Multimodal Interface for an Intelligent Wheelchair

Filipe Joaquim de Oliveira Reis Coelho

July 08, 2019
Multimodal Interface for an Intelligent Wheelchair

Filipe Joaquim de Oliveira Reis Coelho

Mestrado Integrado em Engenharia Informática e Computação

July 08, 2019
Abstract

The use of wheelchairs has increased with the increase of the world population. However, the change from manual wheelchairs to powered wheelchairs only marked a difference in the used technology, while remaining inaccessible to many individuals with specific needs.

The importance of the intelligent wheelchairs is recognized by the change of the wheelchair paradigm, enabling their access to a more significant number of users, as they integrate smart systems and autonomous behaviors, like wall following, obstacle detection and avoidance, automated movement in pre-calculated trajectories or controlling external devices.

However, the higher number of functionalities consequently increases the degree of complexity of the system, an essential factor for people with special needs. Regarding this, adaptive multimodal interfaces can reveal to be crucial, because of the way they can adapt to the specific necessities of each user and thanks to the multimodality they can control multiple systems with more straightforward commands due to the input devices redundancy.

This dissertation focuses on the study and analysis of state of the art in the subjects related to the theme, referring to topics as interaction, interfaces, intelligent wheelchairs, and the analysis of the Intellweels project, and describes the creation and development of a new multimodal interface. Can be seen an explanation for technological choices as well as a detailed overview of the system architecture. Can also be seen the implemented devices like the joystick or a sensor that detects the movement of the head and translates it into input commands as well as a simulator used in the 30 user experiments. After the analysis of the responses from the user experiments, there is an overview of the overall performance of the system and the future improvements to be taken in the subsequent developments.

Based on an ambitious project such as the Intellweels, but also a project with developments since 2007 and with several technological iterations, this work aims to contribute to the control and interaction module of an intelligent wheelchair in the attempt to become one of the many success stories of the project. The multimodal interface created improves on the previous one by being more expandable, configurable and allowing for input and output actions to be in parallel and consecutive. The evaluation of the implemented solution by 30 volunteers is a good indicator of the validity and integrity of the quality of the solution to be implemented. Their feedback is also beneficial to know what are the key points that this solution does not does as well, and that should be improved.

Keywords: adaptability, intelligent wheelchair, interaction, multimodal interfaces
Resumo

O uso de cadeiras de rodas tem aumentado com o aumento da população mundial. No entanto, a mudança das cadeiras de rodas manuais para as cadeiras de rodas elétricas marcou uma mudança apenas na tecnologia usada, continuando inacessível a muitos indivíduos com necessidades específicas.

A importância das cadeiras de rodas inteligentes é reconhecida pela mudança no paradigma das cadeiras de rodas, permitindo o seu acesso a um maior número de utilizadores, pois integram sistemas inteligentes e comportamentos autônomos, como seguimento de paredes, detecção e desvio de obstáculos, movimento automático em trajetórias pré-calculadas ou controlo de dispositivos externos.

Mas o maior número de funcionalidades aumenta consequentemente o grau de complexidade do sistema, um fator importante para pessoas com necessidades especiais. Neste aspecto, as interfaces multimodais adaptativas podem revelar-se fulcrais, pois graças à adaptabilidade podem ser costumizadas para necessidades específicas dos utilizadores e graças à multimodalidade premitem controlar múltiplos sistemas com comandos mais simples devido à redundância dos dispositivos de entrada de dados.

Esta dissertação foca-se no estudo e análise do estado da arte nos assuntos relacionados com o tema abordado, referindo tópicos como interação, interfaces, cadeiras de rodas inteligentes e a análise do projeto Intellweels, e descreve a criação de uma nova interface multimodal. Pode ser visto uma explicação para as opções tecnológicas, bem como uma visão detalhada da arquitetura do sistema. Também pode ser visto os dispositivos implementados como o joystick ou um sensor que detecta o movimento da cabeça e traduz em comandos de entrada, bem como um simulador usado nas 30 experiências. Após a análise das respostas dessas experiências, é possível obter uma visão geral do desempenho do sistema e as melhorias a serem efetuadas no futuro.

Tendo como base um projeto ambicioso como o caso do projeto Intellweels, mas também um projeto com desenvolvimentos desde o ano 2007 e com várias iterações tecnológicas, este trabalho tem em vista contribuir no modulo de controlo e interação de uma cadeira de rodas inteligente, na tentativa de se tornar em mais uma das muitas histórias de sucesso do projeto. A interface multimodal criada melhora a desenvolvida anteriormente ao ser mais expansível, configurável e permitir que as ações de entrada e saída sejam executadas em paralelo ou consecutivamente. A avaliação da solução implementada por 30 voluntários é um bom indicador da validade e integridade da qualidade da solução a ser implementada. Os seus comentários também são benéfico para saber quais são os pontos-chave que essa solução não satisfaz, e que por isso deve ser melhorados.

Keywords: adaptability, intelligent wheelchair, interaction, multimodal interfaces
Acknowledgements

To my mother and father, for all the help and constant support throughout my entire life. To my brother for all the availability in listening and help. To my girlfriend, Eliana, for the unconditional love, trust, availability, and for helping me believe in myself at times where I did not.

To my supervisor, Luis Paulo Reis for the idea of this project, the opportunity to work in it and all the help during this entire experience, orienting me to make the right choices. Also to Alexandra Oliveira and Brígida Faria for the incredibly generous help in the user experiments.

To my friend José Vieira for all the shared experience, doubts, and lessons learned, and for the many hours working together for our projects.

To João Martins for all the help making the head motion sensor device, creating the printed circuit board, and for all the electronics and 3D printed related discussions.

And to all of my friends that I shared this experience and many group projects in FEUP, I could not have done it without all your help and support.

Filipe Coelho
“I could either watch it happen or be a part of it.”

Elon Reeve Musk
## Contents

1 Introduction
   1.1 Context ................................................. 1
   1.2 Motivation .............................................. 2
   1.3 Research Questions ................................. 3
   1.4 Implemented Solution .............................. 3
   1.5 Structure of the Thesis ......................... 4

2 Human-Machine Interfaces
   2.1 Human-Computer Interaction ....................... 5
   2.2 Human-Robot Interaction ........................... 7
   2.3 User Interfaces ....................................... 8
      2.3.1 Adaptive Interfaces ........................... 9
      2.3.2 Multimodal Interfaces ....................... 10
   2.4 Conclusions ........................................... 11

3 Intelligent Wheelchairs
   3.1 Intelligent Wheelchairs ............................ 13
   3.2 Interfaces for Intelligent Wheelchairs .......... 19
   3.3 Intellweels Project .................................. 20
   3.4 Conclusions ........................................... 22

4 Development of the Multimodal Interface
   4.1 Development Tools and Technologies ............ 25
   4.2 Systems and Devices .................................. 28
   4.3 Input System ........................................... 28
      4.3.1 Input Signatures ............................... 29
      4.3.2 Action Triggers .................................. 29
      4.3.3 Input Events ...................................... 30
      4.3.4 Input Groups ..................................... 30
      4.3.5 Input Sequences ................................ 30
   4.4 Output System ......................................... 30
      4.4.1 Output Signatures ............................... 31
      4.4.2 Output Groups .................................... 31
      4.4.3 Output Sequences ............................... 31
   4.5 Parameter System ..................................... 31
   4.6 Graphical User Interface ........................... 32
   4.7 Channels ............................................... 32
   4.8 Conclusions ........................................... 33
### List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Mandarasz wheelchair prototype</td>
<td>14</td>
</tr>
<tr>
<td>3.2</td>
<td>Omnidirectional wheelchair prototype</td>
<td>15</td>
</tr>
<tr>
<td>3.3</td>
<td>Two-legged intelligent wheelchair prototype</td>
<td>15</td>
</tr>
<tr>
<td>3.4</td>
<td>NavChair intelligent wheelchair prototype</td>
<td>16</td>
</tr>
<tr>
<td>3.5</td>
<td>Tin Man I and Tin Man II intelligent wheelchair prototypes</td>
<td>16</td>
</tr>
<tr>
<td>3.6</td>
<td>MAid intelligent wheelchair prototype</td>
<td>17</td>
</tr>
<tr>
<td>3.7</td>
<td>FRIEND I intelligent wheelchair prototype</td>
<td>17</td>
</tr>
<tr>
<td>3.8</td>
<td>Intelligent wheelchair prototype based on ACCoMo agents</td>
<td>18</td>
</tr>
<tr>
<td>3.9</td>
<td>Intelligent wheelchair prototype and robotic arm controlled by brain waves</td>
<td>18</td>
</tr>
<tr>
<td>3.10</td>
<td>Intelligent wheelchair prototype developed by MIT</td>
<td>19</td>
</tr>
<tr>
<td>3.11</td>
<td>Intellweels project prototypes</td>
<td>20</td>
</tr>
<tr>
<td>3.12</td>
<td>Example of the multimodal interface created for the Intellwhells project</td>
<td>21</td>
</tr>
<tr>
<td>3.13</td>
<td>Architecture of the multi agent systems</td>
<td>22</td>
</tr>
<tr>
<td>4.1</td>
<td>A diagram of the various systems in the multimodal interface and how they communicate.</td>
<td>27</td>
</tr>
<tr>
<td>4.2</td>
<td>An overview of the Multimodal Interface GUI.</td>
<td>32</td>
</tr>
<tr>
<td>5.1</td>
<td>Diagram showing the interactions of the devices with the multimodal interface systems.</td>
<td>36</td>
</tr>
<tr>
<td>5.2</td>
<td>Circuit board on the left and helmet with 3D printed support on the right.</td>
<td>37</td>
</tr>
<tr>
<td>5.3</td>
<td>Node system example.</td>
<td>38</td>
</tr>
<tr>
<td>5.4</td>
<td>Aerial view on the left and a first person view of the simulated environment on the right.</td>
<td>39</td>
</tr>
<tr>
<td>6.1</td>
<td>Annotated aerial view of the simulator</td>
<td>42</td>
</tr>
<tr>
<td>6.2</td>
<td>Demography: Gender.</td>
<td>46</td>
</tr>
<tr>
<td>6.3</td>
<td>Demography: Profession.</td>
<td>46</td>
</tr>
<tr>
<td>6.4</td>
<td>Demography: Educational Experience.</td>
<td>47</td>
</tr>
<tr>
<td>6.5</td>
<td>Demography: How easy technology is.</td>
<td>47</td>
</tr>
<tr>
<td>6.6</td>
<td>Demography: How easy interaction devices are.</td>
<td>48</td>
</tr>
<tr>
<td>6.7</td>
<td>Experiments: Easiness.</td>
<td>49</td>
</tr>
<tr>
<td>6.8</td>
<td>Experiments: Comfortability.</td>
<td>49</td>
</tr>
<tr>
<td>6.9</td>
<td>Experiments: Reliability.</td>
<td>50</td>
</tr>
<tr>
<td>6.10</td>
<td>Experiments: Responsiveness.</td>
<td>50</td>
</tr>
<tr>
<td>6.11</td>
<td>Experiments: Intuitiveness.</td>
<td>51</td>
</tr>
<tr>
<td>6.12</td>
<td>Experiments: Autonomy.</td>
<td>51</td>
</tr>
<tr>
<td>6.13</td>
<td>Experiments: Freedom.</td>
<td>52</td>
</tr>
<tr>
<td>6.14</td>
<td>Experiments: How easiness to perform the required steps.</td>
<td>52</td>
</tr>
</tbody>
</table>
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Number</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.15</td>
<td>Experiments: More difficult in a real-world scenario.</td>
<td>53</td>
</tr>
<tr>
<td>6.16</td>
<td>Experiments: How easy it felt to create or modify input sequences and input</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>groups using the recording system.</td>
<td></td>
</tr>
<tr>
<td>6.17</td>
<td>Experiments: How easy it felt to create or modify the action mappings.</td>
<td>54</td>
</tr>
<tr>
<td>6.18</td>
<td>Experiments: Usefulness was the variety of input devices.</td>
<td>54</td>
</tr>
<tr>
<td>6.19</td>
<td>Experiments: System efficiency with multiple devices.</td>
<td>55</td>
</tr>
<tr>
<td>6.20</td>
<td>Experiments: System efficiency with multiple devices.</td>
<td>55</td>
</tr>
<tr>
<td>6.21</td>
<td>Simulator aerial view with two positional overlays in red and green.</td>
<td>56</td>
</tr>
<tr>
<td>C.1</td>
<td>Schematic of the voltage regulation circuit.</td>
<td>81</td>
</tr>
<tr>
<td>C.2</td>
<td>Schematic for the header connections.</td>
<td>81</td>
</tr>
<tr>
<td>C.3</td>
<td>Schematic for the ESP-01 connections.</td>
<td>82</td>
</tr>
<tr>
<td>C.4</td>
<td>Schematic for the serial clock (SCL) and serial data (SDA) pins on the left and for the MPU6050 on the right.</td>
<td>82</td>
</tr>
<tr>
<td>C.5</td>
<td>Layout and routing of the circuit for the Head Motion Sensor.</td>
<td>83</td>
</tr>
<tr>
<td>C.6</td>
<td>Three dimensional representation of the final board</td>
<td>83</td>
</tr>
</tbody>
</table>
List of Tables

6.1 The average, standard deviation, minimum and maximum values of the experiments in seconds. .......................................................... 56
## Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUI</td>
<td>Adaptive User Interfaces</td>
</tr>
<tr>
<td>CASA</td>
<td>Computers as Social Actors</td>
</tr>
<tr>
<td>DoF</td>
<td>Degrees of Freedom</td>
</tr>
<tr>
<td>ECG</td>
<td>Electrocardiogram</td>
</tr>
<tr>
<td>EEG</td>
<td>Electroencephalograms</td>
</tr>
<tr>
<td>EMG</td>
<td>Electromyogram</td>
</tr>
<tr>
<td>EOG</td>
<td>Electrooculogram</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interfaces</td>
</tr>
<tr>
<td>HCI</td>
<td>Human-Computer Interface</td>
</tr>
<tr>
<td>HRI</td>
<td>Human-Robot Interaction</td>
</tr>
<tr>
<td>IW</td>
<td>Intelligent Wheelchair</td>
</tr>
<tr>
<td>MIT</td>
<td>Massachusetts Institute of Technology</td>
</tr>
<tr>
<td>PW</td>
<td>Powered Wheelchair</td>
</tr>
<tr>
<td>RQ</td>
<td>Research Question</td>
</tr>
<tr>
<td>SCL</td>
<td>Serial Clock</td>
</tr>
<tr>
<td>SDA</td>
<td>Serial Data</td>
</tr>
<tr>
<td>SW</td>
<td>Smart Wheelchair</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Any system, to be usable by a broad audience must have an interface that resembles the objective in hands rather than the raw control of the system’s devices [JC14]. For that, it is crucial to know the device to be controlled to study what interface is more appropriate. With single-purpose devices like the lights or a toaster, the switches and buttons might suffice. However, with multi-purpose devices like computers and smartphones, the interface might need to accommodate a broader range of controlling actions, like touch, keyboard, mouse, voice, gestures. Even more, sharing the same information with such different devices with different sizes and availability, was complicated with traditional interfaces.

Considering these obstacles, adaptive interfaces started being adopted by modern devices and systems, capable of adjusting to different screen sizes and different types of data inputs. However, a system like an intelligent wheelchair that is designed to be used by almost everyone, despite any limitation the user might have cannot rely solely on an adaptive interface. The userbase for these kinds of systems are very distinct, and an intelligent wheelchair for one user might require some input devices that another intelligent wheelchair for another user might not, depending on the user needs. Here, adaptive multimodal interfaces are a component of the system that can allow for multiple different types of input devices to interface with the said system, and moreover, allow for a different combination of inputs to produce the same action. They provide one of the most potent ways of interaction when tightly integrated with a device, since they provide a more natural interaction, closer to the one between humans, and also provide robustness to the system, by increasing the redundancy of the input methods or by complementing the information [RLL+04].

However, a multimodal interface is only as good as its versatility, to accept different types of inputs, and the system it is controlling. Therefore, this dissertation project is an attempt to implement a multimodal interface in an intelligent wheelchair. After studying and considering several types of input methods, a multimodal interface was the most logical choice, to allow a more broad adoption of the system presented in the wheelchair, lowering the barrier of interaction.

The work described in this dissertation approaches the problem of disabled people (or people
with special needs) in controlling intelligent wheelchairs. These wheelchairs can perform more actions than traditional ones, and sometimes their control can be overwhelming for someone who might already struggle to manipulate a powered wheelchair.

1.1 Context

Average life expectancy and quality have been increasing for the last decades, and therefore, the number of elderly and disabled people has also been increasing. These people might have minimal autonomy, especially because they can not easily move around or interact with other devices that might be at their disposal.

One of the most significant technological devices for these people are intelligent wheelchairs. Even manual or powered wheelchairs can provide a higher level of autonomy for those who could not move otherwise, but intelligent wheelchairs are more relevant than powered wheelchairs because they mark a shift in the paradigm and in the ability to do something rather than providing a mere technological shift. They allow for a higher interaction level that would not be possible otherwise, or at least, would be difficult.

Intelligent wheelchairs allow the users to move more freely, to play games, and to run actions without any user interaction. For example, a single command can tell the intelligent wheelchair to follow a wall or to navigate between rooms inside a house. However, even intelligent wheelchairs might not accommodate all types of users. There might be people that have difficulty to say an entire sentence or might not have the accuracy to maneuver a joystick. Here is where an adaptive multimodal interface can make a difference and improve the interaction, by allowing different types of commands to perform the same action and, at the same time, providing more solutions to the user.

1.2 Motivation

The main motivation for this project is linked directly to the contextualization made above. An intelligent wheelchair with all its abilities is an incredible device almost indistinguishable from a smart robot and can provide a great level of autonomy, freedom, and, therefore, quality of life.

Current solutions might not be flexible enough for all types of users that need an intelligent wheelchair. The users have particular problems and needs, and in these cases, it is the device that should adapt to the user and not the user that should adapt to the device.

For a well-implemented solution, the interface should be usable (by facilitating the use of a complex system), interactive (allowing a large and diverse number of inputs), adaptable (should adapt the system to the user needs) and expandable (can integrate with other intelligent systems and appliances).
1.3 Research Questions

This project aims to implement a multimodal interface in an intelligent wheelchair. However, a multimodal interface can be used to control any other system and improve its usability and interactivity. Therefore it is desired that the finished interface integrates well in an intelligent wheelchair with multiple input sources, but can also easily be adapted to other devices, even because with more devices can be controlled by the interface, more versatile the intelligent wheelchair can become.

Therefore, exists two research questions that this project is trying to answer:

- **Research Question 1** - Is it possible to make an interface for an intelligent wheelchair that adapts to different user needs?
  Can this interface allow different combinations of input devices without reprogramming?
  Can the user add new input devices at any time?

- **Research Question 2** - Can the interface be generic and flexible enough also to control other electronic devices?
  Is the process of adding devices only possible for input devices? Would it be possible to communicate and control devices other than the wheelchair?

These questions are at the base of the investigation work, and throughout the document, they are referred to as RQ1 and RQ2, respectively. They were relevant for several stages of the document and served as a justification for several architecture decisions made.

1.4 Implemented Solution

The proposed solution is multimodal interface but an adaptive and configurable one, with a plethora of features like a robust input-output system, a node system to (re)configure input actions and a built-in simulator. Although the interface tailored for an intelligent wheelchair, the system implemented is generic enough, responding to input events and triggering output actions, providing an easy solution to add any input or output device. The current implementation already accounts with several types of inputs, and the user can use them in any way they want to trigger any available action in the multimodal interface.

The implemented node system allows for configuring the input-output mappings, by reading input events to compose input sequences and use them to trigger an output or output sequences. These sequences can be inputs (or outputs) made in parallel, in series or combinations of both of these. It provides a lot more versatility in controlling the systems attached to the interface, by allowing the user to choose the actions he is more comfortable or able to reproduce correctly.

The interface has a channel system, where the user can use any action to activate or deactivate channels. These channels control the actions associated with them, deciding whether or not they should be active. Combining this with all the combinations for the input sequences explained above creates near unlimited ways of configuring the interaction.
Introduction

The built-in simulator provides high quality, realistic environment to evaluate the multimodal interface performance. It resembles an indoor environment with the common obstacles that it might pose, as doors, furniture, and interaction with objects and lights. This way, the several input devices implemented in this project can be tested in the simulator quickly, and its performance measured.

The interface should also have a built-in programming mode, that allows record input sequences performed by the user as input sequences. This method allows for a quicker remapping of actions without the need to configure the interface outside.

1.5 Structure of the Thesis

This initial chapter introduces the problem explaining how current technology in intelligent wheelchairs can improve by using multimodal interfaces. It also goes through the motivation in developing this project, which is coupled tightly with the problem. It then evidences the research questions this project is aiming to study and answer to and showing how the solution is expected to be in the final iteration.

The second chapter (Chapter 2) goes through the current state of the art and related projects about the topics of Human-Machine Interfaces, Human-Computer Interaction, Human-Robot Interaction, User Interfaces, Adaptive Interfaces, and Multimodal Interfaces.

The third chapter (Chapter 3) reviews the already made investigations and projects about intelligent wheelchairs, what they are, and how they can help the patients. It then focuses on a project with similar goals developed previously in this institution, the Intellweelsproject, and explains how this project improves the work already developed.

The next two chapters focus on the development of the project. The first one (Chapter 4), is more about technology and architecture decisions, like the choice of the framework, what are the systems and devices in the context of this project and an overview of the systems in place, how they connect to each other and how they control the information flux of the multimodal interface. The second chapter (Chapter 5) focuses more on the implementation, explains in more detail what is a device for the multimodal interface, what can it do and a description of the several devices implemented. It also describes the built-in simulator created to evaluate the interface performance.

After the project description comes the chapter 6 describing the interface evaluation protocol. An experiment was made with several test subjects in different environments and under different conditions to evaluate the performance of the multimodal interface. The performance of the subjects is analyzed as well as their impressions transmitted via a survey. This chapter describes the evaluation protocol and provides an analysis of the obtained results.

The final chapter (Chapter 7) summarize all the work done in this dissertation project. It succinctly describes the work done, the strengths, weakness, opportunities, and threats of a project like this, if the results were good, the main downsides of the decisions made and an overall overview of how the project is. There is also a proposal of improvements to the prototype that should be taken into consideration when continuing the development of the Intellweelsproject.
Chapter 2

Human-Machine Interfaces

"The term “user interface” refers to the methods and devices that are used to accommodate interaction between machines and the human who use them." (Zhang 2010). In a mechanical system, such as a vehicle, a robot, or an industrial machine, the user interface can be seen as a communication provider between the system itself and the human who controls it. A user interface has two essential functions: transmitting information from the user to the machine and transmitting information back to the user from the machine [Zha10]. Machines provided men with the tools and the power to accomplish more, but as they became more complex, controlling them became harder. So, naturally, the interfaces started being developed and implemented as a mean of hiding the complex actions and information the system has behind simpler, more intuitive access points (or interfaces), allowing a more easy and powerful way to control, monitor or diagnose them from simpler access points like screens and keyboards [Zha10]. Interfaces between the human and the machine are what makes them machines usable, interactive, and useful.

The topic about "interaction" is vast, and this being a project about a multimodal interface, interaction is at its core. For this specific case, three topics cover the interaction subject, one covering the interaction between a human and a computer (section 2.1), another one covering more specifically the interaction between a human and a robot (section 2.2), since the multimodal interface is to be applied to an intelligent wheelchair, and a third section covering the interfaces (section 2.3), including adaptive (section 2.3.1) and multimodal (section 2.3.2) ones.

2.1 Human-Computer Interaction

Stuart Card refers to the human-computer interaction (HCI) as a simple topic. He gives the example of a person trying to accomplish a given task, such as writing an essay or flying an airplane. The person could realize those tasks without any computer, using a quill pen and ink for writing the essay or hydraulic tubes to manipulate the plane’s systems. However, using an human-computer interface resembles the process of their task, and that is what makes HCI distinctive [JC14]. HCI
Human-Machine Interfaces

studies aim to bring human-centered technology discoveries and provide better ways of interaction [CBP16].

Technology is evolving, new methods of interacting with computers have been developed and improved throughout the years, and it is now possible to use voice commands, touch, gestures, eye movement, and neural activity to interact with computers [HY17]. Computers are also not the same as they used to be. We now have wearable computers, portable computers, smartphones, virtual and augmented reality devices [HY17], and all of those developed some new way of interaction between the human and the computer. Their hardware also improved, with small form-factor machines having powerful processors, increased amount of memory, powerful graphical processing units, faster and primary storage devices with higher capacities, as well as better peripherals with higher quality, like trackpads, webcams, microphones and touchscreens in laptops [HY17]. Mobile devices like tablets, smartphones or smartwatches also have suffered from similar improvements, having multi-core processors, large amounts of memory, fast storage, multi-touch screens, global positioning system (GPS) sensors, near field communication (NFC), fingerprint sensors, high-resolution cameras, and high-quality microphones and speakers [HY17]. All this creates new possibilities of providing more natural and feature-rich interactions, and provide an overall better experience for the end user.

Due to a large number of interaction methods available in current days, it is essential to know which ones are the most important and the most reliable. A survey conducted by Churchill, Bowser, and Preece in 2016 on more than 600 people concluded the most essential areas in HCI education, being interaction design the most relevant subject, experience design the most important topic, desktop, mobile, and tablet are the most critical interfaces and gesture, keyboard, sensors, and touch are the most essential input modalities [CBP16]. This survey also highlights the importance of learning fundamental human characteristics and people’s activities to shape the interaction systems to elevate people’s abilities and, consequently, their creativity [CBP16]. This point of view agrees with what Harberland wrote in Human Factors in Information Technology, saying that the interaction between humans should be studied and analyzed, so more natural and intuitive forms of interaction could be designed more easily [Hab99].

Other vital areas of HCI that has undergone several improvements are the virtual and augmented reality. Virtual reality is old technology that enables the user to be fully immersed in a computer-generated reality and be able to interact with it [Lux16]. However, it was not until recent improvements in computer hardware that made it a viable product for the average user, promoting breakthroughs in what was previously considered critical limitations. It led to several studies on how to improve its performance and immersion while reducing its size, weight, and power consumption. Virtual reality is also used for clinical evaluation treatments and is used in the treatment of psychological disorders [RMB11] [SR01], to provide training for people with special needs and disorders through therapeutic computer games [Lux16].

Augmented reality, or mixed reality, is the transposition of virtual reality to the real world, layering computer-generated reality on top of the real world imagery [Cra13]. This way, the world around the user can be intractable and manipulated, especially with the use of GPS, providing
location-based data to the user [Lux16]. This technology is usually present in mobile devices, although there are glasses explicitly designed to provide improved augmented reality experiences. Like virtual reality, it also has been used in health care, not only for training professionals but also to help patients by, for example, providing real-time virtual coaching [Lux16]. Both augmented reality and virtual reality can be considered as a type of multimodal interface, providing the user with many more natural ways of accomplishing a given task [Kip13].

2.2 Human-Robot Interaction

Humans have interacted with computers throughout the years as if they were independent social actors (computers as social actors, CASA). [KBS+16]. The CASA model states that users personify the machines, even if they do not resemble a human at all [KBS+16]. This idea of a human-like machine is due to the human being a social actor and instinctively search for social interaction, a human-like interaction [KBS+16].

Human-robot interaction (HRI) is a substantial vast topic that has been giving more attention in recent years [VKC19], that goes through physical, cognitive, and social interactions between humans and robots. Robots are a way of expanding the human capabilities, either by the increase in strength, in speed or other special characteristics that would otherwise be difficult for a human (like manipulating objects in tiny spaces) [VKC19]. Improving on the interaction between the human and the robot is directly related to improving the way the robot is controlling, and by that, we are improving how much the robot can expand the human capabilities and provide a useful presence. Several approaches were taken to make robots more human-like, such as humanoids [KMO+08], rescue and assistant robots [Mur04] [RDBB05], vehicle navigation [MM17] or collaborative tasks [KH04].

We can split the metrics about human-robot interaction into six different categories [VKC19]: Mission effectiveness, human behavior efficiency, human cognitive indicators, human physiological indicators, robot behavior efficiency, and collaborative metrics. While these metrics are generic and should adapt to each specific task carried out by the robot, they help to give an overview of how human-like the robot actions are. It is also essential to measure the interaction design concepts, like Level of autonomy, nature of the information exchanged, the structure of the team, adaption and how to accomplish the task, as they give vital and fair ideas on how to do the interaction in different HRI scenarios [VKC19]. The interaction taxonomy should also be analyzed considering, among others, these factors [VKC19]:

- Task type: if the task should be performed by a delivery robot, rescue robot.
- Task criticality: what could happen if the task is performed incorrectly, for example, an intelligent wheelchair failing to detect stairs, can the human life be compromised?
- Task morphology: should the robot have a human-like appearance, an animal-like one, or should it be more functional like a car.
Human-Machine Interfaces

- Task decision support: what kind of information does the control unit receive, should the decision rely only on automated models.

HRI is still a relatively new field, where its guidelines depend a lot on the context and environment the robot is in [VKC19]. Despite that, the studies already conveyed proved that a good HRI is about robots learning to interpret human emotions and desires, and also being able to express emotions depending on the situation [VVJ16]. Emotions, especially, are a topic on increasing importance in health care scenarios, especially when dealing with older adults [KBS+16].

2.3 User Interfaces

User interfaces are at the core of the interaction between a human and a computer system, and they are arguably one of the most important factors to the success of an intractable system [KMSB02]. Despite how good the system is, the interface is what makes it usable, and if the interface cannot provide a good, intuitive experience to the user, then the system also becomes not usable [KMSB02].

Interfaces development should consider the relationship between the architecture and the design of a system [DR13]. Nowadays, there are several types of interfaces, being graphical user interfaces (GUI) and web user interfaces the two most common [Zha10]. These interfaces exist in systems like human-robot communication, road vehicles, medical equipment, aircraft and aerospace shuttles [Zha10]. Being one of the most critical areas in HCI research, GUIs are present in almost all computer systems that have a high level of interaction [DR13]. From the time a user turns on a computer, he interacts with some GUI. They are common due to their widespread among all devices and systems, but they have gone under much work and suffered a considerable amount of changes in paradigms and concepts since their early days until the modern user interfaces we know now and are accustomed to [MM13].

User interfaces tend to easy the interaction by providing context and action sensitive options. For example, if a user wants to save a document, the save button should not be enabled until the user makes any changes, and there is something to save [AS03]. Although this easy of use also comes with downsides: a GUI exposes all the available functionality. Meaning that if a system is very complex, the interface must be well designed or it can provide a bad user experience. In contrast, command line interfaces provide an equally high degree of control without overwhelming the user with all the options but offering a poor user experience.

GUI’s technologies can be classified in categories taking into account their structure, data binding, behavior, data formatting, style, and input validation [MM13]:

- **Form-Oriented, Relation Database-Coupled Frameworks**[MM13]:
  Form-oriented user interface design is very successful, largely because they resemble the relational database system structures.

- **Template-Based Approaches**[MM13]:
  The concept of building and structuring user interface layouts resorting to markup languages...
and templates. A very successful approach due to being the most used on the web. It allowed creating dynamic user interfaces compatible across all browsers and operating systems because the interface exists in the markup language instead of being written in native code.

- **Object-Oriented, Event-Driven Approaches** [MM13]:
  This approach compares user interface design with common object-oriented principles commonly found in programming languages. This concept takes advantage of having a library with several components, helping compose the user interface by combining and superimposing several of these components.

- **Hybrids** [MM13]:
  Hybrids are a new trend in user interface development that combine templates and expression languages with object application programming interfaces (which means they combine template-based approaches with object-oriented approaches). Its primary focus is modern web user interface development.

- **Declarative Frameworks** [MM13]:
  These also are a recent trend that combines markup languages with object notation ones. It is are similar to a hybrid approach, but it is usually more commonly found in mobile and desktop platforms.

- **Model-Driven Approaches** [MM13]:
  This approach takes a model of the system interactions to generate a basic user interface for runtime interpretation. These models usually are described in Unified Modeling Language (UML), entity-relationship or other similar modeling languages.

- **Generic User Interfaces** [MM13]:
  A much smaller section when compared to the other user interface technologies described here. This approach consists of deriving the user interface from the data, task, or any other similar application model.

### 2.3.1 Adaptive Interfaces

With the appearance of the new forms of computing devices like smartphones, tablets, and wearables, the information that was usually only present in desktop-like screens needed to adapt to these new devices. However, these devices offered a much more convenient and personal experience, and thus, the interfaces should reflect that. One of the ideas to personalize the experience relies on the use of Adaptive User Interfaces (AUI) [PL15].

AUI design is conceptually hard because it involves constant monitoring and analysis, as well as a deep understanding of the context, the user and his environment in runtime, and adapt to the new parameters without any user intervention [PL15]. Requires a strict set of pre-defined rules as well as a base of understanding to make the proper changes to the user interface [PL15]. However, when done right, these interfaces can provide a much richer experience, specially tailored to user
preferences and needs. Personalized experiences help to improve the usability of the interface, facilitating the interaction, and providing a higher level of user satisfaction [HUHMB18].

Researchers have described a few adaptation rules, with the help of user experience (UX) experts or system designers [HUHMB18]. They considered several adaptation categories like user characteristics, disabilities, cognition level, and culture. For example, one interface can leverage the meaning of a particular color in a given culture, hide complex actions and widgets that are more useful for more power users from novice ones, or change the font size if the user has poor vision [HY17].

There have been several proposes for adaptive user design, from migratable and plastic user interfaces [BDB04], 3-layer architecture [LRBA10], context-aware adaptation using TRIPLET framework [MV13], model-driven approach with Malai [BB10] or CEDAR [ABY16] or adaptive user interface services with Egoki system [GMA15]. These frameworks provide a good starting point to build AUIs, but they provide little to none support to handle user feedback, which is an essential point in the project of this dissertation.

2.3.2 Multimodal Interfaces

Multimodal interfaces provide a more transparent, flexible, and efficient interaction between the human and the computer [RLL04]. While they provide a more natural interaction, closer to the one between humans, they also provide robustness to the system, by increasing the redundancy of the input methods or by complementing the information [RLL04].

They are not just one more tool to improve "interaction" between the user and the system, but a very effective one. These improvements bring better productivity and usability of the system, by being able to combine several distinct types of inputs, for example, ones that require physical contact (keyboard, mouse, joystick, trackpad, etc...) with ones that does not require physical contact (like gestures, speech, eye tracking) [KY18].

This kind of interfaces helps in exchanging information between the system and the user controlling it by providing better experiences based on several principles [KY18]:

- **Intuitiveness** [KY18]:
  Provide more natural ways of interacting with the system, without needing to instruct the user on how it works.

- **Ergonomics** [KY18]:
  Allows the use of the interface (and consequently the system) by various users with different preferences and needs.

- **Friendliness** [KY18]:
  Collects user input and tracks his actions without interfering with normal human behavior, avoiding accidental actions.
Human-Machine Interfaces

- **Reliability** [KY18]:
  Providing several input methods for the same actions allows the user to choose the one he is most comfortable with, maximizing the accuracy and robustness.

- **Efficacy** [KY18]:
  Allows for a faster and shorter number of steps to accomplish an action, reducing the required time and effort.

- **Universality** [KY18]:
  The interface is usable by several categories of different users, from physically and mentally disabled, to older adults or users who need or want to have a fast, reliable way of interacting because they are under some critical task.

- **Multimodality** [KY18]:
  Allows for several and convenient types of tools, giving the option to the user to choose which one he prefers based on his preferences or needs.

Multimodal interfaces have several applications to enhance the quality of life of their users. For example, older people like to live in their own houses but are exposed to some risks if they choose to live alone. To overcome those risks, they could connect to several biological signal sensors, like Electrocardiogram (ECG), Electroencephalogram (EEG), Electromyogram (EMG) or Electrooculogram (EOG). It is one of the aspects where multimodal interfaces can be a more practical approach, because they can handle all different types of data as input and use them to convey better actions to make, rather than having several systems monitoring each of the sensors and making decisions based on single data inputs [KMS17].

### 2.4 Conclusions

Human-Computer Interaction is a key field for this project, since a multimodal interface has, in its essence, an interaction factor between the computer and the person to control it. It is important to study the existing methods of interaction, and those that stand out for being exceptionally important, because of their intuitiveness or easiness to master. Taking this into account, the study conducted in [CBP16] allows this project to improve its focus when developing the multimodal interfaces, by concentrating its efforts on the more important areas of HCI.

HCI is tightly connected to HRI, although HRI has a more human aspect to it because the interaction is with an actual, existing robot that can have more resembles a human or other living being than a standard computer. Since this project aims to implement a multimodal interface in a real intelligent wheelchair, HRI study is essential to improve and evaluate the interaction between the user and the wheelchair.

Having an interface that is aware of the context and the possible actions is also very important. It can be useful in situations where the user of the wheelchair wants to go backward, but there is an
obstacle or stairs that the user is not conscious, providing safety and better interaction experiences by taking out the responsibility for critical actions from the user.

Combining all this in an adaptive and multimodal interface for a system designed specially for people with disabilities like an intelligent wheelchair can provide a lot more means of interaction between the user and the wheelchair, but more importantly, more convenient ways of controlling it, since traditional, that every healthy person can use, can be impossible for some users.
Chapter 3

Intelligent Wheelchairs

People with impairments, disabilities, or special needs rely on wheelchairs to have a higher degree of autonomy. The appearance powered wheelchairs (PW) brought immediate improvements to the overall life quality of their users since using a manual wheelchair can be impossible for some users. However, even PWs can be hard to maneuver depending on the user. For some, controlling the wheelchair with a single joystick may prove to be an impossible task or at least a doable task with the desired degree of accuracy [LL17]. In an attempt to aid help with that, there have been several implementations of forms of control since the appearance of PWs that resort to alternative forms of input: head or chin joysticks, sip-and-puff or control through thoughts ([MKCB14], [PKB14], [DDF+13], [SDD+14], and [dRM14]).

Although PWs are a good step towards individual autonomy, providing improved mobility and independence, they are a mere technological shift from the manual wheelchairs, and while improving a lot on the usability and life quality of the user over the manual wheelchairs, they still do the same or similar functions. Here is where intelligent wheelchairs stood out. They mark a paradigmatic shift, not just a purely technological one [PWA+11].

3.1 Intelligent Wheelchairs

Elderly or disabled people might find it difficult to drive a PW, due to the reduced capacity to perform certain maneuvers that require a certain degree of precision or agility. They also might have reduced vision and increased reaction time, making the obstacle detection and avoidance a complicated task [VDH+10]. Intelligent wheelchairs (IW) or Smart Wheelchairs (SW) are one of the most effective ways to solve this and similar issues that PW users might be facing. The idea is to improve on existing PW technology by adding degrees of intelligent systems to enhance further the users’ ability to control the wheelchair [VDH+10].

What this means is the IW can aid in daily maneuvers like avoiding obstacles, driving through a doorway or dock beside a table [VDH+10]. The intelligent part derives of the equipped hardware,
Intelligent Wheelchairs

being sensors, cameras and other input sensors connected to an intelligent controller, generally a computer-based system, and tweaked to perform a set of particular tasks [HN19]. These tasks can easily be maneuvers the IW user has to do frequently, thus providing better autonomy and independence. There are clinical surveys that support the usefulness of these IWs as can be seen in [FLS00] and [VDH+10].

There have been several implementations of IWs in the last 3 decades:

- **Madarasz Autonomous Wheelchair** [Dia15]:
  Developed by Madarasz in 1986 [HH93], this project had the purpose of creating a computerized system to control a wheelchair in a completely autonomous way. This wheelchair had a microcomputer, a digital camera, and an ultrasonic scanner. It would then operate based on the information collected and took action through a joystick. A prototype can be seen in figure 3.1. The camera was used to make the recognition of the surrounding to obtain an approximate location of the wheelchair, based on existing, pre-collected reference points. The ultrasonic sensor detected obstacles that might exist in the environment, and the microcomputer processed all the collected data and acted based on the information and the joystick commands.

![Madarasz wheelchair prototype](image)

Figure 3.1: Mandarasz wheelchair prototype

- **Omnidirectional Wheelchair** [HH93]:
  H. Hoyer e R. Hölper developed this project in 1993 to create a system with a high level of functionality and flexibility. It tried to achieve that by having an open source architecture approach, having a modular structure, where each module was having a certain degree of autonomy, like trajectory planning or object recognition. The interface module allowed
Intelligent Wheelchairs

inputs by joystick, terminal and voice command [HH93] [Dia15]. In figure 3.2, a prototype of this wheelchair can be seen.

![Figure 3.2: Omnidirectional wheelchair prototype](image)

- **Two legged Intelligent Wheelchair** [Dia15]:
  An innovating project, very different from the previous two mentioned here, because of its appearance and approach, as it was armed with two legs (see figure 3.3 [SWKK94] [Bar11]. Its legs were used to support the wheelchair, but could also be used as robotic arms, allowing to accomplish some daily tasks like climbing stairs or grabbing objects.

![Figure 3.3: Two-legged intelligent wheelchair prototype](image)

- **NavChair Intelligent Wheelchair** [Dia15]:
  Presented by Simon Levine in 1994, NavChair was designed to improve on the adaptability and ease of use. It had a computer, ultrasonic sensors, and a joystick [dF13]. The prototype, visible in figure 3.4, shared the IW’s control between the system and the user, prioritizing the user commands if necessary [Bar11]. The main goal of this project was to give the IW system the capability of avoiding mobile obstacles, like other robots [Dia15].a
Intelligent Wheelchairs

Figure 3.4: NavChair intelligent wheelchair prototype

- Tin Man I and Tin Man II Wheelchairs [Dia15]:
  Tin Man I was developed by Miller and Slak in 1995 and tried a different approach from the other ones by not focusing on mobile robots but robotic wheelchairs [Dia15]. Using as starting point a commercial wheelchair from Vector Wheelchair Corporation, it equipped 5 different types of sensors: contact sensors, proximity infrared sensors, sonars and compass, and flux control encoders [Bar11]. It also incorporated a microprocessor that helped the user to control the IW, with the possibility to use three custom operation modes: User-guided with obstacle avoidance, move along a predefined trajectory or moving from a starting location to a destination.
  Three years later, the Tin Man II project emerged intending to optimize the previous version, reducing the contact sensor reliance, improving the user interface, increasing the operational speed and creating a new system that helped in the testing and validation [HH93] [Bar11]. This iteration also had new built-in functionality, like recharging the battery, following walls, remembering the trajectory, going back to the starting point or passing through doors [dF13]. We can see a prototype of the Tin Man I and II in figure 3.5.

  Figure 3.5: Tin Man I and Tin Man II intelligent wheelchair prototypes

- MAid Wheelchair [Dia15]:
  The MAid project, visible in figure 3.6, was developed in 1998 with the primary goal of helping old or impaired people to perform daily tasks more easily, resorting to mobile obstacle avoidance mode. For that, it had two operation modes: One fully automatic, better used in crowded environments, and another semi-automatic, to be maneuver the wheelchair
Intelligent Wheelchairs

in small spaces [Dia15]. Tests done with this IW had positive results, as the wheelchair avoided obstacles several times in congested environments [Bar11].

Figure 3.6: MAid intelligent wheelchair prototype

• FRIEND I and FRIEND II Wheelchairs [Dia15]:
  With the idea of aiding impaired people that had difficulties in operating upper limbs, the FRIEND I project appeared in 1999, developed by the automation institute of Bremen University. To achieve it, they equipped it with a 6 degrees of freedom (DoF) MANUS manipulator, connected to a control architecture very complex, also used to control the wheelchair (visible in figure 3.7). Since the target audience of this chair was users who have difficulty in moving their arms and fingers, the input interface implemented was a screen and voice commands [Dia15].
  Later in 2005, FRIEND II succeeded the FRIEND I wheelchair, enabling to install better hardware and implementing new software architecture. The goal of this iteration was to enable the IW to interact with other intelligent devices in the environment, like avoiding collisions with them, moving to them and even controlling them [Dia15].

Figure 3.7: FRIEND I intelligent wheelchair prototype

• ACCoMo Intelligent Wheelchair [Dia15]:
  The ACCoMo prototype (visible in figure 3.8) was idealized in Chiba-shi, Japan artificial intelligence division, by Tomoki Hamagami and Hironori Hirata in 2004. The idea behind this
Intelligent Wheelchairs

project was to allow an IW to avoid obstacles, cooperating with other IWs, and collaborating with its user [Bar11]. For that, it would resort to the learning and evolution of ACCoMo agents, through experiences in virtual and real environments [Bar11]. It was projected to move indoors, avoiding collision with another IWs in the same space [Dia15].

Figure 3.8: Intelligent wheelchair prototype based on ACCoMo agents

- **Intelligent Wheelchair Driven by Brain Waves [Dia15]:**
  This concept was launched by H. Lakany at the United Kingdom’s Essex University in 2015, differentiating from the others by resorting to the reading and processing of brain waves to control the IW. This reading was done by recording electroencephalograms (EEG) when the user was controlling a wheelchair using a joystick and using this data to develop a method to extract the space-time properties on the brain waves oscillation [Bar11]. Later, the brain waves collected would be compared against the recorded ones and associated with an action. It is possible to see in figure 3.9 a robotic arm and an IW controlled through this technology.

Figure 3.9: Intelligent wheelchair prototype and robotic arm controlled by brain waves

- **MIT Autonomous Intelligent Wheelchair [Dia15]:**
  The IW visible in the figure 3.10 and developed by the Massachusetts Institute of Technology (MIT) tried to increase the IW autonomy by implementing sensors that would analyze
Intelligent Wheelchairs

the environment surrounding the wheelchair, implementing an interface with voice commands, software calculated location in a closed space and software controlled motors to drive the wheelchair. The biggest innovation presented is its location capabilities, that allowed the IW to move from one point to another, by mapping the building where the IW is and applying path-finding algorithms [Dia15].

![Intelligent wheelchair prototype developed by MIT](image)

Figure 3.10: Intelligent wheelchair prototype developed by MIT

- **Intellweels [Dia15]:**
  This project began in 2007 at Faculty of Engineering of University of Porto, with the second iteration in 2012, brought a high-level multimodal interface to control IWs. Since this project is the one this thesis is contributing to, it is discussed in further detail in section 3.3.

3.2 Interfaces for Intelligent Wheelchairs

Several types of interfaces were already analyzed and discussed in the chapter 2, with the section 3.1 mentioning some of the implementations of those interfaces in IWs. Therefore, this is a small section, only exposing some projects that propose or implement special interfaces in wheelchairs:

- **NavChair project shared control of the IW between the system, and the user [LBJ+99].**
  Some of the performed tests required the users to drive the wheelchair in an environment that would require various control transitions to reach the destination [dF13].

- **Tan [TSP01] proposed an intelligent chair that could analyze the posture of a user resorting pressure patterns characteristic of specific seating postures [dF13].**

- **Another propose came from Rao [RCJ+02], having the IW focusing on video-based human interaction, with demonstrated usability results by Parikh et al. [PRJ+03] in 2003 [dF13].**
Intelligent Wheelchairs

- Another project was TetraNatura [ACGG02], aiming at developing a low-cost automatic driving system to be implemented in IW, requiring low effort from its users to maneuver it [dF13].

- Purki et al. proposed a multi-agent based control. [PEM02] to balance the control in a given environment, taking into account the user preferences [dF13].

- In 2011, Rousan and Assaleh [ARA11] introduced a wheelchair with three types of inputs: a joystick, directional buttons, and voice commands.

### 3.3 Intellweels Project

Intellweels project (visible in figure 3.11) is a collaborative project between Faculty of Engineering of University of Porto, INESC TEC, and the University of Aveiro. Born in 2007, tries to create an IW visualization and simulation platform by designing and specifying the entire system, from the architecture to the wheelchair, the battery of sensors, the interaction methodologies and even simulators and applications like games. The main goal of this project is to develop this platform in a way that can adapt to any commercial powered wheelchair, and different kind of user needs [dF13] [Dia15].

![Intellweels project prototypes](image)

Figure 3.11: Intellweels project prototypes

The initial prototype developed the interface between the computer system and the hardware existing in the wheelchair, like the sensors and motors, being developed over the years to allow new ways of interacting with the system. Despite its high-quality graphics and excellent performance, the differentiating factor is the adaptive multimodal interface, having autonomous planning and driving abilities and the possibility to cooperate with several other intelligent devices [dF13] [Dia15].
Intelligent Wheelchairs

With regards to the input hardware, IW counts on a high number of devices, allowing for a more robust and versatile way of collecting environment information joystick, keyboard, head movements,faction expression detection, and voice commands. It also endowed a very complex but discrete sensory system, composed of ten sonars, two encoders assembled on the wheels and two webcams [BPMR09]. The idea was to have a robust set of sensors, while maintaining the base design of a regular wheelchair, to prevent the formation of psychological barriers [dF13]. All this connected to a system controlled by a multimodal interface, flexible and adaptive to the user (an example of the interface is present in figure 3.12. Each user could also personalize this interface, with each profile having different actions for each command. For example, for a user blinking the right eye could mean to the wheelchair to move to the right, while for other the same command could make the wheelchair start moving forward [dF13] [Dia15].

![Intelligent Wheelchair Interface](image)

Figure 3.12: Example of the multimodal interface created for the Intellwhells project

This platform’s architecture follows a multi-agent system paradigm, enabling more significant interaction between the system and other intelligent devices by allowing a flexible addition and modification of the modules [Dia15]. The four primary agents presented in this project are visible in figure 3.13.

- **Interface Agent:**
  This agent is in charge of the interaction between the user and the IW [Dia15].

- **Intelligence Agent:**
  It is responsible for the planning, cooperation, and collaboration of the IW actions with the rest of the environment [Dia15].

- **Perception Agent:**
  Has the job of choosing the best set of sensors to collect the information at a given space
Intelligent Wheelchairs

Figure 3.13: Architecture of the multi agent systems

and time, and use the received data to calculate information like location and surrounding mapping [Dia15].

- **Control Agent:**
  It handles control actions like obstacle avoidance or wheel control [Dia15].

### 3.4 Conclusions

Intelligent wheelchairs are, without a doubt, a great step towards giving great autonomy and independence to elderly and impaired people. These types of wheelchairs mark a change in the technological paradigm, being a bigger leap in innovation than the evolution from manual to powered wheelchairs.

Throughout the years, several implementations of IWs have developed and explored several alternatives to the typical input devices found in non-intelligent wheelchairs, and have implemented intelligent systems to aid in the control and avoid obstacles that might be in the path, with the help of increasingly more powerful computers and better sensor sets.

All the projects highlighted in this chapter evidenced a new approach by tackling existing problems in previous implementations of IW. Some implemented speech recognition to allow voice commands, others went through an open source architectural route, there were examples of input redundancy allowing multiple types of inputs to command the wheelchair, cases where the wheelchair was equipped with an arm or legs to help in even more challenges. All these projects
Intelligent Wheelchairs

started with good ideas and accomplished good implementations, and all are relevant to where the state of IWs is now.

The Portuguese project Intellweels was a very ambitious attempt to create a new concept of IWs, equipping the wheelchair with robust control systems, multiple sensors, high-quality displays, and a very versatile multimodal interface while keeping the wheelchair look close to stock as possible.

The proposed solution in this document tries to improve on the existing multimodal interface by enabling a higher degree of customization and flexibility, allowing to program new commands using the multimodal interface itself without needing external help and allowing input sequences, with either sequential or parallel commands.
Intelligent Wheelchairs
Chapter 4

Development of the Multimodal Interface

After researching the state of the art relative to the interaction in chapter 2, as well as intelligent wheelchairs in chapter 3, this chapter, describes the development details of the multimodal interface. Firstly, in section 4.1, an explanation is provided about the choice of the development technologies and the hardware where the system ran. The next section provides an overview of the concept of systems and devices in the context of the multimodal interface. After that, there is a detailed explanation of the several systems that compose the multimodal interface.

The section 4.3 refers to the input system, covering all the required components like input signatures in section 4.3.1, action triggers in section 4.3.2, input events in section 4.3.3, input groups in section 4.3.4 and input sequences in section 4.3.5.

The section 4.4 covers the system that is responsible to execute actions on devices. This system is composed by the output signatures described in section 4.4.1, the output groups detailed in the section 4.4.2 and the output sequences portrayed in section 4.4.3.

The next three sections describe more simple systems that only depend on themselves and on the input or output systems to provide their functionality. These are the parameter system in section 4.5, the graphical user interface system in section 4.6 and the channel system in 4.7.

4.1 Development Tools and Technologies

The choice of the development technology should verify a few requirements not to limit the system’s theoretical concept. The technology should allow to create interactive applications, be performant, even under multiple input processing nodes, allow for threading to execute actions in parallel and have an extendable API to allow devices from multiple sources and technologies to communicate with the system. Besides that, it was also desired not to constrain the deployment platforms (at least should support Windows, Linux, and Android) and should have an open license and be open source code, ensuring the continuity of the project even if the original development of the technology ended for any reason.
With this in mind, a few options were excluded: .NET languages since they are very tied to Windows systems. Web technologies, while cross-platform and open source, would require a lower level integration for the devices with the server or through sockets, and neither seemed ideal. Lesser known languages might discourage other developers from continuing the project and building devices upon this system. Java and C/C++ looked right solutions: They integrate well with many other technologies, are cross-platform and well performant. Despite that, building interactive, graphical applications with C++ or Java can be unhandy, especially under short time constraints.

The requirements of being interactive, performant, cross-platform and since the interface itself would benefit from an input processing system, using a game engine or a game framework seemed increasingly appealing, especially after considering the downsides of more traditional approaches.

After searching many game-related frameworks, the open source Godot Engine was picked to develop the multimodal interface. This framework captivated due to its nature as a game engine and its licensing. Being a game engine, Godot is interactive, responsive, portable, and feature-rich, with the possibility of creating high-quality visuals, leveraging an already powerful input system, and can support deploying to many operating systems. Since it also features a C/C++ interoperability, as well as robust networking implementations, it means practically any external system or device can interact with a Godot project with just a thin layer in between at most.

The Godot Engine has many programming languages available to build the functionality, and the better part is that a single project does not need to use a single language. GDScript (a custom language developed by the Godot developers), C#, C, C++ and even visual scripting (a visual, node-based scripting system) can all be used together with several other, community supported, languages.

Besides that, Godot has a powerful GUI system, completely themeable, with labels, buttons, text fields, syntax-highlighting text editors, tabs, containers, menus, dialogs... The GUI of the Godot Engine itself is made resorting to the same system provided to the Godot projects. It helps to show how powerful the GUI can be.

The project created used the built it language GDScript, which has syntax and functionality very close to python, but it is tailored to better real-time processing, fixing issues that other scripting languages have like threads, garbage collection or native types. This language also allows fast iteration times, real-time hot-reloading (changes in the source code affect a running project in real-time) and cross-platform deployment without having to recompile any native modules or modify any of the source code.

The following sections in this chapter describe the several systems that make up the multimodal interface, as well as their interaction with each other and the functionality they expose that any device can take advantage. There is a diagram at the end of the chapter to better understand the events flow and how the systems communicate with each other (figure 4.1).
Figure 4.1: A diagram of the various systems in the multimodal interface and how they communicate.
4.2 Systems and Devices

There are two main concepts in the context of the multimodal interface: systems and devices. They are separate concepts but complement each other while keeping the concerns separate.

Systems are the backbone of the multimodal interface and the main channel of control. They control the input flow, the activation of output actions, the GUI, passing data from input devices to output devices, converting the parameter data types into other data types or adding tabs or buttons to the main user interface. They are what makes the multimodal interface and its functions.

Devices, however, are what builds upon the systems. They provide the functionality the user ends up interacting with by leveraging the functionality exposed by the multimodal interface. They communicate with each other and the user through the input/output systems and the GUI systems. The devices should represent a real device, like a keyboard, joystick or the wheelchair, and use the functions of the multimodal interface to provide the configuration and interconnection it provides.

4.3 Input System

The input system is probably the most complex system of the multimodal interface. Internally, it is responsible for saving, loading, and updating the inputs file on disk (for making the input sequences persistent) and by building a node tree of input groups. Input groups represent the input actions the user must do to produce an output action. Section 4.3.4 explains in more detail the functionality of the input groups. The input groups are laid out like a node tree, creating more input groups for the next sequence iteration. When a sequence reaches its end (which means the input group recognized the last input, but it has no child input group), the output group communicates with the input system informing that it found an input sequence.

The input system also contains two timers that are triggered at specific intervals depending on the input actions provided by the user through the connected devices. One of these timers is to control the triggering of actions that have the same beginning, but one has more extended sequence than the other, for example, pressing A+B or pressing A+B+C. When the user presses A+B, one of the input groups in the tree informs that it found the action, but the action cannot be triggered immediately because the user might want to produce the action A+B+C. So the input group responsible for the sequence A+B+C informs that there is still a possible sequence. The input system then starts the timer to wait for the next input. When the timer reaches its end, the sequence A+B is triggered; If the input system receives another input event during this timeout, the input system automatically cancels the timer. The other timer is responsible for controlling what is considered a parallel input action and a sequence input action. Since this timer is at the core functionality of the input groups, it is described in section 4.3.4.

The input system also detects if the current input sequence matches any of the saved sequences. If none of the input groups communicate that a possible input sequence was found, the input system resets and treat the next input as a new sequence.

The input system has various components to compose all these tasks:
4.3.1 Input Signatures

The input signatures are a representation of a single input action. They are a data class that stores the input device name, the action name and the type of action (press, release or motion), using these three factors to create a unique identifier of the event. Besides that, the input signature also contains the parameters the input device sent. When a device generates an input event from the input system, the system creates an input signature characteristic of the given event, holding all the parameter data it contains. This signature is then passed to all the active input groups to compare with the signatures it holds. The possible action types cover three separate cases:

- **Press:** The press event simulates a one-time event. It can either be a button press or a voice command recognized. All the input actions that can either be active or not should use the press event.

- **Release:** The release event is very similar to the press event. It represents an event that can either be active or not. The purpose of the release type is to distinguish button presses from button releases. Since it is the device, action name, and event type that categorizes an input event, button presses could produce confusing results. This way, an input sequence might be key press + key release, and the interface reflects that it is the same key, with the same action name from the same device. It just is a different event type.

- **Motion:** Motion events are a special kind of events that can be triggered multiple times per second. Input events in which their parameters vary over time should prefer this event type. For example, a joystick motion could trigger a press event when the joystick is over 50% off the center and a release when it is less than 50%. However, this would only provide a binary interaction. With a motion event, every time the joystick position changes, the event can be triggered, passing the new value as a parameter. This method allows for more fine-controlled actions, like the speed of the wheelchair.

4.3.2 Action Triggers

These are a group of functions that input devices call to interact with the input system. These register the input action in the system and initiate the matching process of the input system. There are three functions available for this, one for a button press (can also be used as single action), other for button release (to complement the button press when the action has the same identification) and third for motion inputs (inputs that can be triggered multiple times per second, such as moving a joystick). Each of these functions must receive a device identification, an action identification, and an optional dictionary of parameters. These are then converted to the adequate input signature and sent to the input groups through an input event.
4.3.3 Input Events

An input event is simply an extension of the Godot class InputEventAction, modified to include an input signature and parameters. This event is created and triggered by the action trigger functions and parsed by the input groups.

By extending a Godot class of input events, the multimodal interface’s input system can leverage an already powerful input system provided by the Godot engine. This way, the custom input event created by the multimodal interface can be passed to the engine’s input system, and by taking advantage of the way Godot transmits the events to the scene-tree provides a performant way of dealing with multiple possible input sequences.

4.3.4 Input Groups

Although already described in the previous sections, input groups are a vital part of the input system. Each group represents the actions that must be triggered simultaneously. Since it is nearly impossible to trigger two events at the same time, a timer is used to create a threshold for the sequence of events and considers them to be parallel. This timer only activates in the input groups that have more than one input signature registered, and the input signature is recognized. If there is no sequence with parallel actions, the timer is skipped to provide a more responsive interaction.

There is a parent input group for each action (that can symbolize one or more input events at the same time), and each input group can have a child input group representing the next combination in the sequence, and the input group is responsible for parsing the input event sent by the input system and validate the signature against the ones it has. As long as all the input events are sent within the previously mentioned threshold, the input group considers as if all the input signatures received to be parallel and after the threshold, the parent input groups enables the child input group and the current one disabled to prevent wasted time in processing the input events. If there is no child input group, this means that the current input group met the sequence for the current action, and the last output group informs the multimodal interface that the output action can be triggered.

4.3.5 Input Sequences

Input sequences are just an arrangement of input groups. If each input group represents one or more input events in parallel, an input sequence represents a sequence of input groups that should be in the sequence they are. As an example, for the input sequence composed by the key A and B at the same time and then the key C, the sequence would have two input groups. The first input group would have the input signatures of the key A event and key B event, and the second input group would have an input signature of the key C event.

4.4 Output System

The output system has a structure very much like the input system, although is it easier to describe. As the input system, the output system is also responsible for saving, loading and updating the
outputs file on disk (for making the output sequences persistent) and by building a node tree of output groups. However, unlike the input actions, the output actions must be registered when the corresponding device is loaded. This registration consists of an identification of the device and the output, a reference to the method to be called when that output is to be activated and what parameters it accepts.

When the input system detects a full sequence, it triggers the corresponding action in the output system. The output system then passes the action to its output groups, which triggers specific events on output devices according to the saved output signatures. The components that make the output system are the following:

### 4.4.1 Output Signatures

These are a representation of a single output action. They are a data class that stores the output device name, the action, a reference to a method call and the parameters the output action accepts.

When the output signature corresponds to one that should be triggered, the output system calls the method saved as a reference in the output signature. Within this method call, the device is free to do any action.

### 4.4.2 Output Groups

The output groups, like the input groups, constitute a tree-like structure of the output actions to perform given a valid input sequence. Each output group can have any given number of output signatures and executes them in separate threads in parallel.

### 4.4.3 Output Sequences

Output sequences are, like the input sequences, an arrangement of output groups. When an output group is called to trigger its output signatures, the output group does it in parallel and wait for all the threads. When all threads finish, the output group calls the next output group in the sequence.

### 4.5 Parameter System

The parameter system is a simple system that is composed only by a parameter signature component, diluted in the input system and output system functionality. The parameter signature stores what the value is and where to which device to send it. The job of the parameter system is to monitor all the recognized input actions, read their parameters, and according to the output action they are triggering, associate the value with a corresponding output signature.

It also provides an automatic conversion for some datatypes. For example, if the output signature requires a string and the input signature sends a float, the parameter system converts the float to string. Some data types might lose precision with this method (like converting from float to string), and some parameters might not even be able to convert at all (like a dictionary to an integer).
4.6 Graphical User Interface

The Graphical User interface is a system that while providing less functionality, might provide more interaction to the user. The main GUI is composed by a header that contains the exit and option buttons, among other information widgets, and a tab container. The GUI system then provides methods to the devices to hook up to this tab container, the options menu, and the quick action bar. This way, any device can add a main interface by adding a tab to the tab container and a completely new screen by registering a new entry in the options menu. Some quick actions can also be mapped to buttons and registered to appear in the quick actions bar (at the top of the multimodal interface, at the left of the exit button in figure 4.2).

![Figure 4.2: An overview of the Multimodal Interface GUI.](image)

There are some auxiliary methods to display new screens, providing stack-based navigation. That way, the user can easily navigate forwards and backward in the stack and have a consistent, logical way of interacting with the GUI. Finally, this system also provides access to dialog boxes, such as text input or confirmation, that follows the rest of the GUI theme.

4.7 Channels

Channels are a simple but compelling concept. They can be seen as pipes between the input and output systems to allow or block the communication and can be activated and deactivated through output actions. This system allows the user to have a set of input combinations to control one thing, and by changing the channel through actions (or having the interface changing it automatically,
based on values from sensors such as location) and have the same inputs producing completely different results.

4.8 Conclusions

In this chapter can be seen the reasons behind the technological decisions. Despite being a relatively new game engine, Godot proved to be a powerful and stable tool throughout the entire development and supported the complex interconnected structure of the multimodal interface.

The several systems that compose the interface provide the core functionalities of the multimodal interface, and they are where all the device connect. All the functionality from the input/output processing to the main screen can be changed entirely by the devices the user chooses to plug in. The GUI example in figure 4.2 is a device named dashboard but serves to show the possibilities by leveraging the systems built.

The input system is a very complex system and is responsible for most of the logic, allowing a more versatile and complete input processing. It connects directly the output system which is responsible for triggering the correct output actions in the correct order, and the parameter system that sends the information from the input events to the output events.

The GUI system is an isolated system that provides all the graphical functionality of the interface, from tabs to stack-based navigation to dialogues.

The channel system also ties to the input system, by verifying if the action to be triggered is in an active channel. It is a simple concept but enables powerful combinations.

All these systems together create a very adaptable and expandable functionality where any developer can extend. Some devices were created to evaluate the interface performance and can be seen in more detail in chapter 5.
Development of the Multimodal Interface
Chapter 5

Implementation

This chapter describes the implementation of the devices in the multimodal interface. The devices are what the user uses to produce actions, is what the user sees in the computer screen and is what the user controls through the output actions.

It begins with the section 5.1 explaining in more detail what is a device in the context of the multimodal interface and the sections 5.1.1 to 5.1.5 describe how more simple devices interact with the multimodal interface and take advantage of the implemented systems.

Section 5.2 describes the node system, a more elaborated device that is used to modify the action mappings through the multimodal interface. This device uses all the systems of the multimodal interface and is a part of the multimodal interface concept of being configurable. Moreover, it is where the implementation of the recording mode is.

Section 5.3 illustrates the simulator device, also an elaborated device that uses the output, parameter, and GUI systems and is the device used in the user experiments described in chapter 6.

5.1 Devices

Devices hook to the multimodal interface through the implemented systems and provide an easy way to expand the multimodal interface interaction capabilities and functionality without having to change the way its design or configuration. Devices are loaded in runtime after all the initialization of the systems, and the user can add or remove them freely, adapting to the system the user wants to compose.

Any device can hook up to the input system by calling the action triggers in the input system, to the output system by registering output actions and to the GUI system by registering tabs and options. During the development of the project, several devices were implemented to test the functionality of the multimodal interface.

The following devices interaction can be seen in the diagram of the figure 5.1.
5.1.1 Keyboard

The keyboard device represents a generic keyboard input device. It reads the scancode of the key pressed or released and sends it to the input system by calling the respective action trigger.

It was the most used device in development since it was the easiest available. It also allows for very complex input sequences, since it can create many different input actions, one for each key press and release.

5.1.2 Joystick

The joystick is an input device that can have any number of button presses/releases as well as motion events. Another big advantage of using Godot is that leveraging the already built in input event system to implement the joystick mapping, any human-interface device recognized by the operating system should work without any modification. It also has a configurable dead-zone, allowing to ignore small movements the user might make with the joysticks.

5.1.3 Speech Recognizer

A Windows-only input device because it uses the Windows speech recognition engine under the hood.

It is also an excellent example of interoperability between technologies. When the multimodal interface initializes the device, it starts an executable file and listens to connections. The speech recognition software then connects to the Godot engine and transmits the recognized sentences.
Implementation

This speech recognition engine requires a dictionary of the sentences and uses it to compare with the recorded sound. Since the multimodal interface should allow configuring all aspects of the devices, it also can change the sentences to be recognized through an options panel (connected to the GUI system).

Any recognized sentence that has confidence higher than 80% is sent to the input system as an action press and can be used as any other input event in an input sequence.

5.1.4 Console

The console is a simple output device using during the development to test the functionality. It outputs messages to the console and accepts different kinds of parameters. It is beneficial in the development of new devices but has no useful functionality to the end user.

5.1.5 Head Motion Sensor

The head motion sensor is a custom input device, designed and built specifically for this project, capable of detecting the pitch, roll, and yaw accelerations of the user’s head. It needs a gyroscope to read the orientation, so a printed circuit board was designed to accommodate a gyroscope and a WiFi module mounted on a helmet (schematics and layout in appendix C).

Also, a custom 3D printed support with fittings to a helmet allows the mounting of the printed circuit board and a battery pack with an on/off. The final result can is visible in figure 5.2.

![Circuit board and helmet with 3D printed support](image)

Figure 5.2: Circuit board on the left and helmet with 3D printed support on the right.

The gyroscope communicates the values to the WiFi module via a serial connection, and the WiFi module hosts a web socket server and broadcasts the values to all connected clients. The multimodal interface then takes advantage of the Godot’s built-in network features and creates a web socket client to connect to the WiFi module. When the web socket client receives data, it converts to a value between -1 and 1 for each axis and sends it to the input system, through the motion action trigger.
Implementation

This device also features a dead-zone similar to the joystick implementation, and a calibration procedure. The calibration resets the position value and uses the next 10 readings to set a baseline and a standard deviation. This procedure happens when the device is connected to the multimodal interface, but can also be called through the options menu or from a quick action button.

5.2 Node System

The node system is one of the most complex user interface devices that is used to configure the mapping of the input actions. It leverages all the functionality provided by the input and output systems, such as reading the output devices, creating and deleting actions from both the input and output systems or hooking up to the input processing.

![Node system example.](image)

This functionality allows providing an incredibly intuitive way of mapping new actions because this device can record input sequences and list all the available output devices. It was built upon the Godot’s graph system, a system that provides visual nodes that can be connected. This way, even the most complex action can be mapped by any user without requiring a high technical skill.

5.3 Simulator

The simulator is capable of using all the top-level functions of the interface and proving the usefulness of a system like the one developed, in a real-life context. This simulator allows evaluating the functionality and usability of the interface.

The simulator is a very sophisticated device (since it integrates with the multimodal interface the same way other devices do) that combines the capabilities of input, output, and user interface devices. The simulated environment is a 3-dimensional indoor environment, where the user controls the first-person character and has several interactive objects. This device takes advantage of many of the Godot Engine functionality, since the simulator is in its essence, a simple game. The simulator registers the main viewport as a tab to the multimodal interface and registers several output devices for controlling the lights and doors per division or interacting with the object that is
Implementation

in front of the player. Any of these actions can be associated in action mappings through the node system device and leveraging the channel system and all the devices. This way, the user could navigate around the house with the controls he is the most comfortable.

Figure 5.4: Aerial view on the left and a first person view of the simulated environment on the right

With this in mind, several combinations were made to allow the user to move around the environment, turn on and off the lights, and open and close the doors:

- Using only the keyboard and key presses
- Using only an Xbox One controller with button presses and joystick motions
- Using a combination of head movement, voice commands, and keyboard

In all the settings, the user can move into all the rooms, turn on and off all the lights, open and close all the doors and interact with two televisions. In the last setting, that would probably reflect a more real use, the user used the head to look around, press a button on the keyboard to switch the channels and use the same head motions to move the character and use his voice to turn on/off the lights, open/close the doors and interact with the televisions.

5.4 Conclusions

This chapter provides an overview of the several devices created to interact with the multimodal interface and leverage the functionality of its systems.

Some devices are more simplistic and provide more simplistic functionalities. For example, the keyboard or joystick leverage the built-in input system in Godot and provide the simple functionality of triggering input events in the input system.

Other devices like the node system or the simulator are rarer even in a real-world scenario, but they show to what extent the interactions between the several systems and a single device can go. The node system, for example, disables the input system of the multimodal interface completely and takes over the input processing to provide a robust recording mode. The simulator uses advanced rendering techniques of the Godot engine to render the scene to a viewport, and inject it in a tab pane, to keep the GUI main structure intact.
Implementation

Besides the exemplified devices, many others can be connected to the multimodal interface and even provide more autonomy. For example, a collision detection sensor could disable the channel where the movement mappings are communicating, preventing the wheelchair for colliding, or a light sensor could turn on the lights automatically when the ambient light decreases below a certain amount.
Chapter 6

Evaluation

The proposed evaluation experiments pretend to analyze the efficiency, comfort, reliability, and easiness of the multimodal interface developed when controlling a device with a movement mechanic similar to a wheelchair. More specifically, they pretend to show if the answers to the research questions stated at the beginning of the dissertation are valid, whether or not the interface can adapt to different user needs and if it integrates with other devices.

For this purpose, this project counts with 30 user experiments on the system built after each volunteer signed an informed consent (in appendix A). Each experiment consisted on performing a sequence of daily tasks, resorting to a built-in simulator also developed alongside this project, from which several data was taken to be analyzed. In this chapter, we present the evaluation protocol in section 6.1, a small description of where and how the experiments were conduct in section 6.2, an analysis of the collected data in section 6.3 divided into two subsections, section 6.3.1 about the sample demographic characterization, and section 6.3.2 with the analysis of the experiments themselves, and the final section 6.4 with a small discussion of the results. These experiments should help validate the solution against the RQ1 and RQ2 introduced in the chapter 1, and consequently help to validate (or disprove) the created solution.

6.1 Evaluation Protocol

The multimodal interface system described in chapter 4 and chapter 5 was designed to aid people with different needs, or even different preferences, to achieve the same goals using the easiest and most comfortable configuration the user can conceive.

It is this challenging task that the developed system proposes to solve. By providing robust interconnected systems and a straightforward programming interface for plugging devices while being versatile to the point of being able to reconfigure the action mappings in real-time, the system aims to provide many different ways of achieving a set of goals.

With this in mind, we conducted 30 experiments with volunteers where they have to perform a sequence of daily tasks in the developed simulator using the multimodal interface in the
Evaluation

background to provide access to all the devices and action mappings. The evaluation runs on a simulated environment mimicking an in-doors scene laid out like the figure 6.1

![Figure 6.1: Annotated aerial view of the simulator](image)

The division naming association with the annotations is the following:

- A. Entrance
- B. Bathroom
- C. Hallway
- D. Bedroom
- E. Secondary Bedroom
- F. Living Room
- G. Dining Room
- H. Kitchen

The tests were designed to question about the usability and reliability of the multimodal interface and its input devices on similar tasks like movement and interaction with the environment. With the help of the tests, we would prove that the multimodal interface is, in fact, apt to be used by people with different levels of technological comfort (RQ1). It would also prove that the interface can interact with different devices of the environment (RQ2).

**Course:** A -> C -> D -> C -> E -> C -> F -> G -> H -> G -> F -> C -> A

**Interactable objects:**
Evaluation

- **Lights:** A example of a location-agnostic device that can be interacted with (turned on or off) from anywhere inside the house.

- **Television:** A example of a location-specific device that can only be interacted with (turned on or off) when the user is in front of it, looking directly at it.

- **Doors:** A example of a mixed device, that can be interacted with (opened or closed) by being in front of it or by using a command for a specific door.

**Action sequences:** Open door (D) -> turn on television (D) -> open door (E) -> turn on lights (E) -> turn on television (F) -> turn on lights (H)

All the tests follow the same structure to keep the variation to a minimum. Beginning in point A (entrance) with the televisions and lights turned off and all the doors closed. The user would then need to move to room D (bedroom) from room C (hallway), open its door, enter and turn on the television (example of a location-specific device interaction). He would then proceed to room E (secondary bedroom) through room C, open its door, enter, and turn on the lights while being on the inside (example of a location-agnostic interaction). Then would need to move to room F (living room) through room C and turn on the television in this room (another example of location-specific device interaction). After that, the user would move to room H (kitchen) through room G (dining room) and turn on the lights at this division (another example of a location-agnostic interaction). Also, to finalize, the user would move back to the first entrance door at room A. When the user reaches the door, the test ends. The user would do this test 4 times, each time with different input devices:

- **1st Task** - The first time, the user would only be able to use the keyboard as an input device. He would need to move around, look up and down, and interact with objects only using key mappings.

- **2nd Task** - This second time, the user would only use a joystick with axis and buttons to perform the same tasks.

- **3rd Task** - At the third time, the user would execute the test using voice commands for interaction (and eventual channel switching) and either keyboard or joystick for the movement. That would allow for more complex interactions, but would probably result in slower results. Each interaction should be made resorting to voice commands, and if they fail three times, the user is allowed to use the corresponding command on his/her complementing device (either joystick or keyboard), and the voice command is assumed to have failed at that interaction.

- **4th Task** - This time, again performing the same actions as all the other tasks, the user would be able to use any of the devices available, and combine them any way they like. This task has the same concept as the previous ones, but they serve to evaluate the user opinion on the multimodal interface after the first experience, where he might not know how the system
Evaluation

works, and after the second interaction. This time he is allowed to configuring the interface since in the first 3 tasks the input-output mappings come pre-configured. This time, the user is allowed to change every single configuration, including the voice command sentences.

In every task, the simulator would save the user position in the virtual environment each second (if the user has changed his position from the previous read) and provide a timestamp of the said position and each interaction with the environment. It also saves the user commands used and the corresponding output actions triggered. That would provide enough data to evaluate each device individually and combined, and reconstruct the user steps to evaluate the difficulties felt at each step, in each task. Between the tasks, the user would also answer a survey (in appendix B) asking to grade between 1 to 5 the several categories, being 1 the less favorable quality:

- **Easiness** - Was it easy to navigate/interact with the environment?
- **Comfortability** - Was it comfortable to navigate and interact with the environment using the given device(s)?
- **Reliability** - Was (were) the given device(s) consistent in the kind of action they produced?
- **Responsiveness** - The given device(s) felt quick to perform the action?
- **Intuitivity** - Was it intuitive to produce the desired action, or did it require some thought process?
- **Autonomy** - How much autonomy did you felt with this setup?
- **Freedom** - Did you felt this setup provides a high degree of freedom?
- How easy it felt to perform the required steps?

Besides these questions, the user the user was invested to identify the biggest difficulty felt and one yes or no question with optional explanation:

- Do you think in a real-world scenario, would be harder to perform these steps? (why?)

In the final task, the survey included some question about the introduced modification to the mappings made. One question asking which devices did the user used and others related to the input configuration:

- How easy it was to create or modify input sequences and input groups using the recording system? (from very difficult to very easy)
- How easy it was to create or modify the action mappings? (from very difficult to very easy)
- How useful did you find the difference in the input devices? Do you think they cover a large variety of possible input actions (for example, button presses, motion events)? (from poor to excellent)
- How efficiently did the interface felt when used with several devices at the same time? (from poor to excellent)
6.2 Test Sessions

All the 30 test sessions occurred into two major sessions: 13 at the university, where most volunteers are students, investigators or teachers with a higher education experience and the rest at a home office, with a more diversified user base, both in age, education, and profession, with the gender being almost balanced in both sets.

Despite that, and after a careful analysis of the data, it makes more sense to present the results of the sessions from both environments together, since the conclusions were almost the same, and dividing the experiment results would result in a longer chapter without any real gain.

Besides the normal sessions that follow the protocol described in section 6.1, two more experiments were conducted using as the movement controller the head movement sensor detailed in section 5.1.5. We did not use this device in the other test sessions for two main reasons:

Using this device to control the movement of the simulator is harder than any other provided method. It is probably harder to use it in the simulator than in a real-life scenario because moving the head in the simulator makes the user loose eye contact with the computer screen. Because of that, the experiments would be a lot longer if they included an experiment with the head motion sensor. It would greatly reduce the availability of the volunteers.

Since this device requires a Wi-Fi connection to work properly, it was dependent on the physical environmental conditions. While it worked reliably at the home-office, it did not work at the university, even with a dedicated Wi-Fi network shared only between the head motion sensor and the host. After many tries, and taking into account the first reason, it was decided not to include this device in the major sampling pool.

6.3 Data Analysis

Throughout every test session, at the end of each task, the users provided their feedback on how the experiment was for them, how responsive the devices were and the overall performance felt by the multimodal interface.

This way, the tasks can also be evaluated by their duration of the experiments and the actions performed. This way the tasks can also be evaluated by their duration, and we can analyze the device performance.

We divided the collected information into two sections to ease the data analysis, section 6.3.1 about the demography of the users and section 6.3.2 about the analysis of the data concerning the experiments on the simulator and its log.

6.3.1 Sample Demographic Characterization

This first section gives an overview of the user demography and how comfortable they are with technology pretended to be used to create a experience baseline.
Evaluation

The gender was tried to be the most balanced possible to prevent influencing the results. Since this system is to be used by both men and women, it was important to keep the gender close to 50%, and as can be seen by chart 6.2, it was successfully achieved.

![Gender Chart](image)

Figure 6.2: Demography: Gender.

About the profession and educational experience, the sample pool is a bit biased, since almost half of the experiments happened at the university, where there is not much diversification. As can be seen by chart 6.3 and chart 6.4, most of the volunteers were students (40%) or professors (13.3%), and half of them had at least college/university (50% + 10% PHD) education.

![Profession Chart](image)

Figure 6.3: Demography: Profession.

When asked about how easy they find technology and how comfortable they are with technological devices like computers, smartphones, smart televisions or tables, most of them answered they find them easy or very easy, with only 6.7% answering they find them hard to interact, as can be seen in chart 6.5.
Figure 6.4: Demography: Educational Experience.

Figure 6.5: Demography: How easy technology is.
Evaluation

Now related to the interaction devices like console controllers, keyboards, joysticks or touch-screens, the responses varied a little more, as chart 6.6 shows, ranging from very difficult (6.7%) to neutral (13.3%), to easy (40%) to very easy (40%), with difficult being the only option not chosen by anyone.

![How easy do you find interaction](image)

Figure 6.6: Demography: How easy interaction devices are.

### 6.3.2 Experiment Reports

This section describes the results of the experiments, either by user feedback when answering the survey after each task and by using the data collected by the simulator while the tasks were running.

We grouped the charts by device, and the device name was concatenated to be shorter. The correspondence of the names in the label of the charts to the tasks are the following:

- **KB**: Keyboard task - This task was completed using only the keyboard.
- **JS**: Joystick task - This task was completed using only the joystick.
- **SR**: Speech recognition task - In this task, the interactions with the doors, televisions, and lights happened through speech recognition. The user chose the movement device between the keyboard or the joystick. The users were instructed to try the voice commands three times before proceeding using the physical button(s) on the devices they were using to control the movement. It is important to note that 1 volunteer did not feel comfortable in speaking English and therefore did not perform this task.
- **CT**: Custom task - The volunteers completed this task with a configuration chosen by them, where they could choose any or all of the three devices, and change any mappings or voice recognition sentences.
Evaluation

When asked about how easy it was to perform the tasks with each device, the vast majority felt it was easy or very easy in all the tasks (100% for speech recognition and the custom tasks, 93.3% for the keyboard task) except for the speech recognition (51.7%), as can be seen in chart 6.7.

![Easiness - Was it easy to navigate/interact with the environment?](chart)

Figure 6.7: Experiments: Easiness.

It is also important to note that in the custom task, where the user could change the configuration to its preference, was considered the easiest by all users. It has the highest percentage of very easy, with the remaining considering easy.

![Comfortability - Was it comfortable to navigate and interact with the environment using the given device(s)?](chart)

Figure 6.8: Experiments: Comfortability.

The chart 6.8 reveals similar results, with the vast majority of the users feeling very comfortable in all the tasks (63.3% for the keyboard, 73.3% for the joystick, 63.3% for the speech recognition and 83.3% for the custom task). It is interesting to note that, despite not being considered easy, most users consider the speech recognition option as a comfortable one.

On a final note on this chart, it is important to note once again that the custom task, tailored to the user preferences, provided the most comfortable experience for accomplishing the tasks.
Evaluation

Concerning the reliability in chart 6.9, the keyboard was, without a doubt, the most reliable device, with joystick being a close second. Both of these options had 100% of the results in either often or always, which is remarkable. Speech recognition proved to be very unreliable with values all over the spectrum, with contrast with the custom task, which also had very positive results.

The evaluation of the responsiveness of the system is shown in chart 6.10, and reveals results that are very similar to the reliability. The keyboard is the device that felt the most responsive, with joystick being again a close second. The custom configuration made by the user in the custom task felt a little behind the joystick, and the speech recognition was very unreliable, a trend which has been observed in several topics now.

Although different concepts, it is easy to understand that a device that does not respond consistently to the user commands is also not reliable, and therefore, the correlation. Moreover, while the custom tasks were not as reliable and responsive as the keyboard or joystick, it might be a
compromise the users choose to do between comfort and reliability/responsiveness.

In regards to intuitiveness, the chart 6.11 shows that all the combinations were very intuitive. The custom task was the most intuitive with the highest percentage of "always" intuitive assignments (90% for the keyboard, 70% for the joystick, 20.7% for the speech recognition, and 73.3% for the custom task). Speech recognition, while performing poorly, felt very intuitive, which is understandable since the voice commands used to perform the actions were directly related to the action name. Keyboard and joystick felt the less intuitive with both having at least one classification of being intuitive sometimes.

Figure 6.11: Experiments: Intuitiveness.

About the autonomy, all the options provided a good level of autonomy, with the custom configuration being the better one. The keyboard was also considered to be very good to excellent in providing autonomy. The joystick and speech recognition felt more limited than the other two, as can be confirmed by chart 6.12.

Figure 6.12: Experiments: Autonomy.
Evaluation

The analysis of the chart 6.13 shows that the custom task was again the one that provided the best results, providing freedom at least often 100% of the experiments. Joystick also had excellent results, and this time was the keyboard task and the speech recognition task to feel more restrictive than the other two.

Now related to the task itself and the steps the user had to perform, almost all of them felt that they were easy or very easy after each task. After the speech recognition task, the users felt that it was not as easy as the other three, but overall the simulation did not provide a high challenge, as can be seen in chart 6.14.

The users were then asked to imagine if they were in a real scenario, and had to use a wheelchair. If they had to perform these steps in a real environment similar to the simulated environment using only the input devices and configurations they used to accomplish the task, did
Evaluation

they think it was harder than in the simulator. After looking at chart 6.15, it can be seen that most of them thought it would be easier to perform the same actions in a real environment.

Figure 6.15: Experiments: More difficult in a real-world scenario.

When asked why, most of the answers were about the physical implications of being in the real world, that a simulated environment could not transmit. For example, the perception of the physical space is better in the real world, the vertical movement of the camera in the simulator was associated with an action, while in the real world would only imply to move the eyes or the head.

Before the final custom task (named CT in the charts/tables) the users were asked if they wanted to change any of the input mappings, speech recognition sentences, and for those who did want to change something, had four other questions asking about what they thought of the input mapping system.

Figure 6.16: Experiments: How easy it felt to create or modify input sequences and input groups using the recording system.

The first question was regarding the recording system built in the multimodal interface, which
Evaluation

allows the users to perform the input action they want to associate with the output, and the multi-modal interface would detect the events if they were in sequences or parallel and compose a graph with the recordings.

As can be seen in chart 6.16, most users felt it was very easy to perform these modifications resorting to the recording system.

Figure 6.17: Experiments: How easy it felt to create or modify the action mappings.

Related to creating new actions or editing existing ones, modifying both the input and the output actions, the answers were more disperse, while most also considered the system very easy to interact.

The third question was regarding how useful they thought the variety of the input devices available were. Do they think the provided input devices allowed for extensive coverage of input events (like button presses, motion events, etc...)?

Figure 6.18: Experiments: Usefulness was the variety of input devices.
Evaluation

According to the previous chart 6.17, most of the users thought the coverage was very good or excellent, with only one saying it was good.

The final question was about how did they felt the multimodal interface performed when re-programmed, and used with multiple devices and multiple actions at the same time.

![Multiple device efficiency chart](image)

**Figure 6.19:** Experiments: System efficiency with multiple devices.

Once again, the answers were very positive, with the majority of the inquiries answering the interface felt excellently efficient.

The devices the user chose to use in this last task was also accounted, and the distribution can is visible in chart 6.20.

![Which devices did you use?](image)

**Figure 6.20:** Experiments: System efficiency with multiple devices.

Despite feeling the speech recognition was unreliable and unresponsive, most users preferred to use it again in the last task, maybe due to the comfort and intuitiveness it provides.

55
Evaluation

The joystick was also widely chosen to execute the last task, with 20 out of 30 of the volunteers using it.

To finalize the data analysis, the duration of the experiments was read from the log files of the simulator and compiled into the table 6.1. This time, the two experiments with the head motion sensor were also included, under the column named HMS.

<table>
<thead>
<tr>
<th></th>
<th>KB</th>
<th>JS</th>
<th>SR</th>
<th>CT</th>
<th>HMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>0:02:37</td>
<td>0:02:29</td>
<td>0:02:36</td>
<td>0:02:17</td>
<td>0:06:17</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>0:00:57</td>
<td>0:01:33</td>
<td>0:00:33</td>
<td>0:00:57</td>
<td>0:01:04</td>
</tr>
<tr>
<td>Minimum</td>
<td>0:01:31</td>
<td>0:01:19</td>
<td>0:01:46</td>
<td>0:01:16</td>
<td>0:05:32</td>
</tr>
<tr>
<td>Maximum</td>
<td>0:04:54</td>
<td>0:07:27</td>
<td>0:03:38</td>
<td>0:04:36</td>
<td>0:07:02</td>
</tr>
</tbody>
</table>

Table 6.1: The average, standard deviation, minimum and maximum values of the experiments in seconds.

The figure 6.21 includes an aerial view of the simulator with one experiment from one volunteer that claimed to find technology "very easy" (in red) and one that claimed to find it "difficult" (in green). These experiments also correspond to the global maximum (7:27) and minimum (1:26) duration, respectively.

Figure 6.21: Simulator aerial view with two positional overlays in red and green.

Despite the results throughout the survey, the speech recognition task did not take much longer than the others, although the user experience felt worse and less responsive than in any other, and having the smallest standard deviation from the set.

The custom configuration continued with the excellent results, having both the lowest average and the absolute lowest time in all the experiments.

The head motion sensor task did the same steps as the others and took, on average more than the double of any of the other tasks.
Evaluation

Even to someone that claims to find technology hard, and whose experiment took more than 5 times the duration of the other, the user in the green example in figure 6.21 was able to finish all the steps and was able to produce similar behaviors than the user in the red example.

6.4 Conclusions

The user tests performed provided enough data to evaluate the multimodal interface performance in some daily tasks in an indoor environment. The number of steps to perform in each task was kept low to reduce the time per task, but it provided enough examples.

There were two examples of location-specific devices (televisions), two examples of location-agnostic devices (lights in the bedroom and kitchen) and two examples of mixed interaction devices (the main bedroom door and the secondary bedroom door).

The head motion sensor not working correctly at the university was unfortunate, but by reducing the time per experiment, it allowed to collect more data from more volunteers and provide more results, focusing more on the multimodal interface performance, usability, comfort, and ease-of-use rather than on the performance of the devices.

Doing an overview of the results and having assisted to all experiments, I was expecting the speech recognition task to take longer than the other ones, and the custom task to be distinguishably shorter than the others. Maybe if the experiment were longer, it would have been easier to tell the difference.

Despite that, the feedback provided by the users tells a different story. Also, being faster is not the best if you cannot provide a good user experience. When the user had the option to choose, the majority preferred the joystick together with the speech recognition, which is a versatile choice since it provides a reliable way of moving with a very intuitive way of interacting.

The last question was more controversial than expected, but the responses are quite impressive. While some consider the simulator a safe environment and the real world would provide more challenges, others consider the simulator to limit their senses and principally, their field of vision, and thus provide more difficult navigation than in a real scenario.

Although the difference in experience in using technology and input devices (individual user resources) has been possible for every type of volunteer to have successfully completed the proposed tasks.
Evaluation
Chapter 7

Conclusion

Wheelchairs have been a crucial element in today’s society by disabled people to move around. However, manual wheelchair requires strength and coordination to move around, and some impaired people just were not able to do it, requiring an assistant all the time they needed to move. With the technological development, a new technological shift appeared, marking the beginning of powered wheelchairs, reaching even further on the user base, by allowing people to move with a simpler interface, like a joystick. This change was a welcome improvement, and while it gave more freedom to people in wheelchairs, they were still limited to simple movement actions.

Nevertheless, powered wheelchairs inspired the creation of intelligent wheelchairs, a complete paradigm shift by introducing intelligent behaviors in the system like wall following, obstacle avoidance or autonomous trajectory calculation and movement. Still, these systems were more complex, and impaired people need simple interfaces so they can maneuver all the actions and trigger all the functions available in the system.

So, it was required to study various fields of interaction, from the interaction between a human and a computer to the interaction between a human and a robot, as well as the study of several types of interfaces, enablers of that interaction. Adaptive interfaces are essential in this kind of applications since intelligent wheelchair users have very different capabilities and needs. Multimodal interfaces are also crucial in more complex systems because they allow for input redundancy, increasing the compatibility with such non-identical types of users, and expanding the possibilities for external systems’ integration.

7.1 Project Overview

The present project was very ambitious, with that being a strength and also a weakness. It had a very well thought work plan with clear goals spaced in time, which was a positive indicator of the success. However, the time-frame was short, and it limited some developments and user experiments, discussed in section 7.2 that would give this project other feedback.
Conclusion

It features a very well laid-out structure with the systems providing the core functionality of the multimodal interface and the devices implementing the interactions. The architecture of the project provides an expandable solution not constrained by giving the possibility to add or remove devices at any time and reconfiguring the action mappings when needed. All this was using an open source, free licensed and easily accessible game engine, which can ensure the development of the project even if the game engine development shuts down.

The 30 user experiments provide a solid ground on how the system is performing and what people feel when performing actions with it. Despite having people with so many different backgrounds and ages, being able to perform the same actions with different input devices did present at least one comfortable, reliable and easy configuration to adapt to the users’ preferences.

To conclude, this system can aid its users to control existing technology, responding to a diverse and distinct number of input actions, and improving their quality of life. The modular system with a high level of modality and configuration, with the possibility to add input and output actions quickly, also provides a simple integration point to other systems means that RQ1 and RQ2 are validated and the system provides the expected functionality.

7.2 Future Developments

The user experiments revealed that the system is functional and can be used to perform daily tasks when controlling a device with similarities to a wheelchair. While many preferred the comfort and easiness of the speech recognition device, it was also considered to be unreliable and unresponsive. To improve on this, the voice recognition device should allow changing the recognition engine to others better suited to the user (could also be provided by implementing a new device). Another improvement point could be training the voice recognition engine to the user's voice and not use a pre-trained model.

The parameter system is efficacious and functional. With it, input devices can send information to output devices and, for example, control the wheelchair speed more precisely instead of being merely moving or stopped. However, the parameter system could use some improvements to provide more functionality. For example, some arithmetic operations could help to respond to some negative feedback from user experiments. Inverting axis or increasing the rotation speed could be solved by a simple multiplication operation. Another attractive functionality would be implementing some logic operations directly into the parameter connections like if operations. A third useful improvement is providing default values for output actions. This feature could significantly reduce the number of actions needed to control the device. For example, we could replace the move forward, move backward, and stop actions by a simple one called move and passing a default value of 1, -1 or 0, respectively.

Lastly, the multimodal interface uses the default theme provided by the Godot engine. While the theme does not limit the functionality at all, a more modern theme could make the system more appealing. The possibility to customize the theme would also be an excellent addition, with the option to export and import new ones.
7.3 The Future of Intellweels

Started with prototypes and investigation around the year of 2007, under the same name Intellweels, but now known as Intellweels 1.0, the idea of improving the life quality of cerebral palsy patients and certainly many others that require a wheelchair to move around started gaining shape. Fastfowarding to 2019, several studies, prototypes, projects, and solutions were developed to show and prove that the concept is good and achievable.

The Intellweels is a significant and ambitious project that promises to change the future of intelligent wheelchairs. But not limiting itself to the promises, near the end of the development of this project, the Intellweels received funding approval, ensuring this way the continuity of the development of the several projects under it. Having partnered with two companies and many institutions, and with the possibility of performing experiments with cerebral palsy patients, the project Intellweels is very well forwarded to success.

While this multimodal interface is a small part in a much bigger project like Intellweels, hopefully, it has been a good contribution to it, serving as a good starting point the study of the state of the art and to the subsequent developments of the multimodal interface.
Conclusion
References


REFERENCES


REFERENCES


REFERENCES


REFERENCES


Appendix A

User Experiences’ Informed Consent

TERMO DE CONSENTIMENTO INFORMADO

Vimos por este meio convidá-lo/a a participar no estudo de investigação do projeto Intellwheels 2.0 – Interface Multimodal Completa e Adaptativa para Cadeira de Rodas Inteligente, coordenado pelo Prof. Doutor Luís Paulo Reis, que está a ser desenvolvido por uma equipa de investigação da Faculdade de Engenharia da Universidade do Porto.

O estudo baseia-se na evolução científica e tecnológica na área de sistemas inteligentes, com o objetivo de desenvolver uma interface multimodal completa. Esta interface irá permitir o controlo da cadeira de rodas inteligente e também de múltiplos outros equipamentos e sistemas garantindo a multimodalidade não só no input mas também no output. Nesta fase do estudo, pretende-se realizar o reconhecimento do desempenho do protótipo da interface multimodal desenvolvida. Para tal será necessário testar a realização de diferentes tarefas, em ambiente simulado, com recurso ao uso da interface multimodal e aos dispositivos integrados. Trata-se de um estudo observacional com medição de tempos e facilidade de execução de tarefas através de software de monitorização, reconhecimento de voz e um capacete de deteção de movimentos da cabeça. Estão previstos 4 momentos de avaliação, sendo que o período de cada avaliação não excederá 5 min. Os participantes não serão sujeitos a qualquer procedimento invasivo para além do uso dos dispositivos, supra indicados. Será assegurado o direito de, a qualquer momento, desistir de participar no estudo. Será também salvaguardado o anonimato e a confidencialidade do participante (não haverá identificação nominal do titular, sendo apostado um código). Garante-se a confidencialidade e uso exclusivo dos dados recolhidos para o presente estudo, em condições de anonimato (não registo de dados de identificação), cumprindo os requisitos da Comissão Nacional de Proteção de Dados. Garantindo ainda que a identificação dos participantes nunca será tornada pública.

A participação será de caráter voluntário, não havendo quaisquer prejuízos caso não queira participar e/ou escolha descontinuar a sua participação. Não está contemplado qualquer ressarcimento ou remuneração para participação no estudo. Não haverá qualquer custo para o participante.

Será obtido Consentimento Livre e Esclarecido de todos os participantes do estudo. O consenso...
User Experiences’ Informed Consent

timento informado será obtido antes da inclusão do participante voluntário no estudo. Somente o consentimento de pessoas capazes de compreender e comunicar dúvidas sobre o estudo e/ou procedimentos, após lhes ser provida informação oral e escrita sobre o estudo e/ou procedimentos, será considerado satisfatório para a inclusão no estudo.

Em nome da equipa agradecemos a sua participação.

Investigador responsável: Prof. Doutor Luís Paulo Reis, lpreis@fe.up.pt

Assinatura: ................................................

Declaro ter lido e compreendido este documento, bem como as informações verbais que me foram fornecidas pela pessoa que acima assina. Foi-me garantida a possibilidade de, em qualquer altura, recusar participar neste estudo sem qualquer tipo de consequências. Desta forma, aceito participar neste estudo e permito a utilização dos dados que de forma voluntária forneço, confiando em que apenas serão utilizados para esta investigação e nas garantias de confidencialidade e anonimato que me são dadas pelo/a investigador/a.

Nome: ...........................................
Assinatura: ...........................................
Data: ................../................../..............

ESTE DOCUMENTO É COMPOSTO DE 1 PÁGINA E FEITO EM DUPLICADO:
UMA VIA PARA O/A INVESTIGADOR/A, OUTRA PARA A PESSOA QUE CONSENTE
Appendix B

Multimodal Interface Survey

Multimodal Interface Survey
* Required

1. ID *

2. Age *

3. Sex *
   Mark only one oval.
   - Female
   - Male
   - Prefer not to say
   - Other:

4. Educational Experience *
   Mark only one oval.
   - Elementary School
   - Middle School
   - High School
   - College / University
   - Master’s
   - Doctorate (PhD)
   - Other:

5. Profession *

6. How easy do you find technology? *
   Mark only one oval.
   1 2 3 4 5
   Very difficult
   Very easy

7. How easy do you find interaction devices? (like consoles controllers, keyboards, joysticks, touchscrene.) *
   Mark only one oval.
   1 2 3 4 5
   Very difficult
   Very easy
Complete the Keyboard Task

Keyboard Task
The steps were performed using only the keyboard

8. KB1 - Easiness - Was it easy to navigate/interact with the environment?
   
   Mark only one oval.

   1 2 3 4 5

   Very difficult  Very easy

9. KB2 - Comfortability - Was it comfortable to navigate and interact with the environment using the given device(s)?

   Mark only one oval.

   1 2 3 4 5

   Never  Always

10. KB3 - Reliability - Was (were) the given device(s) consistent in the kind of action they produced?

    Mark only one oval.

    1 2 3 4 5

    Never  Always

11. KB4 - Responsiveness - The given device(s) felt quick to perform the action?

    Mark only one oval.

    1 2 3 4 5

    Never  Always

12. KB5 - Intuitiveness - Was it intuitive to produce the desired action or did it require some thought process?

    Mark only one oval.

    1 2 3 4 5

    Never  Always

13. KB6 - Autonomy - How much autonomy did you felt with this setup?

    Mark only one oval.

    1 2 3 4 5

    Poor  Excellent
14. KB7 - Freedom - Did you feel this setup provides a high degree of freedom?

   Mark only one oval.

   1  2  3  4  5
   Never ☐ ☐ ☐ ☐ ☐ Always

15. KB8 - What was the biggest difficulty you felt while performing this task?

16. KB9 - How easy it felt to perform the required steps?

   Mark only one oval.

   1  2  3  4  5
   Very difficult ☐ ☐ ☐ ☐ ☐ Very easy

17. KB10 - Do you think in a real-world scenario, would be harder to perform these steps?

   Mark only one oval.

   ☐ Yes
   ☐ No

18. KB11 - Why?

   Complete the Joystick Task
   Joystick

   Joystick Task
   The steps were performed using only the joystick

19. JS1 - Easiness - Was it easy to navigate/interact with the environment?

   Mark only one oval.

   1  2  3  4  5
   Very difficult ☐ ☐ ☐ ☐ ☐ Very easy

20. JS2 - Comfortability - Was it comfortable to navigate and interact with the environment using the given device(s)?

   Mark only one oval.

   1  2  3  4  5
   Never ☐ ☐ ☐ ☐ ☐ Always
21. JS3 - Reliability - Was (were) the given device(s) consistent in the kind of action they produced?
   Mark only one oval.
   
   Never   1 2 3 4 5
   Always

22. JS4 - Responsiveness - The given device(s) felt quick to perform the action?
   Mark only one oval.
   
   Never   1 2 3 4 5
   Always

23. JS5 - Intuitiveness - Was it intuitive to produce the desired action or did it require some thought process?
   Mark only one oval.
   
   Never   1 2 3 4 5
   Always

24. JS6 - Autonomy - How much autonomy did you felt with this setup?
   Mark only one oval.
   
   Poor   1 2 3 4 5
   Excellent

25. JS7 - Freedom - Did you felt this setup provides a high degree of freedom?
   Mark only one oval.
   
   Never   1 2 3 4 5
   Always

26. JS8 - What was the biggest difficulty you felt while performing this task?

27. JS9 - How easy it felt to perform the required steps?
   Mark only one oval.
   
   Very difficult   1 2 3 4 5
   Very easy

28. JS10 - Do you think in a real-world scenario, would be harder to perform these steps?
   Mark only one oval.
   Yes
   No
Complete the Speech Recognition Task
Speech Recognition Task
The steps were performed using speech recognition in combination with keyboard or joystick

30. SR1 - Easiness - Was it easy to navigate/interact with the environment?
Mark only one oval.

1 2 3 4 5
Very difficult ○ ○ ○ ○ ○ Very easy

31. SR2 - Comfortability - Was it comfortable to navigate and interact with the environment using the given device(s)?
Mark only one oval.

1 2 3 4 5
Never ○ ○ ○ ○ ○ Always

32. SR3 - Reliability - Was (were) the given device(s) consistent in the kind of action they produced?
Mark only one oval.

1 2 3 4 5
Never ○ ○ ○ ○ ○ Always

33. SR4 - Responsiveness - The given device(s) felt quick to perform the action?
Mark only one oval.

1 2 3 4 5
Never ○ ○ ○ ○ ○ Always

34. SR5 - Intuitiveness - Was it intuitive to produce the desired action or did it require some thought process?
Mark only one oval.

1 2 3 4 5
Never ○ ○ ○ ○ ○ Always
35. **SR6 - Autonomy - How much autonomy did you feel with this setup?**
   *Mark only one oval.*

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poor</td>
<td></td>
<td></td>
<td></td>
<td>Excellent</td>
</tr>
</tbody>
</table>

36. **SR7 - Freedom - Did you feel this setup provides a high degree of freedom?**
   *Mark only one oval.*

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Never</td>
<td></td>
<td></td>
<td></td>
<td>Always</td>
</tr>
</tbody>
</table>

37. **SR8 - What was the biggest difficulty you felt while performing this task?**

   

38. **SR9 - How easy it felt to perform the required steps?**
   *Mark only one oval.*

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very difficult</td>
<td></td>
<td></td>
<td></td>
<td>Very easy</td>
</tr>
</tbody>
</table>

39. **SR10 - Do you think in a real-world scenario, would be harder to perform these steps?**
   *Mark only one oval.*

   | Yes | No |

40. **SR11 - Why?**

   

41. **Which devices did you use to complement Speech Recognition**
   *Check all that apply.*

   | Keyboard | Joystick | Head Motion Sensor |

**Complete the Custom Task**

**Custom Task**

The steps were performed using a custom combination of devices
42. **CT1 - Easiness** - Was it easy to navigate/interact with the environment?
   *Mark only one oval.*

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very difficult</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Very easy</td>
</tr>
</tbody>
</table>

43. **CT2 - Comfortability** - Was it comfortable to navigate and interact with the environment using the given device(s)?
   *Mark only one oval.*

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Never</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Always</td>
</tr>
</tbody>
</table>

44. **CT3 - Reliability** - Was (were) the given device(s) consistent in the kind of action they produced?
   *Mark only one oval.*

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Never</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Always</td>
</tr>
</tbody>
</table>

45. **CT4 - Responsiveness** - The given device(s) felt quick to perform the action?
   *Mark only one oval.*

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Never</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Always</td>
</tr>
</tbody>
</table>

46. **CT5 - Intuitiveness** - Was it intuitive to produce the desired action or did it require some thought process?
   *Mark only one oval.*

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Never</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Always</td>
</tr>
</tbody>
</table>

47. **CT6 - Autonomy** - How much autonomy did you feel with this setup?
   *Mark only one oval.*

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poor</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Excellent</td>
</tr>
</tbody>
</table>

48. **CT7 - Freedom** - Did you feel this setup provides a high degree of freedom?
   *Mark only one oval.*

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Never</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Always</td>
</tr>
</tbody>
</table>
49. CT8 - What was the biggest difficulty you felt while performing this task?

50. CT9 - How easy it felt to perform the required steps?
Mark only one oval.

1 2 3 4 5
Very difficult ☐ ☐ ☐ ☐ ☐ Very easy

51. CT10 - Do you think in a real-world scenario, would be harder to perform these steps?
Mark only one oval.
☐ Yes
☐ No

52. CT11 - Why?

53. CT12 / Which devices did you use?
Check all that apply.
☐ Keyboard
☐ Joystick
☐ Head Motion Sensor
☐ Speech Recognition

Action Mappings

If you used action mappings please answer these questions

54. CT13 - How easy it was to create or modify input sequences and input groups using the recording system?
Mark only one oval.

1 2 3 4 5
Very difficult ☐ ☐ ☐ ☐ ☐ Very easy

55. CT14 - How easy it was to create or modify the action mappings?
Mark only one oval.

1 2 3 4 5
Very difficult ☐ ☐ ☐ ☐ ☐ Very easy
56. **CT15 - How useful did you find the difference in the input devices? Do you think they cover a large variety of possible input actions (for example, button presses, motion events...)?**

*Mark only one oval.*

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poor</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
</tbody>
</table>

57. **CT16 - How efficiently did the interface feel when used with several devices at the same time?**

*Mark only one oval.*

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poor</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
</tbody>
</table>

**Channels**

If you used channels please answer this question

58. **CT17 - How do you evaluate their usability?**

*Mark only one oval.*

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poor</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
</tbody>
</table>
Multimodal Interface Survey
Appendix C

Head Motion Sensor

C.1 Schematics

Figure C.1: Schematic of the voltage regulation circuit.

Figure C.2: Schematic for the header connections.
Head Motion Sensor

Figure C.3: Schematic for the ESP-01 connections.

Figure C.4: Schematic for the serial clock (SCL) and serial data (SDA) pins on the left and for the MPU6050 on the right.

C.2 Routing
Head Motion Sensor

Figure C.5: Layout and routing of the circuit for the Head Motion Sensor

C.3 3D View

Figure C.6: Three dimensional representation of the final board