failure in placing a piece.

As mentioned earlier, there is a desire among disabled players to participate in online communities with others as equals. Yuan et al [11] emphasize on the importance of maintaining game fairness and challenge when adding accessibility to games. To do this, they suggest preserving the original gameplay as much as possible. Although it is important to maintain fairness, it is hard to accurately implement this; the game format for each disability needs to be so distinct that adaptation results in having various different games, as with Access Invaders [39].

To further understand how we can make games accessible, it is necessary to identify and understand the current issues in game accessibility so we can overcome them. Porter and Kientz [12] analyse the current state of accessibility in games, so as to identify and understand issues and barriers in both sides of the game accessibility industry: the gamers and the developers. They do this via a survey to online gamers with disabilities and interviews with game industry individuals. The quantitative results of the surveys indicate that mobile devices are in third place of the most-used gaming platform, following PCs and consoles; this is explained by the adaptation of specific input devices to these platforms, and shows that something is lacking in mobile phone accessibility. In addition, players with motor impairments were the large majority in the surveyed individuals, followed by visual, hearing and, finally, cognitive.

They found that the types of games people with motor impairments play are in this order: single player independent, MMO's, single player collaborative, and multiplayer in person. This suggests that multiplayer games push away gamers with disabilities, which is confirmed by them in the surveys, as they feel an inability to 'keep up' and compete, and have trouble communicating with other players. This shows a need for accessibility features that will allow them to be at the same level of able bodied users, and that will allow them to easily participate.

The main complaints of the surveyed gamers were that some games do not recognize input from assistive devices and software, and the need to ask for external help during gameplay diminished the gaming experience and brought them feelings of reluctance. Porter and Kientz highlight the importance of testing games with individuals with disabilities, so as to identify shortcomings of the current games.

On the developer's side, when accessibility is not a priority for the game, they only implement the simplest accessibility features (colour palette taking colour-blind users into account and captions for hearing-impaired users, for example). They explain the lack of accessibility in gaming with the fact that developers are mostly able-bodied individuals; certain impairments are so foreign a concept that they do not consider it as an important factor in game development. However, as more and more individuals with

disabilities are integrated into the development workforce, this is gradually changing. Developers need to be more sensibilised toward accessibility, and provided tools to make the integration of accessibility easier.

Bierre et al [13] stated that the main problem for disabled people when purchasing a game is that most games do not have any indication about their accessibility features. Because of this, choosing a playable game is an intimidating task.

Bierre et al identified common problems for disabled gamers when playing a new game. Cognitively-impaired players' gameplay is affected by complex storylines, the lack of tutorials and easy-to-understand documentation, no indication of dangerous situations, and by a lack of game speed adjustments. Hearing-impaired players' gameplay is affected by the game lacking subtitles, only providing vital clues to complete game tasks via audio without closed captions, and by only providing audio cues for danger or getting injured. Visually-impaired players' gameplay is affected by only providing clues to complete game tasks via text. Motor-impaired players' gameplay is affected by the game needing precise timing or the ability to be precise in positioning the cursor.

There are various solutions, guidelines and approaches we can follow to address these issues. Yuan et al [11] present their Game Interaction Model, which allows the identification of the parts of gameplay that each impaired individual has difficulties with. They divided gameplay into 3 steps: the reception of stimuli, determining a response, and providing input. According to them, gameplay is an infinite loop of these 3 steps until game completion. In this model, visually- and hearing-impaired players have difficulty with the first step, 'the reception of stimuli', cognitively-impaired players have difficulty with the second step, 'determining a response', and motor-impaired players have difficulty with the third step, 'providing input'.

With this mind set, we can think of solutions for game accessibility. Switches and their scanning mechanism are shown to be an extremely useful tool for motor-impaired users in general device interaction, but also have great limitations, especially in gameplay.

There are several strategies to make gameplay more accessible: reduction (eliminating some aspects of the game), automation (automate some of the more difficult parts of the game) and scanning (uses the switch's scanning mechanism).

Garber [6] provided some suggestions for developers to make games more accessible, the simplest being the use of subtitles, adjusting hues for colour-blind players, and allowing the customization of the text size, characters and game sensitivity. More time-intensive suggestions include: hardware support for assistive controllers and

input devices; implementation of a system that allows players to skip parts that are too difficult to complete; using artificial intelligence to further assist disabled gamers with tasks; automate difficult to provide input; and providing the users a very basic set of controls. He recommends working with specialists in the field and disabled users to improve the game's accessibility.

Barlet and Spohn, from the AbleGamers Foundation, wrote *Includification* [14], which is a set of guidelines for game developers to create accessible games. According to them, despite being unrealistic to include every single type of disability, the more accessibility options we add to a game, the more people are included. The set of guidelines about motor accessibility details three levels.

- 1) The first level targets the minimal features for a game to be accessible to motor-impaired people: remappable keys and alternative configurations.
- 2) The second level focuses on slightly more complex features. Compatibility with third party devices & assistive technology is important for users that require these devices. Allowing the user to move or resize individual elements of the HUD interface alleviates the strain for gamers with low stamina or dexterity by putting the elements in an easily reached location of the screen. Allowing the use of macros (a single button/command that activates a series of commands), helps level the playing field for motor-impaired players, but has been considered cheating by the gaming world when used by able-bodied gamers, and is therefore difficult to include in games. Fail-safes, or auto-pass, are good for when the user is stuck in a certain part of the game; by detecting that the player failed the task a few times and offering to 'skip this part' will diminish frustration in gameplay. Sensitivity sliders are important for both players who need small movements to be recognized by the game, and for players with tremor to ignore accidental input.
- 3) The third level is the most difficult to achieve, and the best for motor accessibility: compatibility with all input devices, and the ability to slow down the game clock.

Includification also offers some smartphone-specific accessibility guidelines: adding a buffer against accidentally touching the same spot more than once and including a 'hit box' which delineates the touchable area of the screen. For multi-touch problems, it suggests grouping together multiple touches into a single button (as mentioned before, the use of macros). It also suggests adding buttons to simulate accelerometer values.

2.2 Input Challenges and Adaptation

We will begin by analysing specific input-related challenges that disabled users face, and the main differences between input of able-bodied users and of users with less motor dexterity. Design recommendations are derived from the analysis results, and existing frameworks that aim to adapt input with various methods are described. We also talk about frameworks that record and replay input, and frameworks that recognize and create gestures, providing examples for each. Finally, we introduce user models and shared user models, also providing examples.

Naftali and Findlater [16] conducted studies that aimed at learning how motor-impaired individuals use smartphones in their daily life, and how these devices present challenges and empower them. The main input challenges experienced are multi touch, text entry and text correction. The participants' main wishes for smartphones were the development of more precise voice-to-text and voice control technology, and provide alternatives to multi-touch.

Anthony et al [18] analysed 187 YouTube videos that depicted users with motor disabilities interacting with a mobile touchscreen, along with conducting surveys on the video uploaders. Touchscreen devices offer interaction that may be difficult for users with disabilities, forcing users to customize devices for their own use. 91% of videos depicted direct interaction (fingers, hands or feet), only 56% of that interaction being through fingers. Most of that interaction was one handed. Challenges found in touchscreen interaction were that some users held their finger on the screen for too long. Dragging and sliding motions also presented challenges, and some users were unable to reach the entire screen.

Direct interaction also included palms or side of hands, of which 83% were small children, knuckles, of which most were babies or young children, noses, and feet. Indirect interaction (8%) included the use of head sticks and mouth sticks. These had the limitations of not being able to perform multi-touch gestures and needing the device's sensitivity or delay time to be adjusted. The user's posture in these videos was mostly seated (71%), followed by lying down (17%) and reclining (8%). The device's position in these videos were mostly lying flat (42%) and standing vertically (41%). Handheld use was only in 8%. Some design implications resulting from this data were: allowing device sensitivity adaptation, providing alternative support for multi-touch interaction and support for constant touch adaptation.

Nicolau et al [19] studied the differences and similarities between motor-impaired and able-bodied users in how they performed a set of interaction techniques: tapping (touching a target to select it), crossing (crossing over target to select it) and directional gestures (gestures in 16 possible directions, which could be performed anywhere on the

screen). These were analysed with two parameters: size and position of the target. The test results revealed that target size significantly affected tapping error-rates. Regarding positioning of targets in tapping, having the target on the edges and within their arm support's reach benefitted motor-impaired users. Crossing error-rates were independent of target position, but target size affected them for both motor-impaired and able-bodied users. Directional gesturing was the least inclusive technique: while benefitting able-bodied users when the target was small, it should be avoided for user interfaces designed for motor-impaired users. The design conclusions derived from this study were: both tapping and crossing are inclusive interaction techniques that can be performed well by both motor-impaired and able-bodied users; directional gestures should be avoided for motor-impaired users; error-rates start to converge when target size is between 7mm and 12mm; and that it is important to keep reach restrictions of motor-impaired users into account when positioning targets on the screen.

This analysis of differences between input of able-bodied users and of users with less motor dexterity includes another player group: children. Anthony et al [20] study the differences between adults and children in their touch and gesture input on touch screens. Via a set of tests using touch targets of varying sizes and the \$N Protractor framework to identify gestures, they identify two main challenges in identifying children's input: unintentional touches inside/outside of the target and low recognition accuracy for some gestures.

Due to children having smaller fingers, weaker arms, and less fine motor control and manual dexterity, the adult-trained touch and gesture recognition technology sometimes fails to accurately register child input, which produces much smaller touch points and exerts much less pressure.

After clicking the target and there being a minor delay in the system recognizing the touch, the child would press the same spot a few more times; since the target automatically advanced to the next view, the child accidentally hit the screen a few more times. These are called holdovers; children frequently performed these on small targets (81% of small targets).

They also missed targets with edge padding nearly double the times (30.2%) they missed targets without edge padding (17.8%). 99% of the misses on edge padded targets occurred in the edge padding 'gutter' (space between the target and the edge of the screen).

The design recommendations based on these results were: use timing and location of the last pressed target to identify and ignore holdovers; use recommended target-sizes; increase target active area for slightly out-of-bounds touches to count; count edge padding as a target touch, or align targets to the edge of the screen to eliminate the

gutter; and to design specific gesture sets to train gesture recognizers for problematic input.

Vickers et al [21] developed a framework that dynamically adapts games to its players' physical or cognitive disabilities, so that they can focus on the intellectual challenges rather than the physical challenges of the game. The first step to the development of this framework was the task analysis from gameplay by expert players; expert players are defined here as players who have found the most efficient way to complete game tasks. This way, it's possible to define common game tasks and the properties associated to them. The developers used the "om" task analysis method, which analyses user attention (eye tracking), intention (think-aloud protocols) and action (input & screen capture).

The framework uses this information to adapt these tasks according to user ability profiles, which are unique to each user. To adjust these user profiles, there are Performance Indicators associated with tasks, which adapts the game tasks according to the players' difficulties in real time. The lack of a need to manually configure these profiles has considerable benefits for these users.

To further adapt gameplay according to the user's needs, they take two more steps: an initial diagnostic test to assess the user's abilities, and the analysis of a log of previous gameplay sessions to identify the user's strengths and weaknesses.

Two possible approaches to the framework's implementation are described: the first implies the use of a middleware solution, which injects input events into the game. Adaptable task- and user-specific components are overlaid over the game interface. This approach became unfeasible, however, because of anti-cheat firewalls that prevented input proxies and the origin of input differed in games (some received input messages directly, others only received them from the OS).

The second approach, and the one adopted by the developers, is the creation of a framework via the implementation of C++ libraries that can be incorporated in games and game engines.

Zhong et al [22] created Touch Guard, a service for motor-impaired users that runs in the background on Android, functioning as an invisible overlay that intercepts and optimizes touch input. The service uses Android's Accessibility API to know every target's position on the screen.

The main functionality of Touch Guard is the Enhanced Area Touch, which increases the users' precision in selecting UI targets. It does this by enlarging the touch point into a customizable-sized circle, which detects all elements that it may intersect. If it intersects more than one element, Touch Guard has two main methods of

disambiguating the intended option: by magnifying the area, and by presenting a full-screen text list of the intersected options by extracting the elements' title via the Accessibility API.

Touch Guard also offers a Click on Lift mechanism, which allows the user to click any part of the screen, and only select an element on finger lift. Another mechanism it offers helps users with hand tremor by filtering high speed movements or movements with a sharp turning angle. It does this by monitoring touch speed and angle in real time, and ignoring all touch events with speed or angle above a certain threshold.

Dynodroid [23] is a system developed to automatically monitor, select and generate appropriate inputs to an Android application. It follows an observe-select-execute cycle, which allows it to only generate input that is relevant to the application in question. The system monitors and generates both user input and system events.

Firstly, it observes which inputs can be relevant to the application by obtaining the view hierarchy of the layout: this way, it can extract the registered call-back methods and the location and size of the visible user interface elements. It also instruments the SDK to monitor when the app registers or unregisters broadcast receivers and system services. This way, we know what the application expects.

Next, with this data, the Selector uses a randomized algorithm which penalizes frequently chosen events to select the input to generate. The Executor generates the selected input event. But, as the system cannot create event objects arbitrarily, it constructs the data associated with the selected input event and obtains an event object from the pool maintained by the Manager System Service, and uses the ADB to send the event to the device.

RERAN [24] is a framework with the purpose of capturing and replaying both GUI and sensor events at a low-level precision and micro-second accuracy. Its goal is to help with development and test debugging.

The developers use the ADB (Android Debug Bridge) as an interface between RERAN and the smartphone. It records input by using the `getEvent` tool, which reads the `/dev/input/events` files to produce a real-time log. After creating this log, RERAN converts the data into a concise form and time delays are calculated.

Initially, to replay the events the developers were going to use the `sendEvent` tool, but this tool had a small lag. Due to this, they decided to only use the `sendEvent` source code as a guide, and implement their own (less resource-intensive) replay agent. This agent directly injects events into the phone's event stream by writing them to `/dev/input/event*`.

Poster: Retro [25] is another record and replay application that reproduces problems encountered in applications, for developers to be able to accurately debug them. The framework records application-layer events (touch, sensor readings, method calls and return values). It also includes a selective logging mechanism which only logs certain event types. The developers have access to the log and replay it in the replayer, which is integrated in Android's development workflow. This replaying interface is also capable of forwarding and rewinding the input events.

Button Blender [26] is a framework created with a record-remix-replay architecture. The aim of this framework is to aid children, elders and gamers with motor impairments during gameplay. The framework captures the player's input in real-time during gameplay, and stores these input events into a log file, the 'play-through file'. Each event is timestamped, and separated by commas.

With these input events, and with a previous recording of expert gameplay, Button Blender intends to use the expert gameplay with a 'sticky-key' logic: the player only plays with one button, while the expert gameplay is synched with current gameplay and replays all other events. For example, the framework can automatically replay the 'walking forward' event, while the player only clicks the 'jump' button.

The main challenge of this framework is the lag between the player's input and the resulting combined output, due to the various processes running asynchronously. Another challenge is synching the expert gameplay with the player gameplay; this requires the ability to detect when the player is at a certain game location in the game world. They accomplish this with hybrid computer vision-based matching techniques.

In our application, we extend a previous record and replay library and Accessibility Service to monitor expert gameplay and adapt the player's touches to what the game expects. This library, SWAT [27], is extensible, adaptable, and allows access to screen content and system I/O events. It also includes a logging mechanism, navigation mechanisms, external device control and assistive macros.

SWAT is an Accessibility Service which uses Android NDK, which allows developers to use the device's native code, and a rooted device. This gives us permission to access the system and access and inject low-level events. Input events are captured before they are processed by the OS and are categorized by the library. SWAT uses macros to replay the input events.

We will need to extend this framework with gesture recognition, as it currently does not recognize the gestures it records and replays. There are various examples of gesture recognition frameworks.

\$N Protractor [28] is the final product of joining the \$N multi-stroke recognizer with Protractor. \$N matches candidate gestures to templates with a geometric matching approach; it checks angular alignment and distances between corresponding points iteratively. Protractor greatly reduces the computing time during the matching process by removing the iterative search over angles and instead evaluating with a distance metric. This metric finds the angle between two vector based representations of gestures. By enhancing \$N with Protractor, the developers alleviated the time cost of representing multi-strokes as uni-stroke gestures, attaining 97% of gesture recognition accuracy in the tests.

gRmobile [29] is a framework for the recognition of touch and accelerometer gestures which uses hidden Markov model. In this framework, recognition is done by comparing the previously prepared and analysed input pattern data with the database gestures. The framework has two modes: gesture training (to build the database), and gesture recognition. To build the database, it is necessary to train and save a set of gestures into the framework. There are various steps to gRmobile's gesture recognition. The first step is Segmentation, which distinguishes the beginning and end of gestures. Next is Filtering, which eliminates the superfluous parts of the data. Quantitizer is only used for accelerometer gestures, and it approximates the data to a smaller set of values. Next, Model computes the probability of gestures. And finally, Classifier, which is used to identify the gesture according to the database.

Some gesture recognizers include a functionality which creates the gesture recognition code based on samples of the gesture. Gesture Coder [30] is a tool that learns multi-touch gestures by demonstration: from sample gestures provided by the developer, it automatically generates user-modifiable code which detects the gesture and provides call-backs to react to the gesture. The detection accuracy varies according to the number of samples provided and as the complexity of the recognition task rises. Gesture Coder was intended to be used as a plugin for an IDE, so as to not interrupt regular implementation.

The lifecycle of a gesture generated by this framework involves 6 states: Possible, which is when the framework receives the first touch event; Failed, which is when the gesture can no longer be a possible match; Began, which is when a gesture is first recognized; Changed, which is the detection of new touch events while the gesture is still recognized; Cancelled, which is when it no longer can be the detected gesture; and Finished, which is when the gesture is concluded.

Proton [31] is a framework used to create, analyse and detect custom multi-touch gestures. Each gesture is specified as a regular expression over a stream of touch events. Proton analyses all the gestures in its gesture set to detect conflicts between gestures,

and automatically creates recognizers for the set as well. This framework provides a gesture tablature, which is a graphical notation of every step toward the formation of a gesture. Rather than considering touch trajectory, Proton tablature uses horizontal tracks to describe the touch sequences. Proton also provides a graphical editor, which can create the tablatures and automatically generate regular expressions to describe them. With this framework, instead of having various gesture call-backs split across the code, the developer writes a single call-back function to react to the recognition of the custom gesture.

To further and more precisely adapt applications to the user, we can use User Models. These hold information about the particular requirements of the user. Kurschl et al [32] describe the different approaches to user modelling: Content- and feature-based, which saves a set of feature-value pairs. Case-based, which saves information about previously problematic situations to later be capable of recognizing similar situations. Collaborative, which matches similar users. Demographic, which matches users based on their demographic background. Knowledge-based, which relies on existing information about items and typical users, and human expertise. For an application aimed at aiding motor-impaired users to use a smartphone, accumulating knowledge on different manifestations of these impairments can be useful.

Kurschl et al create a user modelling wizard which uses a hybridization of content and knowledge based approaches. It uses a series of analytic tests to gather information about the user's input difficulties.

The user model saves data about the users preferred input device (switch or touch input), the user's ability to reach every region of the screen, the minimum size for UI elements, whether to react immediately on finger-down or only on finger-up, the user's ability to perform swipes, and, if the user uses a switch rather than touch input, information such as number of switches, hold time, lock time and scan time. This information is used to generate an application configuration molded to the user's needs.

These user models are useful, but can be strenuous if we are required to create one for every user model-enabled application we use. A solution to this would be to have a single universal user model. Shared User Models [33] (SUM) support the sharing of domain independent user models across applications and devices, to provide system-wide tailored accessibility.

There are two main methods to populate user models: User-initiated, which is the adjustment of settings or preferences, or application-initiated, which submits the user to various exercises to test their abilities. As both of these are tiring processes, by sharing user models the user will not have to be subjected to them for every application they use.

SUM further eases the process by storing the user information both locally and online, periodically syncing the models to stay up to date. This way, the user can use these models in the same way both across applications and devices.

The collection of data for these models is mostly automated, focusing on low level interaction and sensor data. The SUM Client, once embedded into the application, is what parses the user models and tailors the UI according to the user's individual preferences.

SUM lacks the application-initiated approach which, although tiring for users to complete, is much more accurate in measuring user performance. As the test would only be completed once, as SUM intends to share the profile across applications, it is a worthwhile effort.

2.3 Discussion

We explored two main areas related to the mobile game accessibility issue described in the last chapter: accessible gaming and input challenges and adaptation.

As was discussed in the last chapter, there has been various attempts toward mobile game accessibility. We saw that each accessibility tool covers a specific issue. A common trend was that accessible mobile games had that accessibility directly implemented into it, creating that game specifically for a certain group of disabled players. This can create a stigma toward the game, and further isolate disabled players. To avoid this, regular games that are shown to be popular with the masses should be made to include everyone.

Input adaptation is an area that is also still in progress. We saw that motor impaired users have many difficulties with touchscreen device input that still are not addressed properly. We explored various frameworks with different approaches to adapting user input. We found that most frameworks use the application-specific approach, assuming the form of libraries that need to be added into an application in development, rather than being a system-wide approach. We looked at input capture and replay frameworks, which log the user's input to later replay. User models are also introduced as necessary to hold information about the user's particular needs.

Motor impaired people were the main focus of this chapter, but other populations were also mentioned, such as elders and children. We can argue that these populations are also excluded from some applications and games based on their motor abilities, due to mobile applications being created specifically for able bodied adults, using their interaction and dexterity as the interaction baseline, thus not taking into account the specific input differences of children and elders.

We have explored various areas relating to mobile game accessibility and input adaptation, and analysed studies and frameworks relating to these areas. We can conclude from our analysis of the related work that there is still a large gap to bridge in mobile game accessibility. Also, not much is known about the mobile game demands that impede motor impaired people from playing, as the topic still has not been sufficiently explored.

In the next chapter, we will conduct a data collection study so as to determine current mobile game demands and, with these results, create a gesture catalogue.

Chapter 3

Catalogue of Input Demands of Touchscreen

Games

Mobile game accessibility is important, but current accessibility methods fall short of what is needed, and there are gaps that need to be bridged. Due to the fact that game developers have the freedom to define their own gesture recognizers, the possibilities are endless in terms of input demands of games.

In this chapter, we present an analysis of current game input demands. We describe our experimental protocol and report the obtained results, characterizing the most commonly used gestures throughout the games. Finally, we discuss the results and assess what is necessary to evaluate in the second study.

3.1 Data Collection

This study aims to provide a clear view of the landscape of current games by collecting and analysing gameplay data. With this, we will create a gesture catalogue of the most used gestures in games with detailed parameters for each game.

The next sections describe our research questions and experimental protocol.

3.1.1 Participants

Twenty five able bodied participants, 21 males and 4 females, took part in the user study. Their age ranged from 23 to 42 with a mean of 28.1 years old. They were recruited within Newcastle University's Open Lab. 56% of the participants played mobile games, and 88% played games in general. Only 3 participants did not play any type of games.

3.1.2 Apparatus

Hardware Technology

The study was performed on Nexus 5 tablets with a multi-touch capacitive touchscreen, running Android 5.1 and Android 6.0. They were used both in landscape and portrait mode, depending on the game. Input data was captured with the modified TBB accessibility service mentioned in 1.2.5. Throughout the study the use of two portable computers running Android Studio was necessary when the study was done in pairs, since Android Studio can only record one device screen at a time.

Game Selection Criteria

A sample of the top 25 free games in the Google Play Store of March 2016 was taken for the study. An evaluation of these games was performed beforehand according to Ponnada and Kannan's Playability Heuristics for Mobile Games [37]. These playability heuristics evaluate game usability, gameplay and mobility of the game, and are represented in tables 1, 2 and 3.

No.	Game Usability Heuristics
GU1	Audio-visual representation supports the game
GU2	Screen layout is efficient and visually pleasing
GU3	Device UI and game UI are used for their own purposes
GU4	Indicators are visible
GU5	The player understands the terminology
GU6	Navigation is consistent, logical, and minimalist
GU7	Control keys are consistent and follow standard conventions
GU8	Game controls are convenient and flexible
GU9	The game gives feedback on the player's actions
GU10	The player cannot make irreversible errors
GU11	The player does not have to memorize things unnecessarily
GU12	The game contains help

Table 1 Game Usability Heuristics

No.	Mobility Heuristics
-----	---------------------

MO1	The game and play sessions can be started quickly
MO2	The game accommodates with the surroundings
MO3	Interruptions are handled reasonably

Table 2 Mobility Heuristics

No.	Gameplay Heuristics
GP1	The game provides clear goals or supports player created goals
GP2	The player sees the progress in the game and can compare the results
GP3	The players are rewarded and rewards are meaningful
GP4	The player is in control
GP5	Challenge, strategy, and pace are in balance
GP6	The first-time experience is encouraging
GP7	The game story supports the gameplay and is meaningful
GP8	There are no repetitive or boring tasks
GP9	The players can express themselves
GP10	The game supports different playing styles
GP11	The game does not stagnate
GP12	The game is consistent
GP13	The game uses orthogonal unit differentiation
GP14	The player does not lose any hard-won possessions

Table 3 Gameplay Heuristics

As well as this evaluation criteria, we devised a list of game input requirements, which assess the gestures required to play the game. This list is represented in table 4.

No.	Game Input Requirements
GI1	The game requires taps
GI2	The game requires swipes

GI3	The game requires drags
GI4	The game requires double Taps
GI5	The game requires hold
GI6	The game requires pinch or spread
GI7	The game requires rotation
GI8	The game requires the use of an accelerometer
GI9	The game has timeouts
GI10	The game requires agility
GI11	The game allows pauses in games
GI12	The game is time-sensitive
GI13	The game does not require two hands to play
GI14	The game requires touch Precision
GI15	The game requires multi touch input

Table 4 Game Input Requirements

According to these heuristics and the evaluation criteria devised by the research team, we excluded 7 games from the original top 25 games, and added games to the list up to the 32nd top game in the Play Store, so as to have the 25 game samples. The excluded games were removed for the following reasons: the game being solely time-precision based; an accelerometer being necessary for gameplay - this phase of the project only considers touch-based games, or games that can be fully played without needing an accelerometer; remakes or duplicates of games already in the list (Candy Crush Saga and Candy Crush Jelly Saga, for example); as well as an isolated case of an application in the list which was actually a collection of other games.

The final game list is as following:

1. Color Switch
2. Stack
3. Candy Crush Jelly Saga
4. Futurama: Game of Drones
5. Kendall & Kylie
6. Subway Surfers
7. 8 Ball Pool
8. Words Crush: Hidden Words
9. MARVEL Contest of Champions

10. Solitaire!
11. Cooking Fever
12. My Talking Tom
13. DragonSoul
14. Trials Frontier
15. Roll the Ball – slide puzzle
16. Clash of Clans
17. Geometry Dash Lite
18. Crossy Road
19. Gyrosphere Trials
20. Twist
21. Mandala Coloring Pages
22. World Chef
23. Alto’s Adventure
24. Agar.io
25. PAC-MAN

No.	GU1	GU2	GU3	GU4	GU5	GU6	GU7	GU8	GU9	GU10	GU11	GU12
	25	25	25	25	25	24	25	25	25	18	21	25
	0	0	0	0	0	1	0	0	0	7	4	0

Table 5 Game Usability Results

As we can see in table 5, most of the Game Usability Heuristics were met. One game did not have a consistent, logical and minimalistic navigation, in 28% of games irreversible errors could be committed, and 16% of games required memorizing things needlessly. The other game usability heuristics were met by all games.

	MO1	MO2	MO3
	25	25	22
	0	0	3

Table 6 Mobility Results

Of the Mobility Heuristics, 12% of games did not handle interruptions reasonably, as shown in table 6. The other mobility heuristics were met by all games.

No.	GP1	GP2	GP3	GP4	GP5	GP6	GP7	GP8	GP9	GP10	GP11	GP12	GP13	GP14
	25	25	24	25	24	25	15	25	15	25	25	25	25	21
	0	0	1	0	1	0	10	0	10	0	0	0	0	4

Table 7 Gameplay results

Of the gameplay heuristics, one game did not have meaningful rewards, one game did not balance challenge, strategy and pace correctly, 40% of games did not have a meaningful story that supported the gameplay, 40% of games did not allow the players to express themselves, and 16% of games made players lose hard-won possessions. The other gameplay heuristics were met by all games. These results are shown in table 7.

No.	GI1	GI2	GI3	GI4	GI5	GI6	GI7	GI8	GI9	GI10	GI11	GI12	GI13	GI14	GI15
Y	24	15	10	0	5	3	2	0	12	14	22	12	23	12	4
N	1	10	15	25	20	22	23	25	13	11	3	13	2	13	21

Table 8 Game Input Requirements results

Table 8 shows us the Game Input Requirements results. Only 4% of the games did not require taps. 40% required touch precision. 60% of games used swipes and, of those, 54% used up and down swipes, 34% used left and right swipes, and 10% used diagonal swipes. 40% of games used drags. Hold was used in 20% of games. Pinch and spread gestures were used in 12% of games, while general multi touch was used in 16%. Only 8% of games required two handed gameplay.

3.1.3 Tools

The system used to log the input data of the participants throughout gameplay is called TinyBlackBox (TBB), a standalone accessibility service [48]. The original service logged device type 1 user touch interaction to XML files, and scraped application data such as layouts and page elements.

In most games, the layout and page elements are not accessible to the system, and so we view every game as a black box as we are given no information about its internal workings. Due to this, we do not use the application data scraping functionalities of the system.

For the purposes of this study, we extended TBB so as to include device type 2 touch interaction, migrated the logging destination to a SQLite database so as to more easily access and query the data, added a functionality that detects when non-system

applications open and close, and created a preliminary analysis feature which draws the touch interactions to an Android View canvas. Figure 1 shows the structure of the database that was added to the service.

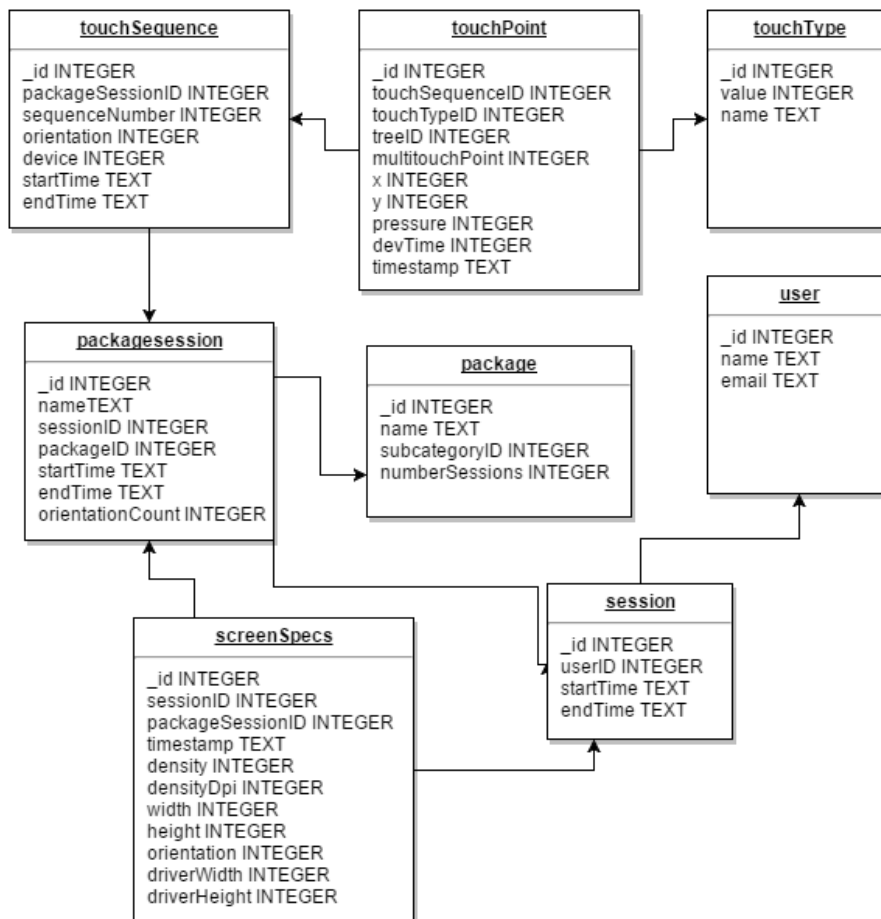


Figure 1 Extended TBB Data Model

To analyse the study results in depth, we extended TBB to reproduce and analyse the touch input. In a first phase, the input touch points are drawn onto an Android Canvas and saved as a PNG image file, to later be used for manual inspection. On the same canvas, with the TBB touch injection functionality, we inject the touch input, which draws onto the canvas. What is drawn onto the canvas is also saved as an image file. We also ran a standard Android gesture listener in the background while the service injected the touch points. This way, we were able to detect many of the gestures independently from the game gesture recogniser, allowing a less exhaustive manual

analysis posteriorly. All data about the gestures, including various additional evaluation parameters, were saved into CSV files.

3.1.4 Procedure

The study was performed in Newcastle University's Open Lab. Each session was 45 minutes to an hour long, and participants were evaluated individually or in pairs, depending on their availability. Ethics were approved by Newcastle University prior to the study, as seen in Appendix D, and the study went according to the script in Appendix B.

The participants were told that the purpose of the study was to collect samples of able-bodied gameplay to identify the required interaction demands to play these games, so as to adapt these interactions to motor impaired users in a later phase of the project.

Next, the participants filled in an online questionnaire about their demographic data and their gaming habits, mobile and otherwise. Participants were then informed about the procedure of the study.

Each participant played five to six randomly picked games from the sample list. They were allowed a short learning phase to get used to the controls, and then played the game for 5 minutes. These 5 minutes of gameplay was recorded; our TBB system logged the input data in the background whilst we recorded the screen with Android Studio's recording option.

3.1.5 Design and Analysis

With the extended TBB accessibility service described above, and for each touch sequence, which we defined as beginning from the first touch point until all fingers are lifted, in the case of multi touch gestures, we collected various parameters about the touch interaction. These parameters were: *Multitouch* (a true or false Boolean that indicated if the gesture was multi touch); *Duration* (measured in milliseconds); *Speed* (measured in pixels per millisecond); *Travelled Distance* (total length of the gesture, measured in pixels); information about DOWN and UP events, which refer to touch down (beginning of gesture) and touch up (finger lift – end of gesture) such as *timestamps*, and *x and y coordinates*. We collected *offset X* and *offset Y* parameters, which were x and y offsets from the beginning of the gesture to the end of the gesture (x and y offset from first and last touch point). We also collected the *number of scrolls and flings*, which was provided by Google's standard gesture recognizer, as well as the *gesture detected* by the recognizer. We collected the *interval from the previous gesture* (in milliseconds), the *gesture direction* (up, down, left or right), and the gesture *Angular*

Offset, which compared the diagonal distance from the first and last touch point to the travelled distance, thus determining the gesture offset from a straight line – this was particularly useful for evaluating swipes.

These parameters as well as the manual analysis of the generated canvas images of the gestures were then used to determine which kind of gesture each touch sequence was. With this, we joined each gesture type from the game and evaluated them separately with IBM SPSS Statistics 23, therefore getting each game’s tap information, swipe information, etc. For this evaluation, we used the following parameters: Duration, Speed, Travelled Distance, Intervals, and Angular Offset. For taps, we removed Speed and Angular Offset from the evaluation parameters.

With SPSS, we extracted the maximum, minimum, standard deviation, mean, median and modes from the gesture data, and used these values to compare the game gesture demands.

3.2 Results

Our goal was to find the most commonly used gestures in today’s mobile games, as well as get detailed data about the gesture demands of each game. The results presented identify the most commonly used gestures, compare these gestures among the games and define the touch requirements for each game.

The complete game demands input catalogue is in Appendix A – Game Input Demands Catalogue.

3.2.1 Taps and Long Presses

All games required target taps, even if only to choose menu items. We identified 8 games in which interval and time-sensitive taps were used: Color Switch, Stack, Geometry Dash, Crossy Road, Twist, Alto’s Adventure, Marvel and Trials Frontier.

Most games’ tap duration median approximated 60ms. Four games had tap duration medians above 100ms: Altos Adventure (m=103), Geometry Dash (m=105), Stack (m=127) and Twist (m=110).

Figure 2 shows tap interval differences among games; five games have intervals below 500ms: 8 Ball Pool (m=445.5), Color Switch (m=238), Crossy Road (m=202.5), Mandala (m=289) and Marvel (m=134). As we can see, with the exception of Mandala, these correspond to games previously identified as interval and time-sensitive games. Five games have median intervals over 2000ms: Alto’s Adventures (m=2175), Candy Crush (m=2502), Futurama (m=3342.5), Roll the Ball (m=4027.5) and Words Crush (m=2542). These correspond to games played with predominantly swipes and drags,

and in the case of Alto’s Adventures, a game with larger wait times throughout the game.

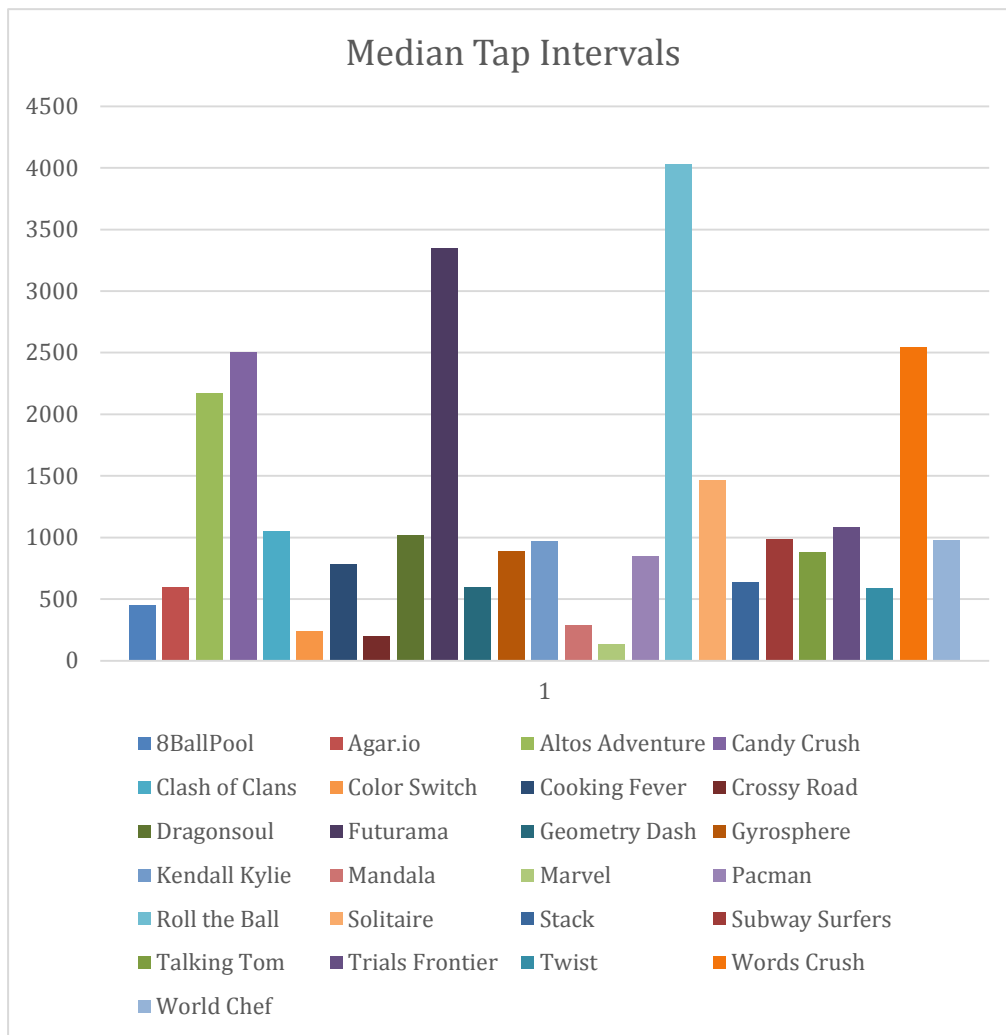


Figure 2 Median Tap Intervals chart

Within these games, target sizes varied from 40mm to 520mm. The most used target size was 50mm – mostly for advertisement exit buttons, in 60% of games – and 80mm to 130mm were used in more than 50% of the games, the most used being 130mm (56% of games).

In some games, long presses provided a different result than a regular tap. Long presses were important to the gameplay of three games: Altos Adventure, Geometry Dash and Trials Frontier. The median duration of these presses were 803ms for Altos Adventure, 630.5ms for Geometry Dash and 955ms for Trials Frontier.

3.2.2 Swipes

Swipes were used in nine games: Candy Crush, Futurama, Kendall & Kylie, and Subway Surfers, Roll the Ball, Crossy Road, Gyrosphere Trials, Pacman and Marvel. The median duration of swipes in most of these games is between 150ms and 350ms, with only Kendall and Kylie having a duration median of 604.5ms.

Swipe speed medians vary between 0.5px/ms and 1.7px/ms. The two games with fastest swipes (above 1px/ms) are Crossy Road (m=1.55), Marvel (m=1.66), Subway Surfers (m=1.38) and Pacman (m=1.21). These are also the games within the smallest median intervals (lower than 1000ms), indicating that these are fast paced swipe games. Two games differentiate themselves for having a large interval median, these being Candy Crush (m=2698) and Futurama (m=3423). This reflects the slower pace of these games.

The games with the largest median travelled distance were Kendall and Kylie (m=496.57) and Subway Surfers (m=321.82). Every other game had a median travelled distance below 300px, the smallest being Candy Crush (m=134.74).

We also measured the angular offset of swipes. This was calculated by subtracting the distance of the beginning point A to the end point B from the gesture's total Travelled Distance. With this we were able to calculate how close to a straight line the gesture was. In terms of angular offsets, only 3 games had an offset larger than 500px: Candy Crush (m=2535.84), Futurama (m=3300.8) and Crossy Road (m=235.55). The game with the smallest offset was Marvel (m=107.59), indicating straighter swipes in this game.

3.2.3 Drags: Regular, Scribbling, Rotation and Shapes

Drags were defined as dragging an object from a start point to an end point – in 4 games - to scribble within a certain area – 2 games - and perform one-finger rotations – 2 games – and one of the games sole gameplay was to draw simple shapes. Drags were used predominantly in 10 games: 8 ball pool, Agar.io, Clash of Clans, Cooking fever, Mandala, Solitaire, Talking Tom, Words Crush and World Chef.

Drag duration median of most games was 200ms to 800ms. Three games had their median above 800ms: 8 Ball Pool (m=1466), Solitaire (m=847) and Words Crush (m=1321). Roll the Ball and Talking Tom had the smallest durations; which can be correlated to Travelled Distance, as these two games as well as Clash of Clans had the smallest travelled distance median.

In terms of speed, 8 Ball Pool ($m=0.1981$) and Clash of Clans ($m=0.3856$) are the games in which drags are performed the slowest, and three games are above 0.8px/ms in speed: Agar.io ($m=0.879$), Cooking Fever ($m=1.0219$) and Words Crush ($m=0.9289$). These can be associated to being faster paced games.

Most games had an angular offset between 400px and 1000px . This offset was largest for Words Crush ($m=1579.55$), and smallest for Mandala ($m=235.81$) and Agar.io ($m=278.99$). This can be associated to the various shapes required by Words Crush, and for the games with smaller offset it suggests a larger use of straighter drag gestures.

Within drags, we also identified specific commonly used gestures: scribbling over a certain area, rotating back and forth with one finger, and drawing shapes.

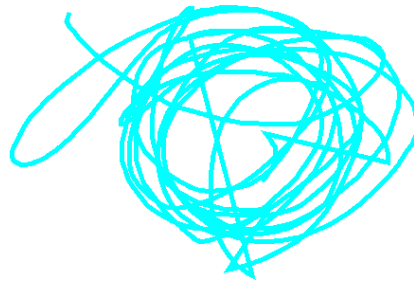


Figure 3 Canvas drawing of a scribble performed in Talking Tom

Scribbling was mainly identified in two games: Talking Tom and Mandala. This gesture usually implies scribbling over a certain area, with the goal being to completely fill in the area - figure 3 is an example of a scribble performed by a user in Talking Tom. The median duration of these scribbles was between 3486ms and 4393ms . The median speed was 0.526px/ms in Mandala, and 0.732px/ms in Talking Tom. The travelled distance median of scribbling was 1833.2px in Mandala and 3096px in Talking Tom.



Figure 4 Canvas drawing of one finger rotation performed in 8 Ball Pool

One finger rotation was found in 8 ball pool and in Agar.io. Figure 4 exemplifies a one finger rotation performed in 8 Ball Pool.

While duration median was similar in both (5937ms and 6323.5ms), every other factor varied largely. Speed median was 0.14px/ms for Mandala and 0.836px/ms for Talking Tom and travelled distance varied from 1238.96px for Mandala to 4506.12px for Talking Tom. This shows the large range for one-finger rotation, as it can be a small curved stroke or various long strokes.

Shapes were only used in one game, Words Crush, but as it was the sole gesture for the main gameplay, we consider these to be important to our analysis as well. We identified 9 different shapes: backwards C, C, backwards N, N, n, Z, S, U, and XI. Figure 5 denotes various shapes performed in Words Crush.



Figure 5 Various shapes performed in Words Crush

The duration median of the shapes varied between 1132ms and 2080ms. The travelled distance median varied between 1148.6px and 1707.6px. The speed median varied between 0.73px/ms and 1.31px/ms. We also counted the frequency of each shape so as to order them by number of uses, as seen in table 9.

Z	Backwards C	n	C	U	S	XI	Backwards N	N
24	17	17	13	13	6	5	4	3

Table 9 Shape frequency

3.2.4 Pinch and Spread

Finally, the last gesture we could identify was pinch and spread. Table 10 and 11 show the statistics of pinch and spread gestures performed in Mandala.

3.3 Discussion

Our goal was to collect gameplay data so as to identify current game input demands, and determine the most used gestures in games. We will discuss the study results and their implications.

We were able to make detailed conclusions about the most used gestures throughout games, as well as relate different games. We concluded that the most used gestures in our sample were taps, long presses, swipes, drags, scribbling within an area, one-finger rotation, dragging in various shapes, pinches and spreads. We were able to relate various games with basis the predominant gestures used and their unique characteristics.

Interval and time-sensitive tap games had the smallest median intervals. Slower-paced games had larger median intervals. Long press duration medians varied from 630.5ms to 955ms.

We were able to correlate games with the fastest swipe speeds to the games with the smallest swipe intervals. We also correlated one of the fast-paced games to the smallest angular offset, indicating speed might influence gesture steadiness in swipes. A faster swipe means a more natural and sleek stroke.

Swipes were generally faster and shorter in length than drags. General angular offset for swipes was significantly lower as well.

We identified drag gestures such as scribble, one-finger rotation and shapes. The shape frequency table allows us to see which the most used shapes in gameplay were.

Finally, we saw that pinch and spread gestures were similar to each other. Pinches had a median duration of 3432ms, and spreads had a median duration of 2693ms. Pinch median speed was 0.1077px/ms, while spread median speed was 0.1062px/ms. The median travelled distance of pinches was 350.7252px and the median travelled distance of spreads was 274.5926px.

3.4 Summary

Our goal was to collect able-bodied gameplay data, and from that derive a list of We have analysed 25 games from the top 25 games in the Play store. With this, we intended to find the most used gestures in today's games, and determine their specific input demands.

We were able to identify the main gestures used in today's games, and narrow them down to various subcategories. We related certain aspects of gestures to aspects of the games, such as gameplay pace.

These results will be relevant for chapter 4, in which we conduct a study to evaluate the feasibility of these gestures for people of different abilities and, in result, determine the playability of each of the games.

Chapter 4

Understanding the Abilities of Unconventional Gamers

In this chapter, we present an analysis of touch capabilities of people with different motor abilities. We first provide an overview of previous works that analysed various characteristics of touch capabilities of various populations. Next, we describe our experimental protocol and report the obtained results, comparing each population's capabilities. Finally, we discuss the results and compare them with the first study.

4.1 Background

We will analyse previous work that studied touchscreen capabilities of people of various ages and abilities. We then present what will be analysed in this study.

According to the United Kingdom's Office of National Statistics, in 2016 33% of people aged 65+ use mobile phones[43], and, according to Australia Bureau of statistics, in April 2012 818,500 children aged 5 to 14 years (29%) had a mobile phone[44].

Nicolau et al [19] studied the input differences between motor-impaired and able bodied users. They analysed how they performed target taps, crossing over targets to select them, and directional gestures in 16 possible directions, on various locations on the screen and in various sizes. The results of this analysis revealed that target size significantly affected tapping and target-crossing error-rates. Having targets within their arm support's reach and on the edges made motor impaired users have a higher accuracy, and directional gestures were found to not be inclusive for motor impaired users.

Anthony et al [20] studied the input differences between children and adults. They performed a set of tests with touch targets of varying sizes, and used the \$N Protractor

framework to identify gestures. The main input challenges for children were that they would unintentionally touch outside the target and that the gesture recognition accuracy was low. This is due to touch and gesture recognition technology being trained with adult input, not taking into account the particularities of child input such as having smaller fingers, exerting less pressure, and having less fine motor control and manual dexterity. Another input challenge were holdovers: since the system had a small delay in recognizing the touch and advancing to the next view, the child would press the same spot a few more times than necessary. Children performed these on 81% of small targets. They also missed targets with edge padding 30.2% of the time, and they missed targets without edge padding 17.8% of the time. 99% of the misses on edge padded targets occurred in the edge padding ‘gutter’ - the space between the target and the edge of the screen. In total, children missed targets 46% of the time, while adults missed 32% of the time. This indicates a need for larger input tolerance for children.

Finally, Kurniawan et al [40] conducted a multimethod study with people aged 60 and older to analyse various aspects of their relationship with mobile phones: their usage patterns, problems, benefits, and desired and unwanted features. They conducted Delphi interviews, group discussions and online surveys. They mostly used their phones for communication, and found that the main issues in mobile phone usage were the small text size, the size and location of buttons - they were usually too small and close together, which affected their touch accuracy and visibility - and phone customisation – elders always had to ask someone else to customise their phone for them.

As seen above, there are input difficulties for motor impaired people, as well as children and elders. The able-bodied, adult-trained touch and gesture recognition technology sometimes fails to accurately register their input. In this study, we will analyse their touch abilities in detail, and then compare the results to the first study, so as to determine the playability of today’s games.

4.2 Data Collection

This study aims at understanding the touch capabilities of people with varying abilities, so as to later assess if current games are accessible to all. The next sections describe our experimental protocol.

4.2.1 Participants

Participants were recruited from 4 different population groups: motor-impaired, elderly, children, and able-bodied. The able-bodied participants were recruited to serve as a baseline. In total, there were 14 participants, 8 male and 6 female, 2 motor-impaired, 4 elders, 4 children and 4 able-bodied. The motor-impaired participants were

recruited at Dundee University, while the rest were recruited at the University of Lisbon.

4.2.2 Apparatus

The study with motor-impaired participants was performed on Nexus 5 tablets with a multi-touch capacitive touchscreen, running Android 5.1 and Android 6.0.

The study with children, elders and able-bodied users was performed on Samsung Tab Pro 10.1' tablets with an LCD multi-touch capacitive touchscreen, running Android 4.4.2. This part of the study was performed with a different device due to the study being performed in another country.

4.2.3 Gesture Prompt Application

An Android application was developed to prompt the users to perform various gestures. This application logged the participants' touchpoints as well as the parameters of each gesture prompted. The results from the previous study were used to decide the parameters for this application. We will now describe each chosen parameter, and explain what influenced the choice of each.

Landscape was chosen as the default orientation due to 52% of the games in the previous study being played in landscape. The gestures chosen for the application have the following main categories: Tap, Swipe, Drag and Pinch/Zoom. Within these, there are further subcategories.

Three subcategories were chosen for tap, based on the previous study: target taps, interval taps and long presses. Target taps includes 3 differing target sizes. The smallest target size is 50x50mm, due to appearing in 60% of the games of the previous study, mostly as exit buttons for advertisements. The medium-sized target is 130x130mm, due to it appearing in 56% of the games, and the largest target is 520x520mm, as it was the largest to appear in the previous study games. These targets were placed at various locations on the screen: top-left, top-centre, top-right, centre-left, centre, centre-right, bottom-left, bottom-centre and bottom-right.

For interval tapping evaluation, the participants were asked to repeatedly tap a large target, at first slowly and then as fast as possible. This target was also positioned in various locations on the screen. For long presses, the participants were asked to hold the target, for 1 second and for 5 seconds.

Swipe subcategories include directional swipes in all directions (up, down, left, right, up-right, up-left, down-right and down-left), and interval swipes. The directional

swipes were generated at different locations on the screen. Interval swipes were fixed to the middle of the screen, but were also in every direction.

Drag subcategories were short drags, long drags, shapes, one-finger rotation and scribble. Short drags were in all directions, at various locations on the screen. Long drags were horizontal, vertical, diagonal and curved, and the covered the entire screen. These drags varied their direction from left to right, and from up to down.

Five shapes were chosen from the previous study, particularly from the 5 most performed gestures in Words Crush: Z, backwards C, n, C and U. The gesture prompts of these shapes are presented in figure 6.

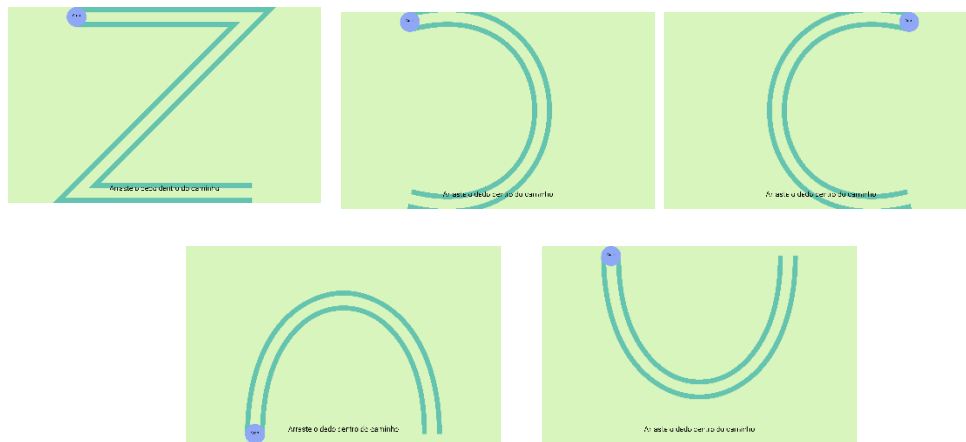


Figure 6 Gesture Prompt shapes

Back and forth rotational drag was used due to two different games from the previous study using one-finger rotation. We observed that these rotations tended to be back and forth.

The last subcategory of drag was Scribble, in which the participant was asked to cover the coloured area by scribbling. These areas gradually became smaller. They were asked to scribble the entire screen, half of the screen, a large target and a small target.

The last category, Pinch and Zoom, had no further subcategories, and were the only to test multi touch gestures.

Due to the possibility of the gestures being performed incorrectly or the application not recognizing gestures accurately, we decided to move between each gesture manually, i.e., the test monitor advances the application to the next gesture by pressing a button.

To do this, we created a second application that communicated with the Gesture Prompt application via TCP/IP. This communication application detected IPs on the

same network, connected to the chosen IP and communicated via Request/Reply messages. We used the zeroMQ framework to perform this communication.

The communication application could send two distinct messages: “next” and “nextGesture”. During a regular test session, we would send the “next” message, and upon receiving this message the Gesture Prompt application would advance to the next iteration of the gesture, or to the next gesture category once the iterations of that gesture were complete.

Between the sending of the message, reception of the message and the execution of the next iteration command within the Gesture Prompt application, there was always a short delay. Due to this study limitation, in the evaluation of the gestures performed we do not include the time between gestures as an evaluation criteria.

The Gesture Prompt application was also extended so as to evaluate the gestures performed by the participants, after the study. The extension to this application was similar to the extension to the TBB service, described in 3.2.3. The touch input was drawn onto an Android Canvas and saved as PNG image files. The application processed various parameters, and saved all of the information from this analysis in CSV files.

4.2.4 Procedure

The study was performed in Dundee University, the University of Lisbon and at the homes of some of the participants, in the district of Viana do Castelo, Portugal. Each session was 30 to 45 minutes long, and participants were evaluated individually. Video recordings were taken of the sessions. . We handled ethics according to the ethical standards of each of the countries; for the study performed in Dundee with motor impaired people, ethics were approved by Newcastle University prior to the study, as seen in Appendix D. The study went according to the script in Appendix C.

The participants were told that the purpose of the study was to collect samples of touch gestures from people with different ages and abilities so as to discern which gestures were feasible for all.

Next, the participants were asked a few questions about their demographics, their touchscreen device experience and their gaming habits. Participants were then informed about the procedure of the study.

Each participant followed the application prompts until they performed all of the gestures multiple times, while the study facilitator used the second communication application to switch between gestures. The application logged the touch interaction in XML files, which we later analysed with an extension of the original application.

4.2.5 Design & Analysis

With the extended Gesture Prompt application described in 4.2.3, and for each touch sequence, which we defined as beginning from the first touch point until all fingers are lifted, we collected various touch interaction parameters. The parameters depended on the gesture being performed, but the basic parameters every gesture analysed were: *Multitouch* (a true or false Boolean that indicated if the gesture was multi touch); *Duration* (measured in milliseconds); *Speed* (measured in pixels per millisecond); *Travelled Distance* (total length of the gesture, measured in pixels); information about DOWN and UP events, which refer to touch down (beginning of gesture) and touch up (finger lift – end of gesture) such as *timestamps*, *x and y coordinates*; *Offset X* and *Offset Y* parameters, which were x and y offsets from the beginning of the gesture to the end of the gesture (x and y offset from first and last touch point); *Interval from the previous gesture* (in milliseconds); gesture *Direction* (up, down, left or right); *Original Path Size*; the *difference between travelled distance and the original path*; *Gesture* (gesture main category); and *Condition* (gesture subcategory).

We then logged information depending on the gesture that was asked of the user. For taps, we logged the *target Diameter*, the *Center X* coordinate, the *Center Y* coordinate, *Smallest X* (left border of the target), *Biggest X* (right border of the target), *Smallest Y* (top border of the target), *Biggest Y* (bottom border of the target), *Touch Down X and Y distance from centre*, *Touch Up X and Y distance from centre*, *average X and Y distance from the centre*, and the *number of times the user touched out of the target*.

For swipes, the same parameters as tap were logged, except these values referred to the initial swipe point where the user was asked to begin the swipe. We also collected *average, minimum and maximum offset from path*, as well as the *total sum offset from the path*. This compared the travelled path to the original Path. The *number of directional changes* was also registered, as well as the *angular offset* of the gesture. Regular drags, one-finger rotation and shapes collected the same information as swipes.

For sequenced taps and swipes, *minimum, maximum and average intervals* were collected as well.

Scribble collected the *number of touch points out of bounds*, the *sum of the distance of touch points out of bounds*, and the *average, minimum and maximum distance out of bounds*. It also collected the *number of directional changes* as well as the *total covered area*, which multiplied the travelled distance by the finger diameter to determine the total scribbled area.

Pinches and Zooms collected *initial and end finger with identifier 0 and 1 distances from the centre of the screen*. It also collected the *initial and end diagonal distance between fingers*, as well as *number of directional changes* for each finger.

We joined the performance of all of the ability groups into each gesture subcategory and evaluated their *Duration, Speed, Travelled Distance, Intervals*, and *Angular Offset* separately with IBM SPSS Statistics 23. We extracted the maximum, minimum, standard deviation, mean, median and modes from the gesture data, and used these values to compare the ability groups with each other and with the game input demands.

4.3 Results

Our goal was to collect, analyse and compare touch interaction data from the different ability groups, and from that assess the feasibility of each gesture, and ultimately assess the playability of each game played in the previous study.

First, we will present the results of the gesture analysis of the four ability groups. Then, we compare these results to the first study's game gesture demands to determine the playability of each.

4.3.1 Measuring user abilities

We logged the execution of the most commonly used gestures in today's games by people with varying abilities. Our goal was to catalogue the differences in gesture performance and to compare this performance to the first study.

Large and medium target taps were performed well by all participants. There was only a single instance of a child missing a medium-sized target. In terms of small targets, participants from every group missed the target a few times. In total, able bodied participants missed once, children missed six times, elderly missed 12 times, and motor impaired participants missed 18 times. The chart below shows the small target misses, taking into account the reduced number of motor impaired participants in comparison to the other groups.

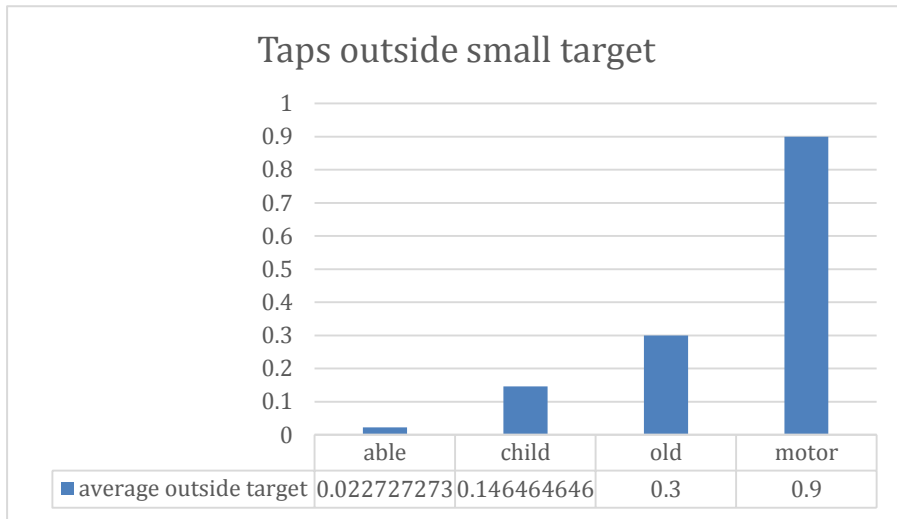


Figure 7 Taps outside small target chart

We will now analyse charts that englobe every gesture and every group. We always use able-bodied performance as the baseline for accurate performance.

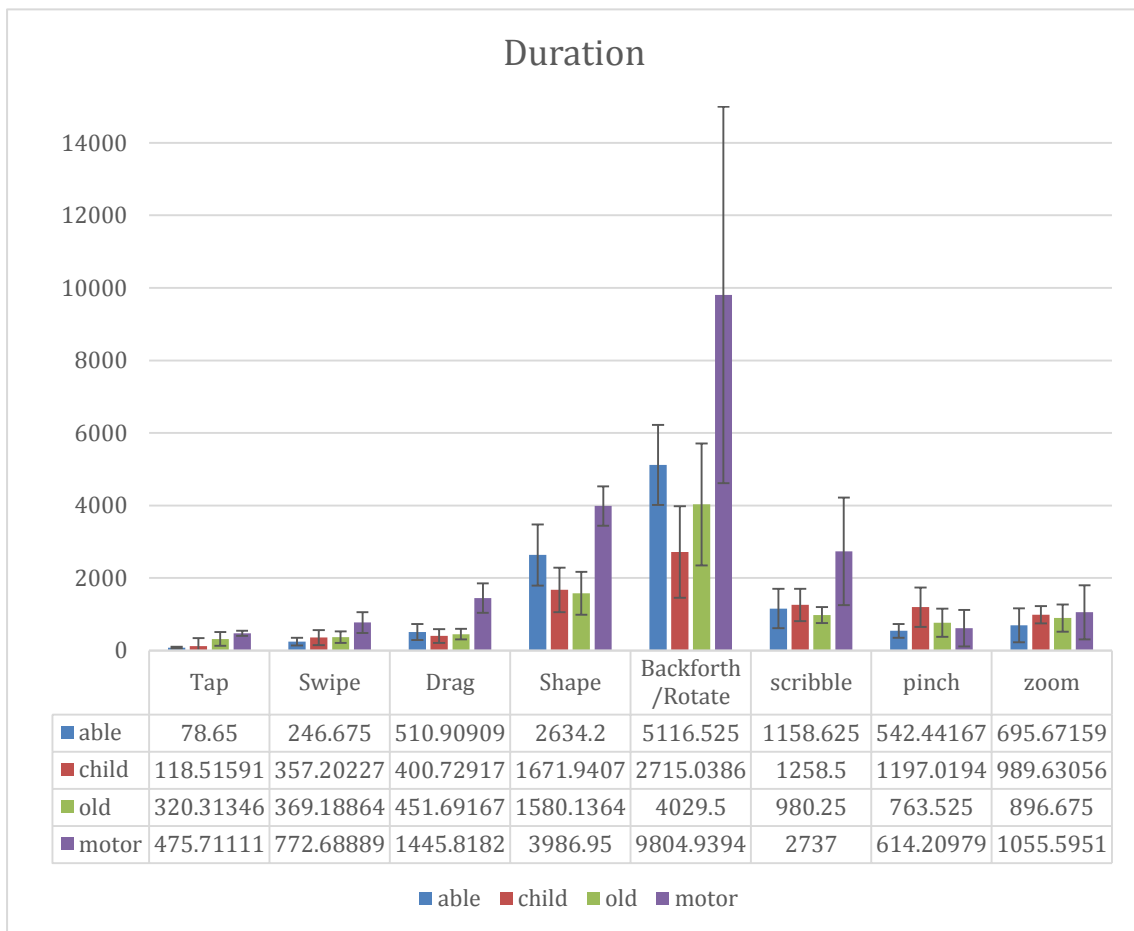


Figure 8 Durations of every gesture performed by every group

As we can see in figure 8, for every gesture, motor impaired participants took the longest to perform them, many times taking over double the time able-bodied participants took. On the other hand, children and elders usually took around the same time as able bodied participants. The exceptions to this were tap, where elders took much longer to perform the gesture, shapes and rotate, where they concluded the gestures in less time, and pinch and zoom, where children and elders had higher duration values. Further ahead we will associate the shapes and rotation to speed and accuracy, to determine if they simply were faster than able-bodied users or if the gestures were performed hastily, affecting the accuracy of the gesture as well.

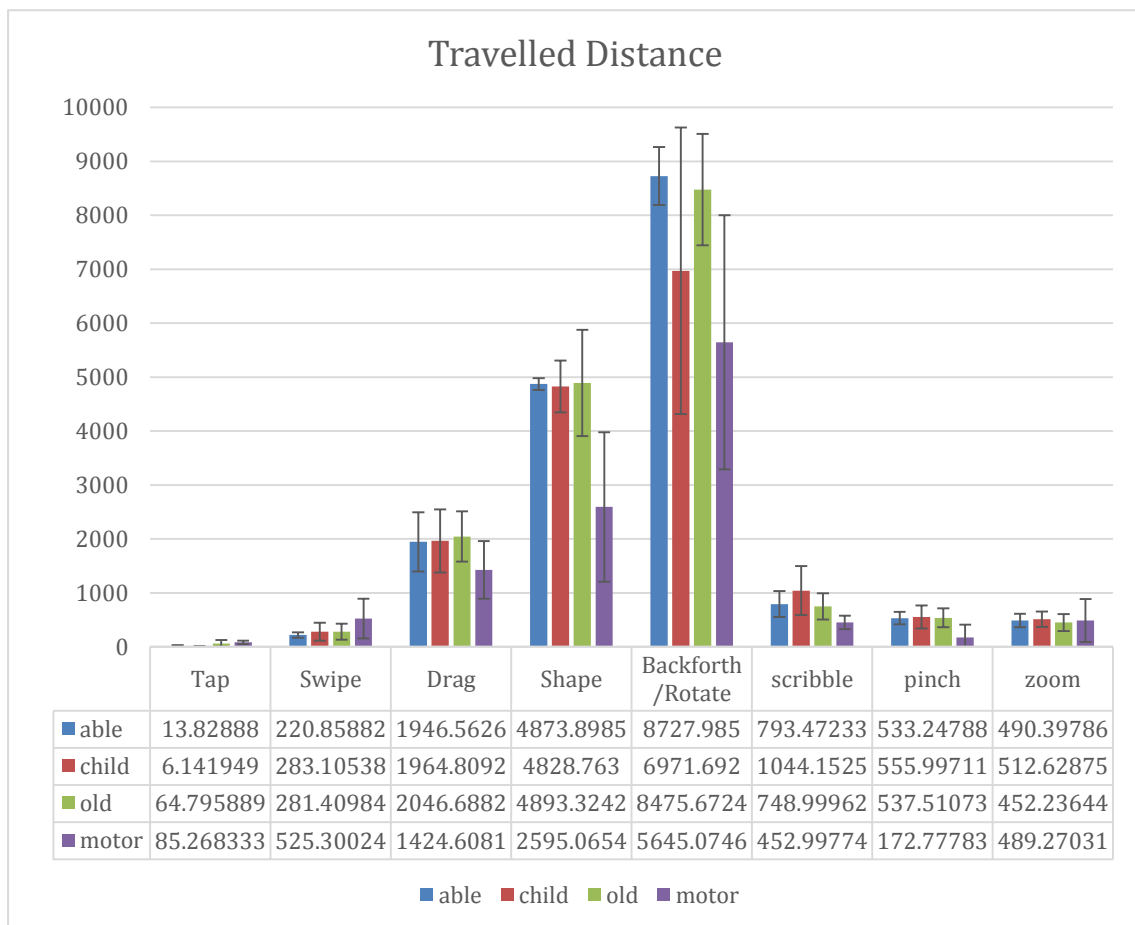


Figure 9 Travelled Distance of every gesture performed by every group

Figure 9 shows differences in average Travelled Distance among gestures. Motor impaired participants had a smaller travelled distance in every gesture except taps and swipes, where the travelled distance was much higher than able bodied participants travelled distance. The other two ability groups were very close to able bodied participants' values, except for tap, where elders had a significantly higher value,

rotation, where children had a much lower value, and scribble, where child had a higher value.

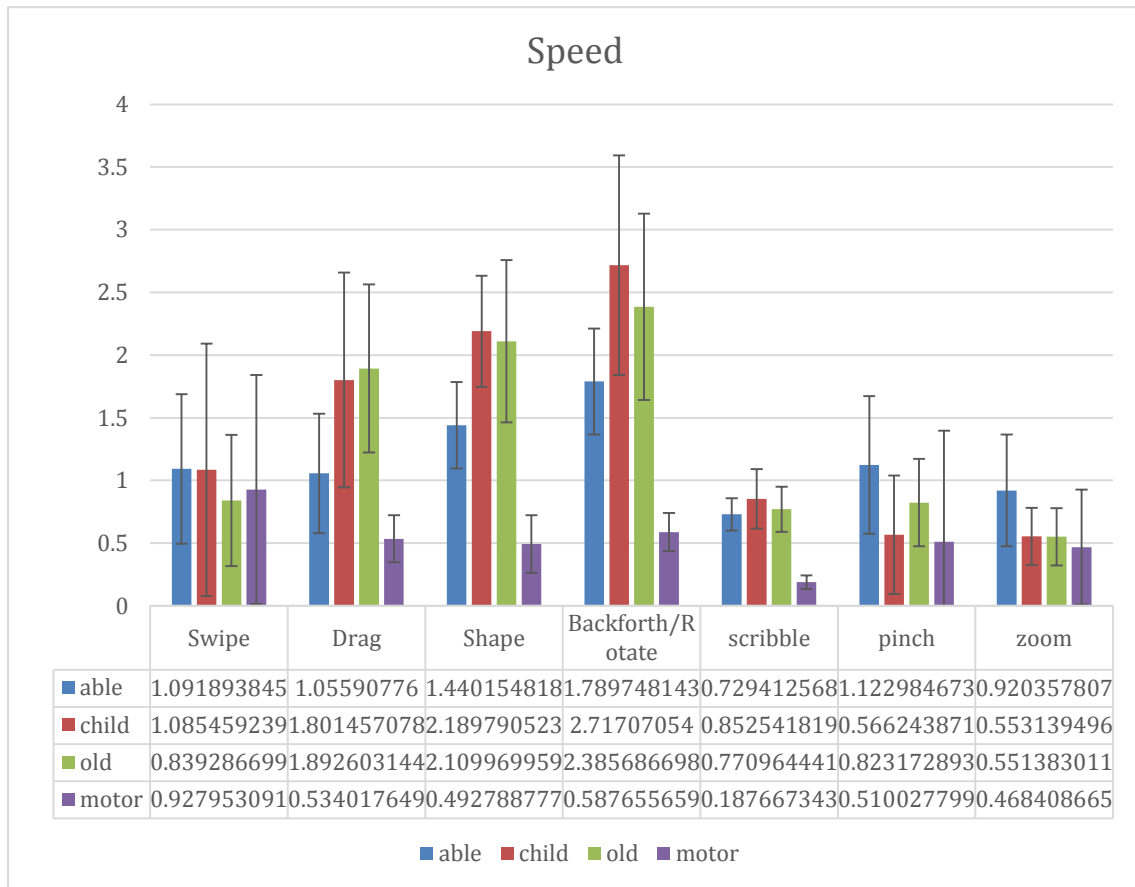


Figure 10 Speed of every gesture performed by every group

Figure 10 shows differences in speed. We can see that motor impaired participants generally always performed gestures the slowest. In swipe, we can see that elders also performed slower, and in drag, shape and rotate both children and elders performed the gesture faster than able bodied participants. For pinch and zoom, elders and children also performed significantly slower than able bodied participants.

We saw earlier how for both shapes and rotation, children and elders were faster and took a shorter time than able bodied participants. We will now compare these to gesture accuracy to determine if they had a better performance in these aspects or if they were simply performing the gesture carelessly.

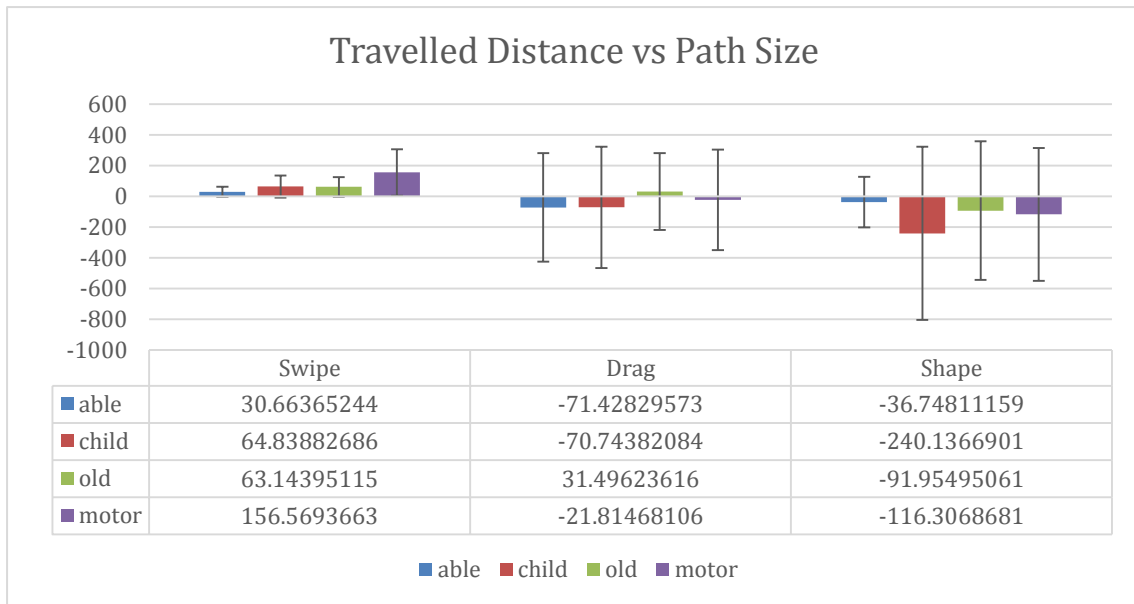


Figure 12 Travelled Distance vs Path Size of swipe, drag and shape

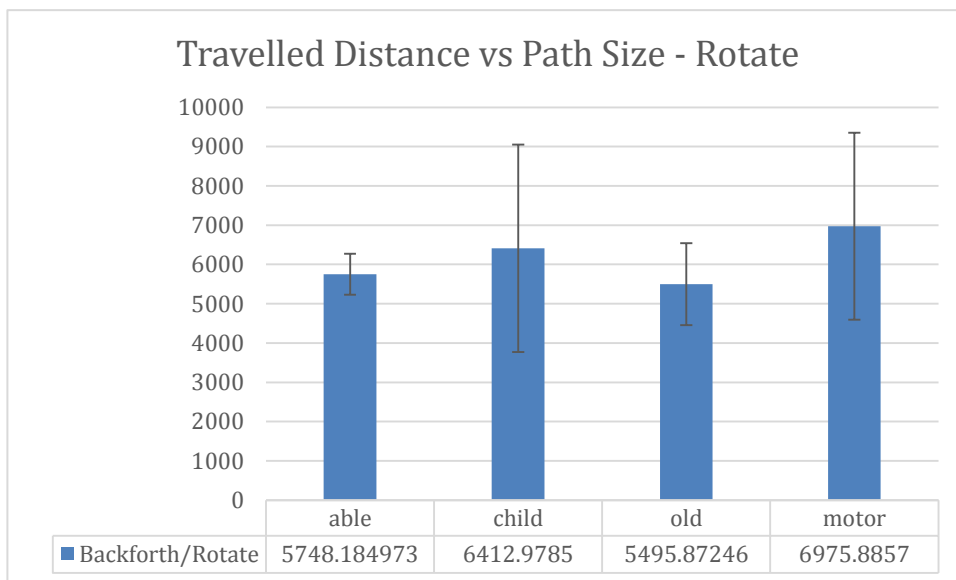


Figure 11 Travelled Distance vs Path Size of rotate

Figures 11 and 12 show the gesture accuracy in terms of the difference between the original path size and the travelled distance. The value is negative if the travelled distance is shorter than the path. As we can observe, in the case of shapes children and elders have a large difference from the original path compared to able-bodied participants, confirming the earlier theory of them having performed the gestures hastily and inaccurately. However, this is not confirmed for rotations, as the difference is similar.

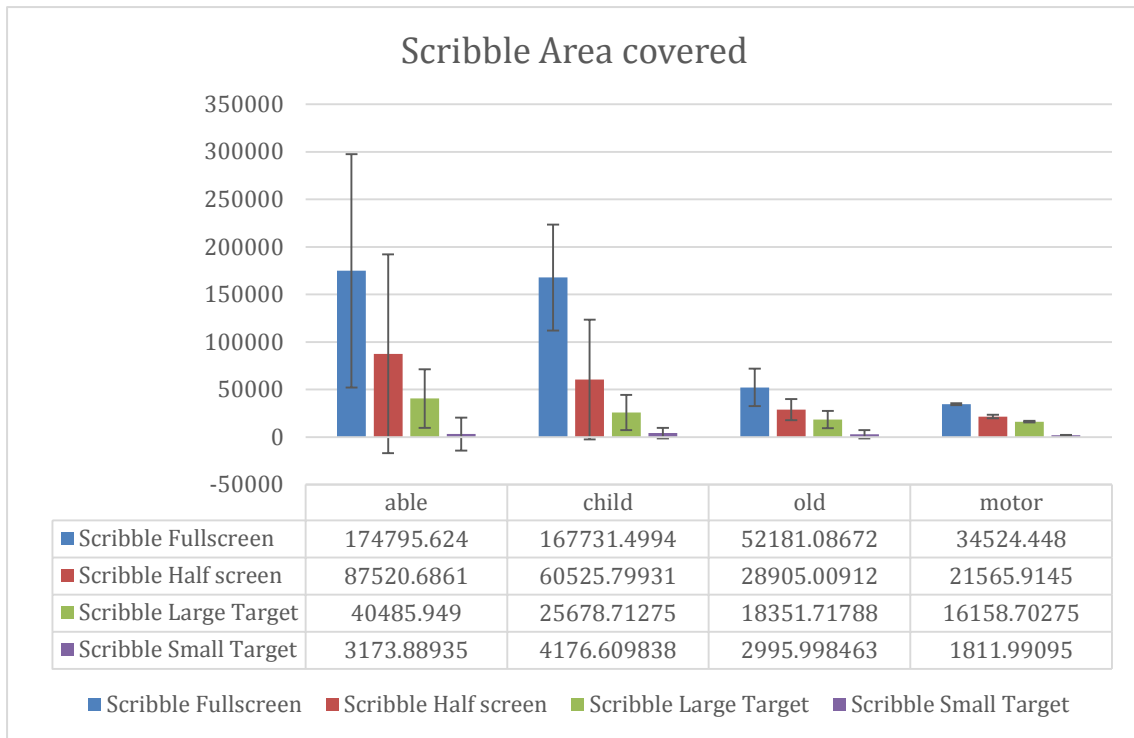


Figure 13 Scribble area covered

Figure 13 shows the differences between areas covered by scribbling gestures. As we can see, for full screen scribble, half screen scribble, and large target scribble, elders and motor-impaired participants have a significantly lower covered area. Children usually cover around the same area as able-bodied participants except for large target scribbling.

Figure 14 shows initial and ending distances between fingers for pinches and spreads.

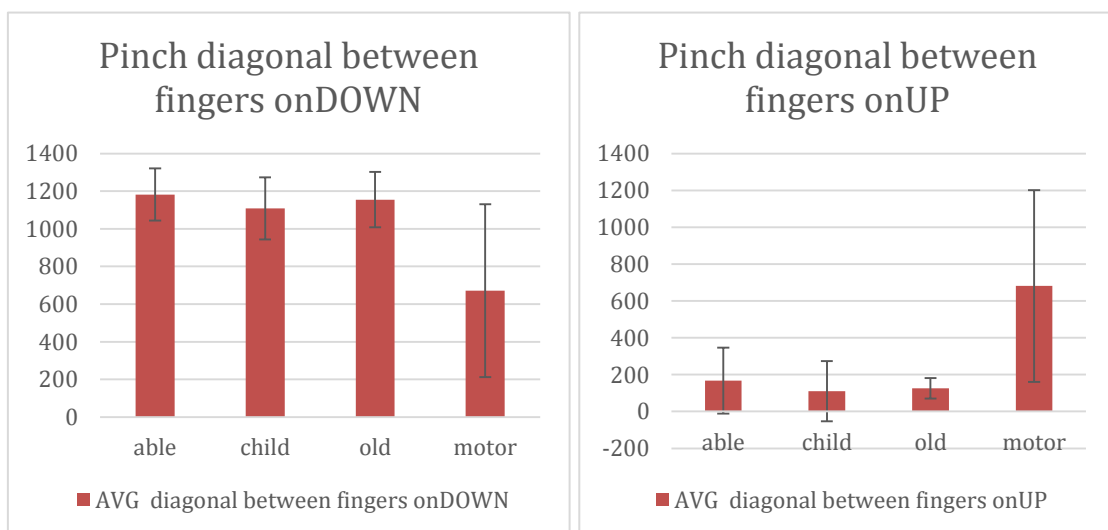


Figure 14 Pinch diagonal onDOWN and onUP

For pinch, the gesture is supposed to begin with a large difference between fingers and end with a small difference. As we can see above, motor impaired participants differ largely from other participants. They begin a pinch with an average of 671px between fingers, and end the gesture with 681px. This shows an inability to perform this gesture by the participants with motor impairments, as the basic requisite for the gesture was not met.

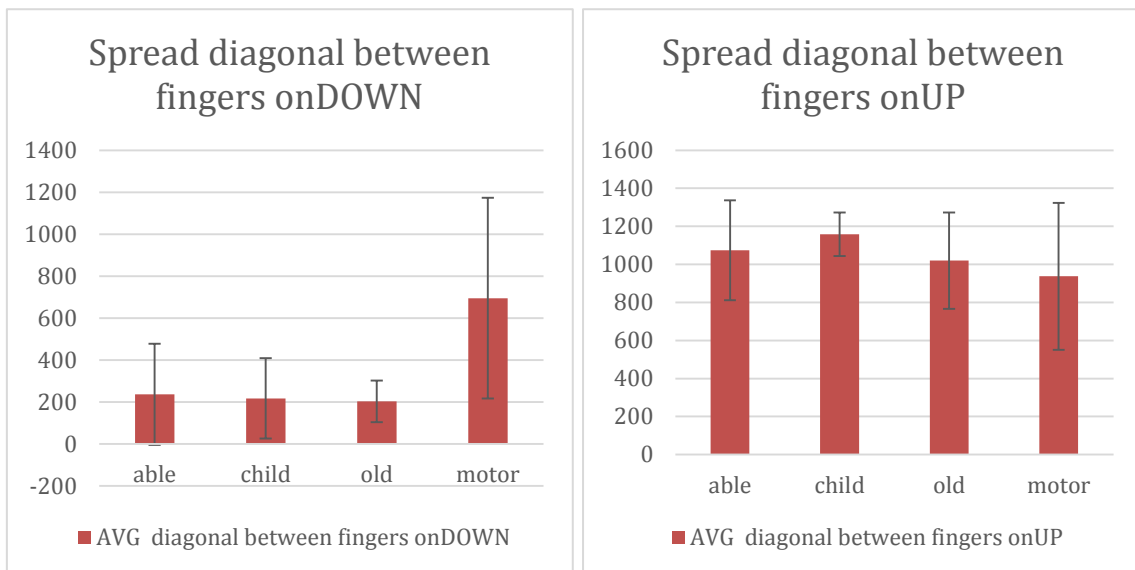


Figure 15 Spread diagonal onDOWN and onUP

For spread gestures, the basic prerequisite is that the gesture begins with fingers closer together, and ends with fingers further apart. In this case, although the initial touch down distance is much larger than other participants, the basic requirement is met as it begins at a distance of 694.66px and ends at 937.58px. However, as observed during the session, only one of the motor impaired participants were able to perform a spread. As can be seen in figure 16, there is a great deal of tremor during the gesture.



Figure 16 Spread performed by motor impaired participant

4.3.2 Comparing abilities and demands

We will now compare the gesture analysis study with the game gesture study. Our goal is to identify which games are playable for all, and which exclude people of varying abilities due to their higher demands. Given the small size of the samples, we chose not to perform statistical analysis and make our conclusions based on anecdotal evidence.

We could immediately exclude pinch gestures for motor impaired participants, as it was shown that they were unable to perform them. For the purposes of this comparison, and given that only 50% of the motor impaired participants were able to perform a spread, we will also consider spread as unfeasible for motor impaired people. This immediately makes Mandala unplayable for this group, as a lot of the gameplay hinges on pinches and spreads.

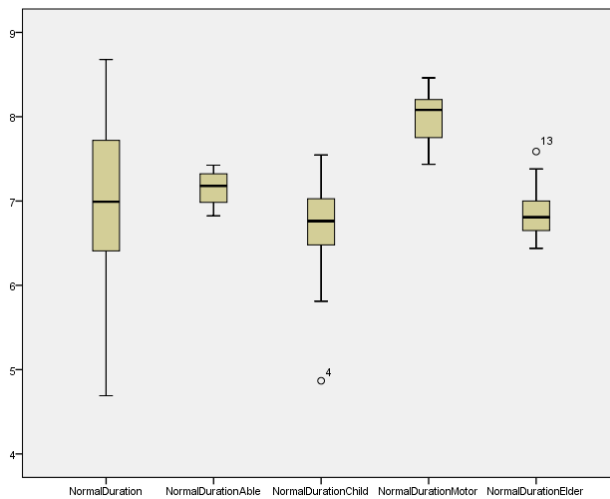


Figure 17 Sample boxplot comparing game demands and the four ability groups

To determine which games are unplayable, we compared the gesture data collected from each game to the gestures performed by the participants of varying abilities. We began by normalising the data by applying an LN technique. Then, we created box plots to visually compare the gestures. Figure 17 shows an example of the evaluation of tap

duration in the game Agar.io. The first variable is the duration of taps performed throughout the game; the

other four are sample tap durations of each ability group, from the gesture evaluation. In this example, the gesture is feasible for every group as they are all within the game's boundaries.

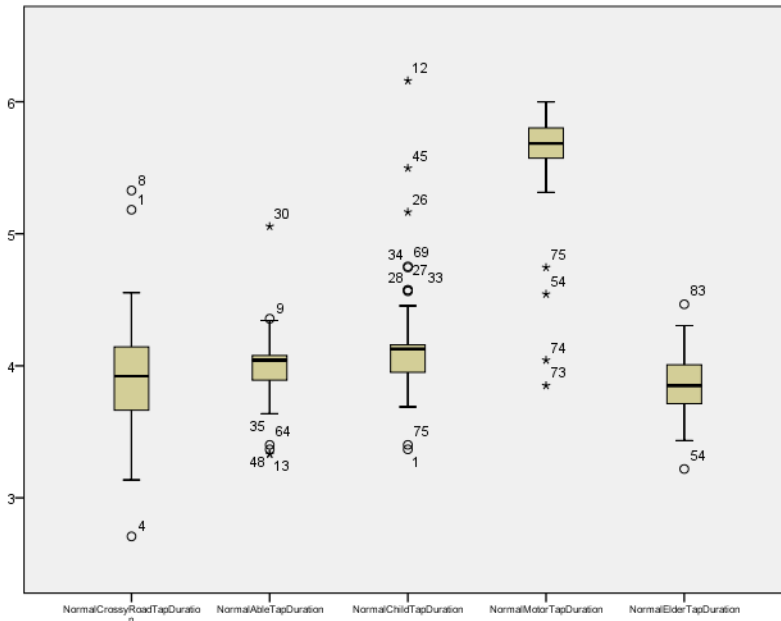


Figure 18 Crossy Road boxplot comparing tap duration game demands with tap durations of ability groups

We found that motor impaired participants’ tap duration exceeded the game boundaries for 15 games. To make a reasonable evaluation, we considered, of those games, which used tap as a primary game gesture, this way excluding games which only needed taps for menu selection, for example. We narrowed the list down to 10 games: Agar.io, Clash of Clans, Color Switch, Cooking Fever, Crossy Road, Geometry Dash, Marvel, Solitaire, Talking Tom and Twist. We decided to analyse these meticulously, using our gameplay experience as factors for evaluation.

Longer taps are not a barrier for Agar.io – the game may interpret it as a drag, but tap and drag have the same effect in the game, therefore not making longer taps a problem. Clash of Clans is a slow paced, target tapping based game which sole gameplay is taps, and is therefore unplayable for motor impaired players. Color Switch, is a game that would be negatively affected by longer taps – it is an interval tapping game, which required quick taps in short intervals. A longer, possibly unrecognized tap would make the player lose control of the game, which makes Color Switch unplayable for motor impaired players.

Cooking Fever is a fast-paced game with timeouts. Taps are used throughout gameplay, and the possibility of an unrecognized tap would negatively affect the player’s performance. Additionally, motor impaired participants’ drags within this game had a speed below the range, which also affects the players overall performance negatively as well. With this, we can conclude that Cooking Fever is also unplayable by motor impaired users.

Crossy Road is a time precision game. A misinterpretation of a tap could mean not moving the character out of harm's way in time. This makes it unplayable for motor impaired players.

Geometry dash cannot afford longer taps; it is a game that uses taps and long presses together to avoid obstacles. The timing of the use of each is crucial for gameplay. Therefore, this game is also considered unplayable by motor-impaired users.

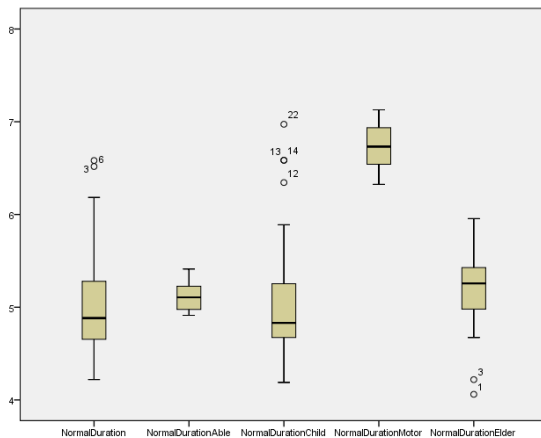


Figure 19 Marvel boxplot comparing swipe duration game demands with swipe durations of ability groups

Marvel is a fast paced fighting game that requires taps and swipes. As well as taps being performed for an extended amount of time, we also saw that swipes performed by motor impaired players had large durations and large intervals compared to the swipes performed in the game. Swipe durations are represented on the chart on the left. Due to this combination of factors, Marvel is also unplayable by motor impaired players.

Solitaire is a slow paced game, for which taps could be interpreted as drags, and in which tapping is a large part of gameplay, making it unplayable. Similarly, Talking Tom also uses taps as a large part of gameplay, excluding this game as well.

Finally, Twist is a very fast paced game in which the misinterpretation of a tap means losing the game. This makes it unplayable for motor impaired users.

In terms of tap intervals, every game had all ability groups within the range.

For swipes, we evaluated duration, speed, travelled distance, intervals and angular offset. Candy crush and Futurama had motor impaired swipe durations above the game ranges, but this did not affect gameplay negatively, as neither game require swiftness.

Swipes performed by motor impaired participants in Pacman and Subway Surfers also had a duration above the game ranges. However, these games are fast paced and require time precision, as well as successive swipes. Due to this, Pacman and Subway Surfers are unplayable for motor-impaired players.

In terms of scribble, shapes and rotation, every player performed within the required range.

4.4 Discussion

We will discuss the results from the gesture analysis study, as well as discuss the game playability analysis.

The gesture analysis study showed that there were performance differences among the ability groups. We used able-bodied participants as the baseline for a correctly performed gesture. Motor impaired participants stood out as generally performing more poorly in comparison to the other groups, and in some cases children and elders performed gestures more hastily than able bodied participants, which meant they performed the gestures quickly but with reduced accuracy. Motor impaired participants were not able to perform pinches, and one of the participants was not able to perform any type of multi touch gesture.

After the first general gesture analysis, we compared the results of this study to the study performed earlier, which evaluated the gameplay of the top 25 games in the Google Play store. We concluded that 48% of the games were unplayable for people with motor impairments: Mandala, Clash of Clans, Color Switch, Cooking Fever, Crossy Road, Geometry Dash, Marvel, Solitaire, Talking Tom, Twist, Pacman and Subway Surfers. This is nearly half of the games that were evaluated, showing that a large change in current games is necessary.

4.5 Summary

We performed a study with people of varying abilities: able bodied, children, elders and motor impaired people. We identified gestures which some of these groups performed more poorly, as well as gestures they were completely unable to perform.

Following this, we compared the results of this study with the results of the previous game demands analysis, and identified 12 games that are unplayable for motor impaired players, making 48% of the games evaluated unplayable for this group.

In the next chapter, we will connect our results to suggest future game design implications, and discuss our accessibility solution, which is based on the study findings.

Chapter 5

Game Design Implications

In this chapter, we frame the design space by recapping previous chapters, we identify game design implications from our study results, and we propose our accessibility solution and present our Annotation Tool prototype.

5.1 Context

We performed two studies to determine the full scope of the issue of accessibility of touchscreen games for underrepresented players.

In our first study, we collected input data from 25 able bodied participants, who were asked to play games from the top 25 games in the Google Play store. We created a catalogue of game gesture demands from the results of this study, and identified the most used gestures in current games – Taps (including long presses), Swipes (directional swipes and sequenced swipes), Drags (including one finger rotation, shape tracing and scribbling), Pinches and Spreads.

In our second study, we asked people of varying abilities – able bodied, motor impaired, children and elders - to perform gestures, based on the results of the previous study. Able bodied input served as our baseline to compare and determine gesture difficulties of other ability groups.

In this study, we concluded that motor impaired participants performed every gesture longer than the other ability groups, with a few exceptions, the largest being that both elders and children took longer to perform pinches and spreads. Elders and children performed shapes and rotations more hastily than the others, meaning that although these gestures had a shorter duration, they also had a higher speed and lower accuracy. We saw that motor impaired participants had a smaller travelled distance in every gesture except taps and swipes, where the travelled distance was much higher than able bodied participants. Motor impaired participants performed gestures the

slowest. Elders also performed swipes slowly, and both children and elders performed pinch and spread slowly. For scribble, elders and motor impaired participants have a significantly lower covered area than the other groups. Motor impaired players were unable to perform pinches, and 50% were unable to perform spread.

We then compared the results of both studies so as to determine if the people of varying abilities met the game demands found in the first study. We concluded that 48% of the games had their demands too high for motor impaired players. These games were unfeasible either because the motor impaired participant's taps took too long, or because their swipes took too long. The other ability groups performed some gestures differently from able bodied players, but every game was still feasible for them.

5.2 Implications for Accessible Gaming

The game design implications that we can conclude from these results are:

Avoid pop up advertisements in games. Standard exit buttons for these advertisements are only 50mmx50mm; 9 out of 10 times motor impaired players will miss this button, leading to them opening advertisements they do not want to and getting more and more frustrated while playing your game. If need be, use top bar or bottom bar advertisements - some people cannot click the tiny button!

Do not assume every player is at same baseline. Most games assume the player is able bodied; provide a way to indicate the contrary. A diagnostic test can be given to the player – disguised as a first level tutorial for example – so as to evaluate the player's gesture performance. Save the results, and automatically adapt the input receivers according to the player's abilities.

Allow input customization so player can choose options that they feel makes their gameplay more comfortable. For example, allow to adjust standard gesture duration, such as how long the user takes to perform a tap, swipe etc., thus avoiding mistaken gesture identifications. This implication is confirmed in our research on the topic as well.

Offer flexible game speed. Fast paced games cannot be played by all. Allow an adjustable mechanic that slows or accelerates the game pace. Design games so that performing gestures at a growing speed is not the sole gameplay goal – maintain game challenge regardless of pace!

Make scribble area thresholds smaller. Games which include scribbling over a certain area until it is completely covered need to lessen the area covered demand; some games do not allow the player to advance until every inch of the

area to fill in is scribbled over. Some players may be unable to cover the entire area, so allow a lower threshold if player is taking too long to cover the area.

Provide alternatives to multi touch gestures. Some players are completely unable to perform multi touch gestures. Games should be designed to offer alternatives to these gestures, such as a button which, on activation, performs the gesture wherever the user taps. This implication is confirmed in our research on the topic as well.

5.3 Human-powered Adaptation of Games

We propose a human-powered, system-wide accessibility solution. Due to the lack of accessibility implementation in games, a system-wide solution is necessary so as to make the solution available for any game, regardless of if the game developer implemented accessibility or not.

A specialised algorithm for game touch accessibility is impossible to create with the current state of games due to the fact that most games are a black box: even with advanced accessibility user interface element detection, game elements cannot be detected. Therefore, a less automatic and more hands-on approach is necessary. One way to provide intelligent, personalized accessibility to users is to use data created manually by humans. The main concept of our proposed solution is to crowdsource gameplay data from expert players, to later use during motor impaired gameplay to aid them in difficult or impossible in-game situations.

This framework would, in a first phase, be used by able bodied game experts (defined as someone who has passed a set number of levels in the game) to record gameplay sessions. In practical terms, it would use our extended TBB service, mentioned in 3.2.3, which is an Android accessibility service that records user input, both by logging their touch points and by recording a low frame-rate video of the screen.

Once the game session is complete, the service would ask the user if they want to create a data set, or if they would like to publish the session to the community so someone else can use to create the data set. This data set would be created by an Annotation Tool, an interface similar to a video editor, which the experts will use to create a data set of game actions. Here we define game actions as actions like “Jump”, “Move Forward”, etc.; a single gesture or an agglomerate of gestures that perform a certain game action. The expert navigates the recorded video and views the gestures he performed, and with this chooses which gestures are a part of the game actions.

Once the data set is complete, the expert player can choose to post it to the community. This means that it would be saved in the cloud and accessible by all users to edit, use and rate.

Another important reason for the gameplay data to be crowdsourced is to make this framework scalable to any and all games in the Play store.

On the other side of this framework, the motor impaired players would be able to visualize every data set created for any game they want to play, ordered and refined by community ratings. For further accuracy in input adaptation, the user would be required to create a user model, where they specify their abilities in detail. This user model can also be created automatically, by prompting them with a diagnostic test to evaluate their gesture performance.

With the data set chosen, together with the user model, the game input would be adapted to the player's needs by injecting the expert player's input. This input can be calibrated to the user, and specific input or switches can be associated to game gestures. To aid this adaptation, interface adaptation can be used as well in the form of button overlays.

5.4 Annotation Tool Prototype

As a proof of concept, we decided to create a prototype of the Annotation Tool. In a first phase, we designed the application by creating interface wireframes for both landscape and portrait orientation. Our main idea was to make it similar to a video editor, so as to allow the user to visualize their previous gameplay session and the dataset that was being created. Figure 20 shows some of the initial interface designs.

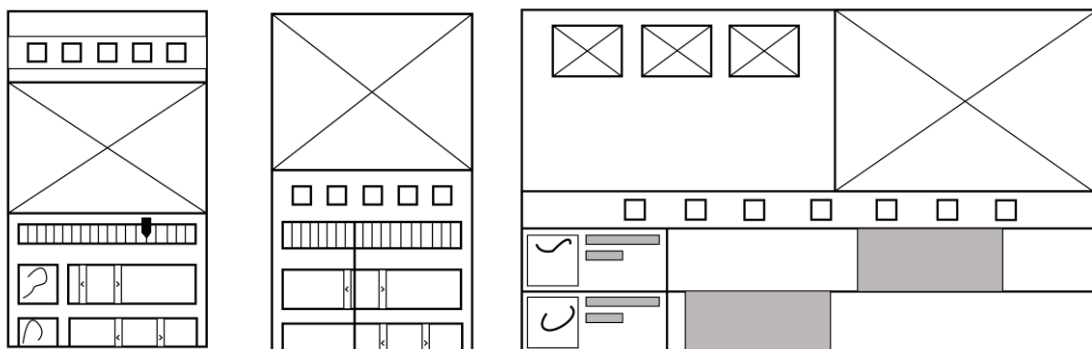


Figure 20 Initial Annotation Tool interface designs

Figure 21 shows a screenshot of the prototype application. The application we developed was an extension of the TBB accessibility service mentioned in 3.2.3. TBB

automatically begins recording the gameplay session once it detects a non-system application opening. Once it detects the onPause event from the application, TBB assumes that the expert player finished the gameplay session, and prompts the player to

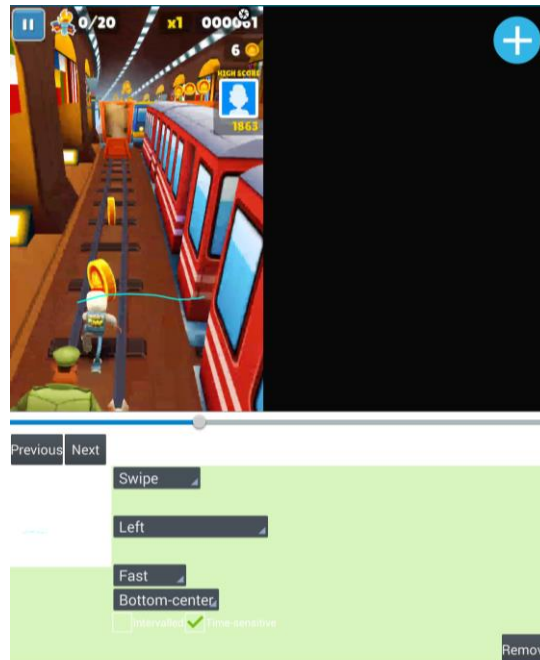


Figure 21 Annotation Tool Prototype screencap

open the Annotation Tool with the recorded gameplay session.

This application loads the video recorded from the session onto an Android Video View. An invisible canvas overlays this video; as the video is replayed or navigated manually, the touch points recorded during the session are drawn onto the canvas. This way, the user can visualize the interactions in real time.

The user has the option to add the current touch sequence to their data set. Once they click the add button, a dialog prompts them to choose an action name, either choosing from a pre-existing name or adding a new one. This action is then added to the data set; the user is able to visualize which action is which due to each being drawn next to the gesture information. The user can optionally refine the action, by choosing options such as position on screen and direction. These refinements will add metadata information to the action, which can later be useful for action cataloguing. Once the user is satisfied with the created data set, it is saved as an XML file.

This data set is similar to the catalogue we manually created in chapter 3 as it logs the game input demands and most used gestures throughout gameplay. Our solution goes through this process automatically and creates its own “game demands catalogue”.

Chapter 6

Conclusions and Future Work

We have performed two studies. In the first study, we analysed the demands of current mobile games and, with the results, created a gesture catalogue with the most commonly used gestures in these games. In the second study, we evaluated the input aptitudes of people of various abilities: motor-impaired, elders, children and able bodied.

With these two studies, we were able to determine the feasibility of current games by contrasting the gesture aptitude results with each game's input demands, and saw that nearly half of the games were unplayable to motor impaired people.

With the findings from these studies and an in-depth analysis of previous work on the topic, we propose a system-wide, crowd-sourcing solution to aid motor impaired players to play any game. As a proof of concept, we designed and implemented a prototype of a part of this system.

It can be argued that the best solution would be to create an application-specific solution instead. However, it has been shown that, despite efforts to create libraries and easy-to-use solutions for the developers to include in their software, they continue to not implement accessibility in most games. Many can simply be unaware of the issue at hand, but it is seen as a waste of time and resources for some. Gradually the game industry mind set is being changed, as the disabled community represent a large number of potential customers. But, at the moment, most games are not accessible to all.

Despite being a large scale project, englobing a subsequent online community and complex input and interface adaptation on the motor impaired person's side, we have all of the parts to create the whole, and therefore know that it is a viable solution. We have functional input logging and injection with TBB and a working first prototype of the Annotation Tool. We have seen in the first stages of TBB extension that interface element overlays during gameplay is possible and does not interrupt the gameplay. Previous work shows that user models are proven to be effective for user-specific adaptation [33].

6.1 Limitations

The proposed solution offers accessibility to any application, as long as there is someone that donates their data. This means that games can only be made accessible if there is someone willing to do so and that, therefore, this framework is very dependent on others. As well as this, our solution does not cover all of our design implications: it does not address the issue of slowing down fast paced games while still maintaining the game challenge. Finally, despite expert gamers donating the most important gestures to play the game, many gestures can depend on the current context of the game. Some games change their required input completely depending on game level or situation, and it is difficult to predict such changes.

6.2 Future Work

In the future, this solution can be completely implemented by joining the various adaptation techniques mentioned above, and adapting it to a cloud platform to create the Data Donors community. Some of the suggestions toward improving the proposed solution, as well as countering the limitations mentioned in 6.1, are the following:

- *Gamify the experience.* To motivate expert players to donate to as many games as possible, and therefore expand the range of our platform, users that donate their data to the platform can be awarded experience points and achievements to their Google Play account, and leader boards can be created within the community to elicit competition.
- *Automate Annotation* with gesture recognition, so as to accelerate the annotation process.
- *System-wide deceleration of games* would be a step in the right direction towards making fast paced games playable for all; the possibility of this needs to be tested.
- *Intelligent image recognition* to aid in identifying game context; annotation could be extended to include “situation training”, in which the user can choose frames of situations in which a particular gesture is to be used. Machine learning techniques could be used to make the image recognition system learn with each annotated situation.
- *Evaluation on a social level.* A study to explore how comfortable motor impaired people would be with a system such as this, and how they feel being aided in this way by others.

Bibliography

- [1] Alphabet. 2016. *Alphabet Announces Fourth Quarter and Fiscal Year 2015 Results*. [ONLINE] Available at: https://abc.xyz/investor/news/earnings/2015/Q4_google_earnings/index.html. [Accessed August 2016].
- [2] International Data Corporation. 2015. *Smartphone OS Market Share, 2015 Q2*. [ONLINE] Available at: <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>. [Accessed October 2015].
- [3] GSMArena. 2011. *Mobile phone usage report 2011*. [ONLINE] Available at: http://www.gsmarena.com/mobile_phone_usage_survey-review-592.php. [Accessed October 2015].
- [4] Lucas Kittmer. 2011. *The Advantages of Mobile Phone Games*. [ONLINE] Available at: http://www.ehow.com/info_8392639_advantages-mobile-phone-games.html. [Accessed October 2015].
- [5] Shaun K. Kane, Chandrika Jayant, Jacob O. Wobbrock, and Richard E. Ladner. 2009. Freedom to roam: a study of mobile device adoption and accessibility for people with visual and motor disabilities. In Proceedings of the 11th international ACM SIGACCESS conference on Computers and accessibility (Assets '09). ACM, New York, NY, USA, 115-122. DOI=<http://dx.doi.org/10.1145/1639642.1639663>
- [6] Lee Garber. 2013. Game Accessibility: Enabling Everyone to Play. *Computer* 46, 6 (June 2013), 14-18. DOI=<http://dx.doi.org/10.1109/MC.2013.206>
- [7] Sebastian Aced Lopez, Fulvio Corno, and Luigi De Russis. 2015. GNomon: Enabling Dynamic One-Switch Games for Children with Severe Motor Disabilities. In Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '15). ACM, New York, NY, USA, 995-1000. DOI=[10.1145/2702613.2732802](http://doi.acm.org/10.1145/2702613.2732802)

- [8] Lauren R. Milne, Cynthia L. Bennett, and Richard E. Ladner. 2013. VBGhost: a braille-based educational smartphone game for children. In Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '13). ACM, New York, NY, USA, Article 75, 2 pages. DOI=<http://dx.doi.org/10.1145/2513383.2513396>
- [9] Dimitris Grammenos, Anthony Savidis, and Constantine Stephanidis. 2009. Designing universally accessible games. *Comput. Entertain.* 7, 1, Article 8 (February 2009), 29 pages. DOI=<http://dx.doi.org/10.1145/1486508.1486516>
- [10] Shari Trewin, Mark Laff, Vicki Hanson, and Anna Cavender. 2009. Exploring Visual and Motor Accessibility in Navigating a Virtual World. *ACM Trans. Access. Comput.* 2, 2, Article 11 (June 2009), 35 pages. DOI=<http://dx.doi.org/10.1145/1530064.1530069>
- [11] Bei Yuan, Eelke Folmer, and Frederick C. Harris, Jr. 2011. Game accessibility: a survey. *Univers. Access Inf. Soc.* 10, 1 (March 2011), 81-100. DOI=<http://dx.doi.org/10.1007/s10209-010-0189-5>
- [12] John R. Porter and Julie A. Kientz. 2013. An empirical study of issues and barriers to mainstream video game accessibility. In Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '13). ACM, New York, NY, USA, Article 3, 8 pages. DOI=<http://dx.doi.org/10.1145/2513383.2513444>
- [13] Kevin Bierre, Jonathan Chetwynd, Barrie Ellis, D. Michelle Hinn, Stephanie Ludi, Thomas Westin. 2005. Game not over: Accessibility issues in video games.
- [14] The Able Gamers Foundation. 2012. *Includification*. [ONLINE] Available at: <https://www.udemy.com/accessibility-features-on-android>. [Accessed October 2015].
- [15] Sami Rahman. 2013. *Accessibility Features on Android*. [ONLINE] Available at: <https://www.udemy.com/accessibility-features-on-android>. [Accessed October 2015].
- [16] Maia Naftali and Leah Findlater. 2014. Accessibility in context: understanding the truly mobile experience of smartphone users with motor impairments. In Proceedings of the 16th international ACM SIGACCESS conference on Computers & accessibility (ASSETS '14). ACM, New York, NY, USA, 209-216. DOI=<http://dx.doi.org/10.1145/2661334.2661372>

- [17] Yoojin Kim, Nita Sutreja, Jon Froehlich, and Leah Findlater. 2013. Surveying the accessibility of touchscreen games for persons with motor impairments: a preliminary analysis. In Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '13). ACM, New York, NY, USA, Article 68, 2 pages. DOI=<http://dx.doi.org/10.1145/2513383.2513416>
- [18] Lisa Anthony, YooJin Kim, and Leah Findlater. 2013. Analyzing user-generated YouTube videos to understand touchscreen use by people with motor impairments. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13). ACM, New York, NY, USA, 1223-1232. DOI=<http://dx.doi.org/10.1145/2470654.2466158>
- [19] Hugo Nicolau, Tiago Guerreiro, Joaquim Jorge, and Daniel Gonçalves. 2014. Mobile touchscreen user interfaces: bridging the gap between motor-impaired and able-bodied users. *Univers. Access Inf. Soc.* 13, 3 (August 2014), 303-313. DOI=<http://dx.doi.org/10.1007/s10209-013-0320-5>
- [20] Lisa Anthony, Quincy Brown, Jaye Nias, Berthel Tate, and Shreya Mohan. 2012. Interaction and recognition challenges in interpreting children's touch and gesture input on mobile devices. In Proceedings of the 2012 ACM international conference on Interactive tabletops and surfaces (ITS '12). ACM, New York, NY, USA, 225-234. DOI=<http://dx.doi.org/10.1145/2396636.2396671>
- [21] Stephen Vickers, Howell Istance, and Michael J. Heron. 2013. Accessible gaming for people with physical and cognitive disabilities: a framework for dynamic adaptation. In CHI '13 Extended Abstracts on Human Factors in Computing Systems (CHI EA '13). ACM, New York, NY, USA, 19-24. DOI=<http://dx.doi.org/10.1145/2468356.2468361>
- [22] Yu Zhong, Astrid Weber, Casey Burkhardt, Phil Weaver, and Jeffrey P. Bigham. 2015. Enhancing Android accessibility for users with hand tremor by reducing fine pointing and steady tapping. In Proceedings of the 12th Web for All Conference (W4A '15). ACM, New York, NY, USA, Article 29, 10 pages. DOI=<http://dx.doi.org/10.1145/2745555.2747277>
- [23] Aravind Machiry, Rohan Tahlilani, and Mayur Naik. 2013. Dynodroid: an input generation system for Android apps. In Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering (ESEC/FSE 2013). ACM, New York, NY, USA, 224-234. DOI=<http://dx.doi.org/10.1145/2491411.2491450>

- [24] Lorenzo Gomez, Iulian Neamtiu, Tanzirul Azim, and Todd Millstein. 2013. RERAN: timing- and touch-sensitive record and replay for Android. In Proceedings of the 2013 International Conference on Software Engineering (ICSE '13). IEEE Press, Piscataway, NJ, USA, 72-81.
- [25] Taeyeon Ki, Satyaditya Munipalle, Karthik Dantu, Steven Y. Ko, and Lukasz Ziarek. 2014. Poster: Retro: an automated, application-layer record and replay for android. In Proceedings of the 12th annual international conference on Mobile systems, applications, and services (MobiSys '14). ACM, New York, NY, USA, 373-373. DOI=<http://dx.doi.org/10.1145/2594368.2601453>
- [26] Karim Said and Shaun K. Kane. 2013. Button blender: remixing input to improve video game accessibility. In CHI '13 Extended Abstracts on Human Factors in Computing Systems (CHI EA '13). ACM, New York, NY, USA, 43-48. DOI=<http://dx.doi.org/10.1145/2468356.2468365>
- [27] André Rodrigues and Tiago Guerreiro. 2014. SWAT: Mobile System-Wide Assistive Technologies. In Proceedings of the 28th International BCS Human Computer Interaction Conference on HCI 2014 - Sand, Sea and Sky - Holiday HCI (BCS-HCI '14). BCS, UK, 341-346.
- [28] Lisa Anthony and Jacob O. Wobbrock. 2012. \$N-protractor: a fast and accurate multistroke recognizer. In Proceedings of Graphics Interface 2012 (GI '12). Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 117-120.
- [29] Mark Joselli and Esteban Clua. 2009. gRmobile: A Framework for Touch and Accelerometer Gesture Recognition for Mobile Games. In Proceedings of the 2009 VIII Brazilian Symposium on Games and Digital Entertainment (SBGAMES '09). IEEE Computer Society, Washington, DC, USA, 141-150. DOI=<http://dx.doi.org/10.1109/SBGAMES.2009.24>
- [30] Hao Lü and Yang Li. 2012. Gesture coder: a tool for programming multi-touch gestures by demonstration. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12). ACM, New York, NY, USA, 2875-2884. DOI=<http://dx.doi.org/10.1145/2207676.2208693>
- [31] Kenrick Kin, Björn Hartmann, Tony DeRose, and Maneesh Agrawala. 2012. Proton: multitouch gestures as regular expressions. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12). ACM, New York, NY, USA, 2885-2894. DOI=<http://dx.doi.org/10.1145/2207676.2208694>

- [32] Werner Kurschl, Mirjam Augstein, Holger Stitz, Peter Heumader, and Claudia Pointner. 2013. A User Modelling Wizard for People with Motor Impairments. In *Proceedings of International Conference on Advances in Mobile Computing & Multimedia* (MoMM '13). ACM, New York, NY, USA, Pages 541, 10 pages. DOI=<http://dx.doi.org/10.1145/2536853.2536860>
- [33] Kyle Montague. 2012. Interactions speak louder than words: shared user models and adaptive interfaces. In *Adjunct proceedings of the 25th annual ACM symposium on User interface software and technology* (UIST Adjunct Proceedings '12). ACM, New York, NY, USA, 39-42. DOI=<http://dx.doi.org/10.1145/2380296.2380315>
- [34] Krzysztof Z. Gajos, Jacob O. Wobbrock, and Daniel S. Weld. 2008. Improving the performance of motor-impaired users with automatically-generated, ability-based interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '08). ACM, New York, NY, USA, 1257-1266. DOI=<http://dx.doi.org/10.1145/1357054.1357250>
- [35] Korozi, M., Leonidis, S., Margetis, G., & Stephanidis, C. (2009). MAID: a Multi-platform Accessible Interface Design Framework. In *Proceedings of the 5th International Conference on Universal Access in Human-Computer Interaction. Part III: Applications and Services (UAHCI'09)* (pp. 725-734). Berlin: Springer-Verlag. DOI= http://dx.doi.org/10.1007/978-3-642-02713-0_77
- [36] Luke Wroblewski. 2010. *LukeW | Touch Gesture Reference Guide*. [ONLINE] Available at: <http://www.lukew.com/ff/entry.asp?1071>. [Accessed 21 January 2016].
- [37] Aditya Ponnada and Ajaykumar Kannan. 2012. Evaluation of mobile games using playability heuristics. In *Proceedings of the International Conference on Advances in Computing, Communications and Informatics* (ICACCI '12). ACM, New York, NY, USA, 244-247. DOI=<http://dx.doi.org/10.1145/2345396.2345437>
- [38] Google Play. 2016. *Top Games - Android Apps on Google Play*. [ONLINE] Available at: https://play.google.com/store/apps/category/GAME/collection/topselling_free. [Accessed 1 March 2016].
- [39] Dimitris Grammenos, Anthony Savidis, Yannis Georgalis, and Constantine Stephanidis. 2006. Access invaders: developing a universally accessible action game. In *Proceedings of the 10th international conference on Computers Helping*

- People with Special Needs* (ICCHP'06), Klaus Miesenberger, Joachim Klaus, Wolfgang L. Zagler, and Arthur I. Karshmer (Eds.). Springer-Verlag, Berlin, Heidelberg, 388-395. DOI=http://dx.doi.org/10.1007/11788713_58
- [40] Sri Kurniawan, Murni Mahmud, and Yanuar Nugroho. 2006. A study of the use of mobile phones by older persons. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '06). ACM, New York, NY, USA, 989-994. DOI=<http://dx.doi.org/10.1145/1125451.1125641>
- [41] Andrei Dumitrescu. 2013. *DICE: Battlefield 4 Will Have More Accessibility to Compete with Call of Duty: Ghosts*. [ONLINE] Available at: <http://news.softpedia.com/news/DICE-Battlefield-4-Will-Have-More-Accessibility-to-Compete-with-Call-of-Duty-Ghosts-388544.shtml>. [Accessed August 2016].
- [42] GameSpot. 2016. *Uncharted 4's Accessibility Options Inspired by Input of Disabled Gamer*. [ONLINE] Available at: <http://www.gamespot.com/articles/uncharted-4s-accessibility-options-inspired-by-inp/1100-6439983/>. [Accessed August 2016].
- [43] Office for National Statistics. 2016. *Internet access - households and individuals*. [ONLINE] Available at: <http://www.ons.gov.uk/peoplepopulationandcommunity/householdcharacteristics/homeinternetandsocialmediausage/datasets/internetaccesshouseholdsandindividualsreferencetables>. [Accessed August 2016].
- [44] Australian Bureau of Statistics. 2012. *Children's Participation in Cultural and Leisure Activities*. [ONLINE] Available at: <http://www.abs.gov.au/ausstats/abs@.nsf/Products/4901.0~Apr+2012~Main+Features~Internet+and+mobile+phones?OpenDocument>. [Accessed August 2016].
- [45] Big Fish Games. 2015. *2015 Video Game Statistics & Trends: Who's Playing What & Why?* [ONLINE] Available at: <http://www.bigfishgames.com/blog/2015-global-video-game-stats-whos-playing-what-and-why/>. [Accessed August 2016].
- [46] Missa Gallivan. 2014. *Why Video Games Are So Important*. [ONLINE] Available at: <http://www.alpinevalleyschool.com/2014/06/why-video-games-are-so-important/>. [Accessed August 2016].
- [47] Jaime Carvalho, Tiago Guerreiro, Luís Duarte, and Luís Carriço. 2012. Audio-based puzzle gaming for blind people. In *Proceedings of the Mobile Accessibility Workshop at MobileHCI (MOBACC)*.

- [48] Kyle Montague, André Rodrigues, Hugo Nicolau, and Tiago Guerreiro. 2015. TinyBlackBox: Supporting Mobile In-The-Wild Studies. In *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility* (ASSETS '15). ACM, New York, NY, USA, 379-380. DOI=<http://dx.doi.org/10.1145/2700648.2811379>

Appendix

Appendix A – Game Input Demands Catalogue

Tap

		8 Ball Pool	Agar.io	Altos Adventure	Candy Crush
Duration	Mean	83.5956	76.366	117.1627	64.5645
	Median	61	66	103	51.5
	Std. Deviation	80.7737	54.18401	66.86046	59.94445
	Minimum	16	12	15	15
	Maximum	460	467	473	458
Travelled Distance	Mean	3.1083	7.3701	24.5446	3.3699
	Median	0	0	0	0
	Std. Deviation	8.20592	11.63385	107.06842	10.34102
	Minimum	0	0	0	0
	Maximum	42.2	48.96	1,343.41	46.74
Intervals	Mean	3,294.3154	1,423.8512	3,369.2335	6,730.8390
	Median	445.5	599	2175	2502
	Std. Deviation	7,992.10005	3,471.75953	3,567.36076	14,525.44616
	Minimum	14	14	15	15
	Maximum	65,226.00	29,999.00	26,406.00	137,825.00

Clash of Clans	Color Switch	Cooking Fever	Crossy Road	DragonSoul	Futurama
72.1951	77.7841	62.9184	75.4217	75.418	71.7532
64	67	55	71	61	64.5
44.11375	40.22	39.13691	35.00968	55.4567	34.2415
14	15	13	15	14	17
453	503	373	432	466	211
1.9675	1.7134	1.7895	2.313	6.1386	2.8408
0	0	0	0	0	0
6.01507	5.5316	5.79021	6.69795	30.35152	6.52948
0	0	0	0	0	0
44.46	134.6	46.21	49.37	426.58	41.92
2,066.8841	677.5587	1,960.4966	874.9794	2,758.3802	5,322.0724
1052	238	785	202.5	1018	3342.5
3,843.25612	2,354.16470	3,105.47403	2,973.74119	4,367.92623	7,070.01833
14	15	14	13	15	16
40,416.00	54,992.00	24,939.00	84,541.00	27,165.00	37,242.00

Geometry Dash	Gyrosphere	Kendall & Kylie	Mandala	Marvel	Pac-man
141.2878	73.2172	83.7517	63.2126	97.6969	75.9929
105	63	72	61	81	67
103.49925	57.08688	52.60137	35.91022	67.01694	41.7377
14	14	13	12	15	17
509	460	464	485	476	257
6.8305	6.6186	3.3239	1.3405	7.8992	10.7787
0	0	0	0	0	0
25.17503	12.39998	6.37884	4.56835	12.43397	16.12157
0	0	0	0	0	0
355.73	49.32	41.14	49.06	49.58	49.95
1,200.1415	2,106.6528	2,360.1394	657.22	1,812.8996	3,600.0148
599.5	886	967	289	134	845
5,497.60734	5,887.34728	4,641.92860	1,488.99857	4,702.89822	6,707.73041
14	14	13	13	16	15
180,517.00	74,043.00	48,193.00	25,003.00	48,285.00	30,847.00

Roll the Ball	Solitaire	Stack	Subway Surfers	Talking Tom	Trials Frontier
82.8591	71.4085	131.6246	75.8468	67.2463	147.21
71	61	127	60	58	90
53.46217	52.2172	65.34124	63.10764	43.48884	197.18153
15	14	13	14	16	15
473	511	449	474	441	3,304.00
3.3485	1.7812	6.1064	3.4549	2.1991	7.317
0	0	0	0	0	0
7.32084	5.50171	7.97304	7.85475	5.97422	33.88522
0	0	0	0	0	0
45.75	48.72	42.66	43.83	47.04	536.33
4,622.1355	2,451.3717	1,009.9552	2,573.4957	2,110.4649	2,790.2073
4027.5	1467.5	637	983.5	883	1084.5
7,263.83176	3,305.21723	2,869.21725	5,218.94623	4,324.08623	6,830.12534
17	15	14	16	15	16
88,061.00	25,610.00	90,536.00	32,671.00	40,370.00	73,168.00

Twist	Words Crush	World Chef
121.6685	90.5642	75.7939
110	75.5	66
64.15721	62.69428	43.48043
14	15	14
515	455	397
7.0604	3.2234	2.0273
0	0	0
9.16857	6.2004	5.68818
0	0	0
48.71	33.07	44.49
938.769	2,166.7595	2,028.4020
587	2542	974.5
2,156.80346	31,075.99964	4,252.33990
15	-515,702.00	15
71,247.00	67,777.00	54,431.00

Long Press

		Altos Adventure	Geometry Dash	Trials Frontier
Duration	Mean	998	799.4857	1,290.3280
	Median	803	630.5	955
	Std. Deviation	559.90689	396.005	952.19171
	Minimum	500	502	511
	Maximum	2,512.00	2,468.00	7,560.00
Travelled Distance	Mean	116.938	39.4274	28.2176
	Median	40	32.5317	22.2361
	Std. Deviation	242.2201	45.87372	31.27988
	Minimum	5	3	0
	Maximum	1,204.60	396.47	235.38
Intervals	Mean	3,249.7027	1,224.2143	2,442.2480
	Median	1803	1061.5	994
	Std. Deviation	3,675.06990	698.66118	4,107.02349
	Minimum	71	63	28
	Maximum	14,268.00	3,762.00	21,894.00

Swipe

		Candy Crush	Crossy Road	Marvel	Futurama
Duration	Mean	292.382	203.2354	333.8657	463.3926
	Median	223	159	153	314.5
	Std. Deviation	236.08999	220.92503	442.34248	415.68092
	Minimum	78	52	45	67
	Maximum	2,205.00	3,061.00	3,739.00	3,661.00
Speed	Mean	0.6416	1.5882	1.8437	0.619
	Median	0.572	1.5477	1.6599	0.5898
	Std. Deviation	0.46523	0.80185	1.34625	0.35617
	Minimum	0.08	0.02	0	0.06
	Maximum	6.19	4.16	6.23	2.95
Travelled Distance	Mean	148.2308	266.4899	338.6584	199.73
	Median	134.7403	235.5551	297.0053	189.4325
	Std. Deviation	101.97013	206.72836	267.5904	79.82551
	Minimum	50.79	11.9	0	55.56
	Maximum	1,589.08	3,321.03	1,395.30	669.78
Intervals	Mean	3,509.3596	847.5212	790.4728	4,428.2333
	Median	2698	402	146	3423
	Std. Deviation	3,410.62276	1,473.43378	3,365.99690	4,185.00339
	Minimum	15	16	14	24
	Maximum	26,311.00	20,591.00	42,439.00	35,396.00
Angular Offset	Mean	3,416.5160	693.7361	730.533	4,301.0449
	Median	2535.8416	245.1356	107.5902	3300.8042
	Std. Deviation	3,416.09873	1,436.88204	3,341.36003	4,185.66734
	Minimum	3.03	1.23	1.79	9.79
	Maximum	26,179.93	20,556.46	42,147.98	35,247.11

Gyrosphere	Kendall & Kylie	Pac-man	Subway Surfers	Roll the Ball
609.1868	742.8723	235.4831	258.2076	300.4581
238	604.5	199	219	246.5
1,101.80054	597.10811	162.92618	155.9758	186.15977
34	96	40	66	65
10,686.00	4,965.00	2,483.00	1,543.00	1,617.00
1.3264	0.9448	1.4602	1.5139	0.9945
0.9829	0.8136	1.2148	1.3826	0.9604
1.13545	0.56039	0.95429	0.67924	0.46562
0	0.03	0.1	0.08	0.01
7.08	3.03	6.02	4.67	4.13
390.1261	534.613	295.7556	348.7935	248.9412
260.8799	496.5716	253.7931	321.8207	239.8984
449.00809	328.1547	179.31057	156.5362	92.12008
2	23.27	50.09	48.94	8
4,241.04	3,582.71	1,628.66	920.12	668.53
641.7681	1,972.9096	768.6963	1,297.2984	1,238.8911
232	574	435	950	368
1,404.86735	5,184.24833	1,133.67989	1,238.15684	2,245.90121
12	15	14	18	15
27,268.00	48,271.00	17,512.00	14,998.00	23,072.00
573.1042	1,783.9735	645.6847	1,079.4172	1,132.8608
224.7028	445.13	308.1862	714.7529	283.879
1,342.12985	5,125.70232	1,120.46503	1,209.53631	2,215.86307
0.04	3.05	0.01	2.65	0.12
26,989.77	48,227.85	17,469.01	14,944.24	23,016.72

Drag

		8 Ball Pool	Agar.io	Clash of Clans	Cooking Fever
Duration	Mean	2,558.9952	2,703.0081	1,156.5932	685.429
	Median	1466	472.5	596	583
	Std. Deviation	2,871.84622	6,810.26204	1,481.66597	477.88388
	Minimum	96	72	73	34
	Maximum	18,192.00	93,377.00	6,957.00	3,870.00
Speed	Mean	0.6501	1.0951	0.6313	1.1062
	Median	0.1981	0.879	0.3856	1.0219
	Std. Deviation	1.06073	0.88974	0.64155	0.65918
	Minimum	0	0.04	0.01	0.05
	Maximum	6.82	6.17	2.3	3.75
Travelled Distance	Mean	620.8434	1,919.8605	372.2347	649.1254
	Median	523.9452	511.6995	276.4374	640.6557
	Std. Deviation	589.03997	4,539.30461	421.65162	385.82374
	Minimum	10	37	5	25.1
	Maximum	3,215.53	63,445.64	2,233.25	1,700.55
Intervals	Mean	6,294.6810	526.0081	1,507.8983	2,078.3576
	Median	854.5	329.5	782	566.5
	Std. Deviation	11,503.49933	1,039.55434	2,636.66104	3,787.13704
	Minimum	14	14	15	15
	Maximum	64,695.00	16,613.00	18,468.00	29,276.00
Angular Offset	Mean	6,216.8360	487.7067	1,371.7974	1,933.2155
	Median	826.875	278.9963	606.343	417.8209
	Std. Deviation	11,457.24263	1,000.71555	2,624.06596	3,737.78132
	Minimum	14.56	0.12	8.11	1.74
	Maximum	64,648.57	16,506.76	18,371.36	29,128.79

Mandala	Solitaire	Talking Tom	World Chef
819.264	975.3139	929.1404	1050.4082
481	847	395.5	689
1,326.11490	569.28799	1,615.38868	1,058.27165
73	63	72	108
12,209.00	3,474.00	11,705.00	8,145.00
0.9427	0.7695	0.8794	0.8255
0.7672	0.6994	0.546	0.6033
0.64985	0.38861	0.8705	0.83238
0	0.22	0.01	0.03
2.97	1.94	5.15	6.44
514.0228	656.1765	573.3495	624.3986
399.552	562.7151	264.5145	462.083
728.25379	346.98553	1,150.78908	541.02235
10.65	51.53	6	25.24
11,150.06	1,532.08	10,833.46	4,093.02
644.2246	1,901.9781	2,029.5670	1170.9213
299	1316	686	817
1,575.42626	3,040.37448	3,555.05257	1,421.36348
15	15	14	15
20,194.00	28,892.00	25,286.00	15,596.00
537.843	1,656.3822	1,910.0863	1004.3527
235.8184	923.4709	572.1282	645.7193
1,539.40068	2,981.47577	3,534.13539	1,405.28467
0.35	3.04	0.32	2.76
20,097.14	28,114.92	25,111.87	15,478.50

Rotation

		8 Ball Pool	Agar.io
Duration	Mean	7,305.0000	9,675.5000
	Median	5937	6323.5
	Std. Deviation	4,225.40407	12,854.33601
	Minimum	1,039.00	541
	Maximum	18,192.00	93,377.00
Speed	Mean	0.181	0.8907
	Median	0.1416	0.8365
	Std. Deviation	0.09813	0.44723
	Minimum	0.04	0.22
	Maximum	0.35	3.08
Travelled Distance	Mean	1,431.4771	6,974.5502
	Median	1238.9625	4506.1184
	Std. Deviation	1,032.77481	8,479.84796
	Minimum	68	911.95
	Maximum	3,215.53	63,445.64
Intervals	Mean	12,418.2632	447.5
	Median	5472	139.5
	Std. Deviation	15,879.30653	531.31084
	Minimum	15	14
	Maximum	52,996.00	2,332.00
Angular Offset	Mean	12,242.3132	431.3196
	Median	5358.4262	304.9745
	Std. Deviation	15,788.27599	407.57276
	Minimum	17.17	3.93
	Maximum	52,642.64	2,286.47

Scribble

		Mandala	Talking Tom
Duration	Mean	5,318.4615	4,985.1304
	Median	3486	4393
	Std. Deviation	4,156.01900	3,253.72303
	Minimum	1,000.00	1,142.00
	Maximum	12,209.00	11,705.00
Speed	Mean	0.5095	0.7836
	Median	0.526	0.7327
	Std. Deviation	0.27464	0.34492
	Minimum	0	0.36
	Maximum	0.91	1.79
Travelled Distance	Mean	2,585.0997	3,598.2717
	Median	1833.205	3096.0178
	Std. Deviation	3,034.24498	2,527.62358
	Minimum	10.65	893.62
	Maximum	11,150.06	10,833.46
Intervals	Mean	781.7692	1,807.3478
	Median	696	393
	Std. Deviation	716.74835	3,316.32271
	Minimum	15	16
	Maximum	2,163.00	14,087.00
Angular Offset	Mean	737.387	1,758.7140
	Median	616.4387	348.6211
	Std. Deviation	713.24403	3,238.46065
	Minimum	4.11	9.69
	Maximum	2,148.20	13,646.79

Shapes

		Z	Backwards C	n	C
Duration	Mean	1356.208333	1829.647059	1246.352941	1310.846154
	Median	1269	1569	1170	1132
	Std. Deviation	619.0004551	1090.302306	698.9160305	606.7798676
	Minimum	585	911	582	663
	Maximum	3278	5619	3693	2554
Speed	Mean	1.430193928	0.78133844	1.329518841	1.329949708
	Median	1.3108983	0.83846545	1.1199474	1.2037919
	Std. Deviation	0.651094926	0.389300119	0.618667702	0.584346019
	Minimum	0.49025723	0	0.3648755	0.5164388
	Maximum	2.9577587	1.2471077	2.6528747	2.2454073
Travelled Distance	Mean	1623.194038	1208.041124	1363.629359	1453.077469
	Median	1715.04675	1357.2804	1377.6844	1474.2046
	Std. Deviation	271.8618618	496.9906099	136.1458503	176.2696144
	Minimum	845.8477	0	1159.1456	1233.759
	Maximum	1994.1968	1740.6754	1562.5432	1827.7976
Intervals	Mean	2010.291667	3018.352941	2076.647059	1836.923077
	Median	1885	2992	1901	1791
	Std. Deviation	1151.019717	2429.794409	1200.262781	1039.193073
	Minimum	45	17	331	117
	Maximum	5031	7282	5496	3766
Angular Offset	Mean	1477.753994	2862.440017	1621.87251	1747.605711
	Median	1122.407225	2311.971324	1402.947794	1756.100143
	Std. Deviation	1027.882447	2439.344901	1183.943113	1041.549121
	Minimum	45.66985509	32.9199359	101.8653246	81.5849944
	Maximum	4411.255698	7219.101783	4977.984556	3602.464071

U	S	XI	Backwards N	N
1767.230769	1814.166667	2129.4	1650	1697
1871	1850.5	2080	1649.5	1605
729.5090991	420.0011508	680.390917	316.4922327	695.5781768
726	1140	1282	1265	1052
2861	2254	2910	2036	2434
0.863784343	0.862136917	0.874449656	0.862764625	0.81054983
0.7369646	0.73921505	0.86474144	0.813969335	0.92080724
0.564451921	0.281324562	0.331540196	0.202072877	0.34871884
0.018605841	0.66118366	0.5164216	0.67971283	0.42002985
2.1659172	1.3720213	1.3518404	1.143407	1.0908124
1190.891867	1471.968433	1683.76908	1388.752225	1216.598187
1287.0193	1506.49035	1707.6382	1455.63445	1148.6255
373.6529176	135.1724245	110.2138138	202.3577141	235.7010842
42.607376	1203.3542	1502.7869	1092.9783	1022.35266
1572.4558	1564.7592	1796.9327	1550.7617	1478.8164
1681.153846	1548.666667	2423	4506	4742.333333
1784	1788.5	2105	3483.5	5837
1428.105321	767.4770789	1559.03608	3702.221405	4272.502467
22	23	333	1342	29
4944	2050	4633	9715	8361
1477.36832	1209.268114	2342.002806	4116.045949	4548.784697
1319.769454	1225.356108	2071.350345	3094.426091	5302.892333
1175.937645	406.64237	1537.525352	3777.25381	3556.199358
244.8875756	542.0504402	216.7803803	864.7474463	676.0127658
4338.72403	1752.583634	4460.268995	9410.584166	7667.448993

Pinch and Spread

		Pinch	Spread
Duration	Mean	3,432.4400	2,630.4400
	Median	2,862.0000	2,693.0000
	Std. Deviation	2,509.52192	1,115.87365
	Minimum	518	985
	Maximum	12,814.00	5,406.00
Speed	Mean	0.1452	0.127
	Median	0.1077	0.1062
	Std. Deviation	0.1277	0.07807
	Minimum	0.02	0
	Maximum	0.67	0.38
Travelled Distance	Mean	478.491	303.6043
	Median	350.7252	274.5926
	Std. Deviation	615.69101	170.08269
	Minimum	40.83	3
	Maximum	3,258.68	672.75
Intervals	Mean	422.32	1,207.2000
	Median	460	655
	Std. Deviation	438.24173	1,626.27268
	Minimum	14	14
	Maximum	1,495.00	5,702.00
Angular Offset	Mean	388.7943	1,100.0238
	Median	338.6684	616.1037
	Std. Deviation	269.92816	1,496.36633
	Minimum	63.82	60.47
	Maximum	1,244.96	5,120.62

Appendix B – First Study Script

INTRODUCTION ORAL SCRIPT: [Hello, my name is Anabela Rodrigues and I'm an Informatics Masters student at Lisbon University. My project is called Assistive Gameplay, and it aims to enable motor-impaired users to be able to play any touchscreen game on mobile devices, regardless of whether the original developers made it accessible or not. This study aims to understand current gameplay demands and most used gestures in games. We will be collecting your gameplay data while you play a few games from the top 25 games in the Google Play store. You'll be playing 5 to 6 games for 5 minutes each. Our framework will be running in the background while you play, and it will record your input. We will also be recording the screen. Your privacy will be protected at all times. Your identity will not be known by anyone other than the people directly involved in the study, and none of your personal details will be stored alongside the data collected. Any input and screen recordings will not be used for any other reason apart from the study. You can withdraw from the study at any time. If you decide to withdraw, the information we hold on you for the research will be destroyed.]

5 minutes: Read information sheet & ask them to sign consent form.

5 minutes: Give them Google Forms form to fill out, which asks about demographics, mobile habits and gameplay habits. Make clear that all questions are optional and the participant can choose not to answer.

30 to 40 minutes: Gameplay session.

DEBRIEF ORAL SCRIPT: [Thank you for your participation. We will analyse the touchscreen data you provided to understand the input demands of each game and to identify the most used gestures. If you want to be kept up to date with the study just give us your email. Let me know if you have any additional comments or questions.]

Appendix C – Second Study Script

INTRODUCTION ORAL SCRIPT: [Hello, my name is Anabela Rodrigues and I'm an Informatics Masters student at Lisbon University. My project is called Assistive Gameplay, and it aims to enable motor-impaired users to be able to play any touchscreen game on mobile devices, regardless of whether the original developers made it accessible or not. This study intends to understand the interaction abilities and difficulties you may have, and which common Android game gestures are difficult to perform. To do this we will ask you to record some gestures with our application. It's a simple interface that will prompt you to perform various gestures. It will record your input, and the application will take some screenshots. With your consent, we will also video record (without audio) you interacting with the tablet – only capturing the tablet and your hands. This video will be used to help us analyse the data later, and we may use some stills for eventual academic publications - if that's alright with you. Your privacy will be protected at all times. Your identity will not be known by anyone other than the people directly involved in the study, and none of your personal details will be stored alongside the data collected. Any input and screen recordings will not be used for any other reason apart from the study. This session will take approximately 50 minutes. You can withdraw from the study at any time. If you decide to withdraw, the information we hold on you for the research will be destroyed.]

5 minutes: Read information sheet & ask them to sign consent form.

5 minutes: Ask questions about: demographics, what kind of impairment they may have, gameplay habits, mobile habits. Make clear that all questions are optional and the participant can choose not to answer.

10 minutes: Practice time: I talk them through the application and what gestures will be required of them. Allow them some time to practice each gesture.

30 minutes: Gestures session

DEBRIEF ORAL SCRIPT: [Thank you for your participation. We will use the touchscreen data you provided to understand which game gestures need to be adapted. The goal of this is to better understand what makes a game inaccessible, and which gestures are more or less difficult to perform. If you want to be kept up to date with the study just give me your email. Let me know if you have any additional comments or questions.]

Appendix D – Newcastle University Ethics Form

University Ethics Form Version 2.1

Date submitted
25/02/2016 17:28:21

Applicant Details

Is this approval for a:
Staff Project [A1]
Name of Principal Researcher:
Anabela Rodrigues
Please enter your email address
annierodrigues@hotmail.com
Please select your school / academic unit
School of Computing Science [A6]

Project Details

Project Title
Assistive Gameplay: making smartphone games more accessible to people with motor impairments
Project Synopsis
<p>The aim of this study is to enable motor-impaired users to be able to play any touchscreen game on mobile devices, regardless of whether the original developers implemented accessibility features. Mobile games have been shown to have numerous health benefits, providing users with stress relief, entertainment, education, manual dexterity improvement and social wellbeing. However, many of these games are complex, fast-paced and require high touch precision and multi-touch gestures. For players with less motor dexterity, these games become frustrating rather than entertaining. There is a need to adapt these games so anyone can play, regardless of age or disability.</p> <p>Our software will use game interaction data collected from able-bodied players to create new gesture models that can be used by motor-impaired users to interact with the same games using alternative interactions. For example, we can remap a simple tap gesture from the motor-impaired user to a more complex swipe gesture collected from able bodied players – thus allowing the motor-impaired user to perform swipes with ease.</p>
Project start date
01/03/2016
Project end date
01/08/2016
Is the project externally funded?
No [A3]
Does your project involve collaborators outside of the University?
Yes [Y]
Please provide a list of the collaborating organisations?
Faculty of Sciences of the University of Lisbon and School of Computing University of Dundee

Existing Ethics, Sponsorship & Responsibility

Has ethical approval to cover this proposal already been obtained?
No [N]
Will anyone be acting as sponsor under the NHS Research Governance Framework for Health and Social Care?
No [N]

Do you have a Newcastle upon Tyne Hospitals (NUTH) reference?
No [N]
Will someone other than you (the principal investigator) or your supervisor (for student projects) be responsible for the conduct, management and design of the research?
No [N]
The Animals (Scientific Procedures) Act defines protected animals as: 'any living vertebrate other than man...in its foetal, larval or embryonic form.....from the stage of its development when— (a)in the case of a mammal, bird or reptile, half the gestation or incubation period for the relevant species has elapsed; and (b)in any other case, it becomes capable of independent feeding'. In practice 'Protected' animals are all living vertebrates (other than man), including some immature forms, and cephalopods (e.g. octopus, squid, cuttlefish). Using this definition, does your research involve the observation, capture or manipulation of animals or their tissues?
No [N]
Will the study involve participants recruited by virtue of being NHS patients or service users, their dependents, their carers or human tissues or the use of NHS & Health/Social Care Facilities or otherwise require REC approval?
No [N]
Does the research involve human participants e.g. use of questionnaires, focus groups, observation, surveys or lab-based studies involving human participants?
Yes [Y]
Does the study involve any of the following? [<small>a. The study involves children or other vulnerable groups, as defined in Section 59 of the Subordinate Vulnerable Adults Act 2006 or those who are relatively or absolutely incapable of protecting their own interests, or those in unequal relationships e.g. participants who are subordinate to the researcher(s) in a context outside the research?</small>]
Does the study involve any of the following? [<small>b. The study requires the co-operation of a gatekeeper (defined as someone who can exert undue influence) for initial access to the groups or individuals to be recruited e.g. students at school, members of a self-help group, or residents of a nursing home? NB. The IoN & School of Psychology volunteer pools are not considered gatekeepers in this case.</small>]
Does the study involve any of the following? [<small>c. It is necessary for participants to take part in the study without their knowledge and consent e.g. covert observations of people in non-public places?</small>]
Does the study involve any of the following? [<small>d. Deliberately mislead participants in any way?</small>]
Does the study involve any of the following? [<small>e. Discussion of sensitive topics e.g. sexual activity or drug use?</small>]
Does the study involve any of the following? [<small>f. The administration of drugs, placebos or other substances (e.g. food substances, vitamins) to the study participant.</small>]
Does the study involve any of the following? [<small>g. Invasive, intrusive or potentially harmful procedures of any kind?</small>]
Does the study involve any of the following? [<small>h. Obtaining blood or tissue samples?</small>]
Does the study involve any of the following? [<small>i. Pain or more than mild discomfort?</small>]
Does the study involve any of the following? [<small>j. Psychological stress, anxiety, harm or negative consequences beyond that encountered in normal life?</small>]
Does the study involve any of the following? [<small>k. Prolonged or repetitive testing i.e. more than 4 hours' commitment or attendance on more than two occasions?</small>]
Does the study involve any of the following? [<small>l. Financial inducements (other than reasonable expenses and compensation for time)?</small>]

Does the research involve the viewing, usage or transfer of Sensitive Personal Data as defined by the [Data Protection Act 1998](#) or data governed by statute such as the [Official Secrets Act 1989](#) / [Terrorism Act 2006](#), commercial contract or by convention e.g. client confidentiality? (If you are unsure please tick YES and complete the sub-questions).

No [N]

Will the study cause direct or indirect damage to the environment or emissions outside permissible levels or be conducted in an [Area of Special Scientific Interest](#) or which is of cultural significance?

No [N]

Will the research be conducted outside of the [European Economic Area \(EEA\)](#) or will it involve international collaborators outside the EEA?

No [N]

Next Steps

Based on your answers your project has triggered none of the high risk flags. Therefore the University requires no further ethical review before your project progresses. Should your project change you may need to apply for new ethical approval.

Supporting Documentation

Please upload any documents (not uploaded elsewhere in the application) which you think are relevant to the consideration of your application.

assistive_play_consent_forms.pdf (252.89KB) Consent Forms - For each phase of the study
assistive_play_information_sheets.pdf (162.409KB) Information Sheets - For each phase of the study
ethics%20assistive%20play.pdf (491.354KB) Ethics Form -

filecount - Please upload any documents (not uploaded elsewhere in the application) which you think are relevant to the consideration of your application.

3

Thank you for completing the University's Ethical Review Form. Based on your answers the University is satisfied that your project has met its ethical expectations and that no further review is required before you begin your research. Please be aware that if you make any significant changes to your project then you should complete this form again as further review may be required. Confirmation of this decision will be emailed to you. Please complete the declaration to submit your application.

Declaration

I certify that:

[the information contained within this application is accurate.]

Yes [Y]

Thank you for completing the University's Ethical Review Form. Based on your answers the University is satisfied that your project has met its ethical expectations and that no further review is required before you begin your research. Please be aware that if you make any significant changes to your project then you should complete this form again as further review may be required. Confirmation of this decision will be emailed to you. Please complete the declaration to submit your application.

Declaration

I certify that:

[the research will be undertaken in line with all appropriate, University, legal and local standards and regulations.]

Yes [Y]

Thank you for completing the University's Ethical Review Form. Based on your answers the University is satisfied that your project has met its ethical expectations and that no further review is required before you begin your research. Please be aware that if you make any significant changes to your project then you should complete this form again as further review may be required. Confirmation of this decision will be emailed to you. Please complete the declaration to submit your application.

Declaration

I certify that:

[I have attempted to identify the risks that may arise in conducting this research and acknowledge my obligation to (and rights of) any participants.]

Yes [Y]

Thank you for completing the University's Ethical Review Form. Based on your answers the University is satisfied that your project has met its ethical expectations and that no further review is required before you begin your research. Please be aware that if you make any significant changes to your project then you should complete this form again as further review may be required. Confirmation of this decision will be emailed to you. Please complete the declaration to submit your application.

Declaration

I certify that:

[no work will begin until all appropriate permissions are in place.]

Yes [Y]