


The Interactive Sphere for Three-Dimensional Control in Games and Virtual Reality

Paulo L. S. Brizolará  [Federal University of Rio Grande do Norte | pl.briz@proton.me]

Leonardo Cunha de Miranda  [Federal University of Rio Grande do Norte | leonardo@dimap.ufrn.br]

Gabriel Alves Mendes Vasiljevic  [Edmond and Lily Safra International Institute of Neuroscience - Santos Dumont Institute, Federal University of Rio Grande do Norte | gabriel.vasiljevic@isd.org.br]

Bruna Camila de Menezes  [Federal University of Rio Grande do Norte | brunacamilamenezes@gmail.com]

Abstract

In electronic games, the controller is the mean through which the player can interact with the game's virtual world, being an essential factor in all of the user experience. New controllers may, therefore, completely modify the player experience, also serving as a tool to investigate new ways of interacting with interactive systems of various purposes. In this context, this paper presents the Interactive Sphere, a spherical device to be employed specially with games and virtual reality environments. This novel device combines the pressing of certain regions of the sphere with gestural interaction, in addition to providing haptic, auditive and visual feedback. The paper describes all of the rationale behind the decisions taken during the design and development process of the device, in addition to the techniques employed for implementing the detection of the acts of pressing and moving the Interactive Sphere. In this project, accessible, low-cost materials and techniques were prioritized, which could be more easily adapted to other contexts. We envision that the lessons learned and the guidelines derived from its design and development process may assist in the idealization and construction of new ways of interacting, by providing a set of methods, techniques and technologies that were employed in the development of a new physical artifact of interaction presented in this work.

Keywords: *Interaction device, controller device, gestural interaction, haptic feedback, VR, organizational semiotics.*

1 Introduction

Controllers are essential components to the player experience in electronic games (Hufnal et al., 2019), acting as a “contact point” with the game's virtual world and fundamentally structuring the player experience (Murphy, 2014). Controllers that are different from those commonly employed today can, therefore, change the way players experience the game (Llagostera, 2019; Brown et al., 2015). Several researchers have already proposed alternative controls and other interaction devices with different formats, such as gloves (Footitt et al., 2014), musical instruments (Ey et al., 2010), rings (Miranda et al., 2010, 2013), trumpets (Sarinho, 2017), belts (Dobbelstein et al., 2015), ropes (Shahmiri et al., 2019), or even a Rubik's Cube (Roudaut et al., 2014). The research and creation of new controllers, in this sense, offer opportunities to explore other means of interaction, allowing to reveal resources capable of also influencing the project of controllers for videogame consoles (Llagostera, 2019).

However, building a different controller may be a challenging task from various perspectives. For example, technical challenges could make it difficult to implement a functional prototype of a device based on the conceived conceptual design. These challenges that arise during the development phase may produce new problems, or even new requirements, which have the potential to induce changes even in the design initially conceived for the controller. In addition, it is important to guarantee, during all of the development process, that the “right device” is being built, and that it actually meets the needs of its target audience, without bringing new interaction barriers (Miranda et al., 2009). When it comes to a project for

an innovative device, it is not always possible to guarantee that widely employed methodologies, as well as the hardware and software technologies employed in current devices, will adapt adequately to the new context.

In this sense, this work presents the project, design and the development of the Interactive Sphere, a spherical controller focused on games and virtual reality (VR) environments. This device aims at abstracting the most basic principle of interaction employed by a large portion of videogame controllers, that is, the “pressing of buttons”, in addition to taking advantage of the movements of the human hand. To this end, a combination of pressure over the device's spherical surface and gestural interaction was explored. To improve the use experience of this controller, the device also provides haptic, visual and auditive feedbacks. Thus, the controller presented in this work may contribute to explore, under various perspectives, new ways to experience and interact in games and VR, and potentially with other types of interactive systems.

In addition, seeking to contribute to the development of other devices using the methods, the techniques, or the technologies employed in this work, all of the design and development process used in the project of the Interactive Sphere are also described in a detailed manner. The methodological background of this process was inspired in a set of activities and artifacts widely employed by the authors in previous works (e.g., Miranda et al. (2007, 2011); Almeida et al. (2009); Santana et al. (2009)). This description facilitates the adoption, by other researchers, of a subset of these activities and artifacts selected according to the needs of each project. For building the prototype of the Interactive Sphere, there was also a concern in using low-cost materials and prototyping tech-

niques that facilitate their replication or adaptation in projects conducted by other researchers, technology developers, or independent makers.

The paper is organized as follows: Section 2 describes other spherical interaction devices found in the literature; Section 3 presents the process of design and development of the Interactive Sphere; Section 4 details the Interactive Sphere under various perspectives; Section 5 presents implementation guidelines, described based on the lessons learned with the development of the Interactive Sphere; Section 6 discusses the results of this work; and Section 7 concludes the paper.

2 Related work

There are a number of works in the literature that describe interaction devices with a spherical shape applied to different domains, such as the “musical balls” proposed by Weinberg et al. (2000). The musical balls were created to allow for children and non-professionals to play music. The device is composed of a soft fabric ball with eight pressure sensors distributed around the sphere. Users may press the sphere to produce sounds. The sensors consist of high impedance electrodes whose capacitance is measured by a micro-controller (PIC16F84), which sends messages through a serial interface to a desktop computer, which in turn is responsible for playing the music (a specific software was developed to aid in this process).

Perelman et al. (2015), on the other hand, presented the RolyPoly Mouse (RPM), an input device based on a spherical shape, which include functions of both a mouse and a controller for three-dimensional environments. The RPM allows for the detection of not only translation over a 2D surface, but also the rotation over its three axes, that is, front-back (pitch), left-right (roll), and around its vertical axis (yaw). The principle of the device imitates a toy known as “roly poly”, but with a spherical shape. Its base is adapted such that it has a low center of mass, allowing it to be automatically repositioned once released by the user. A consumer-grade sensor (i.e., the Polhemus Patriot Wireless Tracker) was employed to detect the positioning and the rotation of the RPM, which is based on a “marker”, that is, a passive component that emits electromagnetic waves, and a receptor, which detects those waves to determine the marker’s position and rotation (Khalfin, 2002). The device also includes a ring-shaped button, constructed based on a resistive potentiometer that emits a value according to the touched position. An Arduino Fio was employed to interface this button, along with a Bluetooth shield and an external battery for power.

Another spherical device, called “CHI-Ball”, was proposed by Heberlein et al. (2003), and developed to assist in teaching martial arts with the support of games. The device presents a group of buttons coated with silicon in the shape of animal footprints, related to fundamental animal forms in Tai Chi and Kung Fu. The sphere can be held by one or two users concurrently, and the buttons allow to detect whether the users are holding the sphere in the right position. In addition to buttons, the orientation and acceleration of the device may also be employed as inputs. Coloured lights are emitted from the inside of the sphere to provide feedback.

Another device that aims at assisting in the rehabilitation of the hands, or more specifically of the wrists, was developed by Karime et al. (2012) to be employed with serious games. This device was built through the instrumentation of a “anti-stress ball”, a kind of flexible sphere utilized by physiotherapists in certain treatments. The authors coupled three movement sensors to an anti-stress ball (i.e., an accelerometer, a gyroscope, and a magnetometer) to allow for the detection of wrist movements based on the rotation on the three axes of the device (i.e., pitch, roll, and yaw). In addition, to allow the detection of pressure on the sphere, a pressure sensor was attached to the device to be positioned under the user’s palm.

Varesano and Venero (2012) proposed a spherical device to be employed in videogame and entertainment activities, called PALLA. The authors opted for a simple shape, without buttons or other visible components, in order to bring the device’s design closer to everyday objects and simplify interaction for the elderly and for users with little affinity with technology. A prototype of the device was built with a rigid, semi-transparent plastic sphere (reused from a toy package). The sphere can be separated in half into two hemispheres, allowing the assembly of electronic components. The semi-transparent surface allows for a high brightness RGB LED to provide visual feedback from inside the sphere. In addition, a vibration motor is employed to provide haptic feedback. A composition of four sensors allows for the detection of movement (i.e., accelerometer, gyroscope, magnetometer, and a high resolution barometer). A LDR (Light Dependent Resistor), capable of detecting light levels, is also available in the device. An Arduino Duemilanove board controls these sensors and actuators, communicating through a Bluetooth module. Using this group of sensors, the PALLA was designed to identify certain different forms of interaction. The device may be rolled over a surface, being able to detect its orientation and the performed movement, or be lifted by a user, detecting the changes in attitude with an accuracy of less than 10cm. The PALLA is also able to identify taps over points on the sphere’s surface from the movement produced on the accelerometer; and the approximation of the user’s hand over the sphere based on the light level detected from the LDR, that is, detecting the shadow of the hand on the device.

Comparing the Interactive Sphere, which will be presented in this work, with other spherical devices found in the literature as previously described, we highlight two comparison aspects: the proposed use and form of interaction. Regarding the proposed use, the Interactive Sphere is closer to the PALLA (Varesano and Venero, 2012). Both devices were designed as general-purpose controllers for games, although the PALLA has a greater focus on users with low affinity to technology, aiming at being similar to everyday objects, while the Interactive Sphere aims at being similar to other controllers specially designed for games. The CHI-Ball (Heberlein et al., 2003) was also designed as a controller applicable for games, but with a specific purpose, while other devices such as the musical balls (Weinberg et al., 2000), the RolyPolyMouse (Perelman et al., 2015) and the sensory stress ball were designed for other domains of application.

In terms of the form of interaction, the Interactive Sphere combines some of the resources employed in the other devices. As with the musical balls, the CHI-Ball and the sensory stress

ball, the Interactive Sphere employs pressure sensors, but also uses multiple sensors to extend this capability over nearly the entire surface of the sphere, allowing it to receive input from multiple fingers. This can facilitate the control of games and other applications that require more complex controls, such as 3D modelling, simulations, and social interactions in virtual environments. The Interactive Sphere also employs movement sensors, similarly to the other analysed controllers (with the exception of the musical balls). However, in contrast with the PALLA and the RolyPolyMouse, the Interactive Sphere was not designed to be used over a surface, but to be kept held under the user’s palm, allowing more freedom for movements and, along with gesture recognition techniques, the Interactive Sphere also allows for the execution of more elaborate gestures, as will be described later in this article.

3 Design and development process

The design and development process of the Interactive Sphere was structured through a series of activities. Each one of these activities involved the production of one or more artifacts¹, with the goal of refining the project in an iterative and incremental manner until arriving at the functional prototype of this novel device. The initial activities were focused on delimiting the scope and the requirements of the device, while gradually shifting this focus to activities that emphasizes the design and development of the Interacting Sphere to, only then, move towards the actual implementation of the prototype of the device.

Figure 1 represents the eight activities in the sequence in which they were performed. Artifacts produced in a certain stage could be updated or refined in posterior stages as required. However, the focus of the process was not on keeping the artifacts, but to utilize them to design and elaborate the device, that is, the artifacts were considered, in this process, as a mean, and not as an end in itself. This set of activities and artifacts produced as part of this process are detailed next. The goal with this detailing is not to describe a “reference method”, but to share experiences and allow the community of researchers to verify its appropriateness for their own projects.

3.1 Brainstorming sessions to outline the project proposal

The objective defined for the project addressed in this work was the construction of a new “physical artifact of interaction”. This objective was motivated by the intention to explore novel ways of interaction, potentially innovative, with interactive systems; naturally, a research topic of interest to the authors of this work. Given the exploratory nature of this objective, it was necessary to best delimit the proposal of the new “physical artifact of interaction”, that is, to define which “type” of device could be developed and for which context of use it could be employed. A simple way to raise ideas for this purpose is through brainstorming sessions. This technique can be more or less structured, and may have a mediator to coordinate the sessions. The sessions may also have a varied number of participants and can be conducted for as long as the participants deem appropriate.

In the case of the project addressed in this work, a single brainstorming session was performed with the authors of the project, with this session having a duration of approximately one hour and half. At the end of this session, it was decided for the proposal of a “spherical controller”. In other projects, for example, these brainstorming sessions conducted to discuss initial ideas could be performed more often, have a longer duration, or involve a greater number of participants, including professionals trained in different areas of knowledge, who could contribute with different views to the discussions. With more than a single session, each one could explore specific “themes”, such as sessions for “initial generation of ideas regarding a given theme”, “validation of initially defined ideas” or “comparison between (alternative) ideas”. It should be noted that having a professional with prior experience in this type of activity conducting the brainstorming sessions enhances its results.

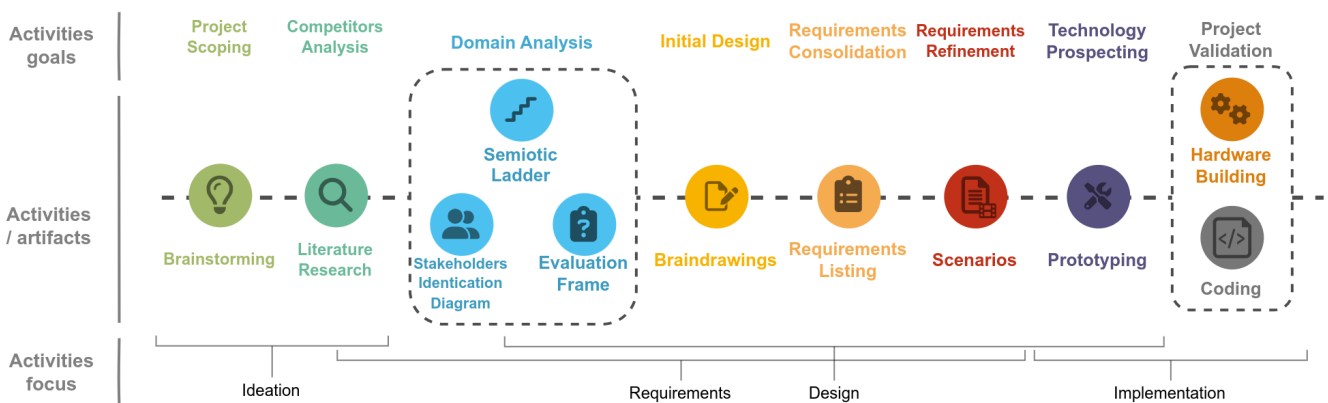


Figure 1. Activities in the process of design and development of the Interactive Sphere.

¹The term “artifact”, in the context of this work, refers to the “product” resulting from a stage of the Interactive Sphere project. Examples of artifacts produced during the development of this process are: technical documents, electronic schemes, various diagrams, and 3D models.

3.2 Competitors analysis through literature research

Once the proposal of a “spherical controller” was defined in the first stage, it was important to verify the existence of works containing similar solutions. This activity is important to ensure the originality of the project, in addition to providing guidelines that could assist in its execution. The main goal was to find works presenting controllers in a spherical shape in the literature.

For this purpose, searches were initially conducted in specialized journals and conferences (e.g., respectively, IEEE Transactions on Consumer Electronics and ACM CHI Conference on Human Factors in Computing Systems). Posteriorly, the searches were expanded using scientific databases and indexers (i.e., Scopus, Web of Science, and Google Scholar), resulting in the identification of part of the related work described in Section 2. It is worth mentioning that the objective of this stage was not to perform an exhaustive or systematic literature search, but to have an overview of the works that presented and developed solutions similar to the proposal of the “spherical controller” defined in the previous stage, and once found, to analyze what are the similarities, development methods, types and forms of interaction, challenges, application domains and limitations, in order to verify the feasibility of implementing a new solution.

3.3 Domain analysis through Organizational Semiotics

After performing the analysis of competitors, the next stage aimed at analysing the domain related to the proposal of a “spherical device of interaction”, that is, the proposal of the “Sphere”. Regarding this aspect, the Organizational Semiotics (Liu, 2000; Stamper, 2001) provides a framework that helps to clarify the investigated domain from a socio-technical perspective. For this, three artifacts of the Problem Articulation Methods (PAM) of Organizational Semiotics were employed (i.e., the Stakeholders Identification Diagram, the Evaluation Frame, and the Semiotics Ladder) in order to identify issues and challenges, in addition to possible solution related to the “spherical device of interaction”.

The **Stakeholders Identification Diagram** assists in identifying the domain stakeholders who directly or indirectly influence or are affected by the proposed solution (Pereira et al., 2013). According to the “level” of this influence, the stakeholders are classified into four layers (or categories): Contribution, Source, Market, and Community. The *Contribution* dimension includes the stakeholders that directly affect the project (e.g., end users and developers). The dimension *Source* comprises possible information and/or technology suppliers (e.g., researchers and repositories). In *Market* are classified possible collaborators or competitors (e.g., other similar solutions, and competitor or partner companies). Finally, the dimension *Community* groups stakeholders that have an indirect influence over the project (e.g., members and venues of scientific communities).

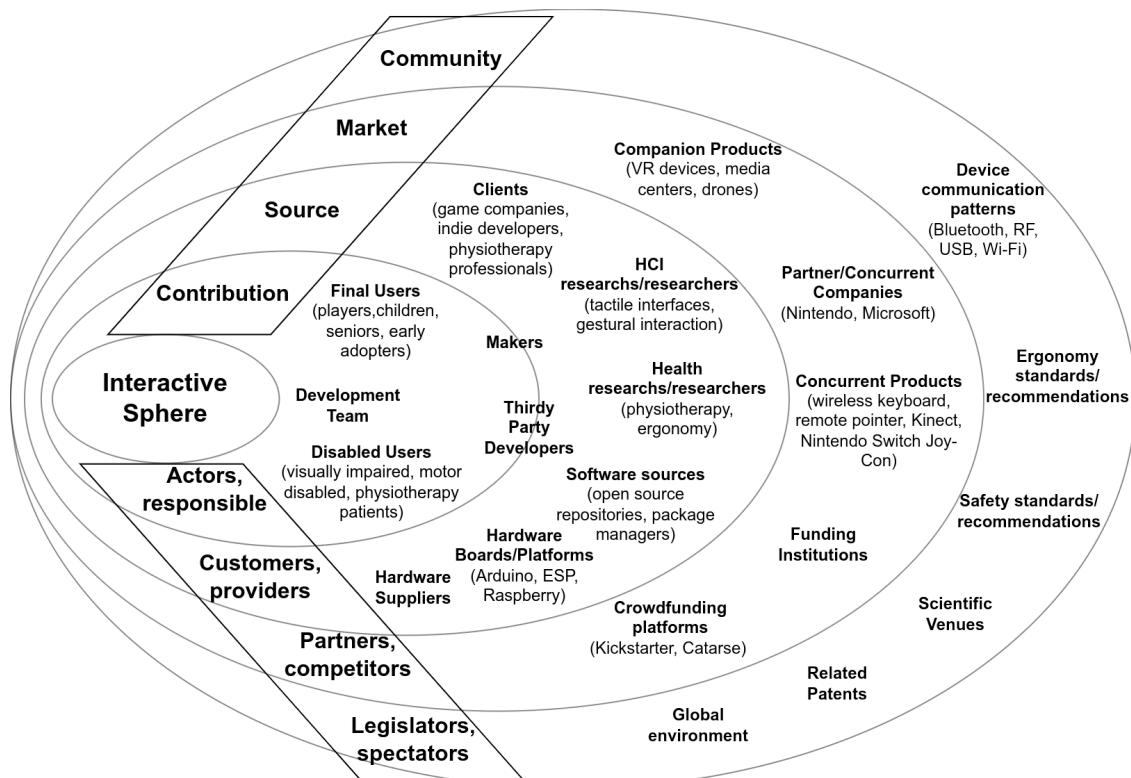


Figure 2. Stakeholders Identification Diagram created for the project.

Considering the Stakeholder Identification Diagram created for the project (Figure 2), we identified several stakeholders that could influence on the design and development of the Interactive Sphere. For example, in the Contribution layer, we identified potential users of the Sphere, such as players, persons with physical or motor disabilities (e.g., physiotherapy patients) and developers. In the Source layer, we identified various suppliers that may affect both the development and in possible applications of the Sphere, such as possible “consumers” of the device, that is, game developers, researchers and physiotherapists, in addition to suppliers for components that may compose the physical project of the Sphere, such as embedded hardware platforms (e.g., Arduino and Raspberry boards) and software repositories (e.g., for gesture recognition functions). For the Market layer, we identified actors that could be both partners (e.g., crowdfunding platforms and funding institutions) and competitors of the product, such as game companies and other game controller devices. Finally, for the Community layer, we listed actors that could indirectly influence on the project and use of the Sphere, such as norms, ergonomics and communication standards between devices, in addition to possible regulatory bodies (e.g., for patents).

For some of the stakeholder roles we also identified sub-categories that may require different needs. For example, in the Contribution layer, we highlight not only players among “end users”, but also children, seniors and early adopters. We also mapped stakeholders who are present in more than one dimension: application developers and independent makers, who both represent categories of users of the solution, as well as possible sources of information.

Based on the Stakeholders Identification Diagram, it is possible to build the **Evaluation Frame**, which allows raising issues and questions as well as ideas and solutions under the perspective of the different, previously identified stakeholders (Miranda et al., 2008).

For the Evaluation Frame created for the project (Table 1), for example, questions were raised in the Contribution layer regarding the adaptability and impacts of the Sphere in relation to the different groups of end users (e.g., persons with disabilities, children, and gamers), and the possibility of manufacturing and maintenance of devices by makers. Some ideas raised based on these issues include the use of multiple interaction modalities to adapt to different users, and the design of the device (hardware) in a way that facilitates its construction and maintenance by makers.

Table 1. Evaluation Frame created for the project.

Stakeholder	Problems and Issues	Ideas and Solutions
Contribution	<ul style="list-style-type: none"> • What implications may the special needs of end users (e.g., elderly, disabled, children) bring to the interaction language? • Which functions must the sphere software interface (API) provide to allow the development of applications? • How to facilitate the manufacturing of the sphere by a “maker” (at home)? • Which implications may groups of users (e.g., children disabled) bring to the physical aspect of the sphere (e.g., resistance, materials, texture)? • Can the interaction language be adapted for users? 	<ul style="list-style-type: none"> • Utilize multiple interaction modalities (e.g., tactile, visual, auditory, gestural) to allow for greater inclusion of user groups. • Include tactile cues on the sphere for the visually impaired. • Design device (hardware) in a way that facilitates the replacement of parts during maintenance. • Use commands that reduce motor effort to assist users with physical limitations (e.g., elderly).
Source	<ul style="list-style-type: none"> • How to adapt the device to the different technologies of consoles and games? • How (and if) the sphere could be employed for physiotherapy (and other applications for health and/or rehabilitation)? • How much can open source hardware platforms available in the market support the development of the sphere? • How the “open” (hardware/software) community can help in the development of the sphere? 	<ul style="list-style-type: none"> • Allow the expansion of commands through software/firmware updates in the sphere. • Have metric measurement mechanisms for the use of the sphere (for applications such as physiotherapy and/or debugging).
Market	<ul style="list-style-type: none"> • How to use the sphere to control other devices (e.g., a drone)? • How can crowdfunding platforms fund the implementation? • How to take/publicize the idea/prototype to the market? 	<ul style="list-style-type: none"> • Facilitate the use of the sphere with other interaction devices (e.g., VR). • Use licenses that facilitate the free use of the solution (at a commercial and user level).
Community	<ul style="list-style-type: none"> • How to market the productions resulting from the development process for benefiting the community? • Which consequences may occur (in the short, medium and long term) for users (in terms of privacy, health, etc.)? • Which impacts for the environment the product may bring once produced in large scale? • Is it possible to adopt measures to reduce this impact? 	<ul style="list-style-type: none"> • To apply mechanisms that avoid possible harm for end users (or for other devices). • Adopt measures that incentive “ethical” uses and manufacturing of the solution (e.g., open hardware, open software). • Design the device in a way to facilitate its unmounting and the reuse of parts.

In the Source layer, we identified issues related both to the development of the Sphere, such as its employment and adaptation to different technologies and contribution by developers from open source communities, and to the adaptation of the Sphere to physiotherapy applications, which included ideas in the sense of allowing the expansion of Sphere’s functionalities with updates and debug modes for more specialized use scenarios. For the Market layer, questions and issues related to the funding and marketing of the Sphere were raised, with solutions including the adoption of licenses for the free use of the device and its compatibility with other electronic devices. Finally, for the Community layer, issues that involve the impact of the Sphere production both for people and for the environment were identified, with ideas for solutions proposing safety mechanisms for the use of the Sphere and means to incentive the reuse of its components.

The **Semiotic Ladder** (Stamper, 2001; Baranauskas et al., 2003; Miranda et al., 2007) also allows to raise ideas and issues related to a information or computational system. However, in the case of the Semiotic Ladder, the ideas and issues are raised from six different perspectives of semiotics which are represented in this artifact in different levels (from bottom to top): Physical World, Empirical, Syntactic, Semantic, Pragmatic, and Social World.

These six levels relate to different aspects in the use of signs to stablish communication. Technical/technological aspects are considered in the lower levels to: produce and transmit signals (*Physical World*); stablish codification schemes (*Empirical*) and define rules to compose language (*Syntactic*). Above these levels are those that consider human aspects, such as: the meaning of the signs (*Semantic*), their use purpose or communicative intention (*Pragmatic*) and the interpretation of these signs in a given context (*Social World*). The envision of a ladder comes from the idea that properties on the upper levels are built from the lower levels (Stamper, 2001).

For the Semiotic Ladder created for the project (Table 2), issues and ideas related to the Physical World were raised regarding materials and dimensions of the Sphere (e.g., size, weight, and temperature), battery use e physical design. In the Empirical level, choices for data communication and technology patterns, and interaction mechanisms (e.g., sensors) were considered. For the Syntactic level, the questions were related to the communication of the Sphere with users and other devices (e.g., protocols, offered features, and interaction language with users).

Table 2. Semiotic Ladder created for the project.

Level	Issues and Ideas
Social World	<ul style="list-style-type: none"> • How different audiences might view the use of the sphere in a social context (e.g., something new, cool, weird)? • Which impacts/rules/restrictions in the interaction language may be imposed by the culture in which the sphere is inserted? • What is the cost to produce/sell the sphere? What is the impact of the cost on the usage of the sphere (e.g., target audience, contexts of use)? • What impacts can the use and production cycle of the sphere bring to the environment? How to design and develop the sphere to make it easy to repair, recycle, reuse, and “repurpose” the sphere throughout its lifecycle?
Pragmatic	<ul style="list-style-type: none"> • For which types of activity would users want to use the sphere? • How to indicate (visually) the capacities and possibilities of the sphere? • How to build the interaction flow of the sphere (i.e., indicate the start, intention/command and end of interaction)? What action should be performed by the user to indicate intention to interact with the sphere?
Semantic	<ul style="list-style-type: none"> • What meanings can the materials and colors used in the sphere produce in users? • How to indicate visually (and for the visually impaired) that the sphere is an interactive sphere? • What meanings/inferences can the user attribute to the weight and size of the sphere? (e.g., fragility, resistance, difficult to use) • How to communicate different states of the sphere (e.g., on, off, ready for interaction) to different user audiences (e.g., people with disabilities)? Idea: feedback (haptic?) on detecting touch (indicating “ready to interact”).
Syntactic	<ul style="list-style-type: none"> • Which protocols are most suitable for communication between the sphere and other computational devices? • What functionalities does the sphere need to have to interact with different types of computational applications? • What is the syntax of the (basic) interaction language between the user and the sphere? What basic actions (commands) compose the language? • How can the user indicate which device/application they want to pair with?
Empirical	<ul style="list-style-type: none"> • What communication technologies to use between the sphere and computational devices? What are the advantages and disadvantages of existing technologies (e.g., Bluetooth, Wi-Fi) for use scenarios of the sphere? • What are the advantages and disadvantages of interaction mechanisms (e.g., gesture, touch) for user interaction? • How much can the (lack of) “accuracy” of sensors interfere in the interaction language?
Physical World	<ul style="list-style-type: none"> • How can the size of the sphere influence on the interaction? • What is the most appropriate size to suit different groups of end users? • How much space do the (minimum) hardware components demand for the construction (of a prototype) of the sphere? • What are the impacts of the sphere’s weight on interaction, users and hardware components? • How to charge the sphere (e.g., USB or induction)? How to indicate sphere energy level? • How to optimize battery usage? Idea: The sphere can detect inactivity and go into standby mode. • What “features” are needed for the external part of the sphere?

In the Semantic level, the focus was on the meanings attributed (by users) or communicated (by designers) in aspects such as appearance, dimensions of the device and its interfaces (actuators and sensors). In the Pragmatic level, the intentions and motivations in the use of the device and on ways of establishing (initial) interactions or commands were analysed. Finally, in Social World, the influence of social aspects on the use and development of the device was questioned, such as: perception and receptivity of the Sphere, influence of culture on the interaction language, and cost and environmental impact.

In the case of the project described in this work, the artifacts from Organizational Semiotics assisted the project team to visualize the solution inserted in society and to perceive other views about its use and construction. This may be done with a relatively low investment of time and other resources, and even without the availability of stakeholders representatives. This naturally implies that the results are limited by the information available for the designers and by their ability to assume other perspectives. In this case, other activities may be important to complement the obtained views. Even so, these artifacts may offer interesting contributions to the development process.

3.4 Braindrawing sessions for the initial design of the device

The artifacts from Organizational Semiotics assisted in raising problems and issues, as well as ideas and solutions. It was then necessary to translate the ideas and solutions found into a more concrete proposal for the device. Thus, the next step was to conduct braindrawing sessions to refine ideas for the Sphere and its applications. Braindrawing is one of the techniques from Participative Design (Muller et al., 1997) which, as its name suggests, is inspired by the technique of brainstorming, but aiming at exploring more visual resources to aid in the construction of collective ideas, thus allowing the rapid generation of a set of candidate proposals. This technique is being employed in various works presented in the literature (e.g., Miranda et al., 2011). In braindrawing, each participant starts with a sheet of paper, where they can explore one or more ideas, preferably through drawings, but written notes can also be made. After a predefined interval (e.g., five minutes), each sheet is passed to the next participant, who continues to work on the ideas/drawings on the paper, until the participants have worked sufficiently with each other's ideas.

Four braindrawing sessions were performed for the project. The first two were carried out with the project authors, in order to explore ideas related to the design, components of the Sphere and possible forms of interaction. The last two sessions sought to explore ideas for digital games designed to be controlled with the Sphere, as a way to validate and refine the capabilities of the device. For this, two other researchers/developers, also identified as potential users (gamers), were invited to participate. An initial explanation of the purpose of the session and the current state of the project was performed at the start of each session. Then, the braindrawing activity itself was performed, as described above. A duration of five minutes was defined for each drawing iteration, with four

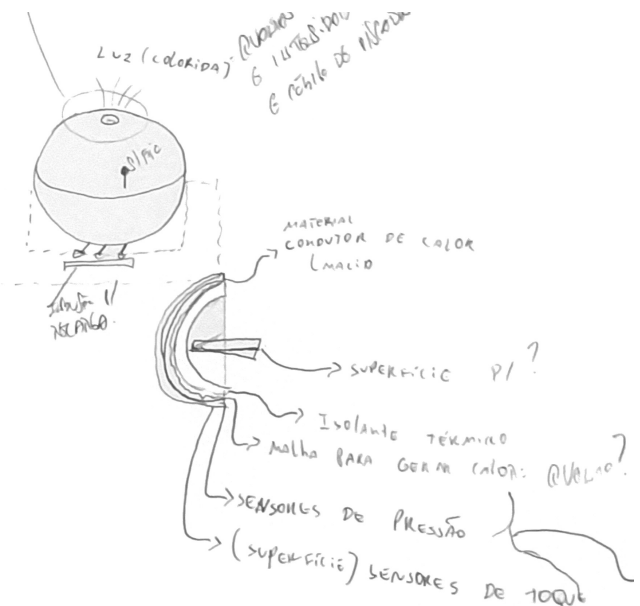


Figure 3. Example of a braindrawing resulting from one of the project sessions.

iterations being performed in each session. At the end of each session, the elaborated ideas were presented and collectively discussed, in order to facilitate their understanding by all participants. It was also attempted to elaborate a consolidated version of the proposed ideas on a new sheet of paper after the discussions. It is worth to highlight that, for some of the sessions, it was not possible to establish a clear consensus within the available duration (about 40 to 50 minutes after the drawings were finished). Figure 3 shows the braindrawing result of one of the project sessions, which outlines a candidate design for the Sphere.

Braindrawing has shown to be an appropriate technique not only for allowing the generation of several ideas in a relatively short time, but also for helping to concretize and explain ideas among the designers, as the drawings allow for another dimension of expression to the participants. Few resources are required to conduct a session (basically, pencil and paper), and it can be done even in quiet environments, in addition to being a dynamic and potentially fun activity, which can help to engage participants. However, converging the ideas and visions of the designers may be a challenge. For example, it was observed that there were different visions of how the device should work during the sessions, not always reaching a consensus. In this sense, according to the needs of the project, it may be interesting to give more emphasis to the convergence stage and to provide more details regarding the characteristics that are already defined about the project at the beginning of the session.

3.5 Consolidation of listed requirements

The activities performed previously allowed eliciting, in different ways, numerous requirements. Thus, to guide the development of the device and determine which of these requirements would be met first, it was important to consolidate and prioritize the requirements. The elicited requirements were then specified and identified, in order to facilitate their reference. These requirements were also prioritized as: necessary, important, desirable, and optional. The requirements were

also divided into three categories: “Interaction Resources”, which include requirements related to ways of interacting with users (inputs and outputs); “Connectivity Requirements”, grouping functions required by the (software) clients to connect to the Sphere; and “Device (hardware) Requirements”, which relates to functions required for the maintenance and functioning of the device.

Listing the requirements helps to provide a clearer view of the requirements that must be implemented, which in turn helps the planning of posterior activities. Thus, the focus was not to describe the requirements in a detailed manner, but to have a general overview about them. The format for listing the requirements was also not very relevant at this stage. Other formats such as, for example, user stories, could also be employed. We sought to use the simplest format that met the needs of the project.

3.6 Detailing of requirements through usage scenarios

The consolidation of requirements allowed to list the requirements elicited in previous stages. Even so, some of these requirements were not sufficiently detailed to be implemented. Thus, usage scenarios were employed as a mean to further detail these requirements. According to Carroll (2003), scenarios are, essentially, stories about people and activities performed by them. Scenarios make it possible to explain the use of the system, that is, the actions taken by users, how the system responds to these actions, and even how users interpret these responses. In this sense, scenarios help to guide the design and analysis of systems with a broader view (Carroll, 2003).

To create usage scenarios for the Sphere, a few digital games idealized in the mind drawing sessions were used as a basis. Each scenario was structured by combining a graphical representation and a textual description. The graphic representation included screen prototypes of the games and, in some cases, sketches of the Sphere usage. The textual description, on the other hand, detailed the context and the intention of the players in the scenario, which steps would be performed by them, and which responses would be sent by the device and by the applications. Based on these scenario representations, a list of requirements and observations was created, including observations and requirements related to their implementation or other aspects of the scenario. If required, new requirements could also be included in this list.

Figure 4 exemplifies one of the images created to describe a usage scenario. This scenario explored how the user could activate the discovering of the device (to other devices). In this case, two discovering actions were imagined: (i) pressing the sphere for a given duration, or (ii) pressing a specific regions (such as the top). Then, the drawing details the feedback provided by the device using a LED strip, during the discovering period and in cases of success or failure. The imagined scenario also included a description detailing these possible sequences of user actions and the device’s responses.

Scenarios were especially useful for making clearer how the device could be employed in a given idealized context, with the possibility of highlighting limitations and new necessities. However, as the elaboration of scenarios still requires a certain effort varying according to the level of details, the construction of scenarios *a posteriori* for all requirements may not compensate. It is possible, however, to utilize scenarios even to guide the development the system as a whole, such as with design based in scenarios (Carroll, 2003) or with Behaviour-Driven Development (North, 2006; Solis and Wang, 2011).

3.7 Construction of partial prototypes to explore technologies

Before starting the construction of a functional prototype for the Sphere, it was necessary to explore techniques and technologies that would be employed in the device. A set of partial prototypes were employed in this sense. According to Hunt and Thomas (2019), prototypes are learning experiences in which software components and other disposable materials (i.e., that will not be directly employed in the final system) are created, to test given aspects of a system, abstracting other aspects that are not relevant to this experiment.

In the development of the Sphere, prototypes were applied to test technologies and hardware components, especially those that had not been tested before, but also to investigate software and hardware components that could present defects. For example, prototypes were created to test the accelerometer/gyroscope module, pressure sensors, buzzer, vibration motor, and charging components. Each prototype was implemented as a mini-project, being composed exclusively of hardware and software components essential for its operation.

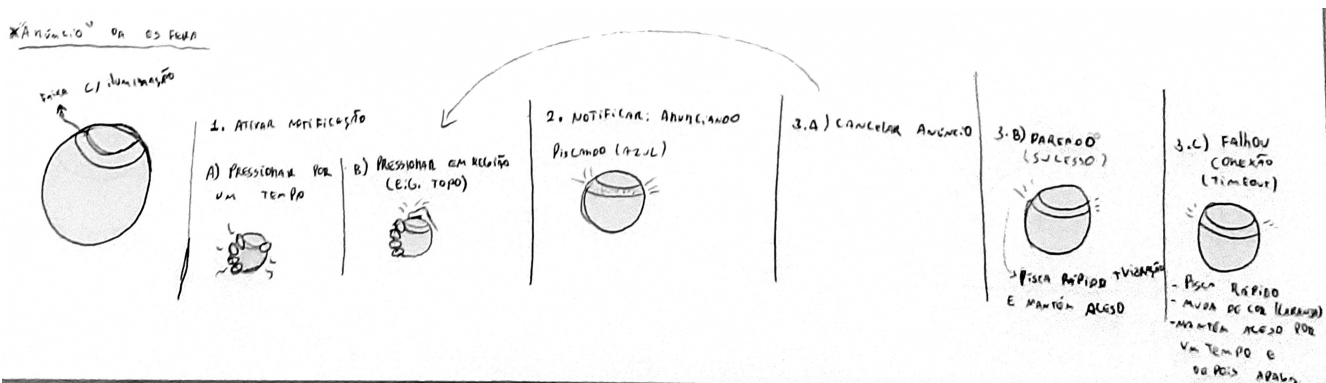


Figure 4. Example of illustration employed with a usage scenario depicting the announcement of the Interactive Sphere to allow its connection with a device (host).

The implemented prototypes helped to test not only the hardware components in an isolated way, but also the related software components. In some cases, part of the code has been adapted or rewritten to be integrated into the device implementation. The experience gained also helped guide the project's code architecture. Naturally, however, the prototypes are not able to highlight possible challenges in the integration of the code or hardware components. Furthermore, this technique may introduce an extra initial time into the development process, but which can be compensated by a reduction in error in later stages.

3.8 Project validation through a functional prototype

The construction of a functional prototype for the device is an important stage of the project. In this sense, the goal was to be as close as possible to the specification described in previous stages. The focus, however, was in the interaction functions, rather than other factors such as product design (e.g., aesthetics and materials). The construction process of this device involved the integration of the technologies that were tested in the partial prototypes into a single device. This process may be divided between hardware manufacturing and software implementation. The manufacturing of software involved the circuit project to integrate the required components, the building of the board, the hull of the Sphere and the sensors and actuators, in addition to the mounting of these components. For the software implementation, an architecture of the components was structured along with the coding of services and an API to access them, in addition to the client (i.e., a library and auxiliary tools) to connect to the device and consume the offered services.

4 Interactive Sphere

The Interactive Sphere was designed aimed at it being employed as a general-purpose physical artifact of interaction, but especially focused at its use with games and virtual reality environments. The concept of a spherical controller, from which the device originated, emerged from the questioning and reflections of the authors about what would be a good shape for an interactive device that, while held in the palm of the users' hand, would better allow them to explore interactive systems for various purposes with their fingers.

This section initially describes the idealized requirements for this artifact, followed by the conceptual project of the device with the presentation of the product design and the interaction language. Additionally, usage scenarios are also presented in order to illustrate the use of the device both in a game and in a virtual reality environment. Then, important hardware and software aspects of the Interactive Sphere are described and, finally, its functional prototype is presented.

4.1 Requirements

The requirements (RQ) and capabilities that the Interactive Sphere should support were defined as part of the design

and development process described in Section 3. These requirements, in this work, were classified into three types: (1) interaction requirements, which involve inputs and outputs (i.e., feedback); (2) connectivity requirements; and (3) hardware requirements. Some of the requirements in these categories that are most pertinent to the context of this article are described below.

For the interaction requirements (RQ1), two types of input were specified: (RQ1.1) gestures, from movements with the device, and (RQ1.2) the pressing of certain regions of the device. As outputs, to provide feedback or to indicate the status of the device, the following feedbacks were specified: (RQ1.3) haptic, that is, vibrations on the sphere; (RQ1.4) visual, that is, lights of multiple colours emitted by the sphere; and (RQ1.5) auditive, that is, sounds emitted by the sphere.

Connectivity requirements (RQ2) indicate the need to (RQ2.1) establish wireless communication between the device and the user's host (by providing a remote API), in addition to (RQ2.2) detection capability (i.e., discovery) of the device via network. The hardware requirements (RQ3) present needs related to the maintenance of the device, such as (RQ3.1) updating the embedded software, and (RQ3.2) providing a support (dock) to accommodate the device in a stable way when it is not in use (which is necessary since the device has a spherical shape), in addition to (RQ3.3) provide basic functions, such as allowing the charging of the device's battery and indicate the Sphere's status (e.g., on, off, charging, about to discharge, and establishing a connection).

4.2 Conceptual project

The conceptual project of the Interactive Sphere is the result of a sequence of choices made during all of the design process, and seeks to meet all of the requirements that were raised for the project, as briefly described in Section 4.1. The conceptual project defines the view of the conceived product and provides, in a following step, its implementation. Thus, this project comprises the product design, that is, its physical and aesthetic aspects, but also the device's interaction language, that is, the interaction resources that it provides. These elements are described next, followed of usage scenarios that illustrate how this device could be employed in real-world situations.

4.2.1 Product design

The physical design of the Interactive Sphere was inspired by the so-called "anti-stress balls", employed in physiotherapy. As with this kind of sphere, the device was designed as to have a size that allows it to be held in hand in a comfortable manner. Ideally, different sizes should be available (perhaps even through an adjustable mechanism) for its correct adjustment for different hand sizes, but spheres ranging from 7cm to 8cm of diameter were employed as reference for the prototype. A smooth and semi-flexible material (e.g, silicon) should be employed for the sphere's coating, such as it may be pressed by the user. The pressure sensors must be bellow this coating, and the electronic components must be in the device's core. The arrangement and organization of these elements within the Interactive Sphere are presented in Figure 5.

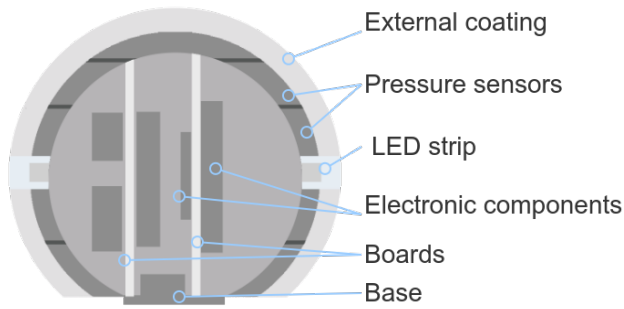


Figure 5. Vertical section of the Interactive Sphere showing its inner layers.

Ideally, the entire surface of the device could be sensitive to pressure, but to deal with hardware limitations, simplify the API offered to the device’s client applications, and even to facilitate user interaction, some regions were specifically specified to be pressure sensitive. Five main regions were defined in the “front” face, visible by the user (Figure 6, left), that is, “Front Up” (FU), “Front Down” (FD), “Front Right” (FR), “Front Left” (FL), and “Center” (Ct). The format of these regions was inspired in the directional pads used in gamepads, and planned based on the distance that they would be from the thumb in a resting position, in such way that the five regions could be controlled using the thumb (Figure 7, top).

Four regions were defined in the “back” of the sphere (Figure 6, right), designed to be controlled by each of the other four fingers, and thus named after them, that is, regions of the “Index” (Ix), “Middle” (Md), “Ring” (Rg), and “Pinkie” (Pk) fingers. The “horizontal” shape of these regions was chosen to better adapt to the natural position of these fingers. At the “Top” (Tp) there is also another region accessible to the thumb or index fingers. It should be noted, however, that the indication of which fingers to use for each area does not necessarily prevent the user from adapting its use in other ways. Furthermore, for being symmetrical, the sphere can be used in both hands. In total, the Interactive Sphere has 10 “press-able” regions, which can thus be employed as different inputs (including in a combined manner) to the client applications. In the “back” face of the sphere, positioned at the center, there is also a LED strip that provides visual feedback and indicates the status of the device.

The lower part of the sphere must have a flat base so that the device can be rested on a surface when not in use, in addition to serving as an output for the buzzer, USB connector, on/off switch, and support for a vibration motor. For maintenance, an access point that would allow the opening of the sphere was also designed, which is accessible through the base and separates the two faces (front and rear). The design of the sphere also proposes a bracelet on the device to prevent it from falling during use (Figure 7).

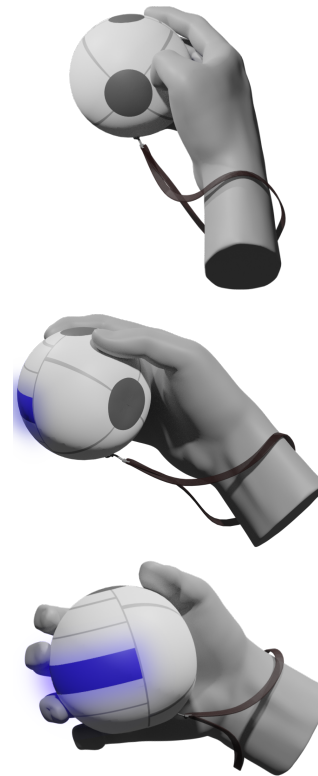


Figure 7. Representation of the use of the Interactive Sphere: (top) frontal view and the position of the thumb while pressing the Sphere; (center) lateral view, with thumb at rest; and (bottom) a lateral view, demonstrating the positioning of the fingers over the pressure regions on the rear face.

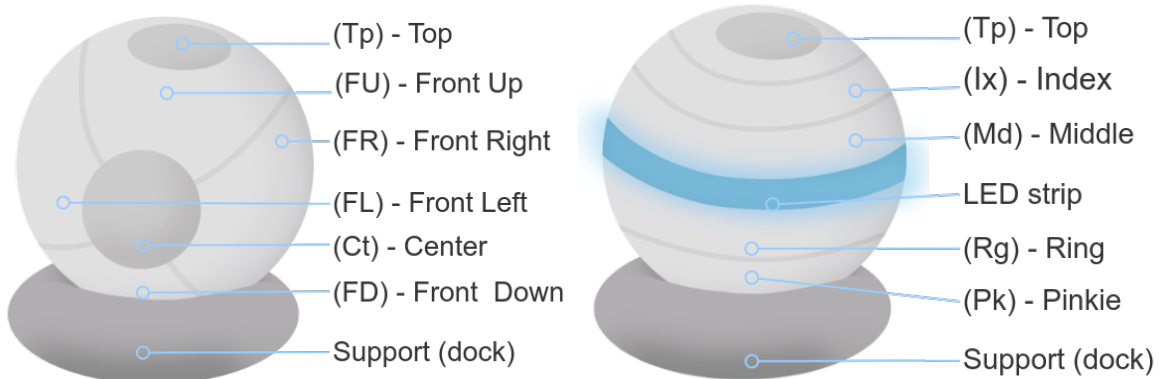


Figure 6. Product design of the Interactive Sphere: (left) frontal view; and (right) rear view.

4.2.2 Interaction language

The interaction possibilities offered between the user and the Interactive Sphere can be seen from the perspective of its interaction language, that is, the commands (inputs) that the device allows the user to perform and the responses (outputs) that users are able to perceive from the artifact. These input (commands) and output (feedback) elements, represented in Figure 8, therefore, delimits the capabilities of the controller and what the applications can perform.

In this sense, according to the requirements briefly described in Section 4.1, the Interactive Sphere allows for users to execute two types of input commands: commands based on pressing the delimited regions on the device (see Figure 6); and gesture commands, based on device movements. This combination fits on what van den Hoven and Mazalek (2011) call tangible gesture interaction, that is, a kind of interaction that utilizes physical objects as support to gestural and tangible interaction. Angelini et al. (2015) decompose and classify this kind of interaction based on three fundamental components: move, hold, and touch, with the pressure level, or force, considered as an additional property of hold and touch. In the case of the Interactive Sphere, the three components are present, albeit touch being supported only with “pressure”. In addition, for common interaction, it is considered that the user would be holding (hold) the device.

The movements with the Interactive Sphere can be composed of inclination, rotation or translation in relation to the center of the device. These movements were defined based on the basic information that can be detected from a gyroscope and accelerometer. Simple gesture commands can be defined directly by tilting or accelerating the device. However, to identify more complex commands, more sophisticated gesture recognition techniques may be required, such as using machine learning algorithms.

The top of Figure 9 illustrate simpler gestural commands, “Walk” and “Rotate Camera”, based on the inclination of the device, and a more complex command, “Jump”, which would involve the movement of lifting the sphere and then returning it downwards (as if it “jumped” an obstacle). For pressure sensors, the Interactive Sphere allows to detect the pressed region at an analogue pressure level (i.e., how much each region is being pressed). This allows for the mapping of this pressure level with application actions, such as, for example, the movement speed of a character. The pressing of the regions of the device can also be conjugated to gestures to avoid false-positives, that is, for indicating the intention to performing a gesture. The bottom of Figure 9 exemplifies a possible mapping to control a character in a 3D environment, using the Interactive Sphere with a single hand. By combining two Interactive Sphere, more complex commands could be recognized.

The capabilities of interaction of the Interactive Sphere also include visual (i.e., RGB colors and lighting patterns, such as blinking and alternating colors), haptic (i.e., intensity and vibration patterns), and audible (i.e., through sound patterns) feedbacks. The device offers an API that allows the control of these feedback resources, in such way that each game or application may define how to use them. The following subsection details usage scenarios and how these resources (highlighted in bold) could be practically employed in a game and in a virtual reality environment controlled by the Interactive Sphere.

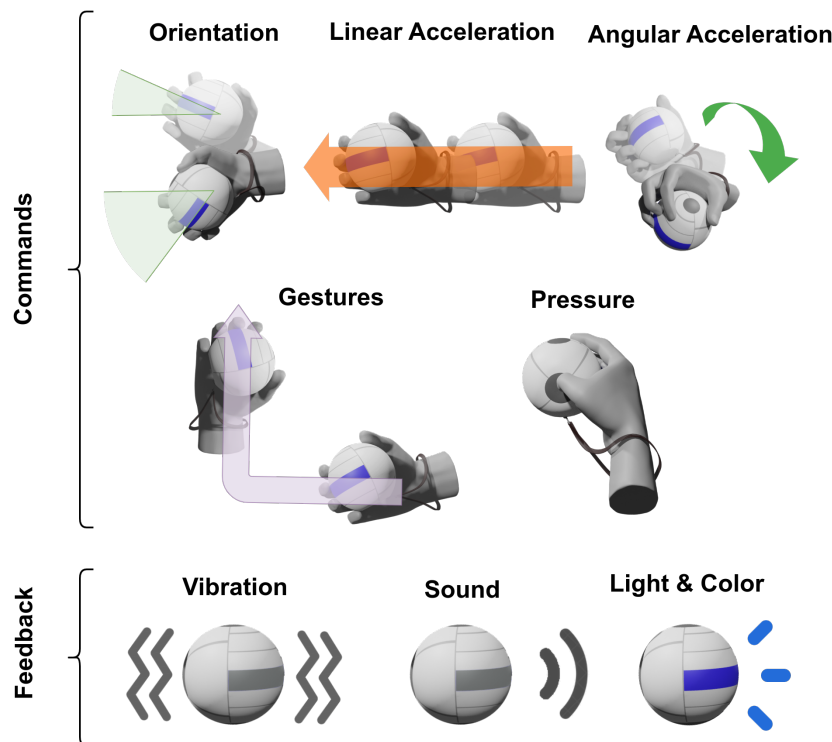


Figure 8. Representation of the elements in the interaction language.

4.3 Usage scenarios

Jedi Sphere The capabilities designed for the Interactive Sphere may be exemplified using an usage scenario imagined for the “Jedi Sphere”, one of the digital games conceived during the brainstorming sessions. In this scenario, each player uses a sphere and the user’s computer controls the game. The concept of the game consists of using one Interactive Sphere as a “light saber”, based on the Star Wars movie franchise, in order to play against another player and defeat them in a battle using the virtual saber. To do so, the player would combine gestural actions with the pressing of regions on the Sphere to execute commands, such as attacking or defending with the saber (horizontally or vertically), moving to dodge attacks or using the “the force” to try to push, pull or destabilize the opponent. According to the combinations of actions of the players, however, one of them can be hit or “open the defence” to receive an attack. The color and intensity of the LEDs would identify the player and indicate their level of health, respectively. Additionally, flashes of light, vibrations, and sounds could be combined to indicate incoming damage, collisions between sabers, and other feedback including player victory or defeat.

In this scenario, when starting a match, the game would allow to configure and connect the Spheres. The pairing and network discovery can be activated by pressing the top of the sphere for given duration, with the LEDs indicating the pairing status: blue blink to **announce presence**; fast blink + vibrate and keep LED on for **established connection**; and fast blinking, changing color (orange) and going off after a short delay to indicate a **connection error**. Upon completing the initial configuration, the host computer starts to act in the “background”, with both the interaction and feedbacks being provided exclusively through the Interactive Spheres, aiming to maximize the users’ experience by directing their focus of attention to the game itself, and with the player’s interaction taking place in the game’s virtual environment via the Interactive Sphere. During the game match, the Sphere would react according to each occurring event: auditive feedback + strong vibration and red blink to indicate **damage**; sound and continuous vibration to indicate **saber collision**; and sound + short vibration to **indicate an attack**. The Sphere could also provide feedback to indicate its charging status: oscillate blue light to indicate **charging**; keep light green to indicate **full charge**; and short sound + change color to red to indicate **discharging**.

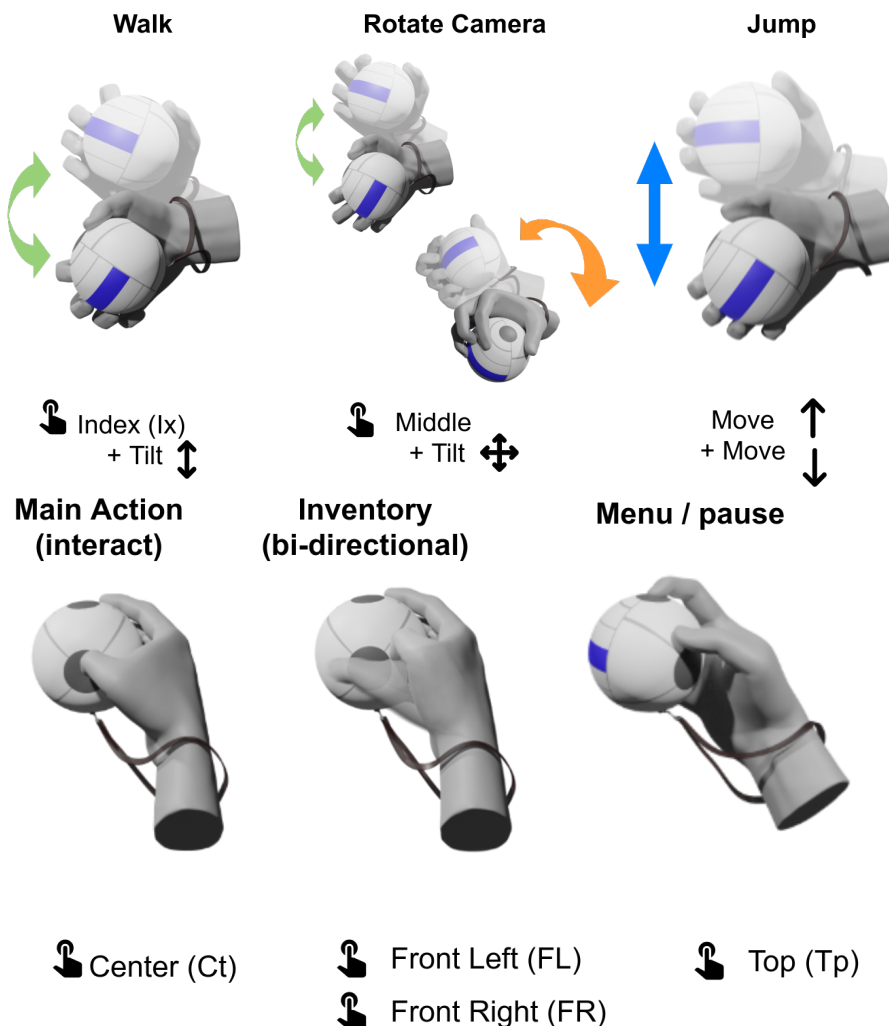


Figure 9. Examples of commands to control movement in a 3D game: (top) shows the gestural commands; and (bottom) show commands based on the pressing of the device (label: ↑ up/down; ⇄ up/down + left/right; ↑ up; ↓ down; ↵ press specified region).

Virtual reality games The previous scenario explored the usage of the Interactive Sphere in a game environment controlled remotely by another device. In this case, the players interact in a real world setting, with the Sphere serving as an “augmented reality” controller. It is also possible to employ the Interactive Sphere as a controller in a virtual reality environment, instantiating its representation in this virtual scenario using various objects. Given the previous example, the Spheres could be mapped into light sabers by using a virtual reality headset, maintaining the same rules as the original game (with the opponents’ avatar representing the targets for each player’s attacks).

Other kinds of games from different genres can also be controlled using the Interactive Sphere. A possible use scenario, in this case, would be an exploration game, in which the player must navigate through a virtual dungeon, defeat enemies and solve puzzles in order to reach its end and win the game. In this case, one or two Spheres could be employed. For this example, the usage of two Spheres will be considered. The Sphere would be responsible for all of the available commands: walk, draw weapon, attack, defend, jump, and interact (with the scenario and its objects). The action of **walking** could be performed using gesture recognition: rapidly moving both spheres vertically in an alternate manner (i.e., while the right hand moves the sphere up, the left hand moves the sphere down, and vice-versa). For **drawing** a weapon, such as a sword, the player would move their hand for either their shoulder or their waist, and press both the middle finger and the thumb, similar to drawing a weapon from a holster. **Attacking** and **defending** would be performed using gestures (depending on the weapon).

Jumping could also be performed using gestures, that is, moving both Spheres up at the same time. This movement could be performed while also moving the Spheres forward or backward at the same time as moving up, to create three possibilities: jumping (at the same location), jumping forwards, or jumping backwards. This action would explore the notion of depth of the virtual scenario. Finally, the Sphere could be employed to **interact** with the environment: putting the Sphere on a door knob and rotating it around its axis would open the door, for example, or pointing one of the Spheres at a button and pressing the thumb sensor could also press this button. Given the spherical shape and the pressure sensors, the Interactive Spheres could be also employed in a multitude of other different actions such as **driving** (with them serving as a steering wheel).

However, considering the virtual reality setting, both the RGB LEDs and the sound emitter on the sphere, in this case, would serve primarily for providing feedback regarding its power status, similarly to the previous use scenario. The haptic feedback, in this case, would be the main source of feedback regarding the game status.

4.4 Prototype of the device

After the conceptual design of the Interactive Sphere was conceived, as presented in Section 4.2, the prototype of the device was built. Below important information related to the hardware and software aspects of the project are detailed, as well as the procedures adopted for the construction of the

functional prototype of the device.

4.4.1 Hardware components

The hardware design for the Interactive Sphere prioritized simplicity and low cost, so that the solution could be replicated by other researchers, developers, or makers, without the need for professional equipment. In addition, the sphere’s reduced internal space was a determining factor in choosing the electronic components that should be embedded in the device. Thus, for example, the ESP32² (Figure 10, left) was chosen as processing unit, not only for its reduced size, but also to possess “onboard” data communication resources, such as Bluetooth transmitters and Wi-Fi, in addition to offering greater computational resources (e.g., processing power, program storage space and RAM) in comparison to popular boards in the maker community (e.g., Arduino Uno or Leonardo). A module with accelerometer and gyroscope (MPU-6050³) was employed to allow for the detection of gestures (Figure 10, middle), and a vibration motor and a buzzer were employed, respectively, to provide haptic and auditive feedback (Figure 10, right).

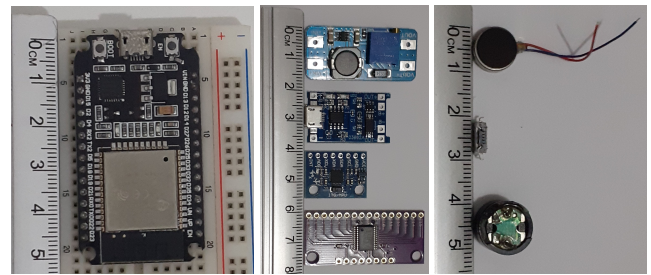


Figure 10. Some of the electronic components employed in the Interactive Sphere: (left) ESP32 micro-controller, in its devkit version; (middle) from top to bottom, voltage converter (MT3608), battery charger (TP4056), gyroscope/accelerometer (MPU-6050), and multiplexer (CD74HC4067); and (right) from top to bottom, vibration motor, micro-USB connector and buzzer.

Addressable RGB LEDs were also employed in order to emit light and color signals, given their potential to build complex “movement” and color patterns. Also, aiming at making it possible for the electronic components of the Interactive Sphere to receive the working voltage necessary for its full operation with mobility (i.e., without the use of cables to power the electronic circuits), a power supply circuit was also designed using rechargeable lithium batteries (TP4056)⁴. The tension produced by this type of battery may vary between 3.0V to 4.2V⁵. Thus, to elevate this tension to a 5V level, which is acceptable for the ESP32 devkit, a Boost (MT3608⁶) tension converter was employed. An LDO converter (Low-Dropout) could be a better alternative, as it would allow to reduce the tension to the ESP32 native level (3.3V), however, the MT3608 was readily available and at a lower cost. The electronic scheme for the Interactive Sphere is presented in Figure 11.

A set of capacitive sensors were employed to detect pressure over the device’s surface, connected to the

²[espressif.com/en/products/socs/esp32](https://www.espressif.com/en/products/socs/esp32).

³[invensense.tdk.com/products/motion-tracking/6-axis/mpu-6050/](https://www.invensense.tdk.com/products/motion-tracking/6-axis/mpu-6050/).

⁴[tp4056.com/datasheet/](https://www.tp4056.com/datasheet/).

⁵learn.adafruit.com/li-ion-and-lipoly-batteries/voltages.

⁶components101.com/modules/mt3608-2a-dc-dc-step-up-power-module.

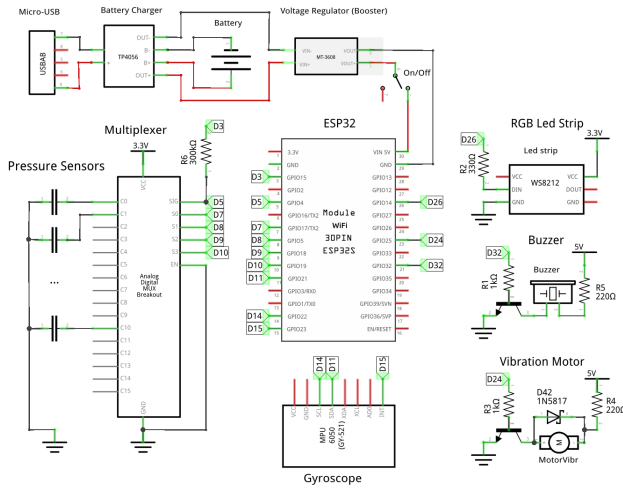


Figure 11. Electronic scheme for the Interactive Sphere.

ESP32 through a 16-channel analogical/digital multiplexer (CD74HC4067). The sensors were made with flexible and low-cost materials, with the first version using paper towel as insulator and aluminium paper as conductor. The scheme presented in Figure 12 represents the composition of this sensor, built based on the work of Nassar et al. (2016). Posteriorly, aiming at obtaining a better response, the sensors were rebuilt with carbon fiber (conductor) and simple paper (sulphite).

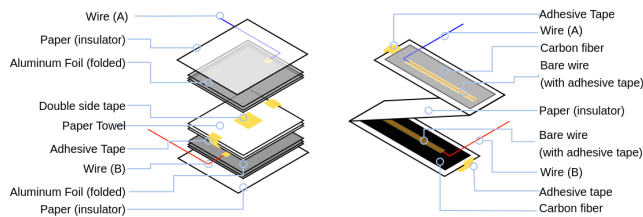


Figure 12. Scheme for the electronic sensors adopted for the Interactive Sphere: (left) sensor made with aluminium paper; and (right) sensor using carbon fiber.

4.4.2 Software components

The software developed for the Interactive Sphere⁷ is divided between the components embedded in the device and those that are executed in the host (e.g., desktop computer or smart-phone) responsible for running the client application that interacts with the Interactive Sphere (e.g., game or virtual reality environment). Embedded components are responsible for interacting with the hardware, offering device services to the host through a restful API. The components on the host are responsible for mediating the communication between the device and the user’s applications, and may also offer higher-level functions, such as gesture recognition.

Internally, embedded components can be divided into three main layers: API, services, and hardware. In the API layer, the components define an interface for services through a RESTful model. That is, routes and methods are used to map the functions offered by a given service (e.g., “GET /pressure/sensors” or “PUT /led/color”). The services layer acts as an intermediary, separating the API from the specific hardware aspects, which are handled by the hardware layer. These layers are separated through interfaces that

decouple the implementations, facilitating their replacement and allowing the performing of hardware-independent tests. The API layer, for example, uses the Stream interface (from Arduino) for data input and output, which can allow for substituting the default implementation (Bluetooth) for another data communication technology (e.g., Wi-Fi or serial). Functions provided by the Interactive Sphere traverse these three layers, having components that are related. For example, “PressureServices” are exposed in the API by “PressureApi” and implemented by a “CapPressureServices”. Figure 13 illustrates these components and presents the solution’s software architecture.

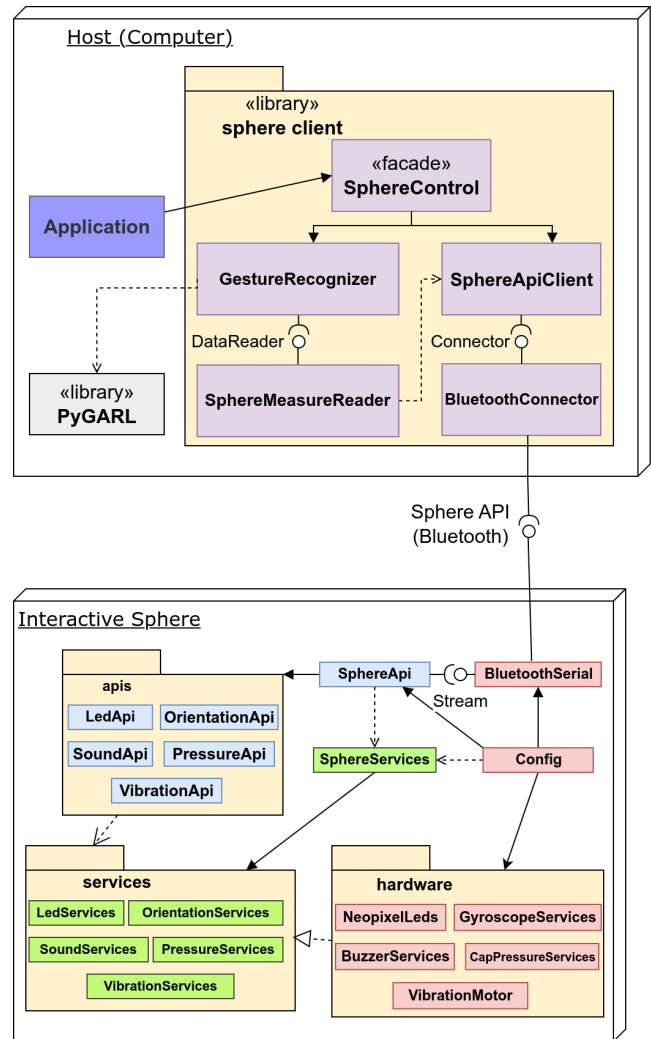


Figure 13. Software architecture for the Interactive Sphere.

Components running on the host were implemented through a Python library. This library implements an Interactive Sphere client (SphereApiClient), which connects to the device through a Bluetooth connector (BluetoothConnector). Gesture recognition is implemented based on an external library called PyGARL (Python Gesture Analysis and Recognition Library). A component (SphereMeasureReader) abstracts the readings of data from the gyroscope/accelerometer through the Interactive Sphere API, allowing these data to be sent to PyGARL. These functions, that is, the communication with the device and gesture recognition, are abstracted by another component (SphereControl), offering an interface to be employed by the applications.

⁷ Available at gitlab.com/interactive-sphere.

The communication between client and device is performed through a protocol that was developed based on HTTP, which is executed over a Stream (Bluetooth, by default). This protocol maintains the textual representation of HTTP messages as a way to facilitate debugging, but with a simplified format that allows reducing the size of messages. This protocol also adds functions for registering and receiving notifications from the device (i.e., the Listen and Unlisten methods).

4.4.3 Functional prototype

The prototype for the device was constructed in an “hand-made” manner based on the conceptual project for the Interactive Sphere (Section 4.2). One of the first challenges in this process was to determine a size for the sphere that would make it comfortable to hold, while also allowing for it to support all of the electronic components of the device (as presented in Section 4.4.1). A size of approximately 7 cm in diameter was defined based on empirical tests performed by the authors. Then, to determine the best internal positioning for the components, a 3D model was created (Figure 14) with Blender3D, based on measurements taken from these components.

Based on this model and the positioning that the sensors should occupy on the surface of the device, the sphere was divided into two hemispheres, following a “cut” in the vertical direction: the frontal hemisphere, where the sensors on the front of the device would be positioned; and the rear hemisphere, with the other pressure sensors and the LED strip. For each hemisphere, a perforated phenolite plate would be used to organize and connect the electronic components. In the frontal hemisphere would be the battery (Figure 14c) and the other components used for power supply and recharging (Figure 14b). In the rear hemisphere would be the microcontroller (Figure 14e) and the modules of the main sensors (Figure 14d), that is, the accelerometer/gyroscope module and the components for the readings from the pressure sensors (such as the multiplexor).

The accelerometer/gyroscope was aligned to the center of the sphere so that its measurements reflected the device’s movements (it is possible to observe the alignment of the module with the center of the sphere through Figures 14a and 14d, under lateral and frontal views, respectively). A small hinge would be used to interconnect the two hemispheres and allow the opening of the device. The buzzer and the vibration motor are positioned at the base of the device (Figure 15), in addition to a mini-switch that would allow the Interactive Sphere to be turned on/off and a micro-USB connector for charging and updating the software/firmware. A set of small openings have been designed into the base of the Interactive Sphere for the ventilation of its internal components.

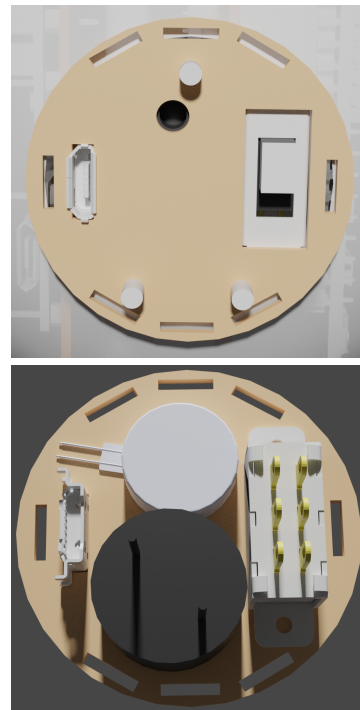


Figure 15. Base of the Interactive Sphere: (top) external view; and (bottom) internal view.

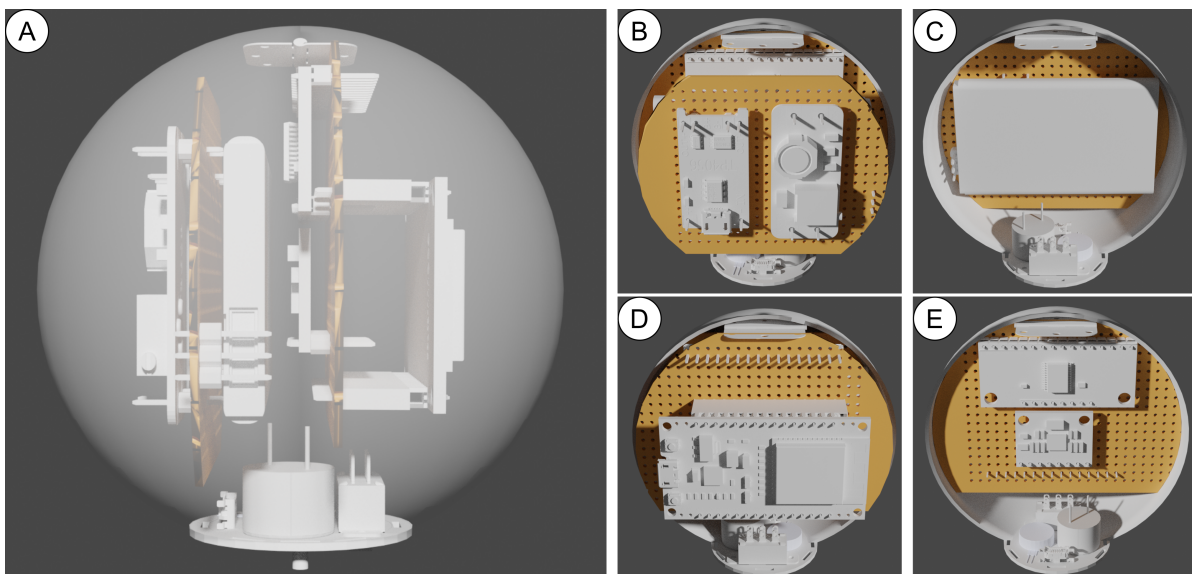


Figure 14. 3D model of the interior of the Interactive Sphere: (a) lateral view with both hemispheres; (b) frontal hemisphere, face of the board with battery charger and voltage converter; (c) frontal hemisphere, face of the board with the lithium battery; (d) rear hemisphere, face of the board with the accelerometer/gyroscope and the multiplexer; and (e) rear hemisphere, face of the board with the ESP32 micro-controller.

To build the prototype based on this model, a Christmas ball that was divided in half in a vertical manner was employed as the structure of the sphere (Figure 16a). The choice of employing this material was mainly due to the difficulty to obtain or to craft a similar structure adapted to this shape. A perforated phenolite plate was cut and sanded, based on the measurements of the model, to allow its positioning inside the sphere. The resistors, connectors, the multiplexer for the pressure sensors, and the accelerometer/gyroscope were soldered on the board (Figure 17).

The ESP32 micro-controller was connected to the board using a set of female connectors, which composed a socket-like structure. The use of this socket structure was aimed at enhancing the device’s modularity and durability by facilitating eventual component replacements, and by avoiding damage to the board during soldering (Figure 16b).

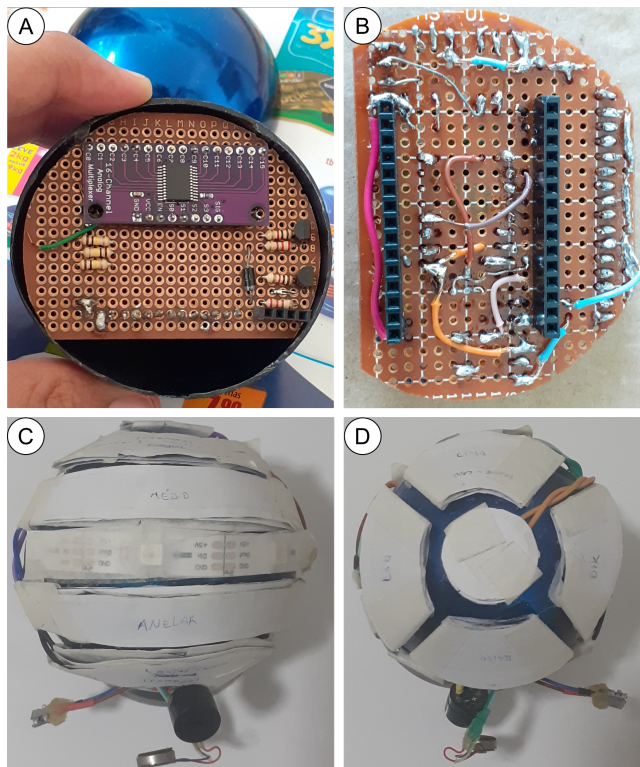


Figure 16. Functional prototype of the Interactive Sphere: (a) view of the frontal face of the board and division of the sphere into two segments; (b) view of the rear face of the board after soldering; (c) rear view with sensors, LED strip, buzzer, vibration motor, and USB connector; and (d) frontal view of the prototype with sensors installed.

Female connectors were also employed to provide more flexibility and allow for the positioning of the actuators (i.e., RGB LED strip, buzzer, and vibration motor), pressure sensors, and a micro-USB connector for power supply (Figure 17). In the specific case of the pressure sensors, these female connectors are important to allow positioning the sensors on the surface of the sphere according to the previously defined regions (e.g., Figures 16c and 16d).

In the following section, the process of crafting these pressure sensors are detailed, in addition to the techniques for detecting pressure over them and methods to record and recognize gestures based on accelerometer/gyroscope.

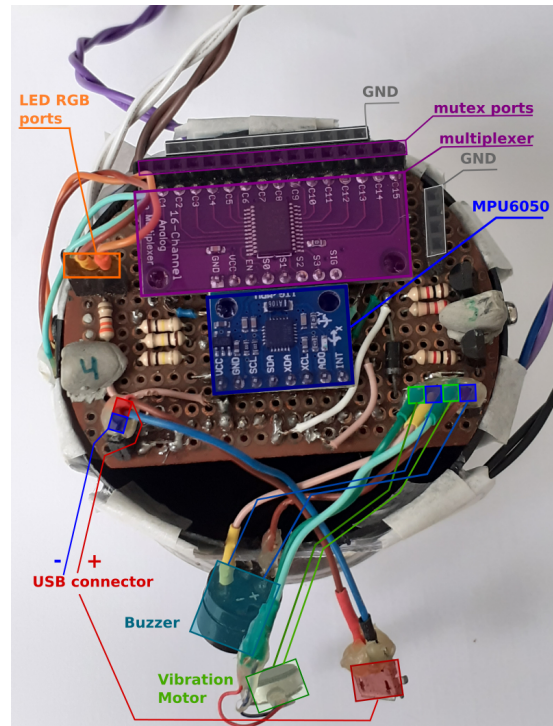


Figure 17. Connectors and components in the frontal face of the functional prototype of the Interactive Sphere.

5 Implementation guidelines

Although the majority of the techniques used in the construction of the Interactive Sphere are already employed by the industry, applying these techniques in the prototype of a new device may be quite challenging. The development of the Interactive Sphere, in special, presented several challenges, such as the manufacturing of pressure sensors (with alternative shapes and materials), the treatment of the signals produced by these sensors, and the gesture recognition process using an accelerometer/gyroscope. Thus, in order to contribute to the development of new interactive artifacts or controllers by other researchers, this section details the techniques employed to implement pressure detection and gesture recognition using the Interactive Sphere.

5.1 Pressure detection

Capacitive sensors can detect various effects from the environment, such as touch, proximity, deformation or pressure (force), by means of measuring the capacitance (i.e., the ability to store charges in an electric field) between two or more conductors, separated by an insulating material. These sensors may be manufactured into different shapes, scales, and with different materials, including flexible and low-cost materials, which facilitates its use for prototyping in several applications (Grosse-Puppenthal et al., 2017). However, these sensors may be less accurate, as they are exposed to electromagnetic noise and interference from the environment. In addition, considering non-industrial manufacturing, the difficulty in standardization may threaten the reliability of the sensors’ measurements. To deal with these issues, some strategies were adopted in the development of the Interactive Sphere for constructing the sensors’ hardware and software, as described below.

5.1.1 Manufacturing of sensors

To facilitate the manufacturing and to allow the replication of the Interactive Sphere, flexible, low-cost and, preferentially, easily accessible materials were required. Initially, paper towels (as an insulator) and aluminium foil (as a conductor) were chosen following the proposal of Nassar et al. (2016). Subsequently, carbon fiber as a conductive material and a simple layer of paper as an insulator were tested to achieve an increase in the sensitivity and reliability of the sensor, in addition to a reduction in its volume, although carbon fiber is more expensive and less accessible than aluminium foil. As shown in Section 4.4.1, two conductor plates, each with a terminal, are separated by a layer of the insulator. This ensemble is coated with another layer of paper, avoiding direct contact with the conductors. Thus, when the sensor is pressed, there is an increase in capacitance, due both to the approximation between the two plates of conductors and to the proximity of the human body.

The manufacturing of sensors, whose process is illustrated in Figure 18, starts with the definition of the sensor format. For this, a technique used in sewing, known as moulage or draping, was adapted. In this technique, a piece of fabric is projected directly onto a mannequin, that is, following a three-dimensional format, and then flattened (Souza, 2006; Rüttschilling et al., 2008). Figure 18a shows an initial application of the technique to obtain a mold using fabric. However, other applications were made using paper directly. In this case, a “model sphere” (made from polystyrene foam) with the same size of the Interactive Sphere was used. The paper could be modelled (folded) over this base, by using a marker to delimit the format of the mold for the sensor, which would then be cut. The sensor’s external and internal moulds were defined through this paper mold. The internal ones, with a slightly smaller size, define the shape of the conductor plates.

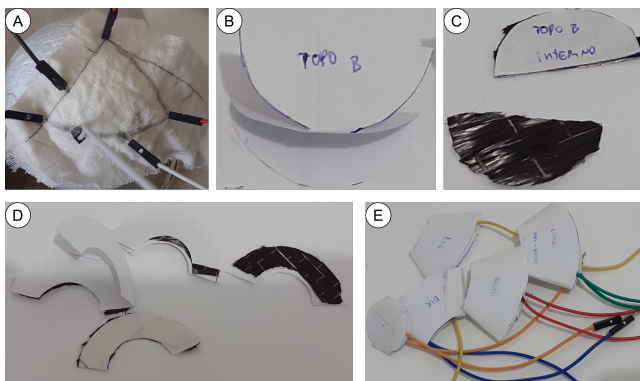


Figure 18. Sensor construction steps: (a) molding the sensor over a polystyrene foam mold; (b) external sensor mold; (c) internal mold already glued with the carbon fiber; (d) gluing the internal molds to the external one using double-sided tape; and (e) sensors after wire gluing and sealed with tape.

In the case of carbon fiber (Figure 18c), these moulds are glued to the fiber in order to keep the wires that compose them united after being cut. The external moulds, on the other hand, are composed into a fold (Figure 18b) on which the internal plates of the sensors are fixed, while also isolating them from the external environment. Figure 18d illustrates this step of gluing the internal boards of two sensors. One sensor (open and positioned to the right on top) already has both internal

boards glued together. The left sensor is missing the gluing of the internal board that is positioned below in the figure, with the double-sided tape used for gluing being visible. After this fixation, the sensor terminal wires are glued, each to one of the internal boards. The wires are stripped and positioned in a way to allow contact with the entire carbon fiber board, aiming to enhance the connectivity between the fiber wires. Finally, the sensor is closed and glued with the help of adhesive tape. Figure 18e illustrates some of the completed sensors.

Each sensor is connected to the micro-controller of the Interactive Sphere through a multiplexor. Industrial solutions, however, compose the capacitive sensors into a matrix, which allows to employ a large number of sensors, but requires a more complex hardware (Pourjafarian et al., 2019). A simpler solution for multi-touch, albeit specific to a few Arduino boards (i.e., Uno and Mega) was proposed by Pourjafarian et al. (2019). Investigating the adaptation of these techniques for detecting pressure in other hardware models could be interesting.

5.1.2 Detecting pressure

Detecting the pressing of the capacitive sensors involves at least three aspects: the reading of the sensors; the classification of these readings to assert whether the sensor is being pressed; and filtering the read values to reduce noise and other interferences. The reading of the sensors on the Interactive Sphere was performed using the CapacitiveSensor⁸ library. This library requires the use of two digital pins with a resistor between them (see pins D3 and D5 in Figure 11), but can also be employed with different micro-controllers, which facilitates the portability of the solution. The ESP32 has specialized pins for the reading of capacitive sensors (ten pins are supported: GPIO4, GPIO0, GPIO2, MTDO, MTCK, MTDI, MTMS, GPIO27, 32K_XN, and 32K_XP), however, in the performed tests, the CapacitiveSensor library achieved better reading times in comparison to the native solution of the ESP32, which is particularly important for the reading of multiple sensors.

A first step to verify whether the sensors are being pressed is to identify the baseline readings for each sensor, that is, the maximum reading value that is obtained when the sensor is not being pressed. This reference value may be determined during the stage of sensor calibration, and a “safety margin” (calculated, for example, as a percentage of the margin of variation observed during calibration) may be added for determining the reading baseline. Any values below this baseline is considered noise and eliminated. Values above the baseline should indicate the influence of external factors. However, depending on the sensibility of the sensor, even the proximity or touch of the hand may be detected (some applications may employ these sensors to also detect proximity or touch). Thus, it is necessary to determine a minimum value (threshold) that indicates the start of a pressure event. As this value may be different for each sensor, it is also necessary that a common value to be defined, calculated as a relative measure (percentage) based on the range of already observed values (minimum and maximum). This requires for the sensor to be pressed at

⁸github.com/PaulStoffregen/CapacitiveSensor.

least once so that this range of valid values can be “learned” by the device.

Errors can also occur due to noise from the measurement caused, for example, by imperfections in hardware or software, or even by environmental factors. To reduce local oscillations in these measurements, it is possible to use a moving average of the last read values. To eliminate peaks (outliers) in these values, it is possible to apply a moving median (O’Sullivan and Igoe, 2004). Still, larger oscillations may cause undue changes in state (i.e., from “pressed” to “released” and vice-versa). Two other strategies may be used to deal with this problem. The first one consists of changing the state only after a sequence of measurements in the same state, that is, a “debouncing” algorithm (Margolis, 2011; O’Sullivan and Igoe, 2004). The second one consists of employing another threshold that indicates the “not pressed” state, which creates a transition range between the two thresholds; measurements in this range do not change the current state.

A side effect of these filters is the delay or reduction in the effect of (legitimate) signal changes, which can lead to delays or false negatives. Therefore, determining the value of thresholds, as well as other parameters, is currently a challenge, requiring experimentation. In the tests performed by the authors, for example, it was decided to eliminate the moving median in order to increase detection efficiency. However, according to the speed of readings and the frequency of peaks, the results can be different. Exploring machine learning techniques to determine these parameters in a semi-automated way could be an interesting approach. In industrial manufacturing, which manage to standardize the sensor production process, it may also be possible to determine standardized parameters.

5.2 Gesture recognition

The Interactive Sphere supports gestural interaction through an electronic module with accelerometer and gyroscope, as presented in Section 4.4.1. This module allows, for example, to calculate the inclination and acceleration of the device, and these informations may be mapped into commands in given applications. However, recognizing more complex gestures require more sophisticated techniques, such as machine learning approaches. The execution of such algorithms may be performed in the host or directly in the micro-controller, using libraries such as TensorflowLite⁹. In this case, however, it may be harder to adapt the recognition of gestures to different applications (as it would require changes in the device’s code), in addition to having to deal with greater resource constraints (e.g., processing, memory and storage space). Thus, for the Interactive Sphere, it was decided for the execution of the gesture recognition code in the host, using a Python library called pyGARL¹⁰ (Python Gesture Analysis and Recognition Library). This library employs a SVM algorithm (Support Vector Machine) (Noble 2006), available at scikit-learn¹¹, for recognizing gestures produced by an accelerometer/gyroscope.

5.2.1 Recording gestures

The first stage for recognizing gestures consists in the recording of a set of representative samples of the gestures that should be recognized, and of the model training based on these samples. Each sample corresponds to a sequence of readings from the sensor. The readings are composed by the data obtained from the gyroscope and from the accelerometer in a given moment. For the Interactive Sphere, the raw value from the gyroscope and the “real” acceleration of the device (which estimates and remove the effect of gravitational acceleration) were employed. A simple tool was developed based on the pyGARL code for the recording of samples, which register various samples of a same gesture. A keyboard command was employed to indicate the start and the end of the gesture. The readings received during this interval are recorded into a data file, along with an indication of which gesture was performed. Once the samples are recorded, pyGARL is used to train a model, which will be used during the gesture recognition stage.

5.2.2 Recognizing gestures

The recognition process involves segmenting the stream of readings received from the accelerometer/gyroscope into discrete samples, and executing the classification algorithm on each sample to see which gesture it corresponds to. For the segmentation of samples, it was implemented an approach based on the “volume” of movement. This approach assumes that each gesture consists in a sequence of movements, finished with a pause or the decreasing of movements. This can allow the recognition of gestures with different durations, without requiring the users to explicitly inform the start and the end of each gesture.

The calculation of the “volume” of movement is performed by considering each reading as a multidimensional vector and calculating the magnitude of this vector (i.e., the square root of the sum of each reading squared). Thus, the sample starts when this magnitude exceeds a certain threshold and ends after a certain number of readings below the threshold, indicating the “pause” of the device. Very small samples and values below the threshold are discarded. This threshold may be determined by recording a small number of samples when the device is idle or with low movement, similarly to the determination of the baseline value for the pressure sensors.

Once a sample has been extracted, it will be processed by the pyGARL classification algorithm to identify which gesture it corresponds to. A limitation of this kind of algorithm, however, is that it always classifies a sample into one of the trained gestures. That is, if a user performs an unknown or invalid gesture, the algorithm will produce a false positive classification. The algorithm employed for sample segmentation helps to avoid situations in which the device is idle, but does not avoid the detection of “accidental” movements. To avoid these “accidents”, the application may combine the execution of a gesture with the pressing of a few sensors in the device, indicating the user intention to perform a gesture. Another option is to train the model for gestures that indicate false-positive or other recurring movements that must be discarded. In addition, the classification algorithm from scikit-

⁹tensorflow.org/lite/microcontrollers.

¹⁰github.com/federico-terzi/pygarl.

¹¹scikit-learn.org.

learn also provides a probability measure (i.e., the method “predict_proba”), which may indicate more certainly that the sample corresponds to one of the known gestures. However, this measure also does not consider the possibility of false-positives (the sum of probabilities for known gestures is always 100%). New tests could be performed to assess the suitability of this measure as a cut-off criterion.

6 Discussion

The design of the Interactive Sphere allowed exploring interaction principles and techniques that may be applied to projects of other kinds of physical artifacts of interaction. The pressure sensors employed in the Interactive Sphere, for example, are low-cost, may be manufactured with easily accessible materials such as paper and aluminium, and also do not require complex hardware components. In addition, the materials are flexible and adaptable, allowing for the manufacturing of sensors with different shapes and sizes. To facilitate the adoption of these sensors in other researchers’ projects, we described in Section 5 the techniques employed for the manufacturing of these sensors and for processing the signals generated by them, such as the draping (or moulage) technique adapted from sewing, which was employed to help moulding the sensors to three-dimensional surfaces. For the software development, techniques such as calibration and the use of filters such as moving median and debouncing algorithms were also employed, which assisted in the removal of signal noise and to detect when the sensors are actually being pressed.

In addition to the pressure sensors, the techniques for gesture recognition based on machine learning employed for the Interactive Sphere were also highlighted. These techniques allow for recognizing gestures by recording samples of valid gestures, which simplifies the definition of gestural commands while also allowing the definition of more elaborated commands. At a design level, the gestural grammar could be established by taking into account the questions posed in the framework of Correia et al. (2013). In addition to this, a sample segmentation algorithm was defined through the movement sensors’ data stream and based on the magnitude of movement, in order to continually identify gestures without necessarily requiring an explicit command to indicate the start and the end of the gesture.

The methods adopted in the design and development process can also contribute to other projects. Ideation and research/analysis of related work are important to better delimit the scope, verify originality, and maximize the innovation of the project. Artifacts from Organizational Semiotics employed in this project help designers to raise questions and ideas through different perspectives and domain abstraction levels, with relatively smaller time and effort. Braintesting sessions may contribute to raise new ideas as well as to materialize and make explicit the visions of the designers. Once a set of requirements has been set for the project, activities of requirements consolidation can be important to synthesize, organize and prioritize them.

Allied to this, usage scenarios contribute to unifying possible divergences, providing a more concrete view (based on

examples) of the usage of the product under development, acting as a kind of simulation of the proposed solutions. As such, they may also help in identifying gaps in the understanding of the project, to evidence unidentified requirements, and even to serve as a preliminary evaluation of the solutions. In the other hand, the usage and testing of partial prototypes can help to validate the employed components and technologies, serving as a learning tool, but also providing a basis for the actual implementation.

The development process of the Interactive Sphere also highlighted some challenges for the construction and prototyping of similar devices. One of the encountered challenges is in the construction of the flexible coating for the device. Techniques for modelling with construction silicone were explored¹², but this material is not easily mouldable, having a short drying time that makes it difficult to mould detailed shapes. In addition, the resulting material may still be not very flexible. One possibility is to explore the use of liquid silicon, which could allow defining finer details, although it requires a more complex process with the generation of “negative” moulds to establish the desired shape. Another possibility would be to investigate DIY (Do-It-Yourself) techniques to produce conductive silicon (see Nagels et al., 2018), in order to integrate the coating layer and the sensors into a single material, being potentially more reliable than the sensors presented in this work, although having much more complex manufacturing process. In this sense, it is possible to investigate how to make techniques that build matrices of pressure sensors more accessible, such as those presented by Rosenberg et al. (2009), which requires a layer of a flexible force resistive sensor.

Another challenge, related to the detection of pressure over the surface of the Interactive Sphere, is detecting the force applied by a user to the opposite side of the pressed sensor. That is, as the Interactive Sphere presents sensors in nearly all of its surface, pressing a given region could cause the user to accidentally press the “other side” of the sphere. Calibration based on the level of applied force may be a solution to this issue, by determining the levels that corresponds to intentional or accidental pressing. Another possibility is to investigate machine learning techniques to recognize pressure patterns.

Some of the challenges encountered, especially in prototyping, are directly related to the spherical shape of the device. For instance, constructing the casing, the circuit board, and the capacitive sensors of the Interactive Sphere themselves required adjustments to this shape. These challenges help explain the limited number of similar works highlighted in Section 2. That is, simpler shapes tend to be more easily adopted in industry or academia. However, part of this work’s contribution lies precisely in highlighting these challenges, as well as identifying opportunities in the construction of unconventionally shaped controls. For example, advancements in the development of flexible sensors may pave the way to explore other shapes, adapted to the target applications.

Regarding another perspective, the development process of devices such as the Interactive Sphere also brings challenges. While Software Engineering has a vast literature on development methods, with a special focus on the software industry,

¹²Through the recipes to create silicone moulds available at: [wikihow.com/Make-a-Silicone-Mold](https://www.wikihow.com/Make-a-Silicone-Mold).

methods for building interaction devices are not widespread. In addition, the context of development in an academic setting can be significantly different from a software company. For example, in academic settings, development teams are usually small (even with a single member), as well as the availability of time and resources may be more scarce, and stakeholders representatives will not always be accessible. In this sense, a future research possibility would be to explore which methods are employed in the academia for the development of interaction devices. Specifically, it would be interesting to research how to employ techniques that assist in the early evaluation of the built device prototypes (i.e., following the test-first principle), but considering factors from the academic setting, such as the scarcity of time, resources and personnel.

In relation to the applications of the Interactive Sphere, although this work has focused on the domain of digital games and virtual reality applications, it would be fruitful to investigate how the Interactive Sphere could be employed in other types of applications. For instance, it is possible to explore the creation of gestures and applications with a more communicative aspect, that is, representing more complex signals, rather than just manipulative gestures, that is, representing actions. van den Hoven and Mazalek (2011) describe a taxonomy that classifies gestures based on these aspects and review related literature. According to their study, tangible interactions and gesture interactions in games tend to focus more on manipulative functions than on communicative ones (perhaps due to the very nature of these interfaces and applications), although many studies point out the relation of gestures to communication, even suggesting that gestures could have given rise to verbal expression (Tomasello, 2008). In this sense, a question to be investigated is to what extent the Interactive Sphere, and other tangible gesture interaction devices, could support gestures with a communicative function.

6.1 Limitations

The implemented prototype for the Interactive Sphere present some limitations, such as the absent of the external silicon coating. In addition, despite having been designed, a few non-essential elements also were not implemented, such as the power supply module and the base of the device. The daemon designed in the device's software architecture is also not available. However, as an Interactive Sphere client was already implemented as a Python library, the absence of this daemon also does not prevent the use of the device. Although the applications can access the device directly through the Python client, the daemon could act as an intermediary layer between the device and the applications, thus simplifying access to the device by these client applications. Finally, another identified limitation regards the reliability of the pressure sensors in the functional prototype. That is, in some cases, pressing a sensor is not correctly identified, or even another sensor is identified instead. However, individual tests have already been performed with each of the sensors, with no apparent problem being identified in this configuration. New investigations would have to be performed to solve this issue. Another important aspect, which could not be addressed in this article, concerns conducting experiments to assess the capabilities of the Interactive Sphere. Future work should address this issue.

7 Conclusion

This article presented the project, the design and the development of the Interactive Sphere, a spherical controller that was designed for interacting with games and virtual reality environments. This device unites a set of interaction resources in a unique combination to its spherical shape, which can contribute to inspire the design of new physical artifacts of interaction. This device uses pressure sensors and a soft coating to provide a pleasant tactile response to the user. In addition, the distribution of these sensors around the artifact, along with gestural interaction, also allows for the user to execute a wide range of commands, even while using a single hand. For the control of more complex applications, we envision the possibility of the combined use of two Interactive Spheres, being one held in each of the user's hand.

The process of design and development of the Interactive Sphere was structured and described as a sequence of activities, which can be adapted by other researchers and developers in future projects. Aiming at assisting the project of other controllers or interactive devices, this work also detailed a set of guidelines for implementing pressure detection, using capacitive sensors, and gesture recognition, using accelerometer and gyroscope. These guidelines were described based on the lessons learned with the construction of the Interactive Sphere. The methods for building these capacitive sensors could be performed in a manual manner and without the use of specialized equipment, which we envision to allow the replication of this sensor building process in other projects. In addition to this, the gesture recognition techniques presented in this work could also be applied to other contexts, as they can be implemented using a single electronic module, with accelerometer and gyroscope, with a relatively reduced size. The employment of machine learning for gesture recognition also simplifies the creation and recognition of gestural commands, despite of its greater complexity of implementation.

Future work could address some of the limitations in the implemented prototype, such as the lack of the silicone coating for the Interactive Sphere. Tests are needed in order to verify not only the accuracy and the reliability of the sensors and of the techniques for gesture recognition, but also aspects related to Human-Computer Interaction, such accessibility, usability and ergonomics evaluation of the device. Although preliminary tests have already been performed, it would be important to verify the functioning of the device with a larger group of potential users. It is also important to test the Interaction Sphere with different kinds of games as a way of verifying, for example which game genres this device would be most suited for.

Acknowledgements

This work was supported by the Physical Artifacts of Interaction Research Group (PAIRG) at the Federal University of Rio Grande do Norte (UFRN), and partially funded by the Brazilian National Council of Scientific and Technological Development (CNPq Grant #156239/2019-1). We also thank the resources of the PAIRG's Laboratory of Physical and Physiological Computing (PAIRG L2PC) at UFRN. The present article is an extended and revised version

of a work previously presented at the 21st Brazilian Symposium on Games and Digital Entertainment (SBGames'22) under the title "Esfera Interativa: Design de Um Novo Controle para Jogos Digitais" (Brizolara et al., 2022). The authors thank the guest editors to Journal on Interactive Systems (JIS) for the invitation to submit to this journal an extended version of the aforementioned work. The author contributions, following the CRediT taxonomy, are as follows: Paulo L. S. Brizolara (Conceptualization, Data curation, Formal Analysis, Investigation, Methodology, Software, Writing - original draft), Leonardo Cunha de Miranda (Conceptualization, Formal Analysis, Funding acquisition, Methodology, Project administration, Resources, Supervision, Validation, Writing - original draft), Gabriel Alves Mendes Vasiljevic (Writing - original draft), and Bruna Camila de Menezes (Conceptualization, Visualization).

References

- Almeida, L. D. A., de Almeida Neris, V. P., Miranda, L. C., Hayashi, E. C. S., and Baranauskas, M. C. C. (2009). Designing inclusive social networks: A participatory approach. In *Online Communities and Social Computing*, pages 653–662. Springer Berlin Heidelberg.
- Angelini, L., Lalanne, D., Hoven, E., Khaled, O., and Mugellini, E. (2015). Move, hold and touch: A framework for tangible gesture interactive systems. *Machines*, 3(3):173–207.
- Baranauskas, M. C. C., Liu, K., and Chong, S. (2003). Website interfaces as representamina of organizational behaviour. In *Dynamics and Change in Organizations*, pages 63–87.
- Brizolara, P. L. S., Miranda, L. C., and de Menezes, B. C. (2022). Esfera Interativa: Design de Um Novo Controle para Jogos Digitais. In *Anais Estendidos do Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames)*, pages 168–177. SBC.
- Brown, M., Kehoe, A., Kirakowski, J., and Pitt, I. (2015). Beyond the Gamepad: HCI and Game Controller Design and Evaluation. In *Game User Experience Evaluation*, Human-Computer Interaction Series.
- Carroll, J. M. (2003). *Making Use: Scenario-Based Design of Human-Computer Interactions*. MIT Press.
- Correia, A. C. C., Miranda, L. C., and Hornung, H. (2013). Gesture-based interaction in domotic environments: State of the art and HCI framework inspired by the diversity. In *Human-Computer Interaction – INTERACT 2013*, pages 300–317. Springer Berlin Heidelberg.
- Dobbelstein, D., Hock, P., and Rukzio, E. (2015). Belt: An unobtrusive touch input device for head-worn displays. In *Conference on Human Factors in Computing Systems - Proceedings*, volume 2015-April, pages 2135–2138.
- Ey, M., Pietruch, J., and Schwartz, D. I. (2010). "Oh-No! Banjo": A case study in alternative game controllers. In *Proceedings of the International Academic Conference on the Future of Game Design and Technology*, Futureplay '10, pages 215–218, New York, NY, USA. Association for Computing Machinery.
- Foottit, J., Brown, D., Marks, S., and Connor, A. M. (2014). An Intuitive Tangible Game Controller. In *Proceedings of the 2014 Conference on Interactive Entertainment*, IE2014, pages 1–7, New York, NY, USA. Association for Computing Machinery.
- Grosse-Puppenthal, T., Holz, C., Cohn, G., Wimmer, R., Bechtold, O., Hodges, S., Reynolds, M. S., and Smith, J. R. (2017). Finding Common Ground: A Survey of Capacitive Sensing in Human-Computer Interaction. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 3293–3315, New York, NY, USA. Association for Computing Machinery.
- Heberlein, M., Hayashi, T., Nashold, S., and Teeravarunyou, S. (2003). CHI-ball, an interactive device assisting martial arts education for children. In *Proc. of CHI '03*.
- Hufnal, D., Osborne, E., Johnson, T., and Yildirim, C. (2019). The Impact of Controller Type on Video Game User Experience in Virtual Reality. In *GEM '19*, pages 1–9.
- Hunt, A. and Thomas, D. (2019). *The Pragmatic Programmer: Your Journey to Mastery, 20th Anniversary Edition, 2nd Edition*. Addison-Wesley Professional.
- Karime, A., Eid, M., Gueaieb, W., and El Saddik, A. (2012). Determining wrist reference kinematics using a sensory-mounted stress ball. In *IEEE ROSE 2012*, pages 109–114.
- Khalfin, I. (2002). Method and apparatus for electromagnetic position and orientation tracking with distortion compensation employing modulated signal. United States Patent US6624626B2. filed 4 June 2002, and issued 23 September 2003.
- Liu, K. (2000). *Semiotics in Information Systems Engineering*. Cambridge Press.
- Llagostera, E. G. (2019). Critical Controllers: How Alternative Game Controllers Foster Reflective Game Design. In *Companion Publication of DIS '19*, DIS '19 Companion, pages 85–88.
- Margolis, M. (2011). *Arduino Cookbook: Recipes to Begin, Expand, and Enhance Your Projects*. "O'Reilly Media, Inc."
- Miranda, L. C., Hayashi, E. C. S., and Baranauskas, M. C. C. (2009). Identifying interaction barriers in the use of remote controls. In *CLIHC '09*, CLIHC '09, pages 97–104.
- Miranda, L. C., Hornung, H., and Baranauskas, C. (2011). Prospecting a New Physical Artifact of Interaction for iDTV: Results of Participatory Practices. In *DUXU 2011*, Lecture Notes in Computer Science.
- Miranda, L. C., Hornung, H., and Baranauskas, M. (2010). Adjustable interactive rings for iDTV. *IEEE Transactions on Consumer Electronics*, 56(3):1988–1996.
- Miranda, L. C., Hornung, H., Pereira, R., and Baranauskas, M. C. C. (2013). Exploring Adjustable Interactive Rings in Game Playing: Preliminary Results. In *DUXU'13*.
- Miranda, L. C., Hornung, H., Solarte, D., Romani, R., Weinfurter, M., Neris, V., and Baranauskas, M. C. (2007). Laptops educacionais de baixo custo: Prospectos e desafios. In *XVIII Simpósio Brasileiro de Informática Na Educação (SBIE)*, pages 358–367.
- Miranda, L. C., Piccolo, L., and Baranauskas, C. (2008). Artefatos físicos de interação com a TVDI: Desafios e diretrizes para o cenário brasileiro. In *IHC '08*, IHC '08, pages 60–69.
- Muller, M., Haslwanter, J., and Dayton, T. (1997). Chp. 11 - Participatory Practices in the Software Lifecycle. In *Hand-*

- book of *Human-Computer Interaction*, pages 255–297.
- Murphy, S. C. (2014). Controllers. In Wolf, M. J. P. and Perron, B., editors, *The Routledge Companion to Video Game Studies*, pages 19–24. Routledge, New York.
- Nagels, S., Ramakers, R., Luyten, K., and Deferme, W. (2018). Silicone devices: A scalable DIY approach for fabricating self-contained multi-layered soft circuits using microfluidics. In *Proc. of CHI '18*, CHI '18, pages 1–13.
- Nassar, J., Cordero, M., Kutbee, A., and ... Hussain, M. (2016). Paper Skin Multisensory Platform for Simultaneous Environmental Monitoring. *Adv. Mater. Technol.*, 1.
- North, D. (2006). Introducing BDD. <https://dan-north.net/introducing-bdd/>.
- O'Sullivan, D. and Igoe, T. (2004). Chapter 8: Physical Interaction Design, or Techniques for Polite Conversation. In *Physical Computing: Sensing and Controlling the Physical World with Computers*, pages 205–216. Course Technology Press, Boston, MA, USA.
- Pereira, R., Buchdid, S., and Baranauskas, C. (2013). Values and Cultural Aspects in Design: Artifacts for Making Them Explicit in Design Activities. In *ICEIS'12*, Lecture Notes in Business Information Processing.
- Perelman, G., Serrano, M., Raynal, M., and ... Dubois, E. (2015). The roly-poly mouse: Designing a rolling input device unifying 2d and 3D interaction. In *CHI '15*.
- Pourjafarian, N., Withana, A., Paradiso, J. A., and Steimle, J. (2019). Multi-Touch Kit: A Do-It-Yourself Technique for Capacitive Multi-Touch Sensing Using a Commodity Microcontroller. In *Proc. of UIST'19*, UIST '19, pages 1071–1083.
- Rosenberg, I. D., Grau, A., Hendee, C., Awad, N., and Perlin, K. (2009). IMPAD: An inexpensive multi-touchpressure acquisition device. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '09, pages 3217–3222, New York, NY, USA. Association for Computing Machinery.
- Roudaut, A., Martinez, D., Chohan, A., Otrocol, V.-S., Cobbe-Warburton, R., Steele, M., and Patrichi, I.-M. (2014). Rubikon: A highly reconfigurable device for advanced interaction. In *Conference on Human Factors in Computing Systems - Proceedings*, pages 1327–1332.
- Rüthschilling, A. A., Cunha, J., and Broega, A. C. (2008). The draping technique as a creative phase in the fashion design methodology. Association of Universities for Textiles (AUTEX).
- Santana, V. F., Solarte, D. S. M., Neris, V. P. A., Miranda, L. C., and Baranauskas, M. C. C. (2009). Redes sociais online: Desafios e possibilidades para o contexto brasileiro. In *XXXVI Seminário Integrado de Software e Hardware (SEMISH) / XXIX Congresso da Sociedade Brasileira de Computação (CSBC)*, pages 339–353, Bento Gonçalves, RS. SBC.
- Sarinho, V. T. (2017). Cornestick: Um Joystick de Sopro para Jogos Digitais. In *XVI SBGames*, pages 586–589.
- Shahmiri, F., Chen, C., Waghmare, A., Zhang, D., Mittal, S., Zhang, S., Wang, Y.-C., Wang, Z., Starner, T., and Abowd, G. (2019). Serpentine: A reversibly deformable cord sensor for human input. In *Conference on Human Factors in Computing Systems - Proceedings*.
- Solis, C. and Wang, X. (2011). A Study of the Characteristics of Behaviour Driven Development. In *2011 37th EUROMICRO Conference on Software Engineering and Advanced Applications*, pages 383–387.
- Souza, P. d. M. (2006). A modelagem tridimensional como implemento do processo de desenvolvimento do produto de moda. Master's thesis, Universidade Estadual Paulista, Faculdade de Arquitetura, Artes e Comunicação.
- Stamper, R. K. (2001). Organisational Semiotics: Informatics without the Computer? *Information, Organisation and Technology*, pages 115–171.
- Tomasello, M. (2008). *Origins of Human Communication*. The MIT Press.
- van den Hoven, E. and Mazalek, A. (2011). Grasping gestures: Gesturing with physical artifacts. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 25(3):255–271.
- Varesano, F. and Vernero, F. (2012). Introducing PALLA, a novel input device for leisure activities: A case study on a tangible video game for seniors. *ACM International Conference Proceeding Series*, pages 35–44.
- Weinberg, G., Orth, M., and Russo, P. (2000). The embroidered musical ball: A squeezable instrument for expressive performance. In *Proc. of CHI EA '00*, CHI EA '00, pages 283–284.