INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

MESTRADO EM ENGENHARIA INFORMÁTICA

isep



Apple Vision Pro and Spatial Computing: A Feasibility Study for Game Development

PEDRO NUNO LAPA SANTOS

Setembro de 2025

POLITÉCNICO
DO PORTO

ISEP INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

Apple Vision Pro and Spatial Computing: A Feasibility Study for Game Development

Pedro Santos

A dissertation submitted in partial fulfilment of the requirements for the degree of Master of Science, Specialisation Area of Software Engineering

Advisor: Prof. Doutor Carlos Vaz de Carvalho
Supervisor: João Jacob, Ph.D

Porto, September 13, 2025

Statement of Integrity

I hereby declare that I have conducted this academic work with integrity.

I have not plagiarised or applied any form of undue use of information or falsification of results along the process leading to its elaboration.

Therefore, the work presented in this document is original and authored by me, having not previously been used for any other end. The exceptions are explicitly recognised in the section "Ethical considerations" of the first chapter. This section also states how AI tools were used and for what purpose.

I further declare that I have fully acknowledged the Code of Ethical Conduct of P.PORTO.

ISEP, Porto, September 13, 2025

Abstract

Apple has recently launched a new spatial computing and Extended Reality (XR) platform that aims to combine virtual, augmented and mixed reality to offer users an unique experience in the Apple ecosystem. The Apple Vision Pro and its visionOS operating system mark an exciting new player in the market. As this is a new platform with a mostly unexplored market, organizations like Mindera have an exciting opportunity to be early adopters of this platform, if it proves to be a worthy investment.

This project explores the Apple Vision Pro's feasibility from a game development perspective through a systematic literature review, as well as the design and development of an original tabletop game that leverages the platform's unique capabilities. Semi-structured interviews with XR and game development experts were conducted to validate the prototype and gather insights on the platform's technical and practical aspects. Findings reveal that, while it offers exciting features such as controller-free natural interactions that use eye-tracking and gesture recognition, it has some noticeable limitations like its limited availability and high price. Overall, the study concludes that, although the Apple Vision Pro demonstrates promising potential for XR game development, it still is at an intermediate stage of technological maturity (TRL6).

Keywords: Apple Vision Pro, Extended Reality, Game Development, Spatial Computing, Tabletop, visionOS

Resumo

Os rápidos avanços nas tecnologias de realidade estendida (XR) abriram novas oportunidades para criar experiências imersivas com estas tecnologias. Apesar disso, este mercado ainda apresenta alguns desafios únicos. Em 2024, a Apple lançou um novo sistema relacionado com computação espacial e XR que combina vários aspetos destas tecnologias: realidade virtual, realidade aumentada e realidade mista. O Apple Vision Pro e o seu sistema operativo visionOS marcam a entrada neste mercado de uma marca com histórico de ter produtos lucráveis em mercados inicialmente limitados. Como se trata de uma nova plataforma com um mercado pouco explorado, poderá ser uma excelente oportunidade para organizações como a Mindera entrarem num mercado com pouca oferta que poderá evoluir consideravelmente no futuro. Esta entrada irá depender se é um mercado em que o investimento seja justificável. Neste estudo foi explorada a viabilidade do Apple Vision Pro e do seu sistema operativo, num contexto de desenvolvimento de jogos.

A literatura existente foi revista para obter mais informações sobre a plataforma, as suas vantagens, desvantagens, comparações com outros sistemas de XR, bem como tendências e desafios atuais na área de desenvolvimento de jogos XR. Seguidamente, foi desenhado e desenvolvido um protótipo de um novo jogo de tabuleiro para visionOS que tirasse partido das funcionalidades únicas do Apple Vision Pro. Com este protótipo desenvolvido, foram realizadas entrevistas semiestruturadas a especialistas nas áreas de XR e desenvolvimento de jogos, com o intuito da recolha de informações acerca de oportunidades e desafios de desenvolver jogos para a plataforma. Com tudo em mente, os resultados sugerem que o Apple Vision Pro apresenta bastantes funcionalidades interessantes que podem ser exploradas em termos de *game design*. Por exemplo, os jogadores podem interagir com o sistema sem necessidade do uso de comandos através das funcionalidades de reconhecimento gestual e monitorização ocular. Apesar de apresentar algumas vantagens, a plataforma também apresenta algumas limitações como o facto de ainda não ter sido lançada em vários países e ter um preço elevado. Em suma, apesar da plataforma apresentar bastante potencial e ser um marco na inovação da área de XR com funcionalidades únicas, ainda é bastante limitada por fatores como não estar disponível em vários países. Com isto em mente, a plataforma ainda está numa fase intermédia de maturidade tecnológica (TRL6).

Acknowledgement

The present document and prototype developed were made possible thanks to the support of many people and institutions, without whom this study would have not been possible. To all, I extend my deepest gratitude.

First and foremost, I would like to thank my partner, Jéssica, for her unconditional love, encouragement and unwavering support throughout my entire academic journey. To my parents, Carmen and Paulo, for being my role models, for their patience and education, which have shaped me into who I am today. To my brother, Rafael, who motivated me to become a better person every day. To my grandparents (Emília, Ramiro, Eufémia and Genésio), I give my deepest appreciation as they were sources of wisdom and inspiration throughout my whole life.

To my academic advisor, Professor Carlos Vaz de Carvalho, I am thankful for his guidance and feedback that helped shape this study into its final result. I am sincerely grateful to Instituto Superior de Engenharia do Porto (ISEP) and all academic staff that I've interacted with during the course of both my Bachelor's and Master's degrees.

From Mindera, I want to thank my supervisor, João Jacob, for his availability, mentorship and for sharing knowledge that was crucial in order to overcome the challenges faced in this study. Special thanks also goes to Pedro Vicente for making this project possible.

Finally, I want to express my gratitude to Mindera for the opportunity to participate in a project that explores technological innovation in fields that I have vast interest in and for allowing me to collaborate with exceptional people along the way.

Contents

List of Figures

List of Tables

Listings

List of Acronyms

AI	Artificial Intelligence.
API	Application Programming Interface.
AR	Augmented Reality.
AV	Augmented Virtuality.
CR	Cross Reality.
DJ	Disc Jockey.
DSR	Design Science Research.
FOV	Field of View.
FURPS+	Functionality, Usability, Reliability, Performance, and Supportability + Design Constraints, Implementation Constraints, Interface Constraints, Physical Constraints, among others.
GDPR	General Data Protection Directive.
GILT	Games Interaction and Learning Technologies.
HMD	Head-Mounted Display.
HUD	Heads-Up Display.
IDE	Integrated Development Environment.
IEEE	Institute of Electrical and Electronics Engineers.
iOS	iPhone Operating System.
iPadOS	iPad Operating System.
ISEP	Instituto Superior de Engenharia do Porto.
MR	Mixed Reality.
MS	Mapping Study.
MVVM	Model-View-ViewModel.
NSPE	National Society of Professional Engineers.
OST	Optical See-Through.
PICOCS	Population, Intervention, Comparison, Outcomes, Context, Study design.
PREPD	Preparação para Dissertação.

PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses.
SoC	System on a Chip.
TRL	Technology Readiness Level.
UI	User Interface.
USA	United States of America.
VAC	Vergence-Accommodation Confict.
VR	Virtual Reality.
VST	Video See-Through.
XR	Extended Reality.

Chapter 1

Introduction

This first chapter aims to introduce the reader to the study in question. Firstly, the context and problem statement will be provided, followed by the objectives, methodology, planning, organizational context, ethical considerations and, lastly, the document's structure.

1.1 Context

Extended Reality (XR) is a hypernym of other technologies that bridges the digital and physical worlds: Virtual Reality (VR), Augmented Reality (AR) and Mixed Reality (MR). This technology creates immersive and interactive experiences that allow its users to interact with digital objects and environments. While VR focuses on immersing users in completely simulated environments, AR presents digital components in the real world and MR is a blend of both technologies that allows both real and generated elements to interact with each other. These experiences need a specially designed headset that serves as a gateway between users and the digital elements displayed by these systems. XR technologies offer innovative ways for user interaction, transforming industries such as gaming, retail, healthcare, engineering, construction, education and others. This is a frontier for technological innovation, pushing the boundaries of spatial computing in order to refine how users interact with digital content. This paves the way for new platforms and organizations to enter this market, either via creating hardware or developing XR software (Ratcliffe et al. 2021).

The Apple Vision Pro is a new spatial computing headset from Apple that allows users to have immersive experiences by combining virtual, augmented and mixed reality (extended reality). This device is powered by visionOS, Apple's spatial computing operating system that was released in the beginning of 2024 along with the headset. This platform leverages technologies like hand and eye tracking, gestures, 3D spatial interactions and immersive audio to provide the user with an innovative experience (Xu et al. 2024).

1.2 Problem Statement

The introduction of the Apple Vision Pro and its visionOS operating system marks a significant new platform entering the spatial computing and XR market. The platform promises new possibilities in spatial computing, yet its potential remains largely unexplored as this is a first-generation device with high entry costs and a large technical learning curve. As such, it presents some challenges and opportunities for the stakeholders of this project. Unlike other more established ecosystems, the new visionOS operating system requires technical proficiency with frameworks like RealityKit and ARKit, which are tools specific for Apple's platforms. The Apple Vision Pro's market is still growing and is mostly unexplored, which presents itself as an exciting opportunity to be an early adopter if proven to be a worthy investment. For organizations, these uncertainties hinder the decision-making process on whether the platform is feasible for software development and if its market is worth investing and exploring, especially since it's an untested one that has limited user adoption. This project aims to explore the feasibility of this platform, particularly from a game development perspective.

The organization in question (Mindera) and its employees are stakeholders of this project as this is a new platform and market that, if feasible, can be explored. There are many stakeholders within Mindera that are interested in this project, such as project managers, developers, and more. Furthermore, the Games Interaction and Learning Technologies (GILT) unit can also be considered a stakeholder. Lastly, the users of products developed by Mindera act as secondary stakeholders for this project. As this project aims to explore the Apple Vision Pro's feasibility for game development, its outcome will impact the stakeholders perception of the platform and its viability in this context. Failure to solve this problem could result in potential missed opportunities to enter a new growing market for the stakeholders. On the other hand, identifying if the platform is feasible or not can allow the organization to make an educated decision if the market is worth investing from a game development standpoint.

1.3 Objectives

As mentioned before, this study seeks to evaluate the feasibility of the Apple Vision Pro platform for game development. In order to complete this research, some objectives need to be achieved:

- Understand the capabilities and limitations of the Apple Vision Pro and visionOS for game development;
- Design an engaging game concept that leverages the platform's unique features;
- Develop a spatial computing/extended reality (XR) game for visionOS, utilizing SwiftUI, RealityKit and ARKit;
- Evaluate the feasibility of game development for the Apple Vision Pro and visionOS.

1.4 Methodology

Firstly, the research area was defined as *"XR technologies and game development"* due to this dissertation aiming to explore cutting edge XR technologies (Apple Vision Pro and visionOS) and their feasibility in game development. Moreover, the chosen research topic for this study was the *"Technical and practical challenges associated with using the Apple Vision Pro and visionOS for game development"*. This topic explores into the unique difficulties developers might encounter when working with the Apple Vision Pro platform and visionOS operating system while developing games.

As we can see in Figure 1.1, *"Design and Creation"* (Oates, Griffiths, and McLean 2022) was chosen as the research strategy for this thesis. This is highly suited as the project aims to design, implement and evaluate a game prototype for the Apple Vision Pro in order to draw conclusions from this artifact regarding the platform's feasibility in game development. Furthermore, the research relies on the review of existing and relevant documentation as a source of data (*"Documents"* in Figure 1.1). This documentation was gathered in platforms like ACM Digital Library, *IEE Xplore*, among others. There was an emphasis on more recent publications in order to gather the most up-to-date information possible, as this is a new market. All sources inside the *Systematic Literature Review* section of the *State of the Art* chapter used academic documentation. On the other hand, some grey literature were used in other sections of the document. This is due to this study exploring newly released platforms with scarce academic documentation. Moreover, developing the prototype will provide indirect key observations on technical challenges, patterns and successful approaches (*"Observations"*). The prototype will be presented to industry game development and XR experts via semi-structured expert interviews (*"Interviews"* in the diagram). These interviews aim to gather insights about the Apple Vision Pro, visionOS and the developed prototype. After each interview, experts were sent a *"Questionnaire"* to gather extra feedback not discussed in the interview. Lastly, the platform's feasibility for game development will be examined based on *"Qualitative"* methods, as it is mostly based on the author's and industry expert's interpretation of the platform's capabilities, limitations and overall viability based on the research, design, development and evaluation done in this study.

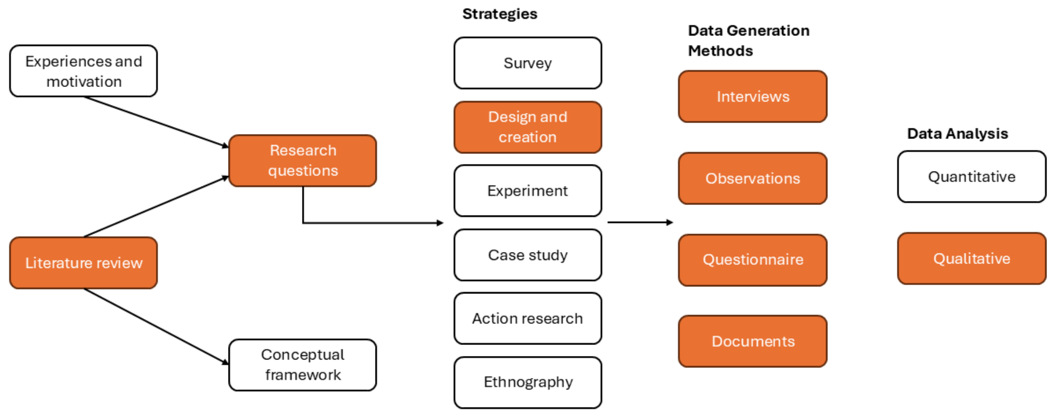


Figure 1.1: Research Process Diagram (Oates, Griffiths, and McLean 2022)

In terms of methodology *"Design Science Research (DSR)"* (Saltuk and Kosan 2014) was chosen as it is particularly suited to this type of study due to its focus on creating and evaluating artifacts to solve recognized issues, generating knowledge that contributes to the research area. With the research area, topic, strategy and methodology already defined, the following question was chosen as the central research question that drives this thesis:

- *What are the technical and practical challenges of using the Apple Vision Pro and visionOS for game development?*

It encapsulates the problem as this project seeks to explore whether the Apple Vision Pro and its operating system are viable options for XR and the development of spatial computing games. Identifying the technical and practical challenges of this platform's use in game development makes this a highly relevant question in order to understand the potential of this new and largely unexplored platform. To comprehensively address this question, several sub-questions are posed:

1. *What are the capabilities and limitations of the Apple Vision Pro and visionOS?*
2. *How does the Apple Vision Pro's capabilities compare to other XR platforms?*
3. *What are the current trends and challenges in XR game development, and how might these apply to the Apple Vision Pro?*
4. *How can the unique features of the Apple Vision Pro be leveraged in game design?*
5. *What are the technical challenges encountered during game development for visionOS, and how can they be mitigated?*

These break down the broader central research question into more specific questions that address technical capabilities, comparative analysis, current trends and challenges in the XR gaming industry, design leverage, and the practicality of developing a game using the platform. In general, these lay the foundation for an in-depth exploration of the Apple Vision Pro and visionOS as a game development platform. Table 1.1 shows the relationship between each objective and the previously defined research questions (the main research question, as well as its sub-questions) so it is possible to identify which questions contribute to each objective.

Table 1.1: Relationship Between Objectives and Research Questions

Objective	**Research Question(s)**
Understand the capabilities and limitations of the Apple Vision Pro and visionOS for game development	Sub-Questions 1, 2 and 3
Design an engaging game concept that leverages the platform's unique features	Sub-Question 4
Develop a spatial computing/extended reality (XR) game for visionOS, utilizing SwiftUI, RealityKit and ARKit	Sub-Question 5
Evaluate the feasibility of game development for the Apple Vision Pro and visionOS	Main Research Question

In terms of development and since the project was developed alone by the author, Kanban was used as a methodology. Its emphasis on limiting work in progress rather than focusing on team roles, ceremonies or fixed sprints is well-suited for solo developers. This methodology also allowed for continuous improvement, one of Kanban's principles.

1.5 Planning

In the order to determine the project timeline, some milestones were set, carefully estimated (in days) and their predecessors were identified. As such, each milestone was placed in a specific order for the project to be delivered. Furthermore, the start date for the project was defined as January 20th 2025. As we can see in Figure 1.2, the project timeline was created and displays all the needed dependencies between tasks or identifies which tasks can be performed in parallel to each other. For example, there is a dependency between the prototype implementation and writing that portion of the document, as one task cannot be done before the other.

With the time estimates in mind, the *Planning* phase was planned to take 20 days from the start of the project. This aimed to create the initial version of the dissertation document. Furthermore, the *Design* phase was planned to take 19 days to analyse the problem, create the concept and design for the prototype game. The *Build* phase was planned to take 80 days to develop the game prototype and create an updated version of the report with the findings for this phase. Moreover, the *Evaluation* phase was planned to take 15 days so that all expert semi-structured interviews are performed, as well as their results analysed and written in a new report version. Lastly, the *Conclusion* phase was planned to take 26 days to finish the document and create a presentation.

Milestones were also identified in this timeline with a specific task with an *[M]* prefix (e.g. *[M] Game Prototype Finished*). These were the identified milestones and their planned dates:

- Initial dissertation document: 14-02-2025
- Document with Analysis, Concept and Design: 03-03-2025
- Game Prototype Finished: 26-05-2025
- Document with Game Prototype: 23-06-2025
- Document with Results: 22-07-2025
- Final Document: 15-08-2025
- Presentation: 27-08-2025

With all these phases and milestones in mind, the project started on *January 20th 2025* and its planned ending was on *August 29th 2025*, meaning that this project was planned to take *160 days* to complete.

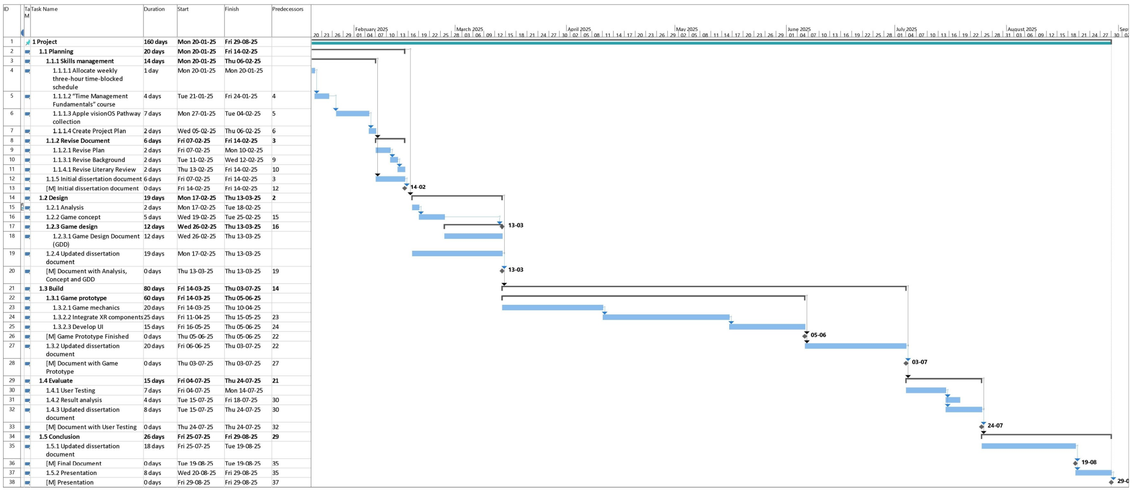


Figure 1.2: Project Timeline

In order to track progress for this project, the communication between the author and the organization's supervisor was done via sporadic status check meetings where progress was reported and showcased. Other than the previously mentioned meetings, communication was also done through the companies private channels on an ad hoc basis. Conversely, the author and the advisor mainly communicated via email and meetings in a non-structured schedule.

1.6 Organizational Context

As mentioned before, this study was carried out in collaboration with Mindera, an engineering company founded in 2014 during a dinner with friends. At present, it has more than a thousand employees across more than 9 countries with clients spread around the world. Mindera works in various industries, including gaming, data, mobile and web development where it applies state of the art technology to solve their clients' needs. Although Mindera works with clients, it also has in-house products as well. The company is known for their self-organization culture, flexible working hours, unlimited holidays, health insurance, the freedom to choose one's own equipment, among other benefits that promote personal autonomy. This approach allows Mindera's employees to focus on their well-being (both personal and family), while actively promoting a healthy work-life balance (Eco 2022).

1.7 Ethical Considerations

Ethics play an integral role in research. This dissertation is no different, as it must ensure that the research follows both academic and professional standards. As such, this study follows the *Code of Good Practices and Conduct of P.Porto*, with a focus on articles 6, 8 and 10. The sixth article refers to the duties of each student. For example, point *2.8* prohibits the use of ideas, sentences, paragraphs, or texts from other works without proper citation or accurate referencing. This is specially relevant as following this point ensures that all references in this document will be correctly acknowledged without plagiarism. Another great point is *2.9*, which prevents presenting work as original if it has already been exhibited or published on another occasion without explicit indication. Since this research builds upon existing research, it is important that said work is not passed off as original by being transparent about its origins. There are more relevant points related to the duties of each student, such as *2.10* and *2.11*. As for Article 8, it refers to a declaration of commitment and is fairly relevant to this dissertation, especially its first point. This mentions a statement of integrity that provides a declaration that no plagiarism occurred, the work is original and created by the author, it was not previously used elsewhere and sources are correctly cited. This statement of integrity is present at the beginning of this document. Furthermore, article 10 outlines integrity and good practices in research. It ensures that said research is conducted in a responsible manner, with impartial analysis and objective conclusions. For example, point *1.c* from Article 10 mentions truthfulness, accuracy and originality during the proposal, execution, or revision of a research project and the communication of its results. This is important in the present study as it must not embellish the Apple Vision Pro's capabilities or feasibility, as it might impact the organization's decision about exploring its market (IPP 2020).

As this is a software engineering study, there are some other relevant codes of ethics, such as the ACM/IEEE-CS Software Engineering Code that sets principles for the software engineering profession regarding public interest, product, professional judgment, ethics in management, profession, their colleagues and themselves (Gotterbarn, Miller, and Rogerson 1997). The principles outlined in the Institute of Electrical and Electronics Engineers (IEEE) Code of Ethics also guide this work as they set integrity and responsibility standards (IEEE 2020). Furthermore, the Code of Ethics for the Portuguese Board of Engineers is also relevant as this is an engineering study made in Portugal. This code tackles duties from an engineering perspective (Ordem dos Engenheiros 2016). The National Society of Professional Engineers (NSPE) Code of Ethics for Engineers also outlines similar principles and rules of practice designed to help engineers uphold the highest standards of ethical conduct in their professional practice (NSPE 2020).

Privacy is also a really important ethical consideration, as this project will implement a prototype that will be presented to industry experts in recorded and transcribed interviews. Any data collection practices must have data protection and privacy in mind. In order to do that, the data collection needs to follow the EU Data Protection Law (European Commission 2024b), with a special emphasis on its General Data Protection Directive (GDPR). For data collection to follow GDPR, users must explicitly consent before any data is collected, be fully informed about what data will be collected, said data must be necessary for this study, stating its purpose, protecting the security and confidentiality of said data. The users must be informed what data is being collected, what its purpose is, and who has access to it. They should be able to access a copy of their data, as well as transfer or delete said data (European Commission 2024a). This was done via a consent form sent to experts alongside their invite. This informs them of the purpose of the interview, that it will be recorded, transcribed and later used in this document preserving anonymity. Participants gave their consent using the consent form, available in full in Appendix B.

Moreover, there are also ethical considerations with the use of XR technologies as they might impact people's social relationships. As the hardware for these technologies is camera-based, there are some ethical considerations regarding it being considered non consensual recording when used in public spaces. These technologies also come with some privacy concerns since they capture the user's personal context (Millard et al. 2024).

The use of content generated by Artificial Intelligence (AI) in this study was restricted to the transcripts gathered during each of the expert interviews. These transcripts were gathered using Microsoft Teams' transcript feature that uses artificial intelligence internally to record the participants' voice and translate it into text. Although these transcripts were gathered using AI, these were reviewed and edited by the author to improve its results before being added to Appendix C.

Finally, some chapters of this document were partially written for the Preparação para Dissertação (PREPD) curricular unit. These include this *Introduction* chapter, as well as the *State of the Art* chapter. While both chapters were originally written for that curricular unit, sections were added and the existing ones were reviewed and updated for this document.

1.8 Document Structure

The present document has six chapters: Introduction (the current chapter), State of the Art, Prototype Analysis and Design, Development, Results and Findings, and Conclusion. Additionally, it presents three appendices: the first displays additional screenshots of the prototype that were not used in the main baody of the document, one appendix detailing the expert interview's process and another with the raw transcripts of each interview.

The second chapter relates to the state of the art. It starts by providing a background regarding technologies that will be mentioned throughout this study: VR, AR, MR, XR and spatial computing. Furthermore, a systematic literature review will be presented where its method will be defined and research questions analysed. This analysis will frame each question using the Population, Intervention, Comparison, Outcomes, Context, Study design (PICOCS) model, inclusion and exclusion criteria will be derived and the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) flow will be followed in order to answer to each question. Finally, some XR technologies will be presented.

Thirdly, the *Prototype Analysis and Design* chapter will present the prototype analysis, which contains the detailing of its concept, rules, a flowchart, target audience, alternative concepts, domain, functional requirements and non-functional requirements. Moreover, its design will be described, starting by its architecture and its many levels of detail, user interface, gesture and branding.

In the fourth chapter, the development process of the prototype will be detailed. Starting by each technical task and then going over every use cases' implementation process. Afterwards, some additional benefits and assets used will be identified.

The *Results and Findings* chapter will go over the platform's challenges and limitations found in the development process of the prototype for visionOS, what the technology allowed to be implemented compared to the original prototype vision and the expert interview process and insights gathered about the platform and prototype. Afterwards, all research questions described in this chapter will be answered with information gathered throughout the study.

Finally, the sixth chapter will provide a summary of all findings, identifying the conclusions made in this study. Afterwards, each objective will be analysed to see to what level were they achieved. This is then followed by the limitations and challenges of this study. Finally, future work and a final appreciation will be described.

Chapter 2

State of the Art

This chapter will present the state of the art in XR technologies, with a specific focus on the Apple Vision Pro, visionOS and game development. Firstly, a *Background* section will give context regarding concepts mentioned throughout this project such as VR, AR, MR, XR and spatial computing. Afterwards, a systematic mapping study and comprehensive literature review will investigate the Apple Vision Pro, visionOS, their benefits, limitations, comparisons to other XR platforms and current XR game development trends and challenges, in an attempt to answer the research questions. Lastly, some technologies related with the Apple Vision Pro, visionOS and XR will be presented.

2.1 Background

This section aims to provide a detailed background of concepts that will be used throughout this study. These concepts include VR, AR, MR, XR, as well as spatial computing.

2.1.1 Virtual Reality (VR)

VR is an immersive technology that simulates digital environments that allow users to interact with virtual elements in three-dimensional spaces as if these were the real world. These environments are separate from the real world as users are transported to a different digital space. It is an amalgamation of sensors, cameras, applications, and technology working together to achieve an immersive experience. This technology has applications in gaming, healthcare, education, architecture, training, among other areas (Kriglstein et al. 2023; Park, Lichtman, and Oyekoya 2023). According to Qian, Deng, and S. Chen 2023, gaming is a breeding ground for VR development that is expected to grow in the future. Moreover, as we can see in Figure 2.1, VR is the technology located the furthest away from reality (Applied Visual Technology Inc. 2020).

The key benefit for using VR is its immersion, as it allows users to engage in scenarios that would be impractical or unsafe in real life. Areas like gaming where the user can be in an otherworldly setting, performing a medical surgery in a remote location or even racing or pilot simulations that help users train or learn without the need to be in a physical and real setting. On the other hand, VR also has some downsides and limitations. Its high cost hinders its user base as not all potential buyers can afford the devices or hardware that they need. Prolonged usage also can contribute to physical discomfort in some cases. These systems can also cause motion sickness because of latency or poor optimization (Fu, Y. Hu, and Veronica Sundstedt 2024).

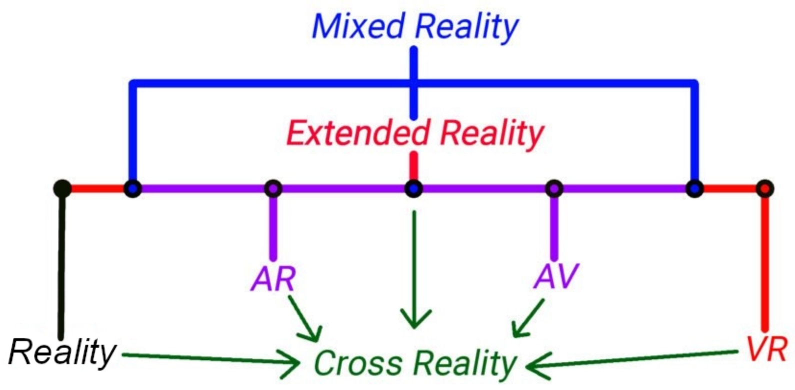


Figure 2.1: VR, AR, MR and XR (Applied Visual Technology Inc. 2020)

2.1.2 Augmented Reality (AR)

Unlike VR, AR overlays digital elements onto the real world, creating an illusion that those objects exist alongside the real environment. This technology enhances human perception and is available through smartphones, tablets, AR glasses and other similar systems (Chitti, Iervolino, and Borghese 2024; Kriglstein et al. 2023). Whereas VR places users in a separate and simulated environment, AR allows its users to remain connected to their surroundings and interact with both physical and artificial objects seamlessly (Manuri and Sanna 2016). Some key components of AR systems include sensors, cameras and displays, which can be either optical or video-based (Lierop, Allard, and Hurk 2021). As we can see in Figure 2.1, AR sits closer to reality than VR as it focuses more on displaying digital objects in the real world (Applied Visual Technology Inc. 2020).

AR offers various benefits across many different domains. This technology improves user perception due to its ability to provide additional information that the user's eye cannot perceive. It is able improve areas such as navigation, where users can see relevant information that otherwise would not be available (Radu and Schneider 2019; Schmalstieg and Höllerer 2016). Other areas include entertainment, education and healthcare. It creates immersive and interactive experiences that improve user engagement, learning and patient care (Minaee, Liang, and Yan 2022). Despite its benefits, AR presents some limitations as well. Reliability and accuracy can be affected by environment mapping or camera localization. There are some constraints in some AR devices such as their limited processing power and battery life (Mekni and Lemieux 2014; Schmalstieg and Höllerer 2016).

This technology is successful in various areas, one of them being gaming. A great example of a popular AR game is *Pokémon GO*, a game that used location services and camera data from the user's smartphone to display virtual game elements in specific locations in the real world (Jasmine et al. 2021; Minaee, Liang, and Yan 2022). Other examples are found in the shopping industry. Users are able to try different clothing or makeup items to check their fit or even see 3D models of house items in the real world to test if it fits their home (Minaee, Liang, and Yan 2022).

Lastly, improvements in areas like machine learning and deep learning technologies will enable more sophisticated application implementations (Minaee, Liang, and Yan 2022). Advancements in cloud technologies may allow for more scalable AR experiences (Y. Chen et al.

2019). There is also a growing interest in social interactions and collaborative work in multi-user AR environments (Sereno et al. 2020).

2.1.3 Mixed Reality (MR)

MR is a technology that blends the real and virtual worlds, combining elements of VR and AR. As we can see in Figure 2.1, MR creates spectrum from AR to Augmented Virtuality (AV) (Speicher, Hall, and Nebeling 2019; Tamura, Yamamoto, and Katayama 2001). It incorporates a wide range of experiences that allow the user to interact simultaneously with physical and digital environments and elements (Ladwig and Geiger 2018; Millard et al. 2024).

It offers similar benefits to other technologies, with applications in gaming (Fu, Y. Hu, and V. Sundstedt 2022), healthcare (Vervoorn et al. 2022) and other areas. Likewise, limitations and challenges regarding its comfort, widespread adoption (Gyawali 2023) and hardware restrictions (Vervoorn et al. 2022) due to a narrow Field of View (FOV) and imperfect hand tracking (Sargolzaei, Rastogi, and Zaman 2024), are also present in this medium.

2.1.4 Extended Reality (XR)

As mentioned before, XR is a hypernym of the previously mentioned technologies, creating a bridge between the real and virtual worlds (Kriglstein et al. 2023). As technologies like VR, AR and MR converge, XR offers immersive and unique experiences to its users by providing seamless transitions between partially augmented and fully virtual experiences. Like its parts, it is being used across various sectors such as gaming, education, healthcare, entertainment, training and more. These technologies are trending in remote collaboration use cases (Chhajer 2021; Dong 2023; Ratcliffe et al. 2021; Vilar, Rodrigues, and Correia 2024; Vivekanandam 2023).

As we can see in Figure 2.1, XR contains all the previous technologies, going from reality to the furthest away from it (VR). On the other hand, there is another key concept: Cross Reality (CR). CR refers to the asymmetric use of XR where multiple of its technologies are used together in an asymmetrical manner (Jung et al. 2023).

In gaming, XR offers the user an unparalleled creative experience, blending immersive storytelling with real-world interactions. This blending of digital and physical elements increases the user's experience. Examples range from entertainment video games to serious games or even gamified apps, providing interactive and educational experiences for all users. Social and interactive games is also a growing area. In them, users interact with other users to promote pro-social collaborative play (Kriglstein et al. 2023; Marín-Vega et al. 2023).

Medical training is a great example of XR implementation. Students can practice procedures and real world scenarios without the need for those scenarios to happen in real life. It gives freedom to test and practice particular use cases without any danger and risk of life. These test simulations can help improve surgical skills and reduce error rates (Ong et al. 2020). XR's learning applications are not only focused around medical education, it can be used to provide learning experiences and simulations in fields such as architecture, construction, engineering and more. It enhances student engagement as complex concepts and scenarios are able to be experienced (Spitzer et al. 2022a).

There are some technological limitations, as its effectiveness can be limited by hardware constraints. Since this a growing market, technological advancements are made at an exponential rate, making existing systems quickly outdated (Cross et al. 2022; Spitzer et al. 2022b). Another challenge of XR is its high entry costs and limited accessibility which affects its widespread adoption, specifically in professional or educational areas (Pons, Medrano, and Domínguez 2022; Spitzer et al. 2022b).

Lastly, the future of XR aims to provide cheaper, smaller and more practical systems that users can use in their day-to-day lives. (Pons, Medrano, and Domínguez 2022; Vivekanandam 2023). Moreover, the addition of more feedback avenues such as smell or haptic feedback aims to improve its users' experience, especially in mental health applications (Pons, Medrano, and Domínguez 2022).

2.1.5 Spatial Computing

Spatial Computing is a technological advancement that enables immersive and intelligent multi-modal interactions in virtual and physical environments (Xu et al. 2024). While it has a similar definition as XR, spatial computing is a broader concept that includes the usage of space in computations. On the other hand, XR is a subset of spatial computing that focuses on bridging technologies like VR, AR and XR to create immersive experiences (Balakrishnan et al. 2021; Kini, Ganeshrao, and Siddalingaswamy 2024; Sampath, Tinnakornsrisuphap, and Hande 2024). It is a growing area with its market valued at USD 97.9 billion in 2023 and project to reach USD 280.5 billion in the four years after that. As such, many tech companies like Apple, Meta, Google and Microsoft have been investing into this area, developing new software and hardware. Apple is one of its pioneers with the release of the Apple Vision Pro, a device brings new possibilities in multi-modal interaction in hybrid spaces where users can utilize spatial audio and state of the art sensors to delve into the world of spatial computing (Xu et al. 2024).

It presents similar gaming, education (Xu et al. 2024) and healthcare (Farhadloo et al. 2024) benefits as its XR subset. Additionally, spatial computing allows for improvements in physical sciences as it optimizes distributed computing for data-intensive simulations and visualizations (Yang et al. 2011). On the other hand, there are some gaps in human-computer interaction research, mainly ethics, accessibility and trust (Xu et al. 2024). Another challenge is its geo-privacy concerns that come from integrating this technology into the daily lives of users (Shekhar, Feiner, and Aref 2015).

2.2 Systematic Literature Review

This section aims to provide a comprehensive literature review that does a mapping study of the already defined research questions, framing the relevant ones using the PICOCS model, deriving inclusion and exclusion criteria and following a PRISMA flow in order to gather information regarding the existing literature when answering these questions.

2.2.1 Systematic Literature Review Method

A Mapping Study (MS) was chosen as this dissertation's literature review method. By systematically investigating the existing literature, this method assesses whether research evidence exists for a specific topic, providing insights into the volume of that evidence (Sampaio 2015). This approach is well-suited to evaluate emerging fields, such as game development with the Apple Vision Pro and visionOS, as it identifies research trends, gaps, and opportunities.

With that in mind, the previously identified research sub-questions were filtered for their relevancy to the literature review as we can see in Table 2.1. Some questions were filtered out due to not being relevant to the literature review. These will be addressed in later phases of this project, such as design, implementation and evaluation. The relevant ones to this study were then mapped into three research questions: *MS-RQ1, MS-RQ2* and *MS-RQ3*.

Table 2.1: Mapping Study

RQ ID	**Research Sub-question**	**MS-RQ ID**	**MS Research Question**
1	What are the capabilities and limitations of the Apple Vision Pro and visionOS?	MS-RQ1	What empirical studies exist discussing the capabilities and limitations of the Apple Vision Pro and visionOS?
2	How does the Apple Vision Pro's capabilities compare to other XR platforms?	MS-RQ2	How does the existing research compare the capabilities of the Apple Vision Pro to other XR platforms?
3	What are the current trends and challenges in XR game development, and how might these apply to the Apple Vision Pro?	MS-RQ3	What empirical studies are there about trends and challenges in XR game development?
4	How can the unique features of the Apple Vision Pro be leveraged in game design?	-	Not relevant to the literature review.
5	What are the technical challenges encountered during game development for visionOS, and how can they be mitigated?	-	Not relevant to the literature review.

Each research question will be framed according to the PICOCS model (Sampaio 2015), where inclusion and exclusion criteria will be derived. Then, a PRISMA flow will be used in the analysis and report phase. The *Comparison* PICOCS component is not relevant for a MS, thus it will not be used for any research question. Only mature sources were considered for this study, from databases such as *ACM Digital Library* and *IEEE Xplore*.

2.2.2 Capabilities and Limitations of the Apple Vision Pro

The first research question (*MS-RQ1*) seeks to identify literature that explores the unique capabilities of the Apple Vision Pro and its operating system. As such, this research question was framed according to the PICOCS model in Table 2.2. Afterwards, the needed inclusion and exclusion criteria were derived for this question (Table 2.3). They help to obtain search strings and define restrictions in order to filter out studies that are not relevant to this question. Records prior to 5 June 2023 were excluded as the Apple Vision Pro and its operating system were announced on that date.

Table 2.2: MS-RQ1: PICOCS Framework

	PICOCS Component	**RQ Part**
P	Population	Studies in XR and Spatial Computing
I	Intervention	Apple Vision Pro and visionOS
C	Comparison	-
O	Outcome	Capabilities and limitations
C	Context	Student projects, final course projects, professional
S	Study design	Empirical studies

Table 2.3: MS-RQ1: Inclusion and Exclusion Criteria

	Inclusion	**Exclusion**	**Sub String**
P	Studies in XR and Spatial Computing	Prior to 5 June 2023	XR; Extended Reality; Spatial Computing; >=(05/06/2023)
I	Literature about the Apple Vision Pro; Literature about visionOS	-	Apple Vision Pro; visionOS
C	-	-	-
O	Literature that mentions capabilities and limitations	-	Capabilities; Limitations; Features; Pros; Cons
C	Student projects; final course projects; professionals	-	Student projects; final course projects; professionals
S	-	Surveys	Empirical study; Case study;
Final Search String		("XR" OR "Extended Reality" OR "Spatial Computing") AND "Apple Vision Pro" AND "visionOS" AND ("Capabilities" OR "Limitations" OR "Features" OR "Pros" OR "Cons") AND Date(05/06/2023 TO *) AND Type("Empirical study" OR "Case study")	

With the inclusion and exclusion criteria defined, queries were performed on *ACM Digital Library* and *IEEE*. Twenty-six records were identified, with zero duplicated being found. Afterwards, those records were screened by analysing their title and abstract. This excluded seven of those records as their were irrelevant to this particular research question. All nineteen records were available for retrieval. Lastly, these records were analysed fully to check if these were relevant based on a further analysis. Five of these records were excluded due to irrelevant content, leaving fourteen reports to be included in this systematic review. This PRISMA process is visualized in Figure 2.2.

With those documents identified, the *MS-RQ1* research question can be answered. The Apple Vision Pro is a **spatial computing platform** released in February 2024 in the United States of America (USA) that has since expanded to other countries and markets. It supports

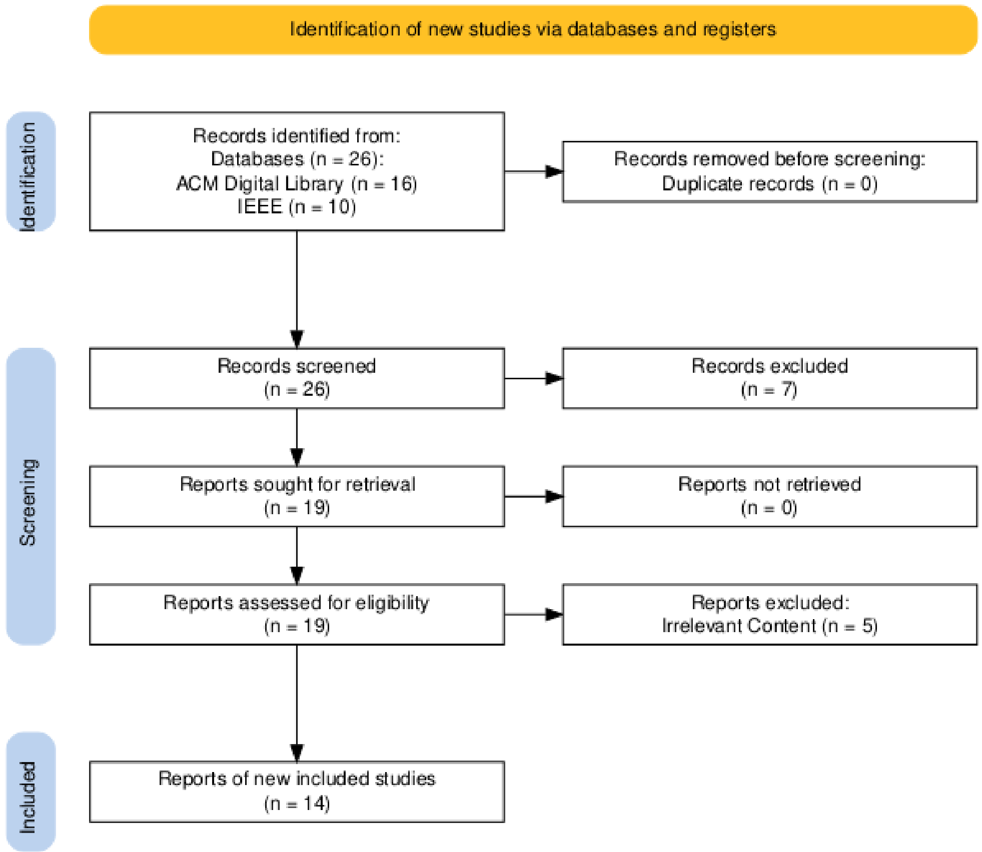


Figure 2.2: MS-RQ1: PRISMA Diagram

both **VR and AR features**, making it an **XR platform** (Hrycak, Lewakis, and Krüger 2024). This device is powered by both an 8-core M2 processor, as well as a dedicated R1 processor that is responsible for processing camera and sensor data. The M2 System on a Chip (SoC) has a configuration similar to a lower-end MacBook Pro (Hrycak, Lewakis, and Krüger 2024; T. Hu et al. 2024).

In terms of capabilities, this new spatial computing platform offers its users a **natural and intuitive interaction** by using multiple sensors that allow **eye-tracking and gesture recognition**, allowing users to make subtle gestures like pitching from comfortable positions, instead of needing to have their hands in front of the device, in an uncomfortable position. Furthermore, using eye-tracking the user is able to interact with the User Interface (UI) by looking at its elements (Estalagem and Esteves 2024; Räthel et al. 2024). The Apple Vision Pro uses its motion and 12 optical sensors to track the user's position and body movements, allowing it to be controller free. Most XR devices have controllers for the users hands but the new Apple system moves away from this with the user's eyes and hands becoming the controller (R. Cheng et al. 2024; Ni 2024; Wang et al. 2024). The aim is to have a more easy-to-learn and practical interaction, while attempting to reduce fatigue by using these downward-facing cameras (Van Der Veer et al. 2024).

The Apple Vision Pro features an **immersive and high-end display**, along with rich **multimodal sensing** (Zhao et al. 2024). Its display uses two Micro-OLED screens, each with **3648 x 3144 pixels** of resolution. This means that each of the two Apple Vision Pro screens is equivalent to a 4K display. With those two 4K equivalent displays, its performance is managed by adjusting the resolution based on what the user is looking at using the previously mentioned eye-tracking.

Furthermore, Apple's new spatial computing device and operating system allows users to integrate these technologies in their day-to-day lives with their **visibility toggle** that transitions between a **VR environment** where users are not aware of their surroundings and a **XR pass-through mode** where the real environment blends in with digital elements. As this pass-through mode is very accurate and nearly indistinguishable from reality without the headset, the Apple Vision Pro can be used in daily activities or while doing some sports. For example, users are able to see recipe videos while cooking, watch videos while walking outside or even record themselves doing extreme sports like snowboarding. This visibility toggle is possible using the digital crown button present of the device (Bondarenko et al. 2024; Chang et al. 2024; Hrycak, Lewakis, and Krüger 2024).

One unique feature that this platform offers is **spatial personas**, 3D human models that track the user's head and hand movements in real time. This persona is used in the Apple FaceTime video conference application and offers an immersive video experience, since users will be displayed as 3D models without the device on the user's head to other participants. For other video conference applications (such as Zoom), a 2D persona will be displayed to the rest of the users. This spatial persona is created during the device's setup process, using their *TrueDepth* camera (R. Cheng 2024; R. Cheng et al. 2024; Wang et al. 2024).

The Apple Vision Pro also displays a computer-generated reconstruction of the user's face to the outside world. This feature is called **Eye-Sight** (Vergari et al. 2024) and will be displayed when the user has vision of their surroundings (i.e. using an MR or XR feature). This is displayed to everyone else if the user can see them. If not, it will display a colour array to identify the user is in a VR environment (Lyons 2024).

Moreover, Apple's spatial computing platform runs on **visionOS**, an operating system derived from the iPhone Operating System (iOS) that focuses on AR and VR capabilities using **Apple's RealityKit and ARKit frameworks**. RealityKit provides AR rendering and simulation that leverages ARKit's features to blend digital and real elements together in a seamless manner.

visionOS has many concepts that are used throughout the system. **Windows** are interactive areas that act as conventional 2D views that apps can present, much like a 2D desktop or mobile app. These can turn into **Volumes** if they represent a 3D object using RealityKit. **Spaces** act as environments where windows and volumes can be presented. A **Shared Space** allows multiple applications to coexist with other applications and the real-world environment. These spaces acts as an AR mode of the Apple Vision Pro where users can reposition windows or volumes freely and interact with them as the user wants. On the hand, **Full Spaces** allow apps to present a more immersive experience, while having complete freedom over the space. This acts as a VR environment where no real-world object is visible. RealityKit is used to present applications in shared spaces, while more access is given to an application presenting in a full space via an Application Programming Interface (API) called Metal (Hrycak, Lewakis, and Krüger 2024).

In terms of programming language, this operating system uses Apple's **Swift** language that provides efficient development with high level APIs. There are wrappers for **C++** provided by Apple in order to ease integration with legacy projects, but conversion to Swift is recommended. Its views are developed using the **SwiftUI** framework (Hrycak, Lewakis, and Krüger 2024).

Other than its hardware capabilities, there are also some other benefits that the Apple Vision Pro brings to the table. The **reputation and market influence** of its brand (Apple) is one of them, as the company has a history of profitable platforms in initially niche markets, such as smartphones and smartwatches. Thus, investing and studying Apple's ecosystem (software and hardware) can be valuable (Hrycak, Lewakis, and Krüger 2024).

On the other hand, in terms of limitations, the platform's **high cost** is one of its biggest downsides as it hinders the number of users. Likewise, its **limited availability**, estimated between **100,000 and 200,000 units** available in the USA launch, also affects the number of users, market adoption and company investments as there are a limited number of devices being sold (Hrycak, Lewakis, and Krüger 2024).

There are also some limitations regarding its **limited FOV**. This can affect the device's immersion as the world is less encompassing (Chang et al. 2024). Another smaller limitation found was regarding the digital personas, as its **utility was found to be limited and their fidelity lacking** (Lyons 2024). Although an improvement and a step in the right direction, the *Eye Sight* feature still is not comparable to interactions between users that are not wearing XR devices (Vergari et al. 2024). Additionally, **privacy concerns** were also raised by Wang et al. 2024 regarding eye tracking and persona data.

Furthermore, the Apple Vision Pro's **size and weight** also are identified as issues. This can impact the user's conform and time spent with the device. Another limitation with the platform is the cause of fatigue due to **Vergence-Accommodation Confict (VAC)** (Chang et al. 2024).

Lastly, according to Ni 2024, the embedded cameras presented in the Apple Vision Pro are vulnerable to **side-channel attacks**. These attacks might allow for malicious injections or potential privacy leaks as the electromagnetic signs emitted by this device might be obtained by attackers, allowing to rebuild high-definition image streams.

2.2.3 Comparison to other XR Platforms

MS-RQ2 aims to make a comparative analyses of the Apple Vision Pro against its competitors in the XR market. Similar to the previous research question, *MS-RQ2* was framed according to the PICOCS model in Table 2.4. Following this framing, Table 2.5 defines this question's inclusion and exclusion criteria so that the correct filters and search strings can be applied to the research.

Table 2.4: MS-RQ2: PICOCS Framework

	PICOCS Component	**RQ Part**
P	Population	Studies in XR and Spatial Computing
I	Intervention	Apple Vision Pro, Meta Quest 3, HTC Vive, Microsoft HoloLens
C	Comparison	-
O	Outcome	Comparison between XR platforms
C	Context	Student projects; final course projects; professionals
S	Study design	Empirical studies

Afterwards, as we can see in Figure 2.3, a PRISMA flow was followed. Queries were performed on the *ACM* and *IEEE Xplore* digital libraries, where a total of eighteen records were found (fourteen on *ACM Digital Library* and 4 on *IEEE Xplore*). No duplicates were found

Table 2.5: MS-RQ2: Inclusion and Exclusion Criteria

	Inclusion	**Exclusion**	**Sub String**
P	Studies in XR and Spatial Computing	Prior to 5 June 2023	XR; Extended Reality; Spatial Computing; >=(05/06/2023)
I	Studies comparing Apple Vision Pro with other XR platforms; Studies about the Apple Vision Pro; Studies about the Meta Quest; Studies about the HTC Vive; Studies about the Microsoft HoloLens	-	Apple Vision Pro; Meta Quest; HTC Vive; Microsoft HoloLens;
C	-	-	-
O	-	Studies not comparing or mentioning these platforms	Comparison; Comparing
C	Student projects; final course projects; professionals	-	Student projects; final course projects; professionals
S	-	Surveys	Empirical study; Case study;
Final Search String		("XR" OR "Extended Reality" OR "Spatial Computing") AND "Apple Vision Pro" AND ("Meta Quest 3" OR "HTC Vive" OR "Microsoft HoloLens") AND ("Comparison" OR "Comparing") AND Date(01/01/2019 TO *) AND Type("Empirical study" OR "Case study")	

between both libraries. After screening the titles and abstract of the found records, four were dismissed as they weren't relevant to the question in mind. Afterwards, the fourteen remaining records were retrieved successfully and assessed for eligibility. Six records were found to contain irrelevant content to *MS-RQ2*. Thus, a total of eight reports were used as sources to answer this research question.

As we can see in Table 2.6, a comparison was made between the Apple Vision Pro and some of its competitors. Firstly, in terms of **FOV**, there is no official number for the Apple Vision Pro but based on the research and studies it has a limited FOV. Other platforms like the Meta Quest 3 have less limiting FOVs at 110°, with the HTC Vive XR Elite being the one closer to the Apple Vision Pro. In terms of **resolution for each eye**, the Apple Vision Pro easily beats the competition as it provides a resolution similar to a 4K screen for each eye. In contrast, systems like the Meta Quest Pro and HTC Vive XR Elite have almost half the resolution for each eye. The Meta Quest 3 is slightly better but still falls behind the Apple Vision Pro. In terms of **stereo camera resolution**, the HTC Vive XR Elite has clearly better resolution with more than double when compared to its competitors. Apple's system comes second with 6.5 MP. Most these system offer **depth sensors**, apart from the Meta Quest

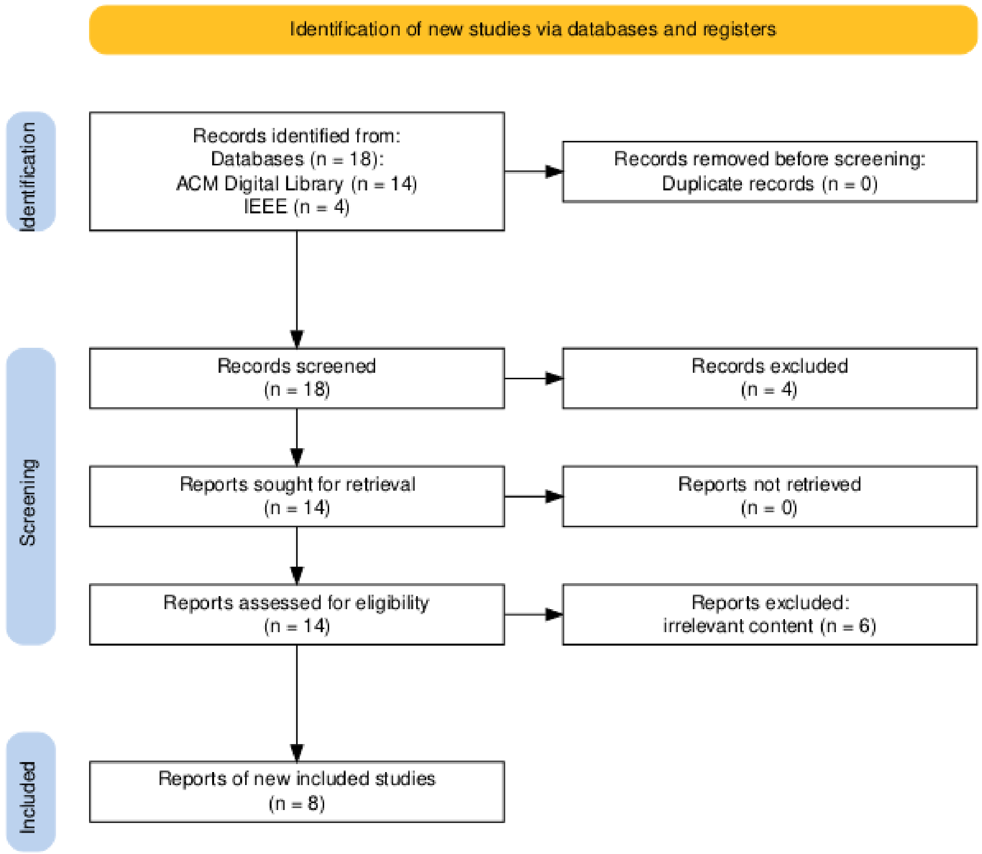


Figure 2.3: MS-RQ2: PRISMA Diagram

Pro. As for the type of lens, every system uses **pancake lenses** as these are closer to the display, thus making the overall system smaller while improving image quality (T. Hu et al. 2024).

Table 2.6: Screen Comparison (T. Hu et al. 2024)

XR System	**FOV**	**Resolution (per eye)**	**Stereo Camera Resolution (each)**	**Depth Sensor**	**Lenses**
Apple Vision Pro	Limited	3648 x 3144	6.5 MP	Yes	Pancake
Meta Quest 3	110°	2064 x 2208	4 MP	Yes	Pancake
Meta Quest Pro	106°	1800 x 1920	4 MP	No	Pancake
HTC Vive XR Elite	102°	1920 x 1920	16 MP	Yes	Pancake

Some XR platforms display virtual elements on the real world by showing a video feed from their cameras, with these elements integrated in it. These platforms can be considered Video See-Through (VST) Head-Mounted Displays (HMDs). Systems like the Apple Vision Pro, Meta Quest 3, Meta Quest Pro and HTC Vive XR Elite are part of this group. On the other hand, there are systems that use transparent lenses to overlay these virtual elements onto the real environment, allowing users to directly see their surroundings. These systems are considered to be Optical See-Through (OST) HMDs. The Microsoft HoloLens 2 is an example of a OST HMD (Bondarenko et al. 2024; R. Cheng et al. 2024).

The user interaction is another difference between XR platforms. A typical system uses controllers for the user to interact with the system. This is the case for HMDs like the Meta Quest 3 or HTC Vive. Unlike these systems, the Apple Vision Pro uses its multiple sensors and cameras to allow users to interact with the system via eye-tracking and hand gestures (Ni 2024).

Moreover, there are different design guidelines for different systems. Platforms more geared towards AR and XR (like the Apple Vision Pro or Microsoft HoloLens 2) recommend their UI to be anchored to the world or the user's body, rather than a Heads-Up Display (HUD) where the UI is anchored to the user's head (Chang et al. 2024).

In a study comparing **tracking accuracy** (L. Cheng, Schreiner, and Kunz 2024), it was found that VST HMDs (including the Apple Vision Pro and Meta Quest 3) performed much better in **outdoor low-light conditions** than OST HMDs (like the Microsoft HoloLens 2). Despite that, there was **visual deterioration, flickering and color issues** for all tested VST HMDs, especially significant in the Meta Quest 3. Contrarily, the Microsoft HoloLens 2 functioned as normal sunglasses in those cases.

Another study about performance tracking found that the Apple Vision Pro is much **less error prone** and more accurate than the Meta Quest 3. These systems had **sub-centimetre accuracy** in most scenarios with errors higher than 10 cm in challenging scenarios. This means that both platforms still have room to improve their accuracy and be more reliable (T. Hu et al. 2024).

Both the Apple Vision Pro and Meta Quest 3 have **privacy concerns and limitations** regarding the camera feed. Apple's device doesn't provide raw eye-tracking information to the developers and gives limited image tracking about the real world, updating it once every second. Meta does something similar as it provides pre-processed data without any specific data for real objects. This limits the development as knowledge about the real-world environment is restricted. On the other hand, this is done by design to protect the user's private data as these devices has cameras and sensors that gather personal data (Feld et al. 2024).

Furthermore, the Apple Vision Pro is slightly **more accepted in public settings** than the Meta Quest 3, likely due to its *Eye Sight* feature. As it displays a virtual representation of the users eyes and simulates eye contact, it can reduce the barrier between its users and non-users, improving the social acceptance of XR systems in social environments. Although a slight improvement over other systems that don't have this feature, like the Meta Quest 3, it still isn't the same as interactions without these XR systems (Vergari et al. 2024).

In summary, the Apple Vision Pro presents a **better display** than its competitors with two 4K resolutions for each eye. On the other hand, it has a **limited FOV** and **decent stereo camera resolution**. As a VST HMD, it has **better outdoor low-light tracking accuracy** than its OST counterparts, but has **worst visual deterioration, flickering and colour issues** compared to OSTs like the Microsoft HoloLens 2. It is **less error prone** than other systems like the Meta Quest 3 and is **slightly more accepted in public settings** because of its *Eye Sight* feature.

2.2.4 Current trends and challenges in XR game development

This last research question tackles the current trends and challenges that the XR industry are facing, specifically in the game development field. As such, *MS-RQ3* was framed according to the PICOCS model in Table 2.7. After it, this question's inclusion and exclusion criteria were derived and sub strings defined so that the correct filters and search strings can be applied to the research (Table 2.8).

Table 2.7: MS-RQ3: PICOCS Framework

	PICOCS Component	**RQ Part**
P	Population	Studies in XR
I	Intervention	Game development
C	Comparison	-
O	Outcome	Trends and Challenges
C	Context	Student projects; final course projects; professionals
S	Study design	Empirical studies

Table 2.8: MS-RQ3: Inclusion and Exclusion Criteria

	Inclusion	**Exclusion**	**Sub String**
P	Studies in XR	Prior to 2019	XR; Extended Reality; >=2019
I	Studies about game development	-	Game development; Gaming; Video Games
C	-	-	-
O	Literature that mentions trends and challenges	-	Trend*; Challenges
C	Student projects; final course projects; professionals	-	Student projects; final course projects; professionals
S	-	Surveys	Empirical study; Case study;
Final Search String		("XR" OR "Extended Reality") AND ("Game development" OR "Gaming" OR "Video Games") AND ("Trend*" OR "Challenges") AND Date(01/01/2019 TO *) AND Type("Empirical study" OR "Case study")	

With these defined, the PRISMA diagram was created and flow followed (Figure 2.4). The *ACM Digital Library* was used to perform the query and fifteen records were found. As no duplicates were found, the screening phase began. After all records' titles and abstracts were examined, seven of those were excluded due to their examined fields being irrelevant to the question. As all reports were retrieved successfully, the remaining eight were analysed for their eligibility. Consequently, two of those records were deemed to be irrelevant in order to answer this research question. Lastly, the remaining six records were used to answer *MS-RQ3*.

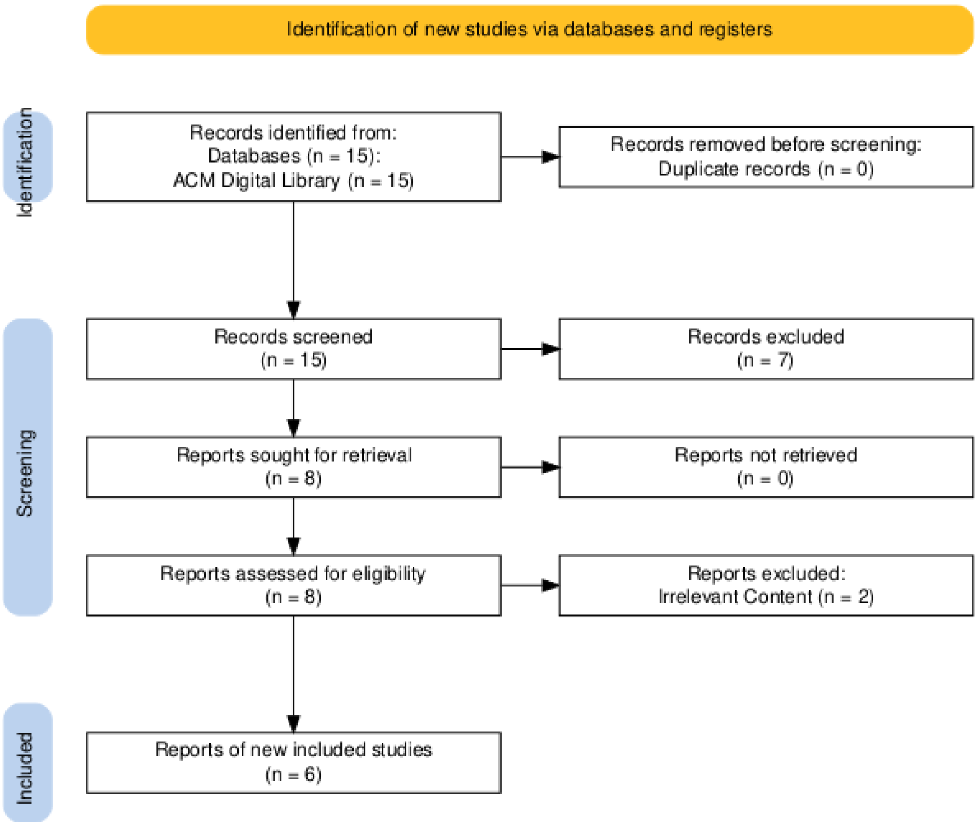


Figure 2.4: MS-RQ3: PRISMA Diagram

With the research concluded, we can identify XR and its encompassing technologies (VR, AR and MR) as an ever growing and developing area. According to Fu, Y. Hu, and Veronica Sundstedt 2024, a recent trend in XR game development is focused around **gaming in the health area**, as people are in constant pursuit of a healthy lifestyle. Development of **exercise XR games** has risen due to this fact. These games can be focused on the user's health as its the main objective or promote a healthy lifestyle even if it is not the game's main focus. These games can also be used in health studies that aim to tackle many diseases, such as childhood obesity.

Another trending area in XR is **multiplayer games focused on collaborative and social aspects**. Games where users interact with each other present a great way promote a collaborative and dynamic experience. These present their own technical challenges as the multiplayer component and the interactions between users must be synchronize in order to not break immersion (Kriglstein et al. 2023).

Other areas like **CR and asymmetrical social games** are also important to consider from a game development perspective. These include social games where users are interacting with each other in different settings. They may be using different systems (cross-play), their location may be the same or a separate one, XR users might be interacting with users with no XR system in the same location. It aims to provide an inclusive and shared experience for all users (Jung et al. 2023; Kitson et al. 2024).

Game development is a complex endeavour that requires great programming skills, principles and structure. In terms of design patterns, the *Singleton Pattern* has widespread usage in the gaming industry. This pattern ensures that a specific class can only have one instance while providing an access point to it. It has many advantages but also it has some challenges similar to the use of global variables. The *Observer Pattern* is also used in some cases as actions are broadcasted to observer objects. Although it can impact performance and cause crashes if implemented incorrectly, it provides a great way to decouple code and improve maintainability. Another pattern used in the game industry is the *Factory Pattern* that allows for object creation without the need to expose that logic other objects. This can be achieved using subclasses or interfaces (*Abstract Factory*). It centralizes object creation, improves code reusability and parallel development. On the other hand, it increases code complexity (Qian, Deng, and S. Chen 2023).

As mentioned before, developing XR projects is challenging and demanding. According to Kriglstein et al. 2023, developers need to be aware of issues such as **tracking precision, latency and realism**, as they may impact the user's experience. Similar challenges were also noted by Sargolzaei, Rastogi, and Zaman 2024, as **subpar hand tracking and narrow FOV** must be carefully considered by developers when implementing XR games. Another challenge that plagues XR game development and the XR industry as a whole is its adoption issue. As these systems are **expensive and require robust hardware**, particularly to run video games, its accessibility and widespread adoption is limited (Kriglstein et al. 2023).

2.3 Tecnologies

This section aims to present some of the existent technologies. These range from XR frameworks (such as ARKit and RealityKit), Swift as the the platform's programming language, SwiftUI to create the UI, XCode to help develop its code and more.

2.3.1 ARKit

ARKit is a powerful framework developed by Apple that allows developers to create AR experiences on devices running iOS, iPadOS or visionOS operating systems. It has many features, such as 3D object recognition, distance measurement, plane detection, among others. These give developers all the needed tools to implement practical AR applications. With these features, immersive and engaging scenarios can be created by blending computer-generated objects onto the physical environment (Khan et al. 2020; Permozer and Orehovački 2019). As we can see in Figure 2.5, a 3D model is being displayed in the real world using ARKit.

2.3.2 RealityKit

RealityKit functions as the rendering engine that leverages ARKit by providing high-performance rendering and animation capabilities, enhancing the visual quality and realism of AR experiences. It allows developers to create scenes with 3D models. Developers can also use physical simulation or manually animate models (Mardhiyyah et al. 2023). Figure 2.6 displays an example of using RealityKit.

2.3.3 Swift

Swift is an object-oriented programming language developed by Apple that implements elements of functional programming. This is focused on software developed for Apple products, such as the iPhone, Apple Vision Pro, and more. It has an easier learning curve than

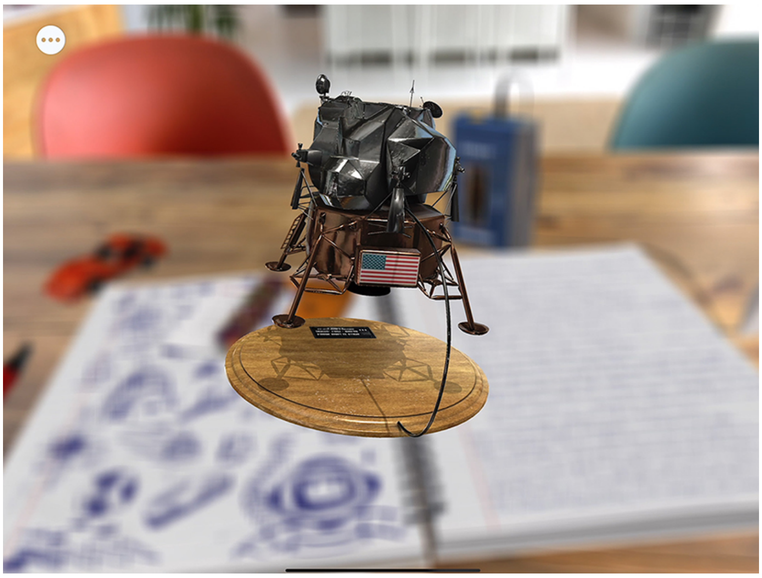


Figure 2.5: ARKit (Apple Inc. 2022a)

Objective-C as it incorporates features from other languages such as Java, C++ and Python (Goodwill and Matlock 2015). This programming language uses type safety and automatic reference counting to prevent errors in its code (P. Deitel and H. Deitel 2015). Furthermore, it has concise and simple syntax that makes it accessible to new developers (Rebouças et al. 2016) and allows to be integrated seamlessly with Objective-C, giving Swift an remarkable interoperability with the older programming language used for Apple systems (P. Deitel and H. Deitel 2015). In terms of performance, Swift is efficient and generally faster than its predecessor (Singh and Kaur 2017; Wells 2015).

2.3.4 SwiftUI

SwiftUI is Apple's newest declarative UI framework that was released in 2019. This framework simplifies the UI building process for applications running on iOS, macOS and visionOS devices. With less code and complexity, developers are able to create responsive, dynamic and modern interfaces requiring less code and complexity. Its an ever-growing framework that is gaining more usage each year, compared to the older and more imperative UIKit (Mardhiyyah et al. 2023).

2.3.5 FaceTime, SharePlay and GroupActivities

FaceTime is Apple's video communication application that is available across various of their platforms, including visionOS. The previously mentioned spatial persona feature is available via FaceTime for visionOS users. It allows users to have high-definition, end-to-end encrypted video or audio calls remotely with other participants via the internet (Apple Inc. 2024e). This application has a feature called SharePlay that allows participants to share activities and synchronizes them for all users in the call. For example, users can watch a video or listen to music together (Apple Inc. 2021b). Applications can add SharePlay support to their apps via the GroupActivities framework. This Apple framework allows apps to integrate with FaceTime, allowing them to be shared and synchronized for participants in a particular call (Apple Inc. 2021a).

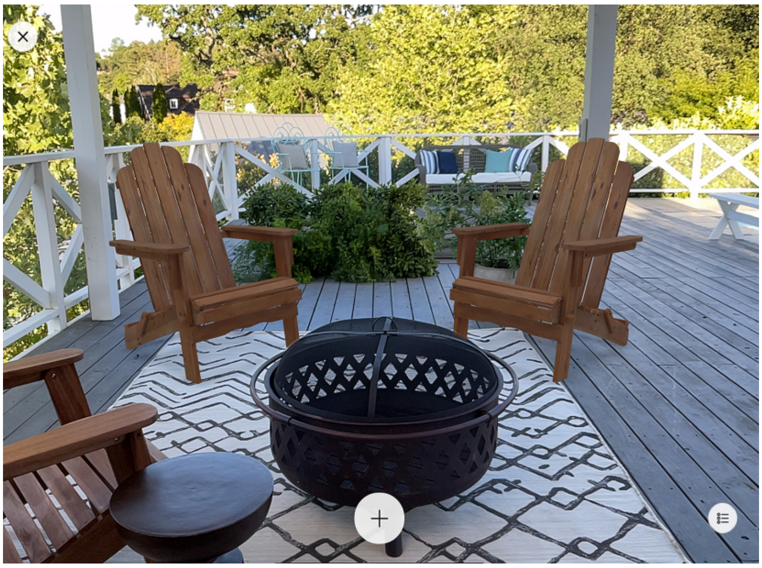


Figure 2.6: RealityKit (Apple Inc. 2024b)

This first party integration is a great opportunity to explore a multiplayer experience in a visionOS application. With that in mind, the SharePlay integration via the GroupActivities framework was chosen to be used when implementing the prototype.

2.3.6 TabletopKit

TabletopKit is a first party foundational framework from Apple that was introduced with visionOS 2.0. It is designed to support the development of tabletop spatial computing games for the operating system. This framework also provides developers with support for multiplayer via FaceTime and SharePlay, as well as equipment state management and interactions (Apple Inc. 2024d). It leverages RealityKit and SwiftUI to allow developers to implement extended reality experiences anchored to flat surfaces, such as a game board (Apple Inc. 2024a). This framework was chosen to be used as it is game development focused, is a first party framework provided by Apple itself and has great support for multiplayer experiences.

2.3.7 Xcode

Apple's Xcode is the main Integrated Development Environment (IDE) when developing applications for Apple products, such as the iPhone, iPad, Apple Watch, Mac or, in this case, the Apple Vision Pro. Since Xcode, the mentioned frameworks, operating systems and hardware were all developed by Apple, it allows for a seamless integration and development process as all components have great synergy. This allows to develop and test applications in an efficient manner (Khan et al. 2020; Mardhiyyah et al. 2023).

2.3.8 Reality Composer Pro

Reality Composer Pro is a tool created by Apple with the aim to let users edit 3D assets and scenes for visionOS applications. This tool is available within Xcode and integrates with RealityKit and ARKit. This is an important tool for visionOS application as it allows developers to create scenes, as well as import and edit models to be use in their apps (Apple

Inc. 2019). This was the main 3D editing tool used to create scenes and edit assets for the prototype, due to its integration with RealityKit, ARKit and Xcode.

2.3.9 Blender

Blender is a free and open-source software. Although with a similar purpose to Reality Composer Pro, Blender has broader applications. For example, while Reality Composer Pro is not meant for creating complex 3D models from scratch, Blender supports the full process of 3D production. That includes modelling, rigging, animation, simulation, rendering, compositing, motion tracking and video editing ("Blender Scripting for Creative Coding Projects" 2023). This is a powerful tool that was used as a enhancer whenever there were some limitations with Reality Composer Pro.

2.3.10 Other technologies that were explored

Unity's XR Interaction Toolkit was one of the technologies considered to be used when developing the prototype. This framework offers an interaction system that aims to help developers create XR experiences. It is available for multiple platforms, including visionOS (Unity 2019). In the end, XR Interaction Toolkit was not used as the previously mentioned native frameworks from Apple offer a deeper integration with the operating system, development tools and frameworks for this particular platform. Furthermore, native frameworks are better optimized for visionOS and utilize the platform's latest features, while providing better stability and long-term support.

2.4 Summary

In sum, XR creates a bridge between the real and virtual world by combining technologies like VR, AR and MR. Moreover, the Apple Vision Pro is a spatial computing platform that offers many exciting features such as eye-tracking and gesture recognition without the need for a controller. While it has its advantages, there are some limitations as well. For example, its high cost and limited availability hinder its user base.

While comparing this platform to other XR systems, there are some instances where the Apple Vision Pro excels and others where it lacks behind others. Its high resolution displays obliterates the competition, offering screens comparable to 4K resolution for each eye. On the other hand, it fails in pricing, limited FOV and it has worst visual deterioration, flickering and colour issues than some OST competitors. Moreover, social multiplayer games and health-related games were identified as some trends in XR game development while issues like motion sickness and latency were identified as challenges.

Lastly, ARKit and RealityKit work together provide tools to developers that allow them to create immersive AR experiences. With Swift as its programming language and SwiftUI as the declarative UI framework, XCode uses all these tools as an IDE developed by Apple. The company's ecosystem provides opportunities for developers. FaceTime's SharePlay allows developers to integrate their applications with the videotelephony application via its GroupActivities framework. Apple offers many other frameworks for visionOS, such as TabletopKit that aids developers when creating spatial tabletop games. Lastly, both Reality Composer Pro and Blender offer tools that allow the editing of 3D assets.

Chapter 3

Prototype Analysis and Design

This chapter will present the analysis and design of a tabletop card prototype game with an original concept that aims to leverage the visionOS unique capabilities identified in the previous chapter.

Firstly, the *Analysis* section will present the concept, rules, flowchart, target audience, alternative concepts, domain, functional and non-functional requirements of the prototype game using the Functionality, Usability, Reliability, Performance, and Supportability + Design Constraints, Implementation Constraints, Interface Constraints, Physical Constraints, among others (FURPS+) model. Moreover, the *Design* section will present the architecture of the visionOS solution through its various levels, as well as the user interface, gestures and branding.

3.1 Analysis

This section starts by detailing the concept for the game that will be developed into a visionOS application.

3.1.1 Concept

In order to create a prototype game for the Apple Vision Pro, a concept was developed for a multiplayer XR tabletop game that allows players to interact elements like cards, dice, immersive 3D environments and more. This concept is a brand new game with its own rules that took some inspiration from existing popular card games, such as *War* (Lakshtanov and Roshchina 2012). This concept also allows to explore some of the unique features and trends that were mentioned in Chapter 2, such as eye-tracking, gesture recognition, switching between AR and VR modes, while providing a social multiplayer experience.

Rules

- Up to 3 players (with multiplayer using SharePlay);
- Other players can observe the game while others play;
- This multiplayer experience must be integrated with Facetime and SharePlay;
- Card game with some other elements, like a die;
- When the game begins, cards are shuffled and each person receives 7 cards that are placed faced down in their hand;

- Players cannot see their cards until they play said card;
- Every round, each player choses one card from their hand and plays them;
- Each played card will be compared to the others and a round winner will be chosen;
- The card with the lowest value wins;
- The winner of the round will collect the other players' played cards;
- If two or more cards have the same value, a tiebreaker die will be rolled;
- The player with the lowest die roll will win the round;
- The board has a stack of cards where players withdraw a card after every round ends;
- Once this stack is empty, no more cards are withdrawn;
- Players are eliminated when they no longer have any cards on their hand and there are no cards on the stack;
- A player wins when they have all the cards in their hand and all other players were eliminated.

Cards

A numbered card deck will be used to play this tabletop game. This deck will include numbered cards from 2 to 10 with a similar structure to a regular card deck without all court cards and jokers. Each card's value is equivalent to their number. In addition to those numbered cards, there will be two special court cards: a queen and an ace. The queen has a value of one and the ace is the best card in the game, with zero as its value.

Flowchart

A flowchart was created to display all scenarios and flows for this prototype's concept, as we can see in Figure 3.1. In it, the flow from starting a game, invite their friends, each round's flows, how each player plays a round or wins the game, is defined and shown.

Target Audience

This game can be enjoyed by anyone who owns an Apple Vision Pro device but its target audience is users that enjoy playing card or board games with their friends, are XR enjoyers and have friends that live far away enough where physical meetings are a pain point.

3.1.2 Alternative Concepts

Although the chosen concept for the prototype was the previously mentioned card game, other original game concepts were considered and explored for the Apple Vision Pro game prototype.

Similar Card Game

Another similar card game was considered, where players would stack cards based on their value, either by choosing to play a specific card from their hand or choosing a random card without knowing its value at a later stage of the game. It also featured some special cards

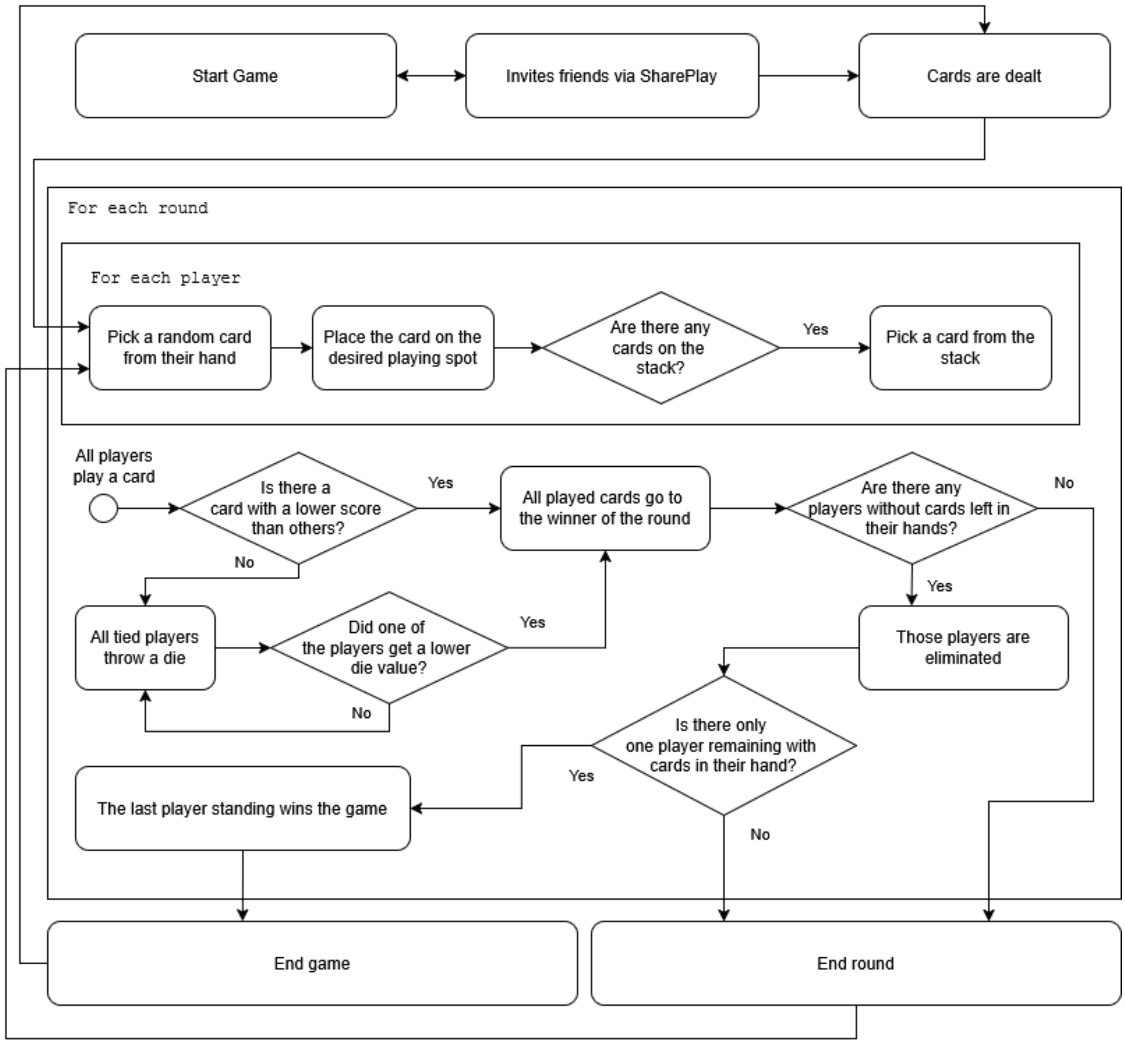


Figure 3.1: Flowchart

with specific benefits that would change up the game. The goal was to be the first player that plays all their cards. These were the considered rules:

- When the game begins, cards are shuffled and each person receives 5 cards faced down and a 5-card hand;
- The table will have a stack where players will place their cards when its their turn to play (referred to as the playing stack);
- Cards can be played if they have a higher value than the top card of the playing stack;
- Special cards can always be played;
- There is also a second stack present on the table where players will pickup cards whenever they do not have a valid card to play in their hand (referred to as the reserve stack);
- Once this stack ends, any player without a valid move will pick up the whole playing stack and add it to their hand;

- After the reserve stack ends, any player who plays all cards from their hand will need to play a card from one of the 5 cards facing down. The player cannot see the card before playing it and the player will pick up the whole playing stack if the card is not a valid move. The player needs to play the whole deck before playing the next card facing down;
- A player wins when all their cards are played.

In terms of cards, a similar numbered card deck would be used to play this game. This time the deck would be from 1 to 10. In addition to those numbered cards, there would also be special cards that allow the player to have specific benefits and play them at anytime.

- Mirror: a card that mirrors the value of the previous card (e.g. a mirror card played after a 7 would have 7 as its value);
- Reset: a card that resets the current playing value to 1;
- Eraser: a card that discards the current playing stack and allows the player to play another card;
- Dice: a card that forces the player to roll two dices to get its value (each dice has a value from 0 to 5);
- Avalanche: a card that forces the player to pick up the playing stack.

Gesture Duel Game

Lastly, a different genre from a tabletop game was also considered, one that would put players in a fight against another player in a multiplayer environment. These two players would battle in rounds against each other by attacking, defending or doing other actions with their gestures. This game would utilize custom gestures of the Apple Vision Pro to identify if the user was attacking (finger gun gesture), defending (clenched fist) or other custom actions.

3.1.3 Domain

In this project, the previously mentioned prototype card game concept will be implemented as an Apple Vision Pro XR game. In order to simplify its comprehension, a domain model diagram was created where the game's concepts and their relationships are displayed (Figure 3.2). Furthermore, these concepts and relationships were further explored and defined via a domain glossary in Table 3.1.

Table 3.1: Domain Glossary

Concept	Definition
Activity	A SharePlay session.
Board	The virtual playing surface where the game unfolds and all game elements are placed (e.g. cards, die, hands, etc.).
Card	A virtual card that users can interact with by picking it up and placing it in specific places. It has a kind, suit and value attached to it.
Die	A virtual six-sided die that players can roll as a tiebreaker in a round.
Elimination (Player)	A player is eliminated when their hand is empty and there are no more cards in the stack.
Game	Game that is being played.
Hand	A player's private set of cards that are used to play in a round. Cards in a hand are placed face-down on the board so that players play their cards without knowing their value.
Immersive Space	A completely virtual environment where the game has full control over that space.
Kind	The type of card. It can be a regular numbered card from 2 to 10, a queen or an ace.
Multiplayer	Game being played by multiple players with various devices via an internet connection.
Played Card	Card chosen by a player to be played in that specific round.
Player	User that is playing the game.
Robot	Companion placed on the board that reacts to the gameplay, either using animations or sound effects.
Round	In each round, players play a card from their hand against each other. The card with the lowest value wins that specific round. Rounds can be won or drawn.
Rules	The game's rule set.
Seat	Place where a player sits and interacts with their game equipment.
Spectator	User that is observing other users playing the game.
Stack	Game element that houses a group of cards on top of each other in a stacked manner. After each round, players withdraw a card from the stack until there are no cards remaining.
Suit	It represents the card's suit: spades, clubs, hearts or diamonds.

Concept	Definition
User	Person that interacts with the game.
Value (Card)	Value of each card. This is used to determine which card wins a given round. Regular numbered cards have their number as their value (ranging from 2 until 10). A queen has a value of 1. An ace has 0 as their value.
Winner	The last player standing with cards on their hand.

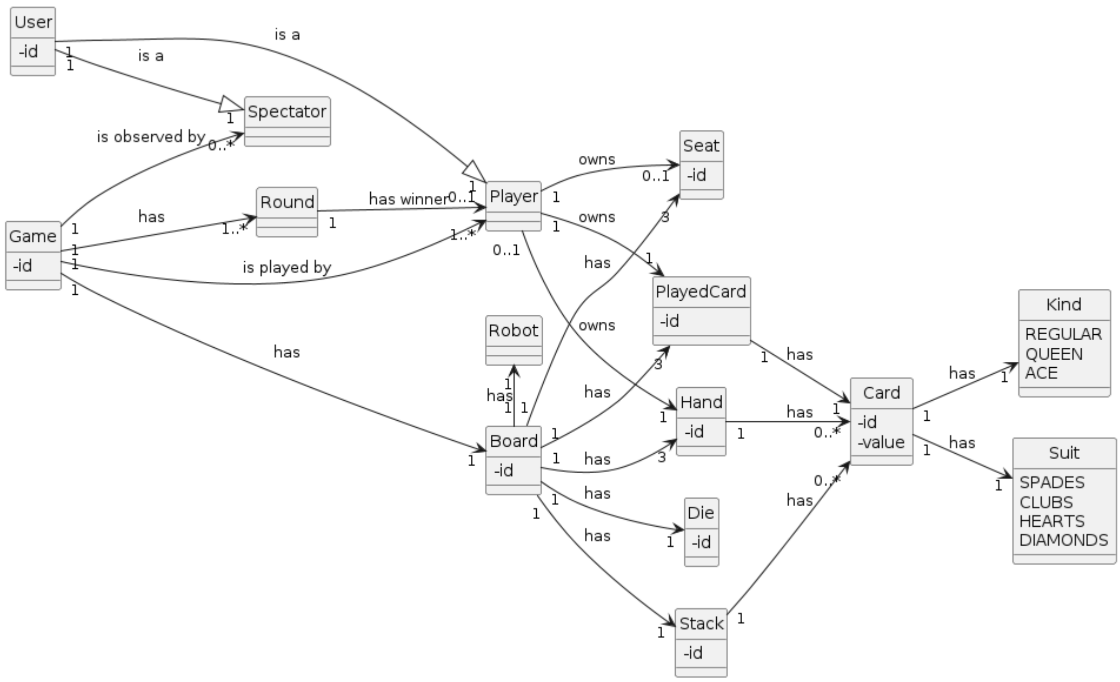


Figure 3.2: Domain Model

3.1.4 Functional Requirements

This project's objective is to evaluate the Apple Vision Pro and visionOS for game development via the implementation of an XR card game. For this to be accomplished, a visionOS application will be created that allows its users to start a game, create a multiplayer session, invite friends, interact with their cards, play rounds and ultimately win the game. Additional features to explore this platform, such as virtual environments, custom rules and windows that can be placed around with the environment (e.g. displaying the game's rules) will be allowed by said application. The previously identified and defined requisites were translated into the following user cases:

1. UC1: Start a game;
2. UC2: See the board;
3. UC3: See the toolbar;
4. UC4: Take a seat in the game;
5. UC5: See my player hand;

6. UC6: See cards in my player hand;
7. UC7: See cards in the board's stack;
8. UC8: Have the cards dealt to the hands and stack;
9. UC9: See where I can place my played card;
10. UC10: Draw a card from the stack;
11. UC11: Play a card from my hand;
12. UC12: Roll a die;
13. UC13: Win a round;
14. UC14: Resolve a draw;
15. UC15: Get eliminated from the game;
16. UC16: Win the game;
17. UC17: Add a Robot that will react to a round win or draw;
18. UC18: Hear sound effects when interacting with equipment;
19. UC19: Reset my game;
20. UC20: Play in a single player mode;
21. UC21: Have a faster game mode;
22. UC22: Show rule set;
23. UC23: Toggle rules off;
24. UC24: Toggle my environment from mixed to virtual reality;
25. UC25: Create a SharePlay multiplayer game session;
26. UC26: Configure the SharePlay session;
27. UC27: Spectate a multiplayer game.
28. UC28: Invite other people to a multiplayer game;
29. UC29: End the SharePlay session;

Furthermore, in order to correlate the identified use cases to the actors that will perform them, a use case diagram was created a displayed in Figure 3.3.

Technical Tasks

Additionally, some technical tasks were also identified that needed to be implemented before starting the implementation of each use case. These following tasks were identified:

1. T1: Create the AceR visionOS application;
2. T2: Set up App Icon and Splash Screen;
3. T3: Implement Model layer.

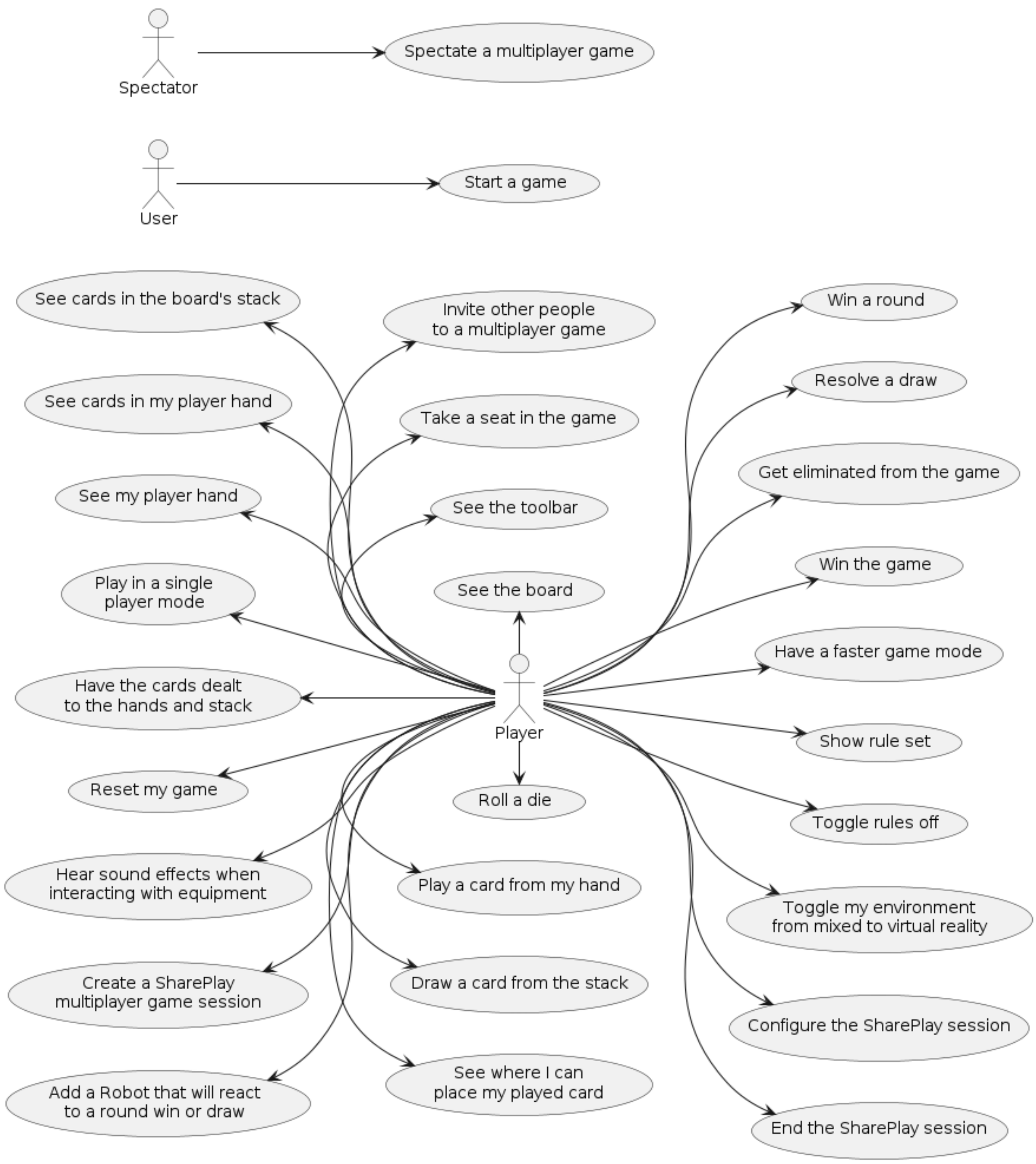


Figure 3.3: Use Case Diagram

3.1.5 Non-Functional Requirements

Next, the non-functional requirements were identified and presented using the FURPS+ model, a model used to classify software quality attributes. It differs from its predecessor defined by Grady 1992 (FURPS model) as it adds some extra categories like *Design Constraints*, *Implementation Constraints*, *Interface Constraints* and *Physical Constraints* to the original categories from the FURPS model: *Functionality, Usability, Reliability, Performance* and *Supportability* (Eeles 2005).

Functionality

- The game's multiplayer sessions should be secure by using Apple's SharePlay and GroupActivities APIs, ensuring that session data is protected and private.

Usability

- The game should have an intuitive and immersive spatial UI that follows Apple's design guidelines for visionOS applications and interactions;
- The interface must also follow Apple's accessibility guidelines, including support for visionOS accessibility features such as gesture recognition;
- New users should be able to easily understand and interact with the game rules and its interface.

Reliability

- The game's state (cards, hands, stack, etc.) must be correctly synchronized for all multiplayer users in real-time with minimal delay;
- If a player unexpectedly leaves a multiplayer session, the game should be able to continue or allow the player to rejoin without losing the game state.

Performance

- The game must utilize the least amount of resources possible (e.g. battery) so the user experience is the best it can be.
- The prototype's architecture should be scalable and support potential future expansion for more players.

Supportability

- The system, as well as its various features, should be properly documented so that it eases its the comprehension of its architecture and implementation in order to help the onboarding of any person that sees the project for the first time.

Design Constraints

- Swift will be the programming language used to develop the prototype;
- Apple's SwiftUI will be the framework used for the prototype's UI;
- Apple's GroupActivities framework will be used for the multiplayer in the prototype, using the SharePlay feature of Apple's Facetime application;

Implementation Constraints

- The game should run on any visionOS operating system with version 2.0 or above;
- The Swift code should follow Mindera's Swift Style Guide (Mindera 2020);
- RealityKit should be used for rendering immersive environments and 3D objects.

Interface Constraints

- The game should provide visual feedback via animations when cards are moved, dice are rolled, or players win/lose rounds;
- The game should provide audio feedback via animations when cards are picked up, dice are rolled, a round is won or drawn, a player is eliminated or when a player wins the game;

Physical Constraints

- As a physical Apple Vision Pro device is not available in Portugal, the application should run on an Apple Vision Pro simulated device via a Mac with Apple silicon running at least macOS Sonoma 14.5 with Xcode 16 IDE or above.

3.2 Design

The requirements and domain analysis presented in the previous section, allows for a solid foundation when designing an effective solution for the required prototype. With this foundation in mind, this section presents the prototype's architecture following the C4+1 model that will be described in Section 3.2.1. This will be illustrated via diagrams representing the logical and physical views of three levels of abstraction.

The first one contextualizes the system as a whole, while the second level describes the system's containers. Lastly, the third level details the relevant components within each container. Finally, the section ends with the description the user interface and guidelines the implementation will follow, the gestures used to interact with the prototype will be defined, as well as its branding. This section will served as the basis for the implementation described in the following chapter.

3.2.1 Architecture

The C4+1 model is an aggregation of two separate models: the C4 model and the 4+1 model. The first model was developed between 2006 and 2011 with the aim to help software development teams to describe architectures. This model uses the following four abstraction levels: Context, Containers, Components and Code. The first one is more broad, describing the whole system and how it interacts with external users and systems. The second level is more specific, describing parts of the internal system in a high level. For example, a system with front-end and back-end applications. These are considered containers for the overall system. The third level represents the components inside each of the system's containers (e.g. a *ViewModel*). Lastly, the *Code* level represents the actual code that was implemented (Vázquez-Ingelmo, García-Holgado, and García-Peñalvo 2020).

On the other hand, the 4+1 model was introduced in 1995 and helps to describe a system's architecture using five different views: logical, physical, process, development and scenarios. The last one represents each use case and is considered separate from the others, thus the 4+1 naming. Each view represents the system from different perspectives. From a functionality-based view to other more focused on software or hardware (Kruchten 1995). Combining these two models allows for a better description of a system's architecture, exploring different views for all levels of abstraction.

Level 1

Figure 3.4 represents the logical view for the first level of the prototype's system. It is possible to see in that diagram that the system interacts with an external API for Apple's *GroupActivities* feature. Furthermore, the system interacts with two types of users: *Player* and *Spectator*. As it is defined in the domain glossary, a player is an user that is actively playing the game, which means the user has a seat and is able to interact with the game's elements. On the other hand, a *Spectator* is an user that is not playing the game but rather spectating it via the FaceTime call and SharePlay. This actor is able to move around instead of having an assigned seat to be able to observe the gameplay. On the other hand, the view related with scenarios for this project is represented by the use case diagram showcased in the previous section (Figure 3.3).

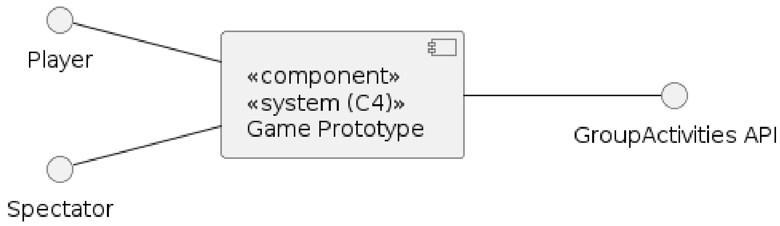


Figure 3.4: Level 1 Component Diagram (Logical View)

Level 2

Moreover, two views are presented for the second level: logical and physical views. The first one (Figure 3.5) is similar to the previous level with a little more detail. Since this prototype only has a visionOS application component, that will be the only container represented in this view inside the system. A non-prototype solution would possibly have more containers in the level as more applications and components will be a part of the system (e.g. backend, Unity or iOS applications). This view represents the actors and external API interacting with the system, particularly the *visionOS App* component.

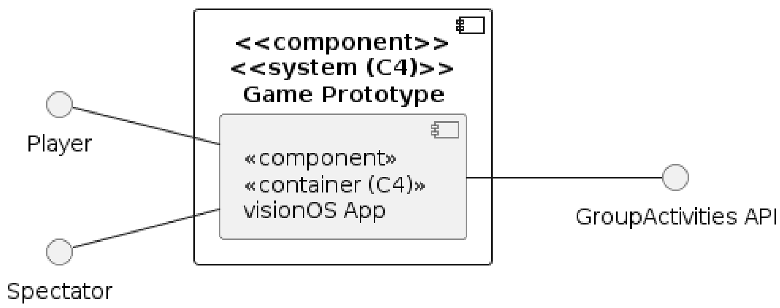


Figure 3.5: Level 2 Component Diagram (Logical View)

On the other hand, Figure 3.6 maps the software components onto hardware. This displays multiple users with their Apple Vision Pro devices and the prototype installed that communicate with each other via Apple's network services and its Facetime application. Some represented users are players while others can be spectators as well.

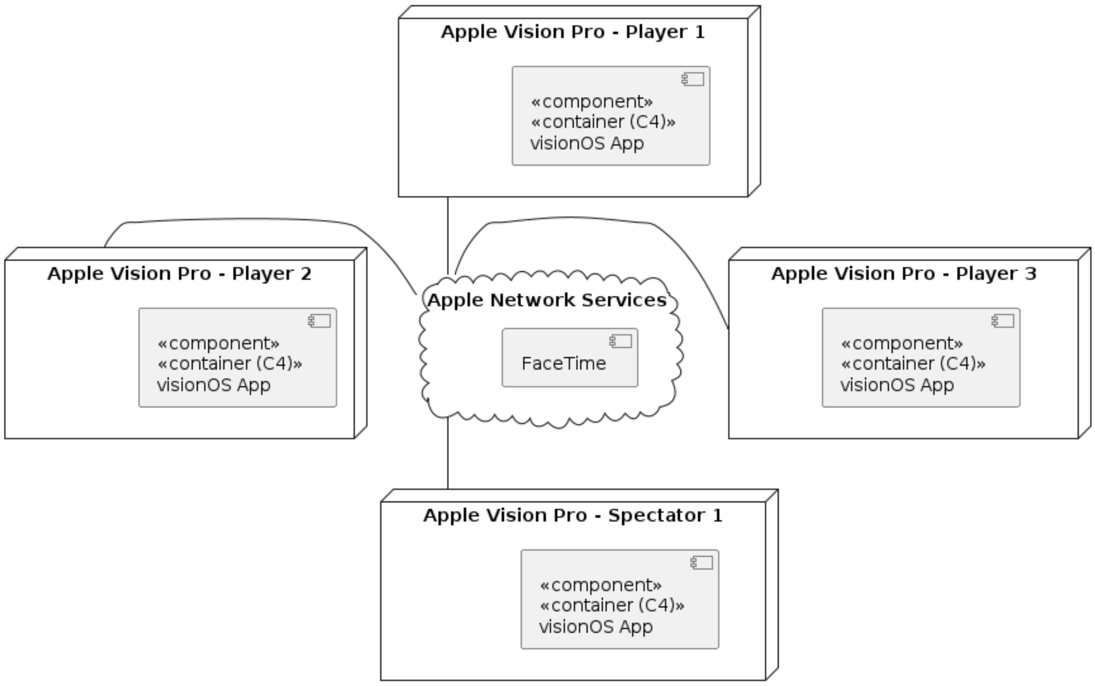


Figure 3.6: Level 2 Deployment Diagram (Physical View)

Level 3

In the third level of C4+1, each container will be described in further detail. In the case of this particular prototype, the *visionOS App* container's logical view is available in Figure 3.7's components diagram. The architecture chosen for the prototype was Model-View-ViewModel (MVVM) with a few adjustments to support TabletopKit's and RealityKit's structure. The MVVM architecture has great synergy with SwiftUI as well.

Thus, we have a *SwiftUI View* that follows a declarative programming paradigm. This kind of view will be responsible to display everything to the user via *Windows* and *Spaces*, where regular UI will be shown. Furthermore, *SwiftUI Views* can also display a RealityKit-powered view where all RealityKit content and models will appear. The *ViewModel* acts as an intermediary that manages state and behaviour for the *SwiftUI View*, exposes a state that the view can observe and bind, and communicates with the *Model*. Much like in regular MVVM, the *SwiftUI View* will observe the state of the *ViewModel* and send user actions that will change that state.

With TabletopKit, a new layer (*InteractionDelegate*) was added that handles TabletopKit-specific user interactions, applies business logic and updates the *ViewModel*. The *InteractionDelegate* will be notified when the user starts to interact with a RealityKit entity and allows the application business logic based on the game state. A similar notification happens when the interactions ends, allowing to update the *ViewModel* and consequentially its state. For example, when a user picks a card, the *InteractionDelegate* will be notified and, based on the starting location, business logic and game state, the allowed destinations where the user can drop that particular card will be vetted and added appropriately. A card from a player's hand can only be dropped in the play card area for that particular player or back to their own hand, being restricted from dropping it into other players' hands and play cards, as well as the table's stack. Another example would be that a card that was picked up in the

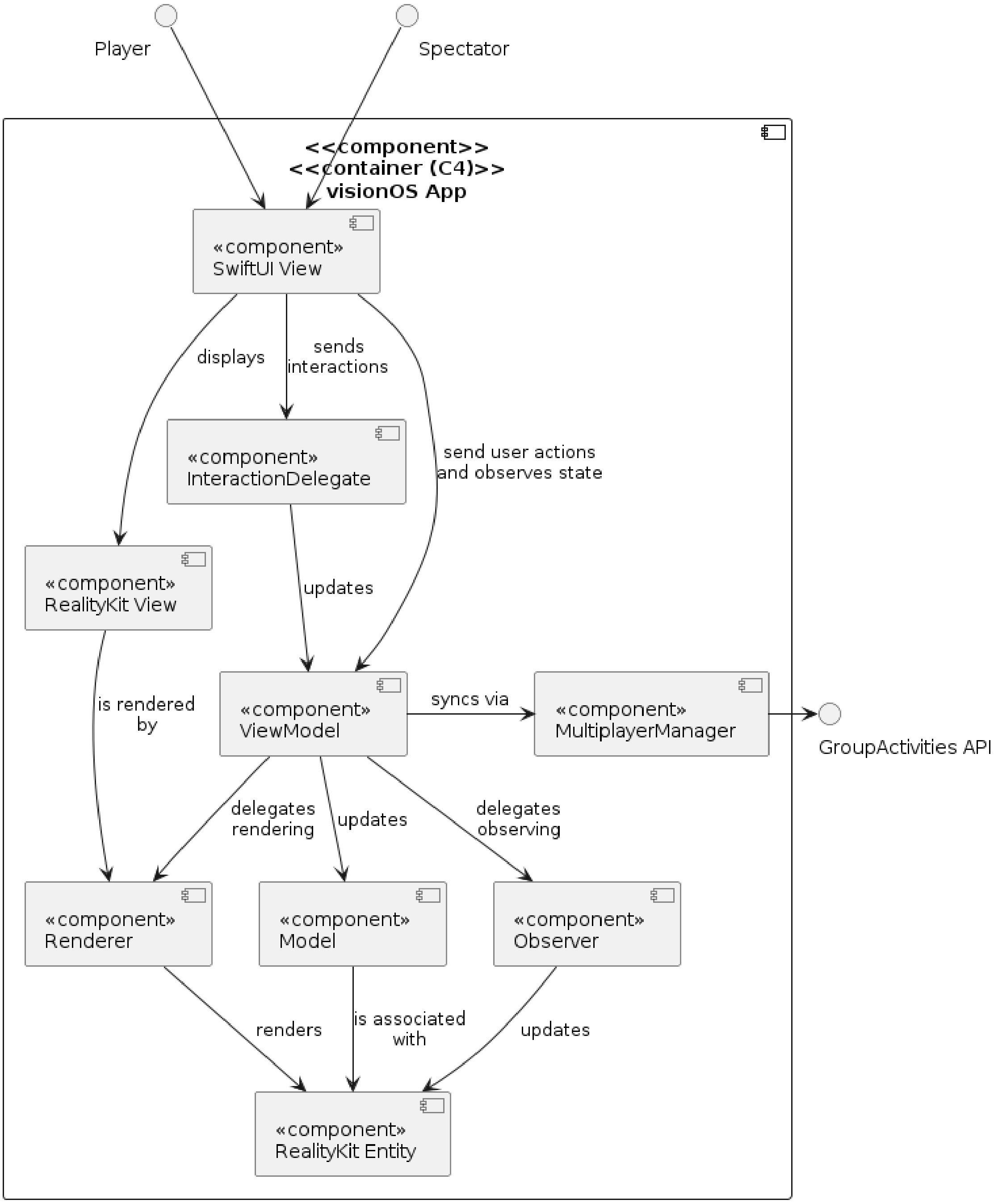


Figure 3.7: Level 3 Component Diagram (Logical View)

table stack can only be placed in that player's hand, not any play card or another player's hand. Restricting a player from placing a card twice in their play card is also an example of this logic being applied. The same applies to logic after a player drops their card, where the *ViewModel* will be notified that a particular card was placed in a specific place, updating the game state and handling rounds outcomes (e.g. all players played a card), handling eliminations when there is a player that has no more cards left in their hand or handling the endgame when a player wins. Moreover, it also handles the equipment permissions and its restrictions when players are interacting with the game. For example, all cards on the table's stack are available for all players to pick up but once a player picks a card from the stack and places it on their hand, that card is now restricted from other players to interact with. The same applies to any card that is chosen to be the play card for that round, as no player is able to pick it up so that players cannot remove a bad card after being played randomly. All this logic is possible due to the *InteractionDelegate* as it manages the instances where players interact with equipment and updates the *ViewModel*.

On the other hand, the *ViewModel* delegates the rendering of the game's RealityKit content to a custom *Renderer* component that is designed to load all RealityKit entities and animations. For example, the robot model that will react to each round win or draw is not particularly a game equipment but still an entity that needs to be loaded and rendered, with its animation logic executed.

Moreover, the *ViewModel* updates the *Model* layer as a result of all actions that it receives from the view or the *InteractionDelegate*. The *Model* represents the game's core data and rules. Examples of *Models* are a *Card*, the *Die*, etc. Some models are also associated with a *RealityKit Entity* that will be rendered by the *Renderer* and displayed in the *RealityKit View*.

The *ViewModel* also delegates some observing of the game state to a custom *Observer* that will observe the state and react with specific business logic. For example, when a player changes a seat, it can re-assign a particular seat or execute specific logic based on the previous and new seats (or lack thereof). This *Observer* can also update *RealityKit Entities*.

Lastly, the *ViewModel* uses a *MultiplayerManager* to sync the local game state with the external *GroupActivities* API from Apple in order to keep every player's game state up-to-date with everyone's actions. This ensures that every player sees the same game state with minimal delay.

3.2.2 User Interface

Other than the game equipment and models themselves, players will be able to interact with a toolbar with seven options. The first option allows the users to reset the game back to its initial state in a regular game mode. This lets player restart whenever they want, giving freedom to interact with game as they please. This reset will be synchronized with all remote players as well. On the other hand, the second option will also reset the game but in a faster game mode. Allowing players, to quickly choose between game modes given their free time or preference. Furthermore, the toolbar will allow players to see the game's rules in a separate window. This window can be placed in the environment while the game is playing played to allow players to check the rules with just a glance. Sometimes players in a group agree on custom rules when playing tabletop games. In order to support that, rules can be turned off by toggling the fourth option in the toolbar. If players want to play in a virtual and digital environment instead of their real one, the fifth toolbar option will allow player to toggle between their real environment and a fully virtual one. This means that players can choose between AR or VR modes for the game. Lastly, the last toolbar option will allow players in a FaceTime call to start a SharePlay activity in order to play the prototype with their friends via multiplayer. Figure 3.8 showcases a design example of how the toolbar should look in visionOS based on Apple's Human Interface Guidelines (Apple Inc. 2023d).

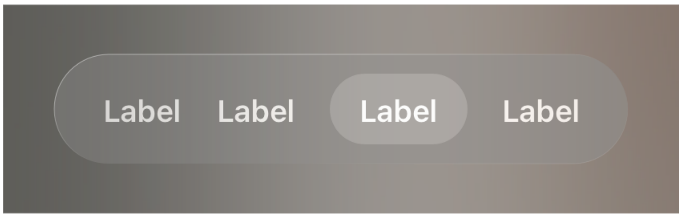


Figure 3.8: Toolbar Example (Apple Inc. 2023d)

3.2.3 Gestures

Players will use gestures in order to interact with the game's elements and user interface. This prototype uses indirect system gestures to navigate its UI. The user is already familiar with these gestures as they are used throughout the operating system and other apps. These indirect gestures allow the user to look at a specific UI element and manipulating it from a distance by doing a pinch gesture with their hand as shown in Figure 3.9.

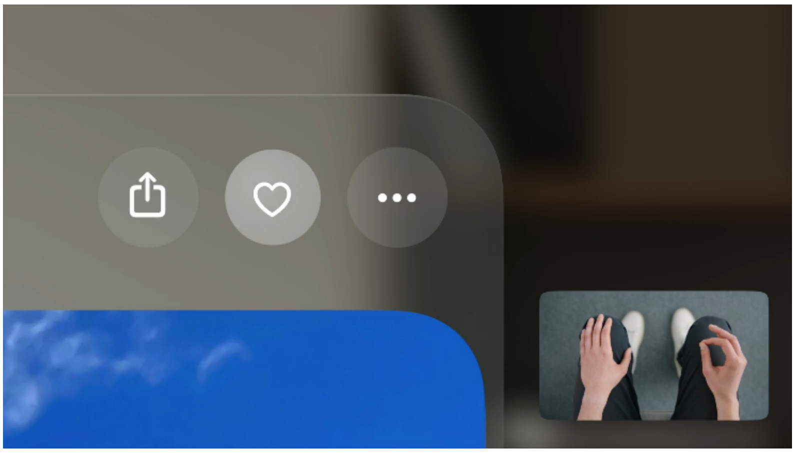


Figure 3.9: Indirect Gesture (Apple Inc. 2023c)

On the other hand, Apple's Human Interface Guidelines for gestures (Apple Inc. 2023c) suggest that direct gestures should be used for physical touch with an interactive object. With this in mind, the prototype's gameplay will use direct gestures. This forces the users to physically touch an object instead of manipulating it at a distance. Figure 3.10 shows the available gestures supported by the prototype while interacting with game elements. Here are some gameplay examples of these gestures being used:

- Pinch: pick up a card or die;
- Drag while pinching: move the selected card or die through the air;
- Let go of pinch: drop the selected card at a determined location (e.g. the player's hand) or throw the die.

3.2.4 Branding

Branding was also something worked on for the prototype. Firstly, the prototype's name was chosen as **AceR**. This is both relevant to the prototype being a card game and a reference to the best card in the game, an *Ace*. It also uses the Augmenter Reality acronym for its upper cased letters (*A* and *R*).

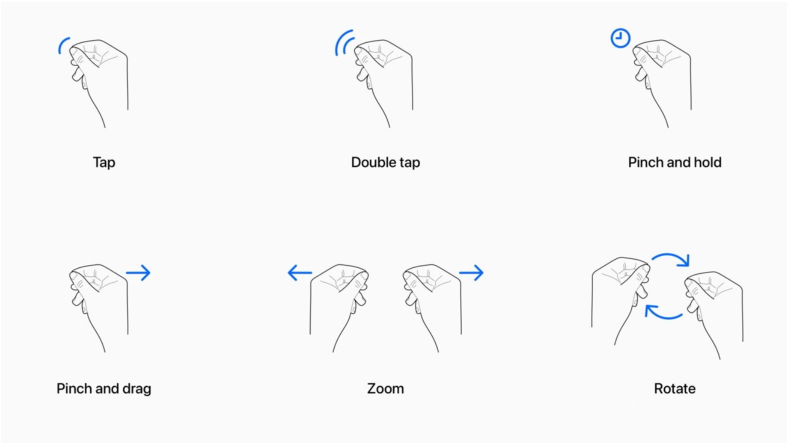


Figure 3.10: visionOS Gestures (Apple Inc. 2023e)

Furthermore, the prototype's icon was explored due to how visionOS handles app icons differently from other flat screen operating systems, such as iOS. Apple's Human Interface Guidelines for visionOS app icons (Apple Inc. 2023b) were followed to create *AceR*'s icon. This defines a visionOS app icon differently from other operating systems as regular app icons are a flat image and usually squared shaped on Apple's operating systems. On the other hand, visionOS icons are circular and feature multiple layers: one background layer and up to two layers on top of it. This requirements exists to allow the operating system to produce a three-dimensional icon that will interact when the user looks at it, as we can see in Figure 3.11.

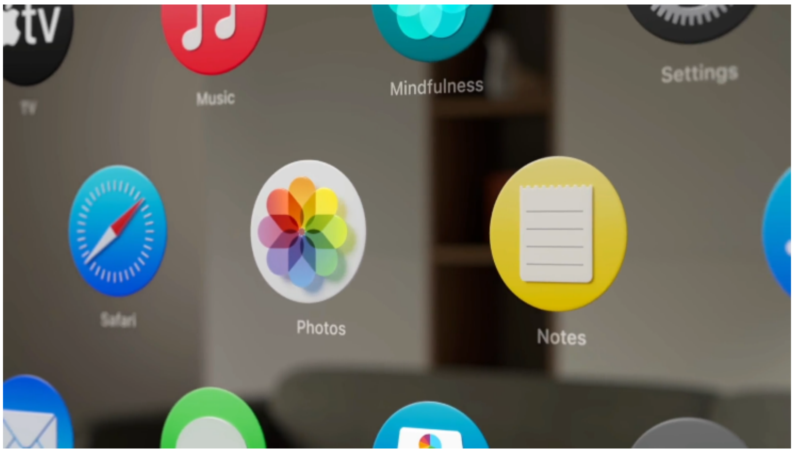


Figure 3.11: visionOS App Icon Example (Apple Inc. 2023b)

Following these guidelines, the author designed an icon with a colourful background and a layer to be placed on top of it with the prototype's logo that uses its *AceR* name. These designs are represented in Figure 3.12 and Figure 3.13 respectively.

Figure 3.12: AceR's Icon Background

Figure 3.13: AceR's Logo

3.3 Summary

In sum, this chapter explored a new game concept called *AceR*, a tabletop card game that utilizes the unique features of the Apple Vision Pro. Its rules were presented along with a flowchart. Domain concepts like a hand, round, suit, seat, among others were defined. The projects functional requirements were presented in the form of 29 use cases and 3 technical tasks, along with its non-functional requirements using the FURPS+ model.

While designing a solution for the visionOS game application, its architecture was presented via scenarios, logical and physical views of three different levels of the application. The user interface of the application defined that players will interact with a toolbar that houses many buttons with quick actions. Users will interact with the application via indirect and direct gestures, as well as eye-sight tracking. Lastly, the branding name and icon were defined to follow Apple's guidelines for visionOS.

Chapter 4

Development

The *Development* chapter will present the implementation process of the AceR prototype's visionOS solution that was analysed and designed in the previous chapter. This solution aims to explore the platform for game development, leveraging its unique capabilities to create an engaging tabletop card game.

The first section will detail this process for the three technical tasks. Moreover, the second section will present the implementation process of each use case. An *Additional Benefits* section will describe some benefits that the platform adds to the game experience for free. The last section will mention the third party assets used during the implementation of the project.

4.1 Technical Tasks

This section will go over the technical tasks responsible to create the visionOS project, set up its app icon and splash screen, as well as implement the *Model* layer that houses all classes for each game concept.

4.1.1 T1: Create the AceR visionOS application

The first task aimed to create the visionOS app project and set it up with all the needed settings and configurations. This includes a *Reality Composer Pro* project where all the RealityKit's scenes and assets will be added and edited. These will then be loaded via code in the AceR visionOS application. The app's information property list file (*Info.plist*) was also updated to have its immersion style as mixed so that users can see their real world surroundings using the camera's pass through video.

4.1.2 T2: Set up App Icon and Splash Screen

As mentioned in Chapter 3, visionOS uses multi-layered app icons that allow the operating system to present them with three-dimensional effects when the user interacts with said icon via their vision. In order to add it to the *AceR* visionOS app, both the front and back images were added to the asset catalogue in the app's resources. As we can see in Figure 4.1, that icon design was implemented in the visionOS app and is displayed on the operating system's main menu, along with other apps. This icon also is used in the app's splash screen that is displayed when the system is transitioning to open the app.

Figure 4.1: visionOS Main Menu

4.1.3 T3: Implement Model layer

In a separate task, the model layer was implemented with models representing game equipment, as well as some other relevant parts of the game. Cards, their suit and kind, hands, seats, play cards, the die, the card stack and the board are among the game equipment represented by a model created in this task.

As we can see in Listing 4.1 that showcases an excerpt from the *Card* struct, each game equipment model conforms to the TabletopKit's *EntityEquipment* protocol. This protocol aims to represent game equipment that will be rendered in the game using RealityKit. This protocol ensures each equipment has properties like a *EquipmentIdentifier* to uniquely identify each entity, a RealityKit *Entity* property that will be loaded onto the RealityKit scene and an initial state. This state can have different types depending on the kind of equipment: *CardState* (used in this case) aims to represent a card-like equipment that houses information such as being with its face turned up or down; *DieState* is used for dice and persists the current value data; and lastly, *BaseEquipmentState* is used for generic equipment that does not conform to the other state types, thus without a need to store specific information to the type of equipment.

On some occasions, an *AudioPlaybackController* is needed if there is a requirement for the entity to play sound effects, which is the case for a card. In this particular model, there are some card-specific properties such as its kind and suit, two created enums that represent the custom card logic of the prototype's game.

```
struct Card: EntityEquipment {

    let id: EquipmentIdentifier
    let kind: Kind
    let suit: Suit
    let entity: Entity
    let audioPlaybackController: AudioPlaybackController
    let initialState: CardState

    // More logic to be added later
    ...
}
```

Listing 4.1: Card Model

4.2 Use Cases

This section will describe in detail the implementation of each of the previously defined use cases for the AceR visionOS application. These include starting the game, playing a card, drawing a round, winning the game, playing in a single player or multiplayer modes, among other features. Additional screenshots are present in Appendix A.

4.2.1 UC1: Start a game

This first use case aims to handle the start of a game when the app launches. This includes creating all the needed views for the app, the view models and other needed adjustments. Firstly, a *GameState* model was created in order to house the game state (deck of cards, hands, seats, stack, etc.) and where logic to update this state will be implemented. It has a *TableSetup* object from the TabletopKit framework that defines how many seats to place players and how are they arranged around the game table, as well as adding game equipment to the table. Seats can be added via the *add(seats:)* function and equipment via the *add(equipment:)* function. These will then be used in the framework's logic for the tabletop game.

Afterwards, a *Renderer* delegate class that implements the *TabletopGame.RenderDelegate* protocol was also created. This class is responsible to render the tabletop game and, in later use cases, it will be responsible to load, render and play animations for specific entities that are not gameplay related (e.g. adding the robot asset that will react to a round win or draw and play a specific animation).

Moreover, an *Observer* class was implemented as defined in the level 3 components diagram in Chapter 3. This class implements the TabletopKit's *TabletopGame.Observer* protocol, aiming to observe the tabletop game state's actions and perform specific logic based on those actions. For example, in a later use case, this class will identify when a user throws the die and it will be responsible to trigger the die sound effect logic.

With all these delegates and state classes implemented, the *GameViewModel* can be created. This view model handles all game-related state to be observed by the game view, as well as letting game-related views perform specific actions. As we can see in Listing 4.2, it has instances of the *Renderer*, *Observer* and *GameState* objects that were previously defined, along with a *TabletopGame* instance. This is a TabletopKit-specific object that is responsible for overseeing the tabletop game's gameplay and setup. It will be initiated by receiving the

GameState's *TableSetup* object to manage its gameplay and setup. Furthermore, both delegate objects (*Observer* and *Renderer*) will be added to the object via the *addObserver()* and *addRenderDelegate()* functions. These will be removed from the object in the class' *deinit*. Additionally, an *AppViewModel* was created so that the general app view can observe a general state relevant for the whole app. For example, this will house state related with the immersive environment.

```
@Observable
class GameViewModel {

    let tabletopGame: TabletopGame
    let renderer: Renderer
    let observer: Observer
    let gameState: GameState

    // More properties will be added in later user stories
    ...

    @MainActor
    init() async {

        self.renderer = .init()
        self.state = .init(root: renderer.root)
        self.tabletopGame = .init(tableSetup: gameState.setup)
        self.observer = .init(tabletop: tabletopGame, renderer: renderer)

        self.tabletopGame.addObserver(observer)
        self.tabletopGame.addRenderDelegate(renderer)

        renderer.game = self

        ...
    }

    deinit {

        tabletopGame.removeObserver(observer)
        tabletopGame.removeRenderDelegate(renderer)
    }

    // More game-specific logic will be added later
    ...
}
```

Listing 4.2: GameViewModel

As defined in Chapter 3, an *InteractionDelegate* is needed between the view model and the view. This delegate is responsible to manage all interactions between players and the game, performing specific logic and actions based on the interaction. This delegate also can update the state from the *GameViewModel* based on each interaction. To do so, this object implements the *TabletopInteraction.Delegate* from TabletopKit. For example, the delegate can identify that the player started an interaction (e.g. picked up a card) and define the destinations where that card is allowed to be placed (using logic from the view model based on its state if needed). Specific logic can also be triggered or actions performed when the interaction ends, based on the type of equipment, the destination and the game state. For example, if an interaction ends with a *Card* type equipment, this delegate can move the card equipment via the *moveEquipment* method or update it its state via the *updateEquipment*. This is where the round end or game over logic will be triggered in later use cases.

In terms of views, a *GameView* view was created that will handle all game-related UI. This view initiates the *GameViewModel* and passes it to a subview that handles the RealityKit and TabletopKit content. As we can see in Listing 4.3, this *GameRealityView* creates a *RealityView* and appends the *Renderer*'s root as a RealityKit entity. Additionally, this is where the tabletop game will be added to the RealityKit view, via the *tabletopGame(tabletopGame:parent:)* method. This method receives the *TabletopGame* from the *GameViewModel* and the *Renderer*'s root as a parent entity. This method offers a closure that returns an interaction from the game that can be interpreted by a *TabletopInteraction.Delegate*. The *InteractionDelegate* previously mentioned will be instantiated in this closure so that the app can intersect any interaction with the game and perform state updates, specific actions and logic.

```
struct GameRealityView: View {

    @State var game: GameViewModel

    var body: some View {

        RealityView { content in

            content.entities.append(game.renderer.root)
        }
        .tabletopGame(game.tabletopGame, parent: game.renderer.root) { _
    in

            InteractionDelegate(game: game)
        }

        // The toolbar will be defined here
        ...
    }
}
```

Listing 4.3: GameRealityView

Lastly, the general app view (*AcerApp* seen in Listing 4.4) was updated to include a *WindowGroup* that instantiates the previously mentioned *GameView*. The style for this *WindowGroup* was set to volumetric so that it is created as a 3D volumetric window and a default size of 1.5 meters was defined for this window group. The baseplate (semi-transparent representation of the floor of the volume) was also hidden via the *volumeBaseplateVisibility* modifier.

```swift
@main
struct AcerApp: App {

    @State private var appViewModel = AppViewModel()

    var body: some Scene {

        WindowGroup {
            GameView()
                .environment(appViewModel)
                .volumeBaseplateVisibility(.hidden)
        }
        .windowStyle(.volumetric)
        .defaultSize(width: 1.5, height: 1.5, depth: 1.5, in: .meters)

        // More window groups and immersive spaces will be added here in the rules and environment user stories
        ...
    }
}
```

Listing 4.4: AcerApp

4.2.2 UC2: See the board

The second use case had the goal of creating, loading and rendering the game board. Firstly, a third party chopping board asset was chosen to be used (Baker 2023). The chosen board asset was "Chopping Board" by Dominic Baker, which is licensed under CC BY 4.0 (Commons 2013). This asset was then converted from its original format (*FBX*) to a format that Reality Composer Pro supports (*USDZ*). Afterwards, the converted asset was added to the Reality Composer Pro's project and a new *BoardScene* scene was created that will be loaded and rendered in the visionOS app. The board asset was added to the scene, adjusting its scale and position to better suit the gameplay. Lastly, some changes specific for this project were made to the asset, as permitted by its *CC Attribution* license (Commons 2013). These changes included changing its material from a wooden texture to a different blue wood texture in order for it to be more on theme with the tabletop game. In Figure 4.2, we can see the *Board* scene open in Reality Composer Pro.

In the visionOS application, a *Table* model was created to conform to *Tabletop*, a TabletopKit protocol that represents the game's table surface. This protocol implements a shape and an identifier for the game's table. For the shape, a *TabletopShape* was created that uses a *rectangular* shape with specific width, height and thickness to match the board asset. Moreover, a *tableID* was created to identify the table. This identifier will be used to add game elements using this *tableID* as its parent for the RealityKit scene.

As shown in Listing 4.5, an initial state computed variable was added to the *Board* model. This *BaseEquipmentState* variable defines the *Equipment*'s *parentID* as the previously defined *tableID*, as well as some other *TabletopKit* related properties. Equipments' control can be defined in such a way that all seats are able to interact with it, only a restricted few, none at all or this value can be inherited from its parent. In this case, no seat should be able to interact with the board directly (only with certain equipment place on it. such as cards). To achieve this, the *seatControl* property was defined as *restricted* with no seats being able to interact with it. Furthermore, the board position and bounding box was also defined.

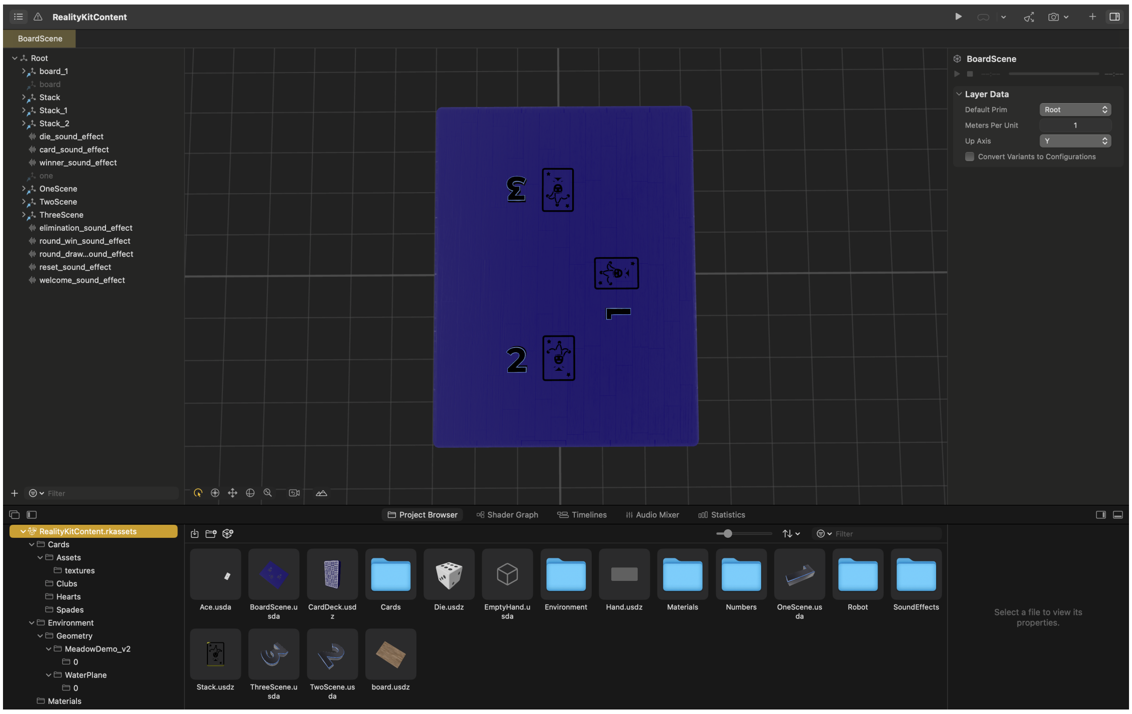


Figure 4.2: Reality Composer Pro's Project

```
struct Board: Equipment {

    ...

    var initialState: BaseEquipmentState {

        .init(
            parentID: .tableID,
            seatControl: .restricted([]),
            pose: .init(position: .init(), rotation: .degrees(45)),
            boundingBox: .init(
                center: .zero,
                size: .init(Constants.boardEdge, Constants.boardHeight,
    Constants.boardEdge)
            )
        )
    }
}
```

Listing 4.5: Board Model

Lastly, an instance of newly created *Table* model was injected in the *GameState*'s *TableSetup* object. In the same file, the board was created and added to the *TableSetup* via the *setup.add(equipment: board)* function. The board scene was loaded in the *Renderer* file as well. As we can see in Figure 4.3, the board is displayed to players in their environment (cards and other game elements seen in the image were added in later use cases).

Figure 4.3: AceR Prototype

4.2.3 UC3: See the toolbar

This use case intends to implement a toolbar, a view that is displayed above the operating system's window-management controls. This toolbar appears closer to the user than the game's board, allowing to perform certain pre-defined game-related actions. These actions will be implemented in other use cases and include resetting the game state (either in a faster game mode or as a regular full game), checking the game's rules, toggling said rules on or off, changing the environment and creating a multiplayer session with friends via Facetime and SharePlay. These are made available in the toolbar as it stays always near the board, this ensures that the player has constant access to these and can press any of them at any time.

In order to fulfil this requirement, a new view was created that will represent the toolbar. This view conforms to the *ToolbarContent* protocol and adds a *ToolbarItemGroup* with all buttons needed for the game actions. Their actions where empty as they were implemented in later use cases. These buttons use *SF Symbols* (Apple Inc. 2022b) as their images because they offer an extensive library of standardized and adaptable icons. Using these icons has some advantages such as having great synergy with all font weights and dimensions, ensure visual coherence and legibility, allow to use multiple colours or styles, and they are free to use inside the Apple's ecosystem. These can be added to any view using their unique key. This view also receives both the *AppViewModel* and the *GameView* as these will be needed to perform the action and observe their state in order to display different icons (e.g. display the light switch icon as *on* if the rules are turned on or the icon as *off* if the user is playing without rules). Similar to the actions, the different icon logic is also part of other use cases.

The implementation details for the toolbar view can be seen in Listing 4.6, where an excerpt shows the view and multiplayer button declaration.

```
struct Toolbar: ToolbarContent {

    let appViewModel: AppViewModel
    let gameViewModel: GameViewModel

    ... // Init

    var body: some ToolbarContent {

        ToolbarItemGroup(placement: .bottomOrnament) {

            ... // Other Buttons

            let gameModeImage: String = gameViewModel.isSinglePlayer
              ? "shareplay.slash"
              : "shareplay"

            Button("GameMode", systemImage: gameModeImage) { ... } //
    Action added later
        }
    }
}
```

Listing 4.6: Toolbar View

In order to display this toolbar view in the game window, the *GameRealityView* was updated with a *.toolbar* modifier that instantiates the *ToolbarContent* custom view that was previously described. Lastly, in Figure 4.4, we can see the outcome of this use case's implementation. It is placed in bellow the board and displays all action buttons that allow the user to quickly alter some game settings.

4.2.4 UC4: Take a seat in the game

The fourth use case aims to let players take a seat in the tabletop game. In order for this to happen, an array with all three *Seat* models was created. This defines their *TableSeatIdentifier* identifiers, and position in the table (one seat on each side of the table, leaving one side empty). This array was then stored in the *GameState* model and added to the scene via the *setup.add(seat: seat)* function. In the *GameViewModel*, the player was automatically assigned to a seat in the game when joining via the *tabletopGame.claimAnySeat()* function.

4.2.5 UC5: See my player hand

This use case implements a player hand for each seat previously defined. In order to achieve this, the *Hand* model was updated to have a *BaseEquipmentState* initial state with the position being relative to the hand's seat. Afterwards, logic was added to the *GameState* where all hands were created based on each of the table's seats, adding them to the *TableSetup* via the *setup.add(equipment: hand)* function. Since the hands were not represented by a visual entity, no asset was needed for it.

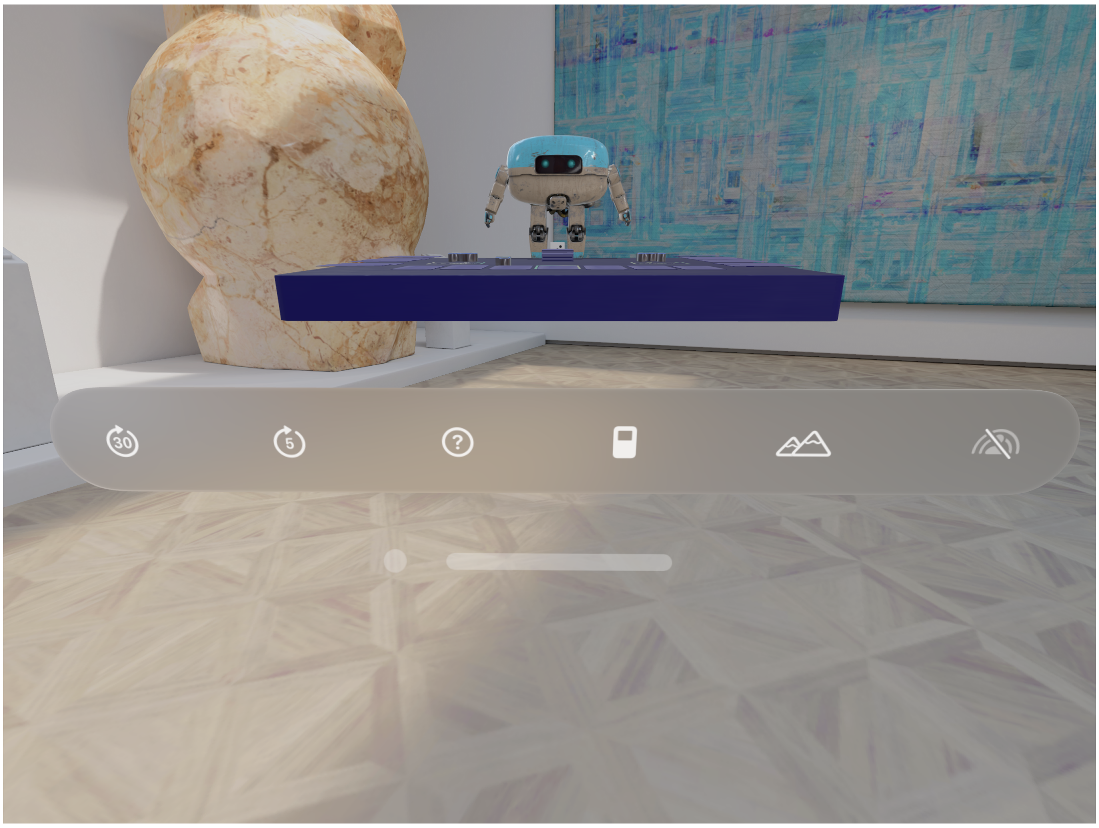


Figure 4.4: Toolbar

4.2.6 UC6: See cards in my player hand

This use case aimed to add cards to each player hand, laying them over each other in a way that the player can see them all at any time. This allows them to be picked up in a future use case. Firstly, the third party cards assets were chosen (Yanez 2018). The chosen card assets was "52-Card Deck" by Anthony Yanez, which is licensed under CC BY 4.0 (Commons 2013). These assets were slightly modified as it had all cards from a deck bundle together in single *FBX* file. This is where the first limitation of developing for visionOS appeared. While converting assets with other formats for singular assets was, for the most part, a straightforward process. More complex assets, such as this one, proved to be more of a challenge. Converting this deck to *USDZ* (*Reality Composer Pro* preferred format), introduced some misconfigurations and limitations that didn't allow for the conversion to be successfully used in the RealityKit project. In order to prevent that, an external program (*Blender*) was used to extract each card asset separately and then export it as *USDZ*. After this whole process, all the card assets were added to the prototype's RealityKit project, ready to be loaded by the visionOS application.

In the visionOS project, a static deck with all cards from each suit and value was created. This deck was then used in the *GameState* object to create all needed card entities, with each card using their suit and value to programmatically load their respective asset from the RealityKit project. These cards were then added to the *TableSetup* object, in a similar way to other equipments.

Furthermore, a *layoutChildren(for:visualState:)* function was implemented in the *Hand* model. This function provides a custom layout for the hand's children entities. In this case, the hand's children are the cards that will be added to it during gameplay. This function uses *EquipmentPose2D* and each children to provide a *planarOverlapping* layout for the cards in each hand layered on top of the others, as we can see in Figure 4.5.

Figure 4.5: Cards Stacked in the Hand of a Winner

4.2.7 UC7: See cards in the board's stack

In order for the game to have the cards in the table's stack, the *Stack* model created in the third task was updated to have a *BaseEquipmentState* initial state that corresponds to the intended location of the card stack in the RealityKit scene. This initial state also defines a bounding box that will capture when cards are moved and dropped on top of it. This model also has an array of cards that represent the cards that are currently on the stack. Moreover, this stack object was created in the *GameState* and was added to the *TableSetup*. As cards will be moved to the stack as children, the default *layoutChildren(for:visualState:)* implementation will place them on top of each other, as we can see in Figure 4.6.

4.2.8 UC8: Have the cards dealt to the hands and stack

The eighth use case aims deal all of the defined deck's cards to each one of the player's hands, as well as dealing the remaining to the table's stack. Firstly, a *cardsPerPlayer* constant was defined with the number of intended cards per player (seven, in this case).

Additionally, in the *GameViewModel*, a *dealCards* function (Listing 4.7) was created to deal all the cards to the table's stack and seven cards for each of the player's hands. Firstly, the function fetches the deck of cards from the *GameState* object, shuffling so that it looses its order. Moreover, for each of the player's hands, seven cards will be removed from

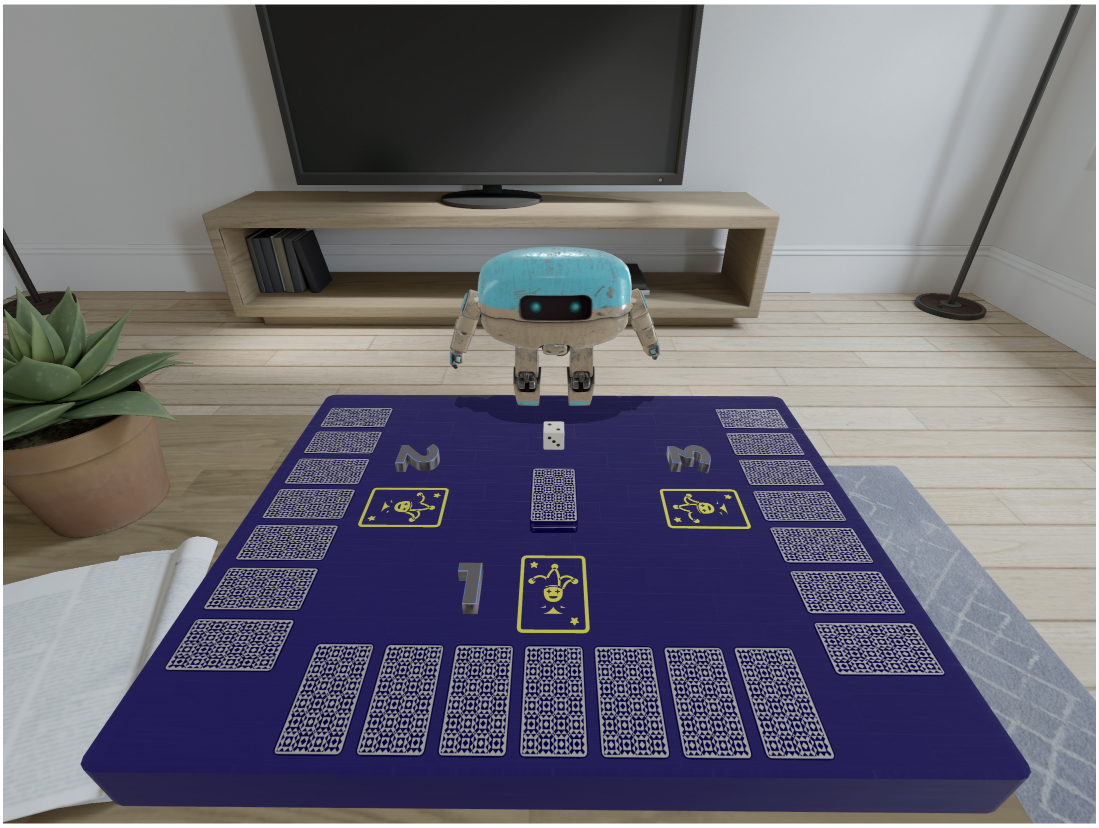


Figure 4.6: Top-Down View

the deck. Each card equipment will then be updated via the *TabletopGame*'s *addAction* function, where an *updateEquipment* action will be sent to the TabletopKit management object. This action will be executed, updating the equipment's state. The card's seat control will be restricted to only allow the hand's owner to interact with the card. This prevents other players to be able to pick up, move or interact in any way with a card from another player's hands. This proves to be one factor where the game can be enhanced by being played in an XR context, rather than a real world physical game. In a physical game, nothing prevents players from cheating and seeing or grabbing another player's cards. This is not the case in the XR world. Since this is a *CardState* equipment, we are able to also define if the card is facing up or down. This is a great advantage of TabletopKit as it has built-in support for card and dice equipments, with specific logic for each. In this case, the card was updated to be face down as players are not supposed to see the card's value before playing it. Then, another action is added to move the card to be a child of that specific hand. This will trigger the *layoutChildren(for:visualState:)* function implemented in UC6.

Now with all cards shuffled and moved into the player's hands, the remaining cards follow a similar process to be added to the stack. Contrary to the cards in a player's hand, the stack's seat control is marked as *any*, since all player can interact with the cards in the table's stack. Some helper functions were also implemented and aim to add a certain card to a hand or the stack, so that the *GameState* object keeps track of the game's state. Lastly, the *dealCards* function is called when the *GameViewModel* is initialized so the game is ready to be played and all cards are randomly dealt at the beginning of the game.

```
@MainActor
func dealCards() {

  var shuffledCards = gameState.deck.shuffled()

  guard !shuffledCards.isEmpty else { return }

  for hand in gameState.hands {

    for _ in 1...Constants.cardsPerPlayer {

      let card = shuffledCards.removeFirst()
      tabletopGame.addAction(
        .updateEquipment(
          card,
          faceUp: false,
          seatControl: .restricted([hand.owner.id])
        )
      )

      tabletopGame.addAction(
        .moveEquipment(matching: card.id, childOf: hand.id)
      )

      gameState.add(card: card, to: hand)
    }
  }

  for card in shuffledCards {

    tabletopGame.addAction(
      .updateEquipment(card, faceUp: false, seatControl: .any)
    )

    tabletopGame.addAction(
      .moveEquipment(matching: card.id, childOf: gameState.stack.id)
    )

    gameState.addToStack(card: card)
  }
}
```

Listing 4.7: Deal Cards to Stack and Player's Hands

4.2.9 UC9: See where I can place my played card

In order for players to see where they can place their chosen card to be played in that round, another third party asset was added to the RealityKit project (Buradkar 2024). This asset represents a golden joker card with a metallic reflection effect and will be used to show players where they can place their chosen cards for the round. Similarly to previous use cases, the model (*PlayCard* in this case) and *GameState* were updated so that the asset is rendered in the RealityKit view, in front of each player hand. As we can see in Figure 4.6, this means that every player has a single *PlayCard* equipment where they will be able to place their chosen card for the round.

4.2.10 UC10: Draw a card from the stack

This use case intended to allow players to pick up a card from the stack and drop it in their hand. The implementation of this use case introduced logic to the *InteractionDelegate*. By overriding the *update(interaction: TabletopInteraction)* function from the *TabletopInteraction.Delegate* protocol, it allows to introduce custom logic when the user starts interacting with an object (e.g. picks up a card) and when the player ends the interaction (e.g. drops a card in the stack). This is based on the value that comes from the *TabletopInteraction* parameter. If the interaction's value's phase is *.started*, it means that the interaction started. On the other hand, if the interaction's value's phase is *.ended*, it means that the interaction ended.

When an interaction starts for an equipment of the type *Card*, we are able to define where that equipment can be placed. As we can see in Listing 4.8, this is possible via the *setConfiguration* function on the interaction object, defining an array of allowed destinations where the card can be dropped. In this case, the allowed destinations is only the player's hand that is fetched via its identifier.

Next, logic was added to call a function that handles when a player ends the interaction with a card (drops the card). This function (Listing 4.9) identifies what type of equipment is the interaction's proposed destination. In this particular case, if the destination is the player's hands, the card's seat control will be updated to be restricted to its owner. This means that only the owner of that hand can interact with that card from that point on (previously was set to *.any* as any player could pick this particular card up from the stack). The card is also placed face down so the player can't see its value. This is another advantage from playing in the real world as players can't cheat and see their cards before playing them. Lastly, the card is updated with these values and moved to the player's hand using actions similar to previous use cases.

In Figure 4.7, we can see a player picking up a card using a tap direct hand gesture. After picking the card up, the user can move the card by moving their hand but keeping the fingers holding the virtual card. By releasing the fingers above the player's hand, the card will be dropped and added to that pile.

```
struct InteractionDelegate: TabletopInteraction.Delegate {

    ...

    private func onPhaseStartedCard(
        interaction: TabletopInteraction,
        card: Card
    ) {

        var allowedDestinations: [EquipmentIdentifier] = []

        game
            .tabletopGame
            .withCurrentSnapshot { snapshot in

                guard
                    let localSeat = snapshot.seat(
                        of: Seat.self,
                        for: game.tabletopGame.localPlayer
                    )
                else {
                    return
                }

                for hand
                    in game.tabletopGame.equipment(of: Hand.self)
                    where hand.owner.id == localSeat.0.id
                {
                    allowedDestinations.append(hand.id)
                }

                ...

                interaction.setConfiguration(
                    .init(
                        allowedDestinations: .restricted(allowedDestinations)
                    )
                )
            }
    }

    ...
}
```

Listing 4.8: Define Card Allow Destinations

```swift
struct InteractionDelegate: TabletopInteraction.Delegate {

    ...

    private func handleCardEnded(
        card: Card,
        interaction: TabletopInteraction,
        proposedDestination: TabletopInteraction.Destination
    ) {

        var seatControl: ControllingSeats = .any
        var faceUp = false

        if let hand = game.tabletopGame.equipment(
            of: Hand.self,
            matching: proposedDestination.equipmentID
        ) {
            ...

            seatControl = .restricted([hand.owner.id])
            Task { await game.setup.add(card: card, to: hand) }
        }

        ...

        interaction.addAction(
            .updateEquipment(
                card,
                faceUp: faceUp,
                seatControl: seatControl
            )
        )

        interaction.addAction(
            .moveEquipment(
                matching: card.id,
                childOf: proposedDestination.equipmentID
            )
        )
    }

    ...
}
```

Listing 4.9: Logic When the Card Interaction Ends

Figure 4.7: Moving a Card

4.2.11 UC11: Play a card from my hand

Similarly to the previous use case, UC11 aims to allow players to play a card from their hand in a given round. In order to achieve this, the previously mentioned *onPhaseStartedCard* function was updated to add the player's play card as an allowed destination when the interaction starts (the player picks the car). This destination is only added if there is no card there already. If the player already played a card, no card can be placed on their play card until the round is over. Furthermore, the *handleCardEnded* function was also updated so that when the player lets go of the card, the card equipment is moved to be on top of the play card asset. In this function the card's state is also updated to be faced up, revealing its value since the player played their card. Lastly, the seat control is set to *restricted* in order to prevent any player to move the card elsewhere, even the one who played it. This prevents the player to remove the played card once its value is seen, another advantage of the XR implementation of the game versus its real world counterpart.

4.2.12 UC12: Roll a die

In this use case, a die will be introduced to act as a tiebreaker when two cards with the same value are played at the same time. Firstly, a third party die asset (Apple Inc. 2024c) was used and added to the RealityKit project. Secondly, the *Die* model was updated to conform to the *EntityEquipment* protocol, using a *DieState* as its initial state. Like previously mentioned, different equipment entities can have different types of state and TabletopKit provides custom states for both cards and dice. In this case *DieState* is a state that stores a value for the die. In the model, the die's position is set, the asset is loaded and the seat control is defined as *any* (so that any player can pick up the die). A *TossableRepresentation* property was also added so that the player can throw the entity, defining its as a cube with its height set be the same as the asset. On the other hand, a *restingOrientation* function was implemented to define the die's rotation based on its value. This function will rotate the asset so that the value display on top is the same as the value on the *DieState*. Moreover, a *Die* object was created in the *GameState* and added to the tabletop game via the *TableSetup*.

With the die rendered and added to the view, the interactions when throwing the die needed to be implemented. In the *InteractionDelegate*, the allowed destinations when the player picked up the die were restricted to none. This was done so that the die could not be placed anywhere on the table, just thrown. In order to add logic when the die is rolled, a new case was added to observe when the gesture ended. This different from the card behaviour as we observed when the overall interaction ended via checking if the *interaction.value.phase* was set to *ended*. In the case of the die, we wanted the interaction logic to be triggered when the gesture ended instead, so that the toss animation was triggered mid-air once the player rolls the die. This was done by observing the *interaction.value.gesture.phase* instead. Listing 4.10 shows the logic triggered when the gesture ends. In it, we can see that the next value is calculated using a custom *nextValue* function that returns a random integer between 1 and 6. This implementation was kept configurable so that if there was a needed to update the die to have more than six sides, it was easily changed within the code. The die's state will be updated with this new value via an update action. Furthermore, the toss animation is triggered via the *toss* function that uses the previously implemented *TossableRepresentation* parameter in the *Die* model. This renders the tossing simulation when the player drops the die. Lastly, the die is moved back to its original position in the board, this time with the value updated to the new one. This signals the players to the value of the die roll. The outcome of this use case's implementation can be seen in Figure 4.8.

```
struct InteractionDelegate: TabletopInteraction.Delegate {

    ...

    private func onGesturePhaseEnded(interaction: TabletopInteraction) {

        guard
            let die = game.tabletopGame.equipment(
                of: Die.self,
                matching: interaction.value.startingEquipmentID
            )
        else {
            return
        }

        interaction.addAction(
            .updateEquipment(
                die,
                state: .init(
                    value: Die.nextValue(),
                    parentID: .tableID,
                    entity: die.entity
                )
            )
        )

        interaction.toss(
            equipmentID: interaction.value.startingEquipmentID,
            as: die.representation
        )

        interaction.addAction(
            .moveEquipment(
                matching: die.id,
                childOf: .tableID,
                pose: die.initialState.pose
            )
        )
    }

    ...
}
```

Listing 4.10: Die Interaction Logic

4.2.13 UC13: Win a round

In this use case, logic was added after each interaction ends where a card equipment is placed in a play card. This logic checks if all players have played a card (all play cards have a card). If so, the end round logic will be triggered. In this case, the game will check each played card's value against each other and find the lowest value (as per the game's rules). If a winner is found (a card has value lower than all other cards), the game will update all played card's state to be faced down and restrict their seat control to the winner, moving them to the winner's hand. This means that the winner receives all of the round's played cards in their hand, faced down. A new round begins and players are now able to play another card.

Figure 4.8: Throwing a Die

4.2.14 UC14: Resolve a draw

Similar to the previous use case, this aims to handle when a round ends in a draw. A draw can happen when two or more players play a card with the same value. For this, logic was added to the round end logic that is triggered after all players play a card. If two or more played cards have the same value, new logic is triggered that instead of updating and moving cards automatically to a specific hand, gives players freedom to handle the draw as they please. In the game's rules, a draw is resolved by rolling a die, with the winner being whoever rolls the lesser value. In order to achieve this, all played cards are updated to be able to be moved by any player after a draw. This allows players who have drawn to roll a die and then move the cards to the respective winner. This implementation does not automatically move the cards after the die toss to give users some freedom to decide how to resolve draws however they want. Sometimes in tabletop games, some house rules are set where all players agree to change the official rules to enhance the game's enjoyment. For example, some users might prefer to have a rock, paper, scissors to resolve a draw instead. This implementation allows for house rules as it doesn't force the players to roll the die, maintaining similar freedom to a real world environment.

4.2.15 UC15: Get eliminated from the game

When a player has no more cards in their hand and there are no more cards in the stack to pick up, they are eliminated from the game. From an implementation perspective, a check was added after a rounds ends that goes over every player hand and checks if the hand is empty and the stack has no more cards left. If so, it removes the player and their equipment from the game, leaving the rest to continue playing. Round ending logic will then be updated to reflect the number of players still in play.

4.2.16 UC16: Win the game

Likewise, this use case has the goal to identify when a player wins the game and display a winning message congratulating the winner. In order to achieve this, after a successful elimination at the end of a round, the game will check if there is only one player left in play. If so, the game will lock itself and display a congratulatory message to the wining player, as seen in Figure 4.5. This message will then disappear when a new game is started.

In order for each player to identify himself, some assets were added to the board RealityKit scene. Three 3D number assets (Jihambru 2022a,b,c) were added near each player's playing card, displaying the number one, two and three corresponding to each player.

4.2.17 UC17: Add a Robot that will react to a round win or draw

This use case adds a 3D companion robot that will react happily when a player wins a round, as well as react sad when a draw happens. Firstly, a third party robot asset was chosen and imported into the RealityKit project, along with three animations: idle, happy and sad (Apple Inc. 2024c). The *Renderer* class was updated to load the robot asset into the RealityKit scene when being initialized. The entity is created with the asset and all three animations are added to a *AnimationLibraryComponent* that will be set in the robot entity. This means that all animations are loaded into the entity when instantiated and then played when the right time comes. After this initial logic is executed and the entity added to the scene, the idle animation is started and set to repeat. This keeps the robot moving in idle until a more expressive animation is played. Moreover, more functions were added to support fetching and playing animations, as well as a function to play an animation based on a specific round outcome. Logic was also added to trigger the animation when a round is concluded.

As we can see in Figure 4.9, the robot was placed on the side of the board that doesn't have a player, overseeing the whole gameplay. The same figure showcases the robot jumping in joy after a player wins a round, playing the happy animation. On the other hand, Figure 4.10 displays a scenario where the robot is sadly reacting to a draw between multiple players.

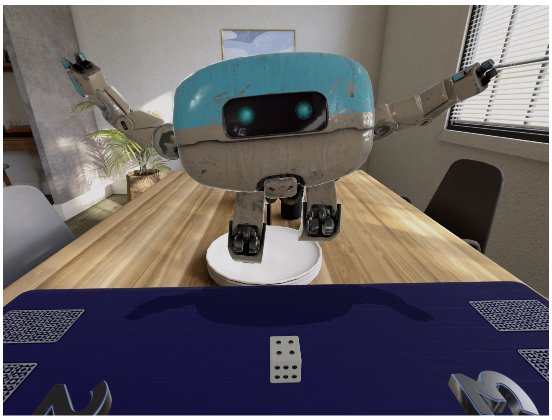


Figure 4.9: Robot Reacting to a Round Win

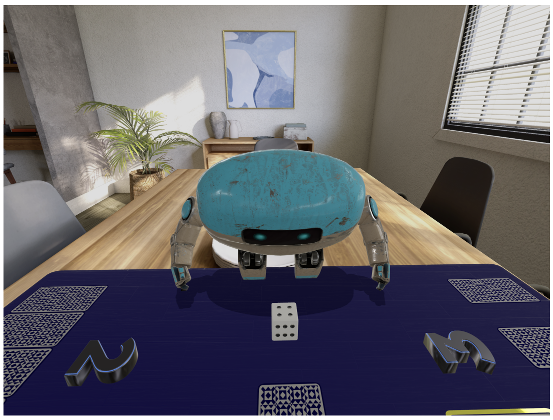


Figure 4.10: Robot Reacting to a Round Draw

4.2.18 UC18: Hear sound effects when interacting with equipment

In this use case, the requirements were to add sound effects when interacting with game equipment and during gameplay. These instances included picking up a card (Splashdust 2022), rolling a die (AbdrTar 2022), winning a round (chieuk 2024), drawing a round (myfox14 2021), a player being eliminated (Nighteller 2022), a player winning the game (uss015dykrt 2023), resetting the game (TuomasData 2023), as well as a robot welcome voice message when first launching the app (xxxTruthxxx 2022). In order for this be possible, all cited royalty free third party sound effects were added to the RealityKit project. Each sound effect should be played spatially to the user from the related entity's location, so that users are immersed in the experience. For example, a card flipping sound effect should come from the specific card that the player is interacting with. Similarly, the die roll effect should come from the die entity. All other general sound effects come from the robot entity, as he acts as the game's Disc Jockey (DJ).

In the codebase, the correspondent models for these equipments (card, die and robot) were updated with an *AudioPlaybackController*. This controller loads the audio asset (sound effect) and prepares the audio to be played when the object is initiated. The audio is then triggered by a custom *playSoundEffect* function in the object that plays the defined audio where this entity is located. This means that the sound effect comes spatially to the player from where entity is located, providing extra immersion.

In the case of a player interacting with a card or die, this *playSoundEffect* function is triggered from the *Observer* class whenever an action is executed. This class observes the actions added in other classes like the *InteractionDelegate* and executes the *actionIsPending* whenever an action is pending. If the action has a *MoveEquipmentAction* type, the function will then identify if the equipment is a die or a card, executing their respective *playSoundEffect* functions. On the other hand, robot related sound effects are triggered whenever the game logic determines a round win, draw, elimination, reset, etc.

4.2.19 UC19: Reset my game

This use case aims to implement the game reset logic that can be trigger by the player at any point via the toolbar. This was implemented in the *GameViewModel* using previously defined functions to deal all cards, as well as resetting some properties (e.g. when a winner is found). This logic was then triggered from the *Toolbar*'s reset button.

4.2.20 UC20: Play in a single player mode

In order for players to be able to play alone, a single player mode was implemented. This mode is active by default and will be disabled when a player creates a SharePlay multiplayer session with other Facetime participants. If the single player mode is enabled, whenever a local player plays a card from their hand, logic was added to move a random card from each of the other players' hands to their play card. This means that there is a bot automation system that observes whenever the local player plays their card and automatically moves and updates a random card for all empty seats.

Furthermore, additional logic was added to automate player's interactions whenever the local player draws a card from the stack to their hand. When a player performs this action, the automation system will also draw a card for the bot players. This allows the user to play a game of AceR without the need to have a multiplayer session active and friends on a Facetime call.

Lastly, the game ends if the local player is eliminated and two bots are still in play, so that the player doesn't need to see the rest of the game being automated.

4.2.21 UC21: Have a faster game mode

An additional game mode was added to the game. This new mode allows users to play a game that takes less time that the full AceR experience for users who have less time to spare or like a faster gameplay. This game mode restricts the game to just 3 cards per player, as well as removing the stack as an interactable element. The stack will hold all other cards that are not being used in the fast game, forbidding the player from picking them up using a *restricted([])* seat control. With less cards, the game is much quicker, allowing for faster gameplay. Logic was added to the toolbar button in order to allow the game to be restarted in a faster game mode so that users can quickly change between a regular pace or a faster game mode at any time. The outcome of this use case can be seen in Figure 4.11.

Figure 4.11: Fast Game Mode

4.2.22 UC22: Show rule set

This use case had a similar aim as the previous one, but with a different action. The toolbar action should open a new window that presents the game's rule set to the player. In order to implement this use case, new SwiftUI views were created to support the intended rule set window. These views were added to the *AcerApp* scene via a new *WindowGroup* that has a specific identifier. This identifier will then be used in the toolbar's button action via the operating system's *openWindow* function. This function opens a *WindowGroup* with a specific identifier. As we can see in Figure 4.12, this approach allows users to read the game's rules before or even during gameplay as the window can be placed anywhere in the room and it stays anchored in the place the user placed it. Since this is local for each player, each player can have their own rules opened in the background behind the board and check them throughout the game without the need to share the rule book. This proves to be an advantage of the XR implementation in the Apple Vision Pro when comparing to a real world implementation where players would have a rule book that needed to be checked each time there is any doubt about the game's rules. It also allows every player to be looking at the rules at the same time, which is not possible in most physical tabletop games. This helps players when they are in their first playthroughs, improving the game's onboarding.

Figure 4.12: Rule Set

4.2.23 UC23: Toggle rules off

Similarly to the previous use case, this aims to implement an action in the toolbar that toggles on or off the game's rules. This allows players to implement their own house rules or even a different game, using the XR equipment built for AceR. Rules are turned on by default. If the player chooses to turn them off, logic restricting card interactions for certain players or for all is turned off, allowing anyone to pick any card. This way players can set their own rules for who wins or draws a round, if any custom house rules are agreed upon. This gives parity to a physical card game as it doesn't force any automatic rules and restrictions, giving players freedom to choose their own rules.

4.2.24 UC24: Toggle my environment from mixed to virtual reality

This use case allows players to toggle between the default mixed reality mode and a virtual reality one. In mixed reality, the user can see the board placed on their real world surroundings, this is done via the Apple Vision Pro's cameras. Users can place the board wherever they want in their environment and it stays anchored. For example, players can place the board on their living room table and see their friend's spatial personas appear around it, as if they were in the same room together. On the other hand, players can also choose to have the board floating in the air if they prefer, allowing for comfort and customization for the player's gameplay. With the new VR mode, users can use the toolbar to toggle between their real world environment and a fully virtual one. This virtual environment will immerse players in a garden where they can play peacefully with their friends. The immersive environment lets players see their immediate surroundings as a security measure, so that players don't accidentally hit something when trying to interact with the game via gestures.

Firstly, in order to implement this use case, a third party garden environment asset (Apple Inc. 2023a) was added to the RealityKit project. Then, a new SwiftUI view was created to render a *RealityKit* view. This view will add the garden environment scene to its content. Moreover, state logic was added to the *AppViewModel* in order to track when this virtual space is being opened or closed. This logic is triggered in the toolbar button action via the *openImmersiveSpace* and *dismissImmersiveSpace* environment values that the operating system provides. Finally, as seen in Listing 4.11, the main *AcerApp* scene was updated to have a new *ImmersiveSpace* that will display the content unbounded in the player's environment. The *immersionStyle* is set to *progressive* so that are lost in the virtual world but still can see their immediate surroundings for safety. Figure 4.13 shows the use case in action with a player playing a game immersed in the garden environment, while still maintaining some parts of their real world.

```
@main
struct AcerApp: App {

    @State private var appViewModel = AppViewModel()

    var body: some Scene {

        ...

        ImmersiveSpace(id: AppViewModel.immersiveSpaceID) {

            EnvironmentView()
                .environment(appViewModel)
                .onAppear { appViewModel.immersiveSpaceState = .open }
                .onDisappear { appViewModel.immersiveSpaceState = .closed }
        }
        .immersionStyle(selection: .constant(.progressive), in: .progressive)


    }
}
```

Listing 4.11: Add Immersive Virtual Environment to Scene

Figure 4.13: Immersive Environment

4.2.25 UC25: Create a SharePlay multiplayer game session

FaceTime's SharePlay feature was chosen to be used for the multiplayer aspect of the game. Participants in a FaceTime call are able share live app experiences with other users through the SharePlay feature. This only is available for apps that add specific logic to support SharePlay. As this uses a feature specific to Apple's ecosystem, it allows for great synergy between frameworks and devices, showcasing an advantage to use the Apple Vision Pro. Integrating this feature with visionOS and TabletopKit allows the game to be shared between players, with actions and game state being synchronized. One of the main advantages of using this feature in this particular game is that we can access each person's spatial persona, allowing for players to see each other as 3D models while playing the game. This enhances the immersion as users can talk and see each other, as well as their gestures during gameplay.

In order to integrate SharePlay into the project, the *GroupActivities* framework was used to provide the app with SharePlay features. Firstly, a custom class was created that implements the *GroupActivity* protocol. This allows the app to advertise its activities to the other FaceTime users. In this class, the group activity's metadata was defined and configured. Then, like mentioned in the designs presented in Chapter 3, a *MultiplayerManager* was implemented and is responsible to coordinate the *TabletopGame* and the group activity session so that game state and interactions are correctly synchronized across all participants. TabletopKit's *coordinateWithSession* function was used for this coordination. Logic was also added to instantiate the *MultiplayerManager* and add the action to the toolbar to start a group activity via the *Activity().activate()* function. Lastly, the app's entitlements files needed to be updated so that it supports group activities. This is a system file used to define the app's service or technologies used so that the operating system grants it the right to use them. In this case, the *com.apple.developer.group-session* key was set as true.

The scenario where a player plays with friends starts with all participants being a part of a FaceTime call. The game can be opened beforehand or after starting the call. Once the user is in a FaceTime call, a new SharePlay button will appear in the UI (below the toolbar) displaying that the game is not being shared yet. To start a multiplayer session, the player needs to tap on the SharePlay button placed on the game's toolbar. This starts a group activity with spatial participants of FaceTime, opening the game on their device with the game state and interactions being synchronized. Once each user accepts the prompt from FaceTime to start playing the game, they will be assigned a seat in the game. As we can see in Figure 4.14, each participant shows their spatial persona and is positioned in their correct seat, simulating the experience of being in the same room and playing together.

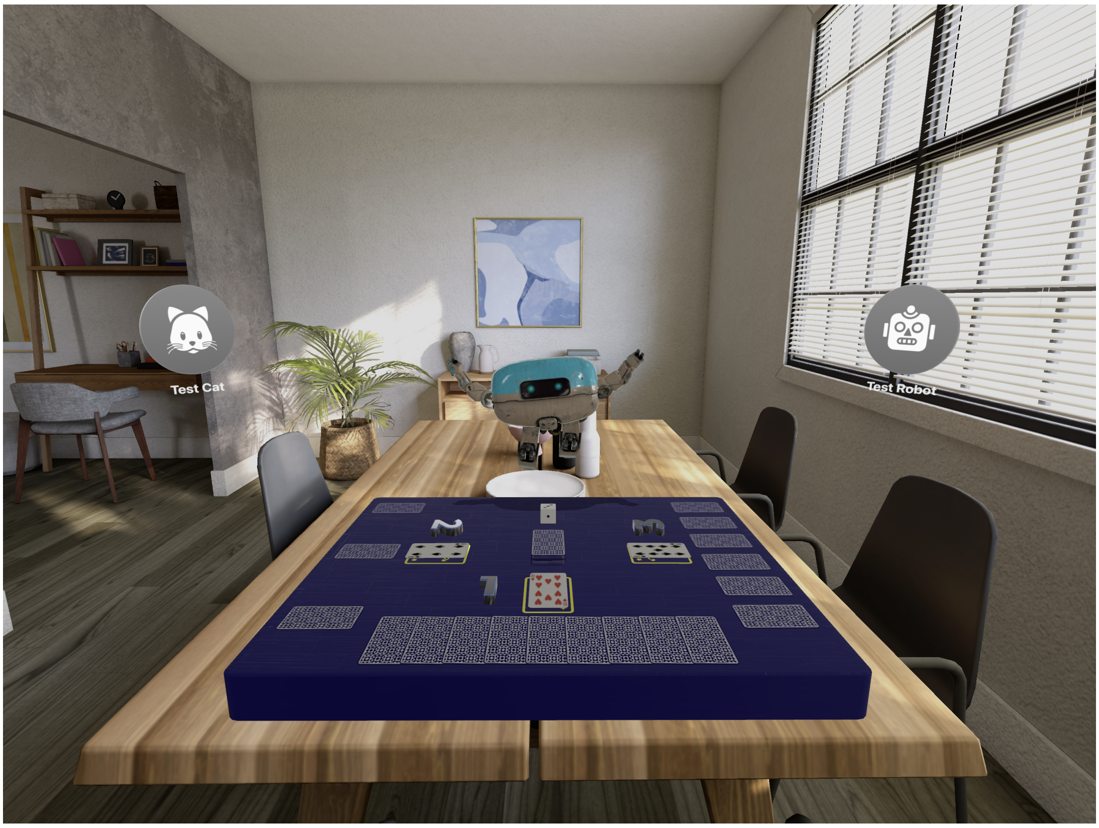


Figure 4.14: Multiplayer Using SharePlay

One of the limitations of the Apple Vision Pro is that real spatial personas are not available on the simulator. Since the device was not yet launched in Portugal, this thesis needed to rely on a simulator to test the application. Thus, the spatial personas from mocked test users in the simulator display a mocked version of a cat and robot emojis. In the real world with real devices, this prototype would be fully functional to use Apple's spatial personas for each user in a similar manner as shown in Figure 4.15 (Stein 2024).

Figure 4.15: Spatial Personas in a Real Device (Stein 2024)

When an activity is successfully started, the SharePlay button turns green displaying to the user that the game is in a multiplayer state and is being shared, as seen in Figure 4.16. When the user taps the spatial persona of any participant of the FaceTime call, the FaceTime menu hovers over their persona, as we can see in Figure 4.17. The same figure also shows that the *AceR* application is now being shared. This information appears to all participants in the call. Other apps features like the rule set can also be shared via SharePlay with other FaceTime participants. This is a great way to allow new players to go over the rules together with their friends.

Figure 4.16: Multiplayer: Active Session

Figure 4.17: Multiplayer: Sharing AceR

This FaceTime integration is great as it provides a multiplayer aspect that takes advantages of Apple's own ecosystem and tools, such as the spatial personas. All participants of the FaceTime call need to be spatial users (use an Apple Vision Pro) in order to be able to share the game, as the application was developed for visionOS. An advantage of using the ecosystem's tools is that, if an iOS or iPadOS application is developed in the future, iPhone and iPad users would be able to shared the same game with spatial users in the same FaceTime call.

Another limitation of the unavailability of using the simulator, is that we are limited to mocked FaceTime calls with mocked participants. In order to test a real FaceTime call with multiple real users and points of view, multiple physical Apple Vision Pro devices are needed. This limited the testing of this feature, especially the game state synchronization between multiplayer users. Some approaches were tested in order to mitigate this constraint. For example, SharePlayMock (X. Chen 2024) is a package that mocks the FaceTime call and uses a local Java server to allow multiple simulator instances to connect to it. Although it was promising, this package was not compatible with TabletopKit and its GroupActivities synchronization methods.

4.2.26 UC26: Configure the SharePlay session

Players are able to configure some SharePlay related settings to allow for some customization. As we can see in Figure 4.18, these appear when the user taps the SharePlay button, located under the toolbar. This will expand into a view that allows the user to define if they want the SharePlay content to be automatically shared in their FaceTime call.

4.2.27 UC27: Spectate a multiplayer game

The game only has three seats available but a FaceTime call can have more users in it. If users accept the game being shared through SharePlay, they will attempt to take a seat in the game using the *claimAnySeat* function. If all seats are full, users will be observers in the game, allowing friends to be together event if the group has more people than supported by the game. This can be seen in Figure 4.16, where the robot test user's spatial persona is not seated in the table but rather spectates the game.

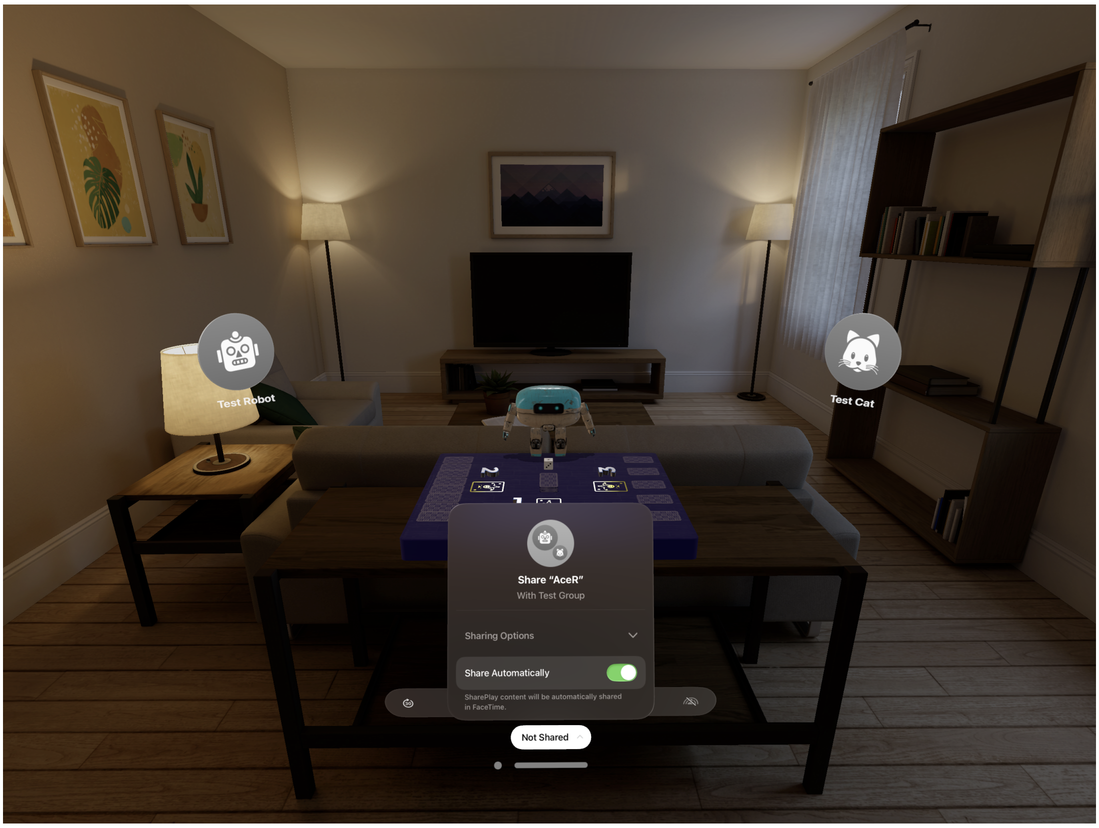


Figure 4.18: Multiplayer: Options

4.2.28 UC28: Invite other people to a multiplayer game

FaceTime also allows for built-in player invites and sharing. In the FaceTime menu, users can open a window that allows users to see what application is being shared, which participants are on the call, share the FaceTime call link and access messages (Figure 4.19). When tapping the *"Add People"* button, users can then invite people from their contacts to enter the FaceTime call and play *AceR* via multiplayer. As shown in Figure 4.20, users can quickly input their friend's name from the contacts and quickly add them to the game. This integration with FaceTime and other visionOS features (such as contacts and messages), allows for a multiplayer game where players can invite more users to join them. From an implementation perspective, it allows games to have a multiplayer system that is built on the operating system's first party APIs, allowing for great synergy.

Figure 4.19: Multiplayer: Share Link

Figure 4.20: Multiplayer: Invite Friends

4.2.29 UC29: End the SharePlay session

In order to end the multiplayer session, the SharePlay button needs to be tapped. Then, the option to stop sharing will be presented to the user, as seen in Figure 4.21. Ending the FaceTime call also stops sharing the game. Game state is preserved if the user wants to continue play alone and it can be restarted by tapping the reset button on the toolbar.

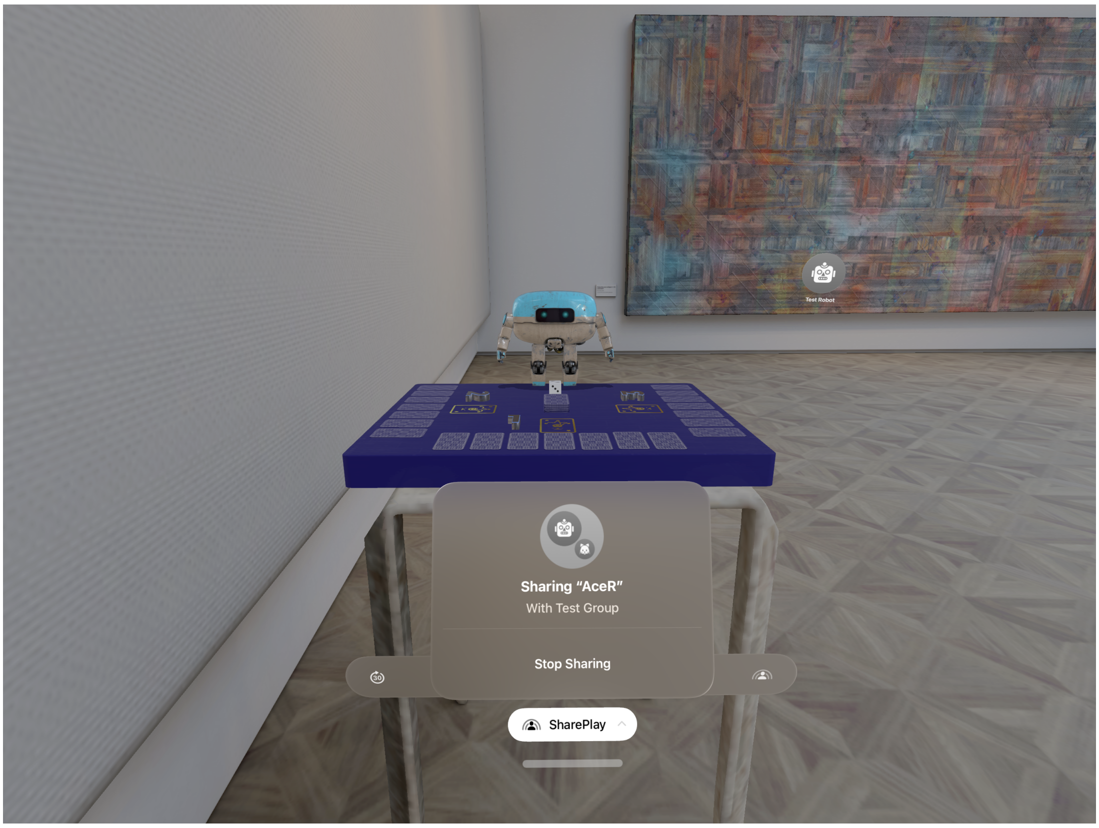


Figure 4.21: Multiplayer: End Session

4.3 Additional Benefits

This section will present some additional built-in benefits of a visionOS application that the prototype can offer its users. These include multitasking and anchoring.

4.3.1 Multitasking

One huge benefit of the game being developed for visionOS is that it allows multitasking. Players can play the game while having other applications opened. For example, as seen in Figure 4.22, users can watch a video while playing the game. The operating system even dims the user's surrounding to fit the atmosphere, while allowing the game to continue to be played. This also applies to watching a movie, listening to music, watching TV, etc. Players can even browse the internet while playing. This multitasking can be done in single and multiplayer modes, with the latter allowing for players to share their apps (like Safari) to other FaceTime participants.

4.3.2 Anchoring

On the other hand, the board can be moved around and placed wherever the player decides. As seen in Figure 4.23, the game will stay anchored where it was placed, allowing the player to move around without it affecting the game's position. For example, the player might place the game on a living room table, grab a glass of water for the kitchen while waiting for his friends to play. The board, all its equipment and the friend's personas will stay anchored at the table.

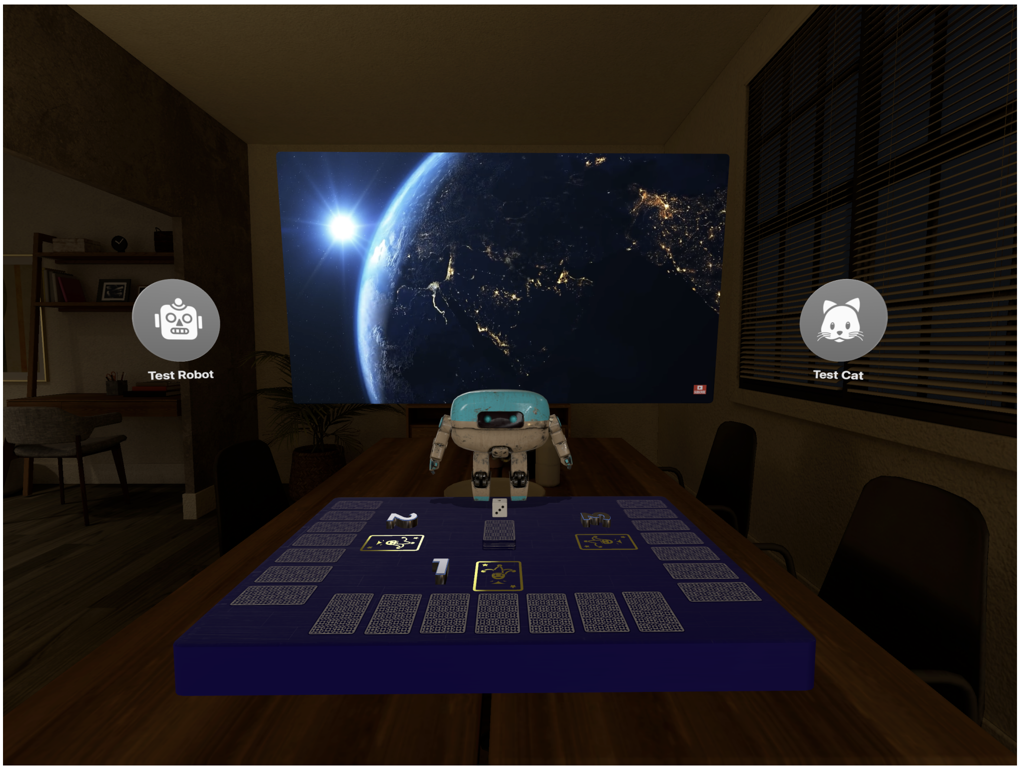


Figure 4.22: Multitasking: Watching a Movie

Figure 4.23: Board Anchored While the Player is Moving

4.4 Assets

As mentioned throughout Section 4.2, multiple third party assets throughout the game's implementation. These included 3D models for each card (Yanez 2018), for the die (Apple Inc. 2024c), the board (Baker 2023), the play cards (Buradkar 2024), the numbers on the board (Jihambru 2022a,b,c), as well as the robot and its animations (Apple Inc. 2024c). Moreover, many sound effects were also throughout the game, including sounds for card interactions (Splashdust 2022), die roll (AbdrTar 2022), game start (xxxTruthxxx 2022), round win (chieuk 2024), round draw (myfox14 2021), game reset (TuomasData 2023), player elimination (Nighteller 2022) and game winner (uss015dykrt 2023). All these assets have licenses that allow them to be used in this thesis.

4.5 Summary

In sum, this section presented the development process for the AceR visionOS application where users can play a tabletop card game that leverages visionOS and its unique capabilities to explore the platform. From creating and setting up of the project to the development of all use cases, the game allows players to play with their friends using its FaceTime integration, as well as in single player mode. Other features include interaction with cards and the die, drawing rounds, fetching from the stack, winning the game, the robot reacting to a round outcome, inviting players, among others. Lastly, the platform itself also adds to the game experience via multitasking and anchoring.

Chapter 5

Results and Findings

With the research into the existing literature, the prototype analysis, design and development concluded, results can be analysed and some findings can extracted.

In this chapter, the author will present the platform's limitations that were found during the development of the prototype. Following that, the prototype vision was compared to what the technology allowed to be implemented. Next, semi-structured expert interviews and their process will be detailed. The expert's insights about the platform and AceR prototype will be presented as well. Lastly, with all information gathered in this study, the research question and sub-questions will be answered.

5.1 Challenges and Limitations

This section will present the platform's limitations found by the author while developing the prototype of the AceR game for visionOS.

5.1.1 Hardware and Simulator

As mentioned previously, the Apple Vision Pro is not currently available in Portugal. This proved to be a limitation for this thesis as it wasn't possible to get a physical device to test the platform and the prototype. Development was done and tested using Xcode's Apple Vision Pro simulator, which simulates an Apple Vision Pro device in macOS. This simulator offers most of the device's features but lacks the immersion of a real device for testing as it presents the user's vision in the computer's flat screen. Furthermore, all environments displayed in the simulator are digital, this meant the immersion users would feel when seeing their own surroundings wasn't present on the simulator.

Since a simulator was used, user testing was not possible for the prototype so other methods (such as interviews) were used to gather feedback from the prototype. Lastly, one limitation of the simulator was related with multiplayer testing. The GroupActivities framework and FaceTime APIs are quite limited on a simulator. Developers are able to simulate Facetime calls with various number of participants, choosing if they are spatial users or not. Unfortunately, there is no way for developers to see the point of view of the other participants in the call, or even controller them. This limited the testing of the multiplayer component of the game as it wasn't possible to control the other participants in the FaceTime call or join a call with other simulator instances. In order to mitigate that, some research was done and multiple alternative approaches were tested. One approach, *SharePlayMock* (X. Chen 2024), mocked the GroupActivity and would use a local Java server for communication between multiple simulator instances. Although this approach was promising, it had some incompatibilities with TabletopKit and how it integrates with the GroupActivities framework. Since the simulator's mocked FaceTime calls used mocked users, the participants show without a real 3D spatial persona. Although these limitations were in place, the prototype was developed so that it would work on real Apple Vision Pro devices, allowing for real multiplayer via FaceTime calls that show each person as a real spatial persona as intended.

5.1.2 Reality Composer Pro

Some limitations were also found with Reality Composer Pro and how assets are imported to the project. For example, the deck asset (Yanez 2018) had all card assets in the same file. The visionOS application required that each card had their own separate asset for loading and movement. After converting the deck asset from their original format (*FBX*) into a supported format (*USDZ*) and importing it into *Reality Composer Pro*, a limitation was found. While *FBX* is an industry standard, *USDZ* is a recent format.

As splitting the assets into separate ones was not possible using this tool, a third party tool needed to be used. *Blender* was used to load the original asset, extract each card and save it as their own separate *USDZ* asset to be imported into *Reality Composer Pro*. Although this task was not complex, it was time consuming and offered some friction as another tool needed to be used to solve a limitation of Apple's first party tool.

5.1.3 Lack of Community Support

The author found the first party documentation that Apple provides about visionOS to be very complete and detailed. With sample projects to explore and video tutorials that go over the platform's features and code. However, since the Apple Vision Pro is a new and emerging platform, the community aspect is scarce. There is a lack of real world examples of third party developers discussing code, implementation details and bugs encountered. Sometimes in more mature platforms, when facing an issue, developers can rely on testimonies of other developers that had the same issue to get over the hurdle or at least give a head start to allow the developer to reach a solution. In this case, the developer is forced to have to solve their own issues without any previous knowledge from the community. On the other hand, older platforms may have third party frameworks developed that offer solutions for certain use cases or limitations. In the case of this platform, developers need to rely on first part frameworks from Apple as there is a scarcity of third party ones. As mentioned before, *SharePlayMock* was one third party framework explored during the development of the prototype. This framework tried to solve a limitation with the Xcode simulator but unfortunately it was not compatible with TabletopKit in this case.

5.2 Prototype Vision vs Technology Readiness

This particular section reflects on the original prototype vision for the prototype (presented in Chapter 3) and what the technology currently supports and allows developers to implement. Initially, the prototype concept envisioned an interactive tabletop card game that used the Apple Vision Pro's eye and hand tracking features to immerse players in multiplayer experience. This concept defined some game rules and other game elements, such as card and immersive 3D environments.

This vision was, for the most part, able to be implemented and tested in the final version of the AceR visionOS application. The game allows users to interact seamlessly with cards using their vision and hand gestures, placing them in a desired location. Sound effects are spatial and play from a specific element's position spatially to immerse users in the game. An option to play the game with a fully virtual environment was also developed, allowing users to choose between AR and VR modes. All game rules were implemented using the TabletopKit framework, using RealityKit and ARKit to accurately render card and other game elements' assets and add them to the player's physical environment in a believable manner.

However, as mentioned before, there were some notable constraints and technical limitations:

- **Hardware availability and simulator restrictions:** due to the lack of the device's commercial availability in Portugal, all testing and implementation were limited to the simulator. This leads to some restrictions regarding testing (e.g. eye and hand tracking are done via mouse click, immersion on a flat laptop screen instead of a real device, etc.). As a physical device was not available, user testing was not feasible.
- **SharePlay constraints:** although the GroupActivities framework was fully integrated and the multiplayer aspect of the game was implemented with game state synchronization and spatial persona support, these features could only be partially tested using a simulator. Multiple physical devices are required in order to fully test the multiplayer part of the prototype and see the other player's perspectives.
- **Reality Composer Pro limitations:** although not a blocker for any feature, the previously mentioned workarounds for some limitation of the RealityKit tool slowed the development process, adding some friction to it.

In summary, although the platform's state currently allowed for every proposed feature and game element in the original vision for the prototype to be delivered, some limitations and constraints lead to the impossibility of testing some features to their fullest extent.

5.3 Expert Interviews

Due to the unavailability of physical Apple Vision Pro devices in the author's country, user testing of the prototype was not feasible. Instead, semi-structured expert interviews were conducted in order to gather insights about the Apple Vision Pro, visionOS and the developed prototype from industry experts in game development and XR.

5.3.1 Methodology

Experts were identified and selected from the industry to take part in this process. Candidates were chosen based on their knowledge in XR development, game development, or both. Initially, seven candidates were approached and showcased interested in being interviewed for this study. Sadly, due to scheduling conflicts only three were able to take part in the interviews. Apart from Expert C, experts were chosen from different organizations, outside Mindera or ISEP in order to minimize bias. On the other hand, Expert C is from Mindera but was not involved in this project. Firstly, these experts were invited formally via an email that described the purpose and structure. The thesis' objectives were not describe in order to not create any confirmation bias. A consent form was also sent along with the email, describing the voluntary nature of the participation, data handling procedures and confidentiality assurances. This form was hosted using Google Forms. Lastly, the *Doodle* platform was used to help with scheduling all interviews with their interviewee's preferred schedules. The email, consent form and scheduling are available in full in Appendix B.

Interviews were conducted remotely, using the Microsoft Teams video meeting platform. The platform's recording and transcription features were used to allow the content of each interview to be used in this thesis. The transcription feature used the platform's artificial intelligence to transcribe the meeting. All participants were made aware and accepted this recording beforehand in the consent form. Many platforms were tested before the interview process started (Google Meet, Zoom, etc.) but Microsoft Teams had the more accurate recording and transcription systems, allowed for transcriptions in Portuguese (Portugal) and allowed to share audio from the prototype, which was needed in order to showcase its sound effects.

Regarding the interviews' structure, it started with a brief introduction of the interview itself. Afterwards, a live demo of the AceR prototype was presented to each participant. As there was no other alternative, the demo was running on a simulator. During each demo, experts were encouraged to ask questions and comment in real-time, allowing for a dynamic demo that was shaped by the expert's queries. A set of four prepared interview questions were common in all interviews. These were used to guide the discussion to cover topics relevant to this thesis and its research questions:

1. Do you think the XR implementation of this game offers unique value compared to a physical version of this game?
2. Do you feel this prototype effectively showcases the Apple Vision Pro is capable of real world game development?
3. Based on this prototype, do you feel the unique features of Apple Vision Pro and the Apple ecosystem (e.g. SharePlay) offer a great opportunity for game development compared to other XR platforms?
4. What opportunities and challenges do you see in designing and implementing games for visionOS?

Depending on each interview, other flexible questions were also posed to each expert. After each interview, a questionnaire was sent to each participant via Google Forms. It requested experts to assess the maturity of the Apple Vision Pro and visionOS for game development using a Technology Readiness Level (TRL) scale. A TRL scale is a metric used to assess the maturity of a particular technology. Although it was originally developed by NASA to evaluate the readiness of technologies related with space, it has evolved to be adopted in various fields and industries including the European Commission (European Commission 2012). There are nine levels ranging from basic research (TRL1) to systems deployed and proven in operational environments (TRL9), as defined in European Commission 2014:

- TRL1: Basic principles observed;
- TRL2: Technology concept formulated;
- TRL3: Experimental proof of concept;
- TRL4: Technology validated in lab;
- TRL5: Technology validated in relevant environment (industrially relevant environment in the case of key enabling technologies);

- TRL6: Technology demonstrated in relevant environment (industrially relevant environment in the case of key enabling technologies);
- TRL7: System prototype demonstration in operational environment;
- TRL8: System complete and qualified;
- TRL9: Actual system proven in operational environment (competitive manufacturing in the case of key enabling technologies; or in space).

Moreover, the post-interview questionnaire also questioned the expert's opinion as to what is stopping the platform from reaching the next level. Lastly, the expert had the opportunity to add any additional comments about the interview, the prototype or the technology. The full questionnaire is available in Appendix B.

5.3.2 Insights

This subsection will detail the insights gathered from each expert interview. Although these insights will be summarised, the raw transcripts of each interview are available in Appendix C.

Expert A

The first interviewee is an expert in game design and development working in the gaming industry. During the demo phase of the interview the expert highlighted the game's concept and implementation as positive aspects. Moreover, the interviewee mentioned this was an excellent investigation into this new emerging technology while exploring it to create a great product. Some aspects of the game were praised as well, such as the robot that reacts to the round result and plays sound effects. On the other hand, some game design improvements were suggested. These included some minor rule adjustments to make the game less casual, as well as having an extra deck so that the player's hand is more scalable and improves the user experience.

Regarding the first question, it is the expert's opinion that the XR implementation of the game is radically different compared to a physical version of the same game.

As for the second question, the interviewee believes that this prototype effectively showcases the platform is capable of real world game development. Some concerns were raised with the market, for example, regarding the number of users and their adoption. Even with these concerns in mind, Expert A thinks the APIs used in this prototype and the Apple Vision Pro itself is ready for development of real world products in the gaming industry, adding that even this prototype can grow to be one.

Thirdly, the specialist pointed to the eye tracking as a great feature of the platform as it allows users to quickly and effortlessly interact with the platform and the game. The multitasking was also highlighted by the interviewee as a unique feature since players can be playing and watching videos or browsing the internet. Lastly, the expert commented that although the immersive environments toggle feature was well implemented and showcases a great feature for the prototype, it was Expert A's concern that users might only try to use that feature once as a novelty.

Regarding question four, Expert A mentioned that there were many opportunities in designing and implementing games for this new and emerging platform, but there are some challenges as well. Firstly, the specialist raised an opportunity and future improvement specifically related to the showcased prototype. This was related to the robot and how it could be used to improve player engagements through unlockables or microtransactions. For example, the player can earn new robot assets, accessories or even new models for winning a determined amount of games or inviting a certain number of friends to play the game. On the other hand, the game can also sell new models or accessories for the robot so that it has a new revenue stream. Both these approaches can drive player engagement as players would be driven to play the game to unlock rewards and these models could be displayed to other players, creating a way for players to showcase their achievements or purchases. Moreover, Expert A also mentioned that the platform may be more suited for games that are not too intricate, with some real world examples given. For example, a simpler *Ticket to Ride* would be a suggestion from this expert, which is a turn-based tabletop game related to trains with a focus on strategy. There are some opportunities as well related with adoption in markets that are more mobile-first from the specialist's point of view, with China being an example.

On the other hand, there are some challenges that were raised by the interviewee. There is risk for studios in developing and publishing games for this platform and the expert wouldn't recommend game studios to focus a whole project on this platform yet. This is due to the fact that studios can publish a great product but low player adoption might not make it worth it for studios. In the expert's opinion, studios must aim to achieve a numbered player base with a monetization strategy that does not drive away said player base and it will be a challenge for studios to find that balance.

After the interview, Expert A filled the post-interview questionnaire and classified the Apple Vision Pro in the TRL scale with **TRL7 (system prototype demonstration in operational environment)**. When asked to comment on what is stopping the platform from reaching the next level the expert said the following:

> *"I'd need to verify a number of projects using that technology thrive in the video games market, and I'd need to assess the development costs."*
>
> — Expert A

Lastly, Expert A said the following when asked for additional comments about the interview, the prototype or the technology:

> *"The demo was very solid. The presenter showcased a deep understanding of the subject matter, effectively developing a wide array of features that highlighted both his technical expertise and the full capabilities of the technology."*
>
> — Expert A

Expert B

Expert B is also a game development expert that currently works in a video game software development company. The interviewee praised the prototype's concept as interesting with a good implementation and end result that showcased well the Apple vision Pro's functionalities.

Regarding the question one, the expert believes without a doubt that the XR implementation of this concept offers unique value compared to the physical version of it. Firstly, the XR implementation does not require any kind of setup. In the physical version of this concept game, players would need to grab a deck of cards, remove the cards not used (kings, jacks and jokers), grab a die, shuffle the cards manually, give each player seven random cards, lay them on the table and placing the remaining in the middle before starting the game. Although this concept is a simple one with just a few types of game equipment, more complex games can have longer setup times (e.g. 30 or 40 minutes). An XR version does this automatically in an instant, which helps to reduce game duration and allows for players to play more games in the same time frame. The same applies to the game's clean-up. In the XR implementation clean-up is not required, contrasting with the physical version which requires players to clean-up and store all game equipment after a game ends. Another benefit of the XR implementation is that the game itself handles the rule verification. In a physical game, players need to be aware of the rules and be constantly monitoring every move to enforce the rules to ensure no cheating is happening. In the XR version, players can focus on their own gameplay as the application itself handles the rule enforcement. With this in mind, the expert enjoyed that the prototype allowed for players to have their own house rules if they pleased as this offers players freedom to choose between official and home-made rules. Another difference stated by the expert was that the immersive environment would not be available in a physical game, as players are restricted to their own environment. Furthermore, Expert B also stated that the robot companion would not be possible in the physical version of the game, which removes some of the game's flair as it reacted to every round win or draw. Finally, the specialist also commented that a physical implementation of the concept also would not have sound effects.

Secondly, the interviewee stated that the prototype showcased in the interview and developed for this project effectively demonstrates that the platform is suitable for the development of real world video games. Expert B also stated that the platform has great potential in this field.

Next, regarding the third question, the expert highlighted the platform's fluidity between systems and applications as the standout unique feature of the Apple Vision Pro. Additional, there was high praise that the system allows users to play the game while doing other activities (such as watching videos or browsing the internet). Expert B stated that this level of multitasking is great for this type of games.

Moreover, the expert stated that there are a great number of opportunities in the design and implementation of video games for visionOS. The XR market is ever growing and many devices are released but the specialist claims that no device has yet to be crowned an undisputed leader. In the expert's opinion, the Apple Vision Pro can sweep the market and become a leader. If that happens, early adopters will have a huge advantage and provides a great opportunity. Expert B also suggested some opportunities for the immersive environments that the prototype showcased. The game could have a mixed environment that would add some elements to the player's surroundings for a more augmented reality experience. The expert gave a medieval game as an example where castle walls, swords or other medieval assets would be added to the player's surroundings to improve immersion. This can be an opportunity for other games developed for the platform in Expert B's opinion.

On the other hand, some challenges were also raised. Firstly, regarding the device availability and how that impacts development and testing of features, such as the multiplayer. Additionally, the expert mentioned that it can be a challenge to avoid an uncanny valley feeling where players feel unsettled and discomfort when virtual characters look almost human but not quite. This platform places 3D models of real people in their physical environment which, if not implemented correctly can lead to an uncanny valley feeling. The same applies to multiplayer and the game assets being placed on real world elements (e.g. a real table with digital cards). Expert B also pointed that there are always challenges with the multiplayer implementation of game and a platform like the Apple Vision Pro will face similar challenges, particularly since it's a new one.

After the interview ended, the expert defined the Apple Vision Pro and visionOS platforms with a **TRL8 (system complete and qualified)** in the TRL scale. Additionally, Expert B stated the following reason that is stopping it from reaching the next level:

> *"From what I could see it still needs some improvements to avoid some clunkiness in it's current state."*
>
> — Expert B

The expert left two additional comments, one related to the technology itself and another with the prototype. About the Apple Vision Pro and visionOS, the expert had the following comment:

> *"Obviously promising, even at it's current state could be a huge success. I believe it's issues are more tied with hardware (glasses still being unwieldy) and affordability."*
>
> — Expert B

Lastly, the interviewee had the following statement regarding the presented prototype:

> *"Works as a prototype. It's a good idea to allow for a manual mode. Integration with other apps (apple music, browser, etc.) was a must and that was achieved. The visuals could use some work, the rules screen was very bland and the way the text was structured didn't make it enjoyable to read (even though there were few rules), the style of the board and cards don't match the visual style of the robot and ideally we would want these components to be harmonious. Working on the visuals for a prototype is not always needed but it could easily impress some 3rd parties if the idea is sharing it as a proof of concept (often they will perceive better visuals as better technology/implementation)."*
>
> — Expert B

Expert C

Expert C is a researcher in XR and game development. Although this interviewee works in Mindera, Expert C was not involved with this study until this interview took place. Regarding the first question, the expert noted that the XR version of this game has use cases where it makes sense over its physical version. One of this cases is when people can't share the same physical environment for a certain reason. For example, people could not be together during the COVID-19 pandemic, which made way for virtual tabletop games where people connected over a digital medium. Another example would be people that live far away from each other or met online. In cases like these, the expert states that the XR version of the prototype game would offer unique value, compared to the physical version. Furthermore, the interviewee compared the XR version to VR-only or a flat screen version (e.g. a computer or smartphone) of the same game. With this in mind, Expert C believes that the XR implementation can offer users value in an use case where some players share the same space and others do not. With spatial personas, it can make players feel more connected with the players that are not physically present in the room. Another example given was when a player wants to be aware of its surroundings while playing (e.g. looking after a small child). The expert also talked about families that have the habit of doing FaceTime calls with their distant relatives, stating that they could add playing a board game to that ritual as it makes for a great family activity. In sum, it has its value for certain use cases over physical, VR-only or flat screen versions of the same game.

When asked about the second question, the interviewee expressed that the presented prototype successfully showcased the Apple Vision Pro is ready from development of tabletop games specifically. Although the prototype was praised as an example for tabletop games, the expert still has some questions regarding other kind of games. For example, video games of other genres with more complex states, environments and interactions to manage. From what was seen by the prototype, it looked intuitive and great but the question still stands for other genres.

Although some concerns were raised for game development with Apple's closed nature, Expert C stated that markets like the USA where Apple products are dominant can be a great opportunity. In these markets people buy and use Apple products more than the competition so there could be an opportunity there if adoption for this platform happens. For example, FaceTime is used regularly for casual communication and this kind of platform can use that market to its advantage. The specialist praised the simulator as being more fluid and complete than some competing platforms with the multiplayer being able to be partially tested. On the other hand, the interviewee mentioned that when a game is being developed and published, it tries to target as many users as possible. The Apple Vision Pro's limited availability, cost and the development being specific for this platform hinders its users base compared to cross-platform or projects for platforms with wider adoption.

Fourthly, Expert C mentioned the limited availability and the need for a simulator as the main challenge for game development in the Apple Vision Pro. Although the simulator offers many features, the lack of physical devices in some countries hinders the development and can be a blocker. The expert also mentions that Apple's platforms sometimes bring some challenges to game development. Contrarily, if the development is happening in a place where the platform is widely available, the expert stated that the development itself seemed intuitive from the presentation, with many frameworks that have great synergy with each other and offer great opportunities (e.g. multiplayer via FaceTime). The interviewee mentioned that although the Apple offers many great frameworks for development, other platforms have some frameworks as well (e.g. Unreal Engine, Unity and Meta). Expert C also discussed limitations with asset editing with Reality Composer Pro, as well as the vast documentation from Apple with a lack of practical examples from other developers. This is common to most new platforms from the expert's point of view.

Lastly, the interviewer believes that the Apple Vision Pro meets the criteria for a **TRL4 (technology validated in lab)** in the TRL scale. Stating the following reason that stops it from reaching a higher level:

> *"As the Apple Vision Pro is still not available in most countries, it does not seem feasible for development beyond proof of concepts."*
>
> — Expert C

5.3.3 General sentiment from the expert interviews

Firstly, all three experts agree that the XR implementation of this game offers unique value compared to a physical version of this game. Experts mentioned the fact that the setup and clean-up are done automatically, the non-existence of human error because of rule validation, allowing users to play together with people that are not able to share the same physical space (live far away from each other, quarantine, etc.) and visual/sound effects as some of examples of that unique value, while keeping some usual advantages of physical games like house rules.

Although all experts stated that the prototype effectively showcased the Apple Vision Pro is capable of real world game development, one expert felt that was the case particularly for tabletop games with still some questions for other genres. One expert stated that the frameworks and APIs allowed for games with bigger scale and another that the platform has lots of potential for game development. There were also some questions regarding the market, number of users and platform adoption.

Moreover, experts praise the platform's multitasking (for example, watching a video and playing the game at the same time), eye tracking, its simulator and system fluidity as unique features when compared to other platforms. The platform's ability to seamlessly switch between augmented and virtual reality was also mentioned.

In terms of opportunities in designing and implementing games for visionOS, experts mentioned the Apple dominant USA market, mobile-first markets like China and the fact that no XR headset has been crowned the undisputed market leader, which leaves the possibility for this platform to sweep the market since it comes from a major brand in technology. If that happens, one expert states that Apple Vision Pro's early adopters have a huge advantage which is a great opportunity. On the other hand, some challenges were mentioned as well. The limited availability was the general consensus for the biggest challenge as it limits development and testing. A consequence of this restricted availability (and its also high cost) can be a limited platform adoption, which hinders the user base. An expert claims that this adoption concern can be a risk for game studios. Another expert stated that Apple's platforms sometimes have challenges in game development. Lastly, concerns were raised about the challenges of preventing an uncanny valley feeling in players, especially when relying on spatial personas in the user's real environment.

Lastly, we can see each expert's chosen TRL for the Apple Vision Pro in Figure 5.1. Expert C, being a researcher, considers the platform to be laboratory tested. Whereas Experts A and B, being game developers, believe it may be close to or at production readiness. Expert C gave a significantly lower score than the first two experts due to the limited availability of the platform in certain markets. Other experts mentioned affordability, the device's unwieldiness, as well as the need to verify more projects and the development costs as the main reasons stopping the platform from reaching higher levels. The figure also showcases the average of all experts interviewed in this study. In this case, the expert's average level in the TRL scale was just above **TRL6 (technology demonstrated in relevant environment)**. This means that the platform is sufficiently matured to enable the development of interactive XR experiences, like the AceR prototype showcased in this study. Its current state allows for innovation and experimental validation of game concepts, allowing meaningfully for its unique features to be leveraged in game design. However, this level also highlights some existing limitations with the system. Due to its limited availability in many regions, the Apple Vision Pro is yet to achieve widespread validation in real world scenarios and developers still face some challenges as mentioned before. Overall, experts believe the Apple Vision Pro and visionOS are great for creating prototypes and XR game development research but is yet to be fully validated for large-scale commercial development.

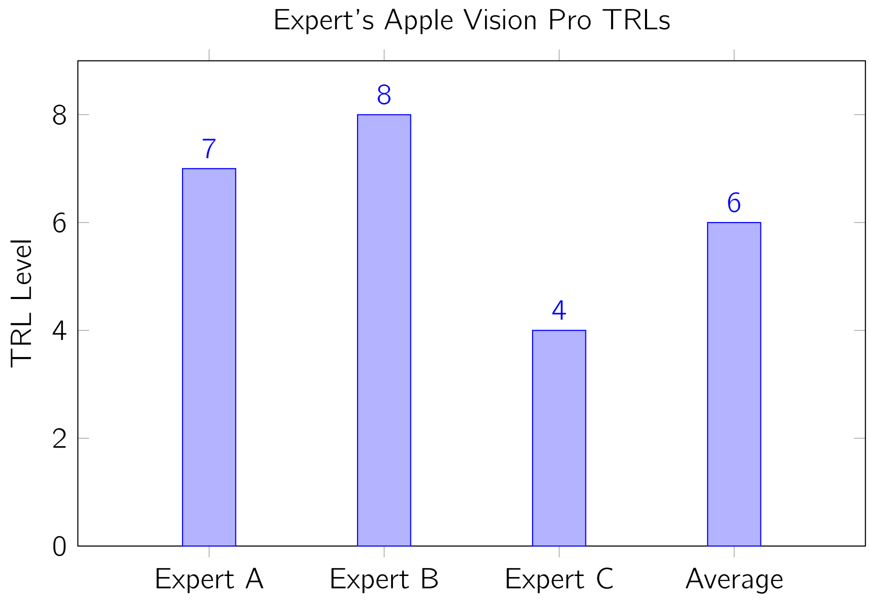


Figure 5.1: Expert's Technology Readiness Levels for the Apple Vision Pro

5.4 Answering Research Questions

This section aims to answer the central research question and all its sub-questions that were defined in Chapter 1, going over the Apple Vision Pro's capabilities, limitations, how these compare to other competitors, how can these leveraged in game designe, what technical challenges were found during the prototype's development and how can these be mitigated.

5.4.1 Sub-Questions

As mentioned before, this sub-section will tackle the response to all sub-questions using the information gathered during the research into the existing literature, analysis, design and development of the prototype by the author and insights gathered during expert reviews.

What are the capabilities and limitations of the Apple Vision Pro and visionOS?

In terms of capabilities, it was found in the existing literature that the platform uses its M2 and R1 chips (Hrycak, Lewakis, and Krüger 2024; T. Hu et al. 2024) to provide the user with state of the art eye-tracking and gesture recognition, while removing the need for a controller (Estalagem and Esteves 2024; Räthel et al. 2024). It has an immersive and high quality display that offers the user the equivalent of a 4K screen for each eye (3648 x 3144 pixels) (Zhao et al. 2024). The platform also has a visibility toggle that allows seamless transitions between a VR environment and an AR pass-through mode where the user is able to interact with digital objects in the physical world (Bondarenko et al. 2024; Chang et al. 2024). This ability to switch between AR and VR was explored during the implementation of the AceR prototype and highlighted by some experts during the interview process. Furthermore, the

Apple Vision Pro displays 3D human models called *spatial personas* that simulate the users' face and hand movements to display a digital lookalike in applications like FaceTime or Zoom (R. Cheng et al. 2024; Wang et al. 2024). With its integration with FaceTime, the prototype displayed each player's spatial personas to the others for a more immersive experience. On the other hand, the device also displays the user's visibility to the outside using the eye-sight feature. This showcases a digital rendering of the user's eyes in an outside display if the user is aware of its surroundings in an AR mode, while displaying a colour gradient when the user is fully immersed (Lyons 2024; Vergari et al. 2024). Moreover, visionOS uses Swift and SwiftUI to present the virtual components for the Apple Vision Pro, where concepts like spaces, volumes and windows are used. Apple's reputation is also a benefit for companies investing in Apple products as they have a history of profitable platforms in initially niche markets (Hrycak, Lewakis, and Krüger 2024). The platform's multitasking that allows users to interact with multiple applications at the same time was highlighted by experts as a great feature of the Apple Vision Pro as it was explored during the development of the prototype.

On the other hand, a general consensus from the expert interviews and existing research is that the biggest limiting factor for the platform is its limited availability. This is due to the platform only being available in specific countries. Even where the platform is available, the number of devices might be limited, as there was an estimated between 100,000 and 200,000 units available in the USA launch. Another limitation raised by experts and the existing literature was its high cost, which along with the previously stated limited availability can hinder the user base and platform adoption (Hrycak, Lewakis, and Krüger 2024). Likewise, its size and weight were also some concerns raised in the existing literature and by one of the specialists that was interviewed. The Apple Vision Pro also has a limited FOV, which can affect its immersion (Chang et al. 2024). Lastly, although Apple makes great effort to mitigate these, there are some privacy concerns and some attacks are still possible (Wang et al. 2024).

How does the Apple Vision Pro's capabilities compare to other XR platforms?

Comparing to other XR platforms, the Apple Vision Pro has a better display with more resolution than other headsets like the Meta Quest 3, Meta Quest Pro or HTC Vive XR Elite (T. Hu et al. 2024). As a VST HMD, it was found to have better outdoor low-light tracking accuracy than OST HMDs (L. Cheng, Schreiner, and Kunz 2024). It is also slightly more accepted in a public setting compared to other systems as it has the *eye-sight* feature where the users eyes can be displayed to other people. Other devices do not have this feature, thus making them slightly less accepted in these situations (Vergari et al. 2024). One expert also stated that the simulator is much more complete and fluid than some of its competitors. It has better integration with the Apple ecosystem than other platforms, which has some great advantages like the FaceTime integration for multiplayer explored in the prototype. Although the synergy with Apple systems is great, Apple products can also have a harder time to use third party tools. On the other hand, its limited FOV compared to other platforms is a downside that can affect the immersion of its users (T. Hu et al. 2024). It was also found to have worst visual deterioration, flickering and colour issues compared to OST solutions as it uses its cameras to display the physical world (L. Cheng, Schreiner, and Kunz 2024). Lastly, one expert mentioned that although Apple's first party frameworks offer great opportunities and very complete, other competitors like Unreal Engine, Unity and Meta also have some great frameworks as well.

What are the current trends and challenges in XR game development, and how might these apply to the Apple Vision Pro?

It was found in the existing research that multiplayer games focused on collaborative and social aspects are trending. As a consequence of that research, it was chosen to explore a tabletop game for the prototype that focused on these multiplayer and social aspects. These games present their own challenges as synchronization between users must be carefully implemented to not break immersion (Kriglstein et al. 2023). Moreover, gaming in the health field is also a growing market as users are in constant pursuit of a healthy lifestyle. With that in mind, the development of exercise XR games has risen, where games either focus on this aspect or promote it as a by-product of the game's main objective (Fu, Y. Hu, and Veronica Sundstedt 2024). Furthermore, asymmetrical social games that tackle cross reality are an important aspect to consider when developing XR games. Users may have different systems or environments, thus cross-play must be considered (Jung et al. 2023; Kitson et al. 2024). The Apple Vision Pro and its FaceTime integration allow for cross-platform integration with other Apple products like the iPhone and iPad. The AceR prototype's FaceTime integration would support for cross-play if iOS and iPad Operating System (iPadOS)'s applications were developed. Some design patterns were also studied and used during the development of the prototype. The singleton pattern ensures a determined class only has one instance, the observer pattern allows actions to be broadcasted to subscribers and the factory pattern allows for object creation logic to be hidden from clients as it is centralized (Qian, Deng, and S. Chen 2023). XR game development also introduces some challenges that can impact the user's experience. This include tracking precision, latency and realism. Limited availability must also be considered due to these systems being expensive and requiring robust hardware (Kriglstein et al. 2023). Some hardware challenges also include imperfect hand tracking and limited FOV (Sargolzaei, Rastogi, and Zaman 2024). An expert also highlighted the uncanny valley, the hardware still being unwieldy and affordability as some challenges in this space.

How can the unique features of the Apple Vision Pro be leveraged in game design?

The previously mentioned unique features offered by the Apple Vision Pro allow for novel approaches to game design. The players interact with the system without any controllers, using their vision, hand gestures and voice as input. This interaction model allows for games to be designed in a way that the interaction between players and game elements is highly intuitive and natural. For example, in the AceR prototype, players pick up cards and the die with their hands in the same way as they would do if it was a physical version of the game with real elements. This improves immersion compared to other platforms that rely on controllers for this kind of interaction. Gestures also provide an unique opportunity for game design as games can use custom gestures to perform certain action within the game. For example, one alternative game concept presented in Chapter 3 had players fight duels where each player would perform a custom gestures to attack the opponent, defend themselves or perform other actions within the game. This allows for games to be designed for the Apple Vision Pro with custom gestures in mind, which offers more opportunities than controllers with a set number of buttons. Additionally, this makes the games more immersive as well. Eye tracking also provides game design opportunities as players can interact with the game and its elements just by looking at them. This can be leveraged in the game's strategy, allowing players to see details or perform actions with just a subtle glance at a specific element. For example, the AceR prototype would highlight game elements (e.g. a card) when a player looked at them. This allowed players to quickly understand which card they were interacting with, before beginning that interaction.

The Apple Vision Pro's high definition pass-through allows for mixed reality games that frictionlessly place digital objects in the user's surroundings with breaking immersion. This can be seen in the prototype as users are able to place their board in their environment and play the game like it is present on the real world. On the other hand, the platform's visibility toggle allows for games to have sections in mixed reality with a mix of real and digital elements, as well as other sections in full virtual reality with a digital environment. Elements like portals to digital worlds from the real one are possible and available for games to leverage in their design. For example, the AceR prototype implemented a toggle between a pass-through mixed reality mode and a fully virtual one. Games can even take it a step further and have this affect the gameplay or its storyline.

Finally, the platform's great synergy with the Apple ecosystem allows for unique design strategies for games. For example, one of the approaches tested in this study leveraged the Apple's FaceTime video calling application to integrate it with the game, powering its multiplayer mode. This integration allowed for an invite system, game state synchronization, as well as audio and video interaction between remote players. Furthermore, the FaceTime/Shareplay integration allowed for the game to present spatial personas within the game so that players are playing with their remote friends and see them actually placed in the exact position they would if the game was being played in a physical form. This is great for immersion and a great feature that games can take advantage of. This seamless flow of players joining a FaceTime call with remote friends and quickly launching a tabletop game that all participants can play simultaneously, provides a unique opportunity for games in this platform. Application multitasking is also a great feature that games can leverage to their advantage. For example, games can be designed in a way that allows players to listen to music, browse the internet, watch a video or a movie. Experts highlighted the multitasking during their interviews as a great feature of the platform that can enhance the game's experience.

What are the technical challenges encountered during game development for visionOS, and how can they be mitigated?

Although the platform offers a great opportunity for game development with many unique features, there are also some technical challenges that were encountered while developing the AceR game prototype for visionOS. At the moment, the biggest challenge is the limited availability of physical devices for sale in some countries. This is the case in Portugal, where this study was conducted. It forces developers to rely on the simulator provided by Apple. Although the Apple Vision Pro simulator has many great features and allows to develop and test applications for visionOS, there are some restrictions that may affect game development. On one hand, the game's immersion and mixed reality is not properly tested as the application is running on a simulated environment and not on the tester's real world environment. These two experiences are not equal, which may lead to developers having a different experience from real users while developing and testing the game. This is evident in the interaction with the application as well, as physical devices rely on the user's eyesight and gestures and the simulator uses the computer's mouse. This leads to completely different ways of interaction with the platform and its applications, including the one being tested. Custom gestures are also hard to translate using a simulator. The simulator is also presented on the developer/tester's flat computer screen instead of being attached to the user's head and completely surround their eye-sight. On the other hand, there could also be bugs only found on physical devices that are not reproducible on the simulator, so there could be challenges regarding the product's quality.

Furthermore, as shown with this particular prototype, multiplayer games can be hard or even impossible to test on a simulator, depending on the multiplayer system and frameworks used. Although the integration with Facetime, SharePlay and the AceR prototype was tested, testing the multiplayer interaction from the other player's point of view was not possible in this case. There are some alternative that try to mitigate this for SharePlay, such as *SharePlayMock*, but they are not compatible with TabletopKit for example. Other alternatives include using their own multiplayer system with back-end support. This increases costs of development and maintenance, with the addition of losing the first party integration with Apple's ecosystem. For example, the spatial persona support would be affected.

Another simulator challenge is user testing. With limited availability for physical devices, testing the game with real users is not possible since testers will need to have a physical device in order to test the application. Each device is also more expensive than the norm, which can increase development costs. This also means that the user base can also be smaller, since this is an expensive platform with limited availability at the moment.

A technical challenge was also found regarding asset editing. This is related with Reality Composer Pro and its limitations. For example, during the prototype's development, another tool (Blender) was needed in order to perform certain editing tasks that Reality Composer Pro could not support. This showcase some limitations with the tool provided by Apple, especially in more complex projects.

Lastly, since this is a brand new platform with a new operating system, the documentation (mainly unofficial) can be scarce. Apple provides great documentation, many sample projects, videos and tutorials about visionOS but there are a limited number of real world examples of third party applications and games developed specifically for visionOS, compared to other platforms in other areas (e.g. mobile or desktop) or even in the XR space. This leads to experimentation on the developers part or attempts to adopt iOS or macOS paradigms since visionOS uses the same programming language and UI framework. With time and platform adoption, this challenge will be mitigated as more people develop for the platform.

5.4.2 What are the technical and practical challenges of using the Apple Vision Pro and visionOS for game development?

The findings of this study revealed that the Apple Vision Pro and its operating system represent an exciting and highly innovative new entry in the XR market from one of the biggest technology companies in the world. Although it presents developers with many opportunities, there are some technical and practical challenges that can hinder game development for the platform. From a practical standpoint, the limited availability of the hardware is its major challenge. As it is only distributed in certain countries in limited numbers, its market is limited. The platform's high cost also increases development costs which may restrict adoption from smaller or independent game studios. This might slow down the ecosystem's growth as well.

From a technical perspective, the limited availability also plays a factor. It restricts development in countries where the hardware is not available, which forces developers to use the simulator provided by Xcode. Although the simulator was praised by one expert for being more complete and fluid than other competitors, there are still some limitations compared to using a physical device for development and testing. One example, was testing of the multiplayer in the AceR prototype. Although the FaceTime integration (start multiplayer, invite, end session, etc.) was able to be tested with mocked users, the other user's point of view and interactions were not possible to test due to the usage of a simulator. There can also be bugs present on physical devices that are not reproducible in a simulator. User testing is also not possible using a simulator. On the other hand, the immaturity of the visionOS ecosystem may offer developers some friction when developing games. For example, the Reality Composer Pro is a great tool when developing for this platform but has some limitations with asset editing. During the prototype's development, Blender needed to be used to fulfil some of these limitations. Although Apple's documentation is vast and complete, there is a lack of third party real world discussion and examples. One expert said that this issue is not specific to the Apple Vision Pro and is common on every new platform, which improves with maturity and adoption. Still, this is something to keep in mind when considering the Apple Vision Pro for game development in its current state.

In order to evaluate its maturity, a TRL scale was used. As mentioned before, experts were asked to use this scale to represent the Apple Vision Pro based on their existing knowledge, the prototype and research presented during the interview process. On average, experts graded the platform with a **TRL6 (technology demonstrated in relevant environment)**. This means that experts believe the Apple Vision Pro is sufficiently matured for innovation and experimental validation of prototypes, but has yet to be validated for large-scale commercial projects due to some of its limitations (such as its limited availability). The author agrees with this level for the platform as the prototype was developed using a simulator as a consequence of its limited availability. The Apple Vision Pro's level can increase to TRL7 (System prototype demonstration in operational environment) in the author's opinion if the device is released and becomes available in more countries around the world. The same can be applied if Apple launches a new Apple Vision Pro model that runs on visionOS in more regions.

5.5 Summary

In sum, some limitations were found during the prototype. These included the need to use a simulator due to the limited availability, some limitations with Reality Composer Pro and lack of community support. These also affected the prototype vision compared to what the platform allowed to develop and test. Due to the simulator use, multiplayer testing was restricted to mocked users, interactions were tested via mouse and keyboard and user testing was not possible. As Reality Composer Pro had some restrictions, Blender needed to be used. Furthermore, semi-structured expert interviews were conducted where insights about the platform were gathered. In average, experts believe that the platform's maturity can be graded as a TRL6 (technology demonstrated in relevant environment) in the TRL scale. Lastly, all research questions were answered based on the information gathered from the state of the art, prototype analysis, design and development, as well as the insights gathered from expert interviews.

Chapter 6

Conclusion

Finally, in the last chapter of this document, a summary of the study's findings will be presented, followed by an analysis of the project's objectives, limitations and challenges, an outline of future work and, lastly, a final appreciation will be made.

6.1 Summary of Findings

In sum, this study explored the Apple Vision Pro and visionOS' capabilities such as its eye-tracking, gesture recognition, spatial personas, AR and VR toggle, eye-sight, high quality display and multitasking between applications. On the other hand, the platform's limitations were also explored, identifying limitations like its limited availability, high cost, size, weight, limited FOV, as well as privacy concerns and some attacks being possible. These capability were then compared to the platform's competitors in the XR market, such as the Meta Quest 3, Meta Quest Pro and the HTC Vive XR. It was found that the Apple Vision Pro has a better display, better outdoor low-light tracking accuracy, it is more accepted in a public space, it has better integration with the Apple ecosystem, its simulator is more complete and fluid. Contrarily, the platform has a much more limiting FOV, worst visual deterioration, flickering, colour issues, when compared to other competitors.

Current trends and challenges in XR game development were also explored, as well as how these might apply to the Apple Vision Pro. Some of these trends include multiplayer games that have a focus on social aspect and collaborative experiences. Other trends include health related games, asymmetrical games that tackle CR with cross-play and some design patterns (singleton, observer and factory). The hardware being unwieldy, uncanny valley, affordability, limited availability, latency, tracking precision, realism, imperfect hand tracking and limited FOV are some of the challenges with XR game development. The Apple Vision Pro's unique features can be leverage in game design to create innovative game experiences for players. The controller-free gesture interactions allow for games to be designed in a way that allow for natural and intuitive interactions between players and game elements. There are also opportunities for game design related with custom gestures, eye-tracking, high definition pass-through, visibility toggle, multitasking and the platform's great synergy with the Apple ecosystem.

A concept for a XR game that leveraged these capabilities was created, analysed and designed to be developed into a visionOS application. This game was a tabletop card game that allowed players to play in single player and multiplayer modes. The multiplayer component integrated Apple's FaceTime application and allowed for the game to shared while players were talking via voice chat and seeing each other's spatial personas. The FaceTime

SharePlay integration also allowed players to invite other players to play the game. The development of this game allowed the author to identify some technical challenges encountered during development. The biggest one was related to the platform's limited availability that forces developers to use the simulator in countries where it is not yet available for purchase. Although the simulator is better then most competitors, it still has some limitations like not allowing for user testing, not allowing developers to test the multiplayer with multiple real users, it does not allow to see the full spatial persona (only a mocked version) and bugs may only be present on real devices. This can be mitigated when the platform launches in more regions. Other challenges included limited asset editing with Reality Composer Pro and lack of real world third party examples. This last challenge can be attenuated with more people developing for the platform.

Lastly, experts were interviewed about the platform and the developed prototype to gather more insight about both. With all the research made, prototype analysis, design and development, as well as the expert's insights, the study evaluated the Apple Vision Pro and visionOS' maturity using the TRL scale with a TRL6 (technology demonstrated in relevant environment). This means that the platform is sufficiently matured for prototyping and innovation but still has some limitations due to its limited availability in order allow for large scale commercial projects. It can reach higher levels if this availability issue is solved by releasing the platform in more regions.

6.2 Objectives Achieved

As mentioned in the first chapter, this project aims to evaluate the feasibility of the Apple Vision Pro platform for game development. As a part of this study, a prototype game was created, designed, developed, tested and showcased to industry experts. In this section all objectives presented in Chapter 1 will be analysed.

- **Understand the capabilities and limitations of the Apple Vision Pro and visionOS for game development - Achieved (100%)**

The study started by analysing the existing literature to identify the platform's capabilities, limitations and how they compared to its competitors in the XR market. This was an extensive study that identified unique features that the Apple Vision Pro offers its users that later could be utilized during development. With this in mind, this objective was fully achieved.

- **Design an engaging game concept that leverages the platform's unique features - Achieved (100%)**

In Chapter 3, the AceR game's concept was created, detailed and a solution was designed for a visionOS application that leveraged the Apple Vision Pro's unique capabilities. The tabletop game utilized the platform's eye-tracking, controller-free hand gesture interactions, detailed frameworks and its fluid integration with the vast Apple ecosystem. As a part of the integration with the Apple ecosystem, the game was designed to work with Apple's FaceTime for its multiplayer, where players can see their friends' spatial personas, talk to their friends while playing or invite other friends. Thus, this objective can be considered to have been fully completed.

- **Develop a spatial computing/extended reality (XR) game for visionOS, utilizing SwiftUI, RealityKit and ARKit - Achieved (100%)**

This objective was fully achieved since the AceR game was developed as a prototype application for visionOS that uses SwiftUI, RealityKit and ARKit. This prototype fully implemented all the defined use cases for the game so that users have a plethora of features to enhance their user experience. Users can play regular or fast game modes, play the game in AR or VR modes, see the rules, interact with game equipment (cards and die), play in a single player experience or in a multiplayer mode via FaceTime where they can invite, talk to their friends and see their spatial personas.

- **Evaluate the feasibility of game development for the Apple Vision Pro and visionOS - Achieved (100%)**

Lastly, the results gathered from the *State of the Art* chapter, AceR concept, analysis, design and implementation were taken into account in order to evaluate the feasibility of game development for this platform. Furthermore, industry experts were consulted via semi-structured interviews where the prototype was showcased and questions were posed regarding the technology and its readiness for game development. A TRL scale was also used in order to define the Apple Vision Pro and visionOS' maturity. Bearing this in mind, this objective was also fully completed.

6.3 Limitations and Challenges

This study has some limitations and challenges. Firstly, although experts were consulted from different organizations and backgrounds, they were all based in Portugal. Experts from other regions may have provided different opinions and perspectives. For example, experts from regions where the Apple Vision Pro is available may have more hands-on experience with the device itself. Another limiting factor can be the number of interviews. Although initially seven experts were lined up to be interviewed, scheduling conflicts restricted the number to just three experts. A higher number of experts may have provided more insights about the platform or the prototype.

Furthermore, a prototype was developed for the study with specific genre (tabletop). A single prototype can only test its specific genre and mechanics, other genres of games may encounter different challenges when developing for this platform. The limited availability of the Apple Vision Pro caused the prototype to be developed and tested using a simulator, like mentioned throughout the document. Although the implementation was developed for physical device usage, using a simulator does not fully replicate the user experience. Interactions are performed via mouse and keyboard instead of gesture and eye-tracking, environments are digital instead of showcasing the user's real environment which changes the immersion and the multiplayer was not able to be tested from multiple points of view. The platform's limited availability also restricted the study in terms of user testing. If the physical device was available, the prototype could've been tested in user's hands and data collected from that process. Instead the study needed to rely on remote expert interviews, which limits the insights in terms of usability and engagement.

Even though Apple Vision Pro development is highly intuitive, the author still faced some challenges as it is a new platform that he did not use before and had to learn over the course of this study. Additionally, visionOS development was not only a novelty for the author but also for the wider community. Furthermore, during this study, vast research into

Apple's documentation and sample projects was made, as well many experimental projects were created to test and become familiar with the system. The scarcity of real world Apple Vision Pro developers and their discussions was a challenge as more mature platforms have an active community that has discussions regarding the platform itself or about bugs that were encountered with the software and its frameworks. Since this is an emerging platform, this community is not fully present yet which means that most if not all bugs that the author encountered during the development of the prototype had to be solved without relying on the existing community's findings.

6.4 Future Work

Firstly, an article related with this study is planned to be published in the future. Moreover, interviewing more experts or using other validation methods will provide a deeper comprehension of the platform's readiness for game development. One of these methods could be user testing of the prototype whenever the platform becomes available in Portugal or testing it in a country where it is already available. This would gather different feedback compared to the expert interviews in terms of usability, user experience, ergonomics and immersion.

Although this study explored the feasibility of game development for the Apple Vision Pro and visionOS, future research could extend this work. One possible direction is the exploration of other game genres and different interaction paradigms, including the alternative gesture duel game concept presented in Chapter 3 where custom gestures are explored. Implementing iOS and iPadOS versions of the AceR game is also another avenue of exploration that would tackle the asymmetric use of XR. Another possibility of future work would be investigating the feasibility of using cross-platform frameworks with the Apple Vision Pro and comparing them to the first party tools investigated in this project. Lastly, similar studies can be made for fields other than gaming. These different future work possibilities can strengthen the understanding of visionOS game development, as well as broaden the research into other areas.

6.5 Final Appreciation

This study was a pleasant experience that allowed the author to explore a new and emerging technology in fields (XR and game development) where he had great interested but did not have any experience before. Although most technologies used in this study were not lectured during the master's or bachelor's degrees, this project provided a great opportunity for the author to utilize the tools and processes learned throughout those and applied them to new technologies.

Moreover, the project achieved all its set objectives, identifying the platform capabilities and challenges, designing a game concept that leveraged those capabilities, developing a prototype of that game for visionOS and evaluating the feasibility of the platform based on that prototype while identifying its limitations and challenges.

Finally, all people involved with the project were pleasant to work with. Both the company's supervisor and the advisor were helpful during the duration of the study. Experts made themselves available for interviews and their insights were important for the study. Mindera as a company also provided with all the tools needed for the project to reach its completion. All participants, including the reader, allowed this project to be possible and reach its completion.

Bibliography

AbdrTar (2022). *Dice*. url: https://pixabay.com/sound-effects/dice-95077 (visited on 07/12/2025).

Apple Inc. (2019). *Creation tools for spatial apps*. url: https://developer.apple.com/augmented-reality/tools (visited on 05/14/2025).

– (2021a). *Group Activities*. url: https://developer.apple.com/documentation/GroupActivities (visited on 05/13/2025).

– (2021b). *Use SharePlay to watch, listen, and play together*. url: https://support.apple.com/guide/iphone/shareplay-watch-listen-play-iphb657eb791/ios (visited on 05/13/2025).

– (2022a). *Augmented Reality: More to explore with ARKit 6*. url: https://developer.apple.com/augmented-reality/arkit/ (visited on 12/17/2024).

– (2022b). *SF Symbols*. url: https://developer.apple.com/design/human-interface-guidelines/sf-symbols (visited on 07/05/2025).

– (2023a). *Construct an immersive environment for visionOS*. url: https://developer.apple.com/documentation/realitykit/construct-an-immersive-environment-for-visionos (visited on 07/13/2025).

– (2023b). *Human Interface Guidelines: App Icons*. url: https://developer.apple.com/design/human-interface-guidelines/app-icons (visited on 05/11/2025).

– (2023c). *Human Interface Guidelines: Gestures*. url: https://developer.apple.com/design/human-interface-guidelines/gestures (visited on 05/10/2025).

– (2023d). *Human Interface Guidelines: Toolbars*. url: https://developer.apple.com/design/human-interface-guidelines/toolbars (visited on 05/09/2025).

– (2023e). *WWDC2023: Design for spatial input*. url: https://developer.apple.com/videos/play/wwdc2023/10073/ (visited on 05/10/2025).

– (June 2024a). *Apple empowers developers and fuels innovation with new tools and resources*. url: https://www.apple.com/cm/newsroom/2024/06/apple-empowers-developers-and-fuels-innovation-with-new-tools-and-resources/ (visited on 05/10/2025).

– (2024b). *Augmented Reality: RealityKit*. url: https://developer.apple.com/augmented-reality/realitykit/ (visited on 12/18/2024).

– (2024c). *Creating tabletop games*. url: https://developer.apple.com/documentation/tabletopkit/tabletopkitsample (visited on 07/10/2025).

– (2024d). *TabletopKit*. url: https://developer.apple.com/documentation/tabletopkit (visited on 05/10/2025).

– (2024e). *Use FaceTime with your iPhone or iPad*. url: https://support.apple.com/en-us/105088 (visited on 05/13/2025).

Applied Visual Technology Inc. (Sept. 2020). *Cross Reality Confusion: Extended, Mixed, Cross — How to Differentiate?* url: https://www.avtsim.com/cross-reality-confusion-extended-mixed-cross-how-to-differentiate/ (visited on 12/04/2024).

Baker, Dominic (2023). *Chopping Board*. Licensed under Creative Commons Attribution 4.0 International (CC BY 4.0). url: https://sketchfab.com/3d-models/chopping-board-02efecfecf81437f865520de7ccfa714 (visited on 07/07/2025).

Balakrishnan, S. et al. (2021). "Interaction of Spatial Computing In Augmented Reality". In: *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)* 1, pp. 1900–1904. doi: 10.1109/ICACCS51430.2021.9442010.

"Blender Scripting for Creative Coding Projects" (2023). In: *SIGGRAPH Asia 2022 Courses*. SA '22. Daegu, Republic of Korea: Association for Computing Machinery. isbn: 9781450394741. doi: 10.1145/3550495.3558222. url: https://doi.org/10.1145/3550495.3558222.

Bondarenko, Volodymyr et al. (2024). "Motion sickness comparison of 2 XR approaches: camera+screen or hologram on transparent lens (Meta Quest 3 or Microsoft HoloLens 2)". In: *2024 International Electronics Symposium (IES)*, pp. 728–733. doi: 10.1109/IES63037.2024.10665757.

Buradkar, Saurabh (2024). *Playing Card Black Joker*. url: https://sketchfab.com/3d-models/playing-card-black-joker-15cf93fb1dac40a29490f34a51a09dc7 (visited on 07/10/2025).

Chang, Chiao-Ju et al. (2024). "Exploring Augmented Reality Interface Designs for Virtual Meetings in Real-world Walking Contexts". In: *Proceedings of the 2024 ACM Designing Interactive Systems Conference*. DIS '24. Copenhagen, Denmark: Association for Computing Machinery, pp. 391–408. isbn: 9798400705830. doi: 10.1145/3643834.3661538. url: https://doi.org/10.1145/3643834.3661538.

Chen, Xinyi (2024). "The Easiest Way to Test SharePlay on visionOS Apps". In: *Medium*. url: https://medium.com/@xinyichen0321/the-easiest-way-to-test-shareplay-on-visionos-apps-7bf8a1753d8e (visited on 07/18/2025).

Chen, Yunqiang et al. (2019). "An overview of augmented reality technology". In: *Journal of Physics: Conference Series* 1237. doi: 10.1088/1742-6596/1237/2/022082.

Cheng, Long, Michael Schreiner, and Andreas Kunz (2024). "Comparing Tracking Accuracy in Standalone MR-HMDs: Apple Vision Pro, Hololens 2, Meta Quest 3, and Pico 4 Pro". In: *Proceedings of the 30th ACM Symposium on Virtual Reality Software and Technology*. VRST '24. Trier, Germany: Association for Computing Machinery. isbn: 9798400705359. doi: 10.1145/3641825.3689518. url: https://doi.org/10.1145/3641825.3689518.

Cheng, Ruizhi (2024). "Towards Network-friendly and Privacy-preserving Immersive Computing". In: *Proceedings of the CoNEXT on Student Workshop 2024*. CoNEXT-SW '24. Los Angeles, CA, USA: Association for Computing Machinery, pp. 3–4. isbn: 9798400712555. doi: 10.1145/3694812.3699920. url: https://doi.org/10.1145/3694812.3699920.

Cheng, Ruizhi et al. (2024). "A First Look at Immersive Telepresence on Apple Vision Pro". In: *Proceedings of the 2024 ACM on Internet Measurement Conference*. IMC '24. Madrid, Spain: Association for Computing Machinery, pp. 555–562. isbn: 9798400705922. doi: 10.1145/3646547.3689006. url: https://doi.org/10.1145/3646547.3689006.

Chhajer, Arihant (2021). "Applications of Extended Reality (XR) for Creation of Real-and-Virtual Combined Environments and Human". In: *Technoarete Transactions on Internet of Things and Cloud Computing Research*. doi: 10.36647/0805014iotccr.

chieuk (2024). *coin*. url: https://pixabay.com/sound-effects/coin-257878 (visited on 07/20/2025).

Chitti, Eleonora, Riccardo Iervolino, and N. Alberto Borghese (2024). "Exploring AR Experience with Thermojelly: a Competitive AR Board-game with Tangible Interfaces". In: *Companion Proceedings of the 2024 Annual Symposium on Computer-Human Interaction* *in Play*. CHI PLAY Companion '24. Tampere, Finland: Association for Computing Machinery, pp. 37–42. isbn: 9798400706929. doi: 10.1145/3665463.3678809. url: https://doi.org/10.1145/3665463.3678809.

Commons, Creative (2013). *Attribution 4.0 International*. url: https://creativecommons.org/licenses/by/4.0/ (visited on 07/07/2025).

Cross, Jamie I. et al. (2022). "Using Extended Reality in Flight Simulators: A Literature Review". In: *IEEE Transactions on Visualization and Computer Graphics* 29, pp. 3961–3975. doi: 10.1109/TVCG.2022.3173921.

Deitel, P. and H. Deitel (2015). "Swift for Programmers". In.

Dong, Xiangyu (2023). "Simulation or Fake: Will Extended Reality Provide a More Vivid Learning Experience?" In: *ECE Official Conference Proceedings*. doi: 10.22492/issn.2188-1162.2023.117.

Eco, Brands' (2022). *Os benefícios da Mindera que dão que falar*. url: https://eco.sapo.pt/2022/07/05/os-beneficios-da-mindera-que-dao-que-falar (visited on 07/12/2025).

Eeles, Peter (2005). "Capturing architectural requirements". In: *IBM Rational developer works*.

Estalagem, Nuno and Augusto Esteves (2024). "Between Wearable and Spatial Computing: Exploring Four Interaction Techniques at the Intersection of Smartwatches and Head-mounted Displays". In: *Proceedings of the 2024 Symposium on Eye Tracking Research and Applications*. ETRA '24. Glasgow, United Kingdom: Association for Computing Machinery. isbn: 9798400706073. doi: 10.1145/3649902.3656365. url: https://doi.org/10.1145/3649902.3656365.

European Commission (2012). *About Technology Readiness Levels*. url: https://euraxess.ec.europa.eu/career-development/researchers/manual-scientific-entrepreneurship/major-steps/trl (visited on 07/23/2025).

– (2014). *Technology readiness levels (TRL); Extract from Part 19 - Commission Decision C(2014)4995*. url: https://euraxess.ec.europa.eu/career-development/researchers/manual-scientific-entrepreneurship/major-steps/trl.

– (2024a). *General Data Protection Regulation (GDPR)*. url: https://gdpr.eu/ (visited on 11/08/2024).

– (2024b). *Legal Framework for EU Data Protection*. url: https://commission.europa.eu/law/law-topic/data-protection/legal-framework-eu-data-protection_en (visited on 11/08/2024).

Farhadloo, Majid et al. (Sept. 2024). "Spatial Computing Opportunities in Biomedical Decision Support: The Atlas-EHR Vision". In: *ACM Trans. Spatial Algorithms Syst.* 10.3. issn: 2374-0353. doi: 10.1145/3679201. url: https://doi.org/10.1145/3679201.

Feld, Nico et al. (2024). "Perceptual Issues in Mixed Reality: A Developer-oriented Perspective on Video See-Through Head-Mounted Displays". In: *2024 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pp. 170–175. doi: 10.1109/ISMAR-Adjunct64951.2024.00044.

Fu, Yu, Yan Hu, and V. Sundstedt (2022). "A Systematic Literature Review of Virtual, Augmented, and Mixed Reality Game Applications in Healthcare". In: *ACM Transactions on Computing for Healthcare (HEALTH)* 3, pp. 1–27. doi: 10.1145/3472303.

Fu, Yu, Yan Hu, and Veronica Sundstedt (2024). "A Pilot Study of User Preferences of Posture and Display Technologies in Virtual Reality Exercise Games". In: *Proceedings of the 2024 8th International Conference on Virtual and Augmented Reality Simulations*. ICVARS '24. Melbourne, Australia: Association for Computing Machinery, pp. 22–27.

isbn: 9798400709012. doi: 10.1145/3657547.3657562. url: https://doi.org/10.1145/3657547.3657562.

Goodwill, James and Wesley Matlock (2015). "The Swift Programming Language". In: pp. 219–244. doi: 10.1007/978-1-4842-0400-9_17.

Gotterbarn, Don, Keith Miller, and Simon Rogerson (1997). "Software Engineering Code of Ethics". In: *Commun. ACM* 40.11, pp. 110–118. doi: 10.1145/265684.265699. url: https://doi.org/10.1145/265684.265699.

Grady, Robert B (1992). *Practical software metrics for project management and process improvement*. Prentice-Hall, Inc.

Gyawali, Dipesh (2023). "Mixed Reality: The Interface of the Future". In: *ArXiv* abs/2309.00819. doi: 10.48550/arXiv.2309.00819.

Hrycak, Camilla, David Lewakis, and Jens Krüger (2024). "Investigating the Apple Vision Pro Spatial Computing Platform for GPU-Based Volume Visualization". In: *2024 IEEE Visualization and Visual Analytics (VIS)*, pp. 181–185. doi: 10.1109/VIS55277.2024.00044.

Hu, Tianyi et al. (2024). "Apple v.s. Meta: A Comparative Study on Spatial Tracking in SOTA XR Headsets". In: *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*. ACM MobiCom '24. Washington D.C., DC, USA: Association for Computing Machinery, pp. 2120–2127. isbn: 9798400704895. doi: 10.1145/3636534.3696215. url: https://doi.org/10.1145/3636534.3696215.

Institute of Electrical and Electronics Engineers (2020). *IEEE Code of Ethics*. url: https://www.ieee.org/about/corporate/governance/p7-8.html (visited on 11/07/2024).

Instituto Politécnico do Porto (2020). *Regulamento do Código de Boas Práticas e de Conduta do Instituto Politécnico do Porto (Despacho n.º 11171/2020)*. url: https://www.iscap.ipp.pt/regulamentos/CodigoboaspraticasedecondutaIPP.pdf (visited on 11/06/2024).

Jasmine, S. et al. (2021). "Augmented and Virtual Reality and Its Applications". In: pp. 68–85. doi: 10.4018/978-1-7998-4703-8.ch003.

Jihambru (2022a). *3D Number - 1 (ONE)*. url: https://sketchfab.com/3d-models/3d-number-1-one-ad1f5a58f50b4176ac6684a7ad8eac2f (visited on 07/20/2025).

– (2022b). *3D Number - 2 (TWO)*. url: https://sketchfab.com/3d-models/3d-number-2-two-885febcfa1d749a3a3d982ae7bea9b45 (visited on 07/20/2025).

– (2022c). *3D Number - 3 (THREE)*. url: https://sketchfab.com/3d-models/3d-number-3-three-ae5779c280a54dbd8cf7e79526115464 (visited on 07/20/2025).

Jung, Sungchul et al. (2023). "Cross-Reality Gaming: Comparing Competition and Collaboration in an Asymmetric Gaming Experience". In: *Proceedings of the 29th ACM Symposium on Virtual Reality Software and Technology*. VRST '23. Christchurch, New Zealand: Association for Computing Machinery. isbn: 9798400703287. doi: 10.1145/3611659.3615698. url: https://doi.org/10.1145/3611659.3615698.

Khan, Sallar et al. (2020). "Augmented reality based gesture detection & object creation system using XCode & ARKit". In: *3C Tecnología Glosas de innovación aplicadas a la pyme*. doi: 10.17993/3ctecno.2020.specialissue6.79-91.

Kini, Vishwas G., Shonraj Ballae Ganeshrao, and P. C. Siddalingaswamy (2024). "XR Review: A Comprehensive Analysis of Visual Function Testing and Gamification in Extended Reality Environments". In: *IEEE Access* 12, pp. 79714–79730. doi: 10.1109/ACCESS.2024.3408677.

Kitson, Alexandra et al. (2024). "Virtual Games, Real Interactions: A Look at Cross-reality Asymmetrical Co-located Social Games". In: *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. CHI EA '24. Honolulu, HI, USA: Association for Computing Machinery. isbn: 9798400703317. doi: 10.1145/3613905.3650824. url: https://doi.org/10.1145/3613905.3650824.

Kriglstein, Simone et al. (2023). "Making A Real Connection: Pro-Social Collaborative Play in Extended Realities – Trends, Challenges and Potentials". In: *Proceedings of the 22nd International Conference on Mobile and Ubiquitous Multimedia*. MUM '23. Vienna, Austria: Association for Computing Machinery, pp. 544–547. isbn: 9798400709210. doi: 10.1145/3626705.3626708. url: https://doi.org/10.1145/3626705.3626708.

Kruchten, P.B. (1995). "The 4+1 View Model of architecture". In: *IEEE Software* 12.6, pp. 42–50. doi: 10.1109/52.469759.

Ladwig, Philipp and C. Geiger (2018). "A Literature Review on Collaboration in Mixed Reality". In: pp. 591–600. doi: 10.1007/978-3-319-95678-7_65.

Lakshtanov, Evgeny and Vera Roshchina (2012). "On Finiteness in the Card Game of War". In: *The American Mathematical Monthly* 119.4, pp. 318–323. doi: 10.4169/amer.math.monthly.119.04.318. eprint: https://www.tandfonline.com/doi/pdf/10.4169/amer.math.monthly.119.04.318. url: https://www.tandfonline.com/doi/abs/10.4169/amer.math.monthly.119.04.318.

Lierop, Meike van, V. Allard, and Jelle van den Hurk (2021). "Augmented Reality". In: *Industry, Innovation and Infrastructure*. doi: 10.1007/978-3-319-95873-6_300006.

Lyons, Kent (2024). "The Road to Ubiquity: Unpacking Barriers to Mass Adoption of Heads-Up Computing". In: *Companion of the 2024 on ACM International Joint Conference on Pervasive and Ubiquitous Computing*. UbiComp '24. Melbourne VIC, Australia: Association for Computing Machinery, pp. 634–636. isbn: 9798400710582. doi: 10.1145/3675094.3678993. url: https://doi.org/10.1145/3675094.3678993.

Manuri, Federico and A. Sanna (2016). "A Survey on Applications of Augmented Reality". In: *Advances in Computer Science : an International Journal* 5, pp. 18–27.

Mardhiyyah, Alya Dhiya' et al. (2023). "LonelyScape: Increasing Attractiveness of Escape Room Game Using Augmented Reality Technology". In: *2023 International Conference on Information Management and Technology (ICIMTech)*, pp. 795–800. doi: 10.1109/ICIMTech59029.2023.10277954.

Marín-Vega, Humberto et al. (2023). "Extended Reality (XR) Engines for Developing Gamified Apps and Serious Games: A Scoping Review". In: *Future Internet* 15, p. 379. doi: 10.3390/fi15120379.

Mekni, M. and A. Lemieux (2014). "Augmented Reality : Applications , Challenges and Future Trends". In.

Millard, David et al. (Aug. 2024). "The Ethics of Mixed Reality Games". In: *ACM Games* 2.3. doi: 10.1145/3675806. url: https://doi.org/10.1145/3675806.

Minaee, Shervin, Xiaodan Liang, and Shuicheng Yan (2022). "Modern Augmented Reality: Applications, Trends, and Future Directions". In: *ArXiv* abs/2202.09450.

Mindera (2020). *The Mindera Swift Style Guide*. url: https://github.com/Mindera/swift-style-guide (visited on 05/28/2025).

myfox14 (2021). *Game Over Arcade*. url: https://pixabay.com/sound-effects/game-over-arcade-6435 (visited on 07/20/2025).

National Society of Professional Engineers (2020). *NSPE Code of Ethics for Engineers*. url: https://www.nspe.org/resources/ethics/code-ethics (visited on 11/08/2024).

Ni, Tao (2024). "Sensor Security in Virtual Reality: Exploration and Mitigation". In: *Proceedings of the 22nd Annual International Conference on Mobile Systems, Applications and Services*. MOBISYS '24. Minato-ku, Tokyo, Japan: Association for Computing Machinery, pp. 758–759. isbn: 9798400705816. doi: 10.1145/3643832.3661389. url: https://doi.org/10.1145/3643832.3661389.

Nighteller (2022). *ciao ciao*. url: https://pixabay.com/sound-effects/ciao-ciao-88030 (visited on 07/20/2025).

Oates, Briony J, Marie Griffiths, and Rachel McLean (2022). *Researching information systems and computing*. Sage.

Ong, C. W. et al. (2020). "Applications of Extended Reality in Ophthalmology: Systematic Review". In: *Journal of Medical Internet Research* 23. doi: 10.2196/preprints.24152.

Ordem dos Engenheiros (2016). *Código de Ética e Deontologia*. url: https://www.ordemengenheiros.pt/fotos/editor2/regulamentos/codigo_ed.pdf (visited on 11/06/2024).

Park, Habin, Daniel Lichtman, and Oyewole Oyekoya (2023). "Exploring Virtual Reality Game Development as an Interactive Art Medium: A Case Study with the Community Game Development Toolkit". In: *Companion Proceedings of the 2023 Conference on Interactive Surfaces and Spaces*. ISS Companion '23. Pittsburgh, PA, USA: Association for Computing Machinery, pp. 5–9. isbn: 9798400704253. doi: 10.1145/3626485.3626530. url: https://doi.org/10.1145/3626485.3626530.

Permozer, Ivan and T. Orehovački (2019). "Utilizing Apple's ARKit 2.0 for Augmented Reality Application Development". In: *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 1629–1634. doi: 10.23919/MIPRO.2019.8756928.

Pons, P., Samuel Navas Medrano, and José Luis Soler Domínguez (2022). "Extended reality for mental health: Current trends and future challenges". In: 4. doi: 10.3389/fcomp.2022.1034307.

Qian, Yulin, Juntao Deng, and Siyi Chen (2023). "Application of Multiple Design Patterns in Virtual Reality Game Development". In: *Proceedings of the 2023 7th International Conference on Big Data and Internet of Things*. BDIOT '23. Beijing, China: Association for Computing Machinery, pp. 103–108. isbn: 9798400708015. doi: 10.1145/3617695.3617696. url: https://doi.org/10.1145/3617695.3617696.

Radu, Iulian and Bertrand Schneider (2019). "What Can We Learn from Augmented Reality (AR)? Benefits and Drawbacks of AR for Inquiry-based Learning of Physics". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Glasgow, Scotland Uk: Association for Computing Machinery, pp. 1–12. isbn: 9781450359702. doi: 10.1145/3290605.3300774. url: https://doi.org/10.1145/3290605.3300774.

Ratcliffe, Jack et al. (2021). "Extended Reality (XR) Remote Research: a Survey of Drawbacks and Opportunities". In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. CHI '21. Yokohama, Japan: Association for Computing Machinery. isbn: 9781450380966. doi: 10.1145/3411764.3445170. url: https://doi.org/10.1145/3411764.3445170.

Räthel, Fabian et al. (2024). "An Evaluation of Targeting Methods in Spatial Computing Interfaces with Visual Distractions". In: *Proceedings of the 30th ACM Symposium on Virtual Reality Software and Technology*. VRST '24. Trier, Germany: Association for Computing Machinery. isbn: 9798400705359. doi: 10.1145/3641825.3687712. url: https://doi.org/10.1145/3641825.3687712.

Rebouças, Marcel et al. (2016). "An Empirical Study on the Usage of the Swift Programming Language". In: *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)* 1, pp. 634–638. doi: 10.1109/SANER.2016.66.

Saltuk, Ozan and Ismail Kosan (2014). *Design and Creation as a Research Methodology (Ludwig Maximilian University of Munich)*. https://www.medien.ifi.lmu.de/lehre/ss14/swal/presentations/topic2-saltuk_kosan-DesignAndCreation.pdf.

Sampaio, Alberto (Nov. 2015). "Improving Systematic Mapping Reviews". In: *SIGSOFT Softw. Eng. Notes* 40.6, pp. 1–8. issn: 0163-5948. doi: 10.1145/2830719.2830732. url: https://doi.org/10.1145/2830719.2830732.

Sampath, Hemanth, P. Tinnakornsrisuphap, and Prashanth Hande (2024). "Enabling Extended Reality Over 5G with Distributed Computing". In: *IEEE Communications Magazine* 62, pp. 32–37. doi: 10.1109/MCOM.003.2300747.

Sargolzaei, Parisa, Mudit Rastogi, and Loutfouz Zaman (Oct. 2024). "Advancing Mixed Reality Game Development: An Evaluation of a Visual Game Analytics Tool in Action-Adventure and FPS Genres". In: *Proc. ACM Hum.-Comput. Interact.* 8.CHI PLAY. doi: 10.1145/3677055. url: https://doi.org/10.1145/3677055.

Schmalstieg, D. and Tobias Höllerer (2016). "Augmented reality: Principles and practice". In: *2017 IEEE Virtual Reality (VR)*, pp. 425–426. doi: 10.1145/2897826.2927365.

Sereno, Mickael et al. (2020). "Collaborative Work in Augmented Reality: A Survey". In: *IEEE Transactions on Visualization and Computer Graphics* 28, pp. 2530–2549. doi: 10.1109/TVCG.2020.3032761.

Shekhar, S., Steven K. Feiner, and W. Aref (2015). "From GPS and virtual globes to spatial computing - 2020". In: *GeoInformatica* 19, pp. 799–832. doi: 10.1007/s10707-015-0235-9.

Singh, Bikramjit and R. Kaur (2017). "Raising Performance of iPhone using Swift Language over Other Programming Languages". In: *International Journal of Advance Research, Ideas and Innovations in Technology* 3, pp. 991–994.

Speicher, Maximilian, Brian D. Hall, and Michael Nebeling (2019). "What is Mixed Reality?" In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Glasgow, Scotland Uk: Association for Computing Machinery, pp. 1–15. isbn: 9781450359702. doi: 10.1145/3290605.3300767. url: https://doi.org/10.1145/3290605.3300767.

Spitzer, Barbara Oliveira et al. (2022a). "Framework for the Use of Extended Reality Modalities in AEC Education". In: *Buildings*. doi: 10.3390/buildings12122169.

– (2022b). "Framework for the Use of Extended Reality Modalities in AEC Education". In: *Buildings*. doi: 10.3390/buildings12122169.

Splashdust (2022). *Flip Card*. url: https://pixabay.com/sound-effects/flipcard-91468 (visited on 07/11/2025).

Stein, Scott (2024). *Spatial Personas Have Turned My Apple Vision Pro Into a Telepresence Machine*. url: https://www.cnet.com/tech/computing/spatial-personas-have-turned-my-apple-vision-pro-into-a-telepresence-machine (visited on 07/14/2025).

Tamura, H., Hiroyuki Yamamoto, and A. Katayama (2001). "Mixed Reality: Future Dreams Seen at the Border between Real and Virtual Worlds". In: *IEEE Computer Graphics and Applications* 21, pp. 64–70. doi: 10.1109/38.963462.

TuomasData (2023). *Game Over 31*. url: https://pixabay.com/sound-effects/game-over-31-179699 (visited on 07/20/2025).

Unity (2019). *XR Interaction Toolkit*. url: https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@3.0 (visited on 05/16/2025).

uss015dykrt (2023). *Brass fanfare with timpani and winchimes, reverberated*. url: https://pixabay.com/sound-effects/brass-fanfare-with-timpani-and-winchimes-reverberated-146260 (visited on 07/20/2025).

Van Der Veer, Gerrit Cornelis et al. (2024). "Experience 2.0 and Beyond - Engineering Cross Devices and Multiple Realities". In: *Companion Proceedings of the 16th ACM SIGCHI Symposium on Engineering Interactive Computing Systems*. EICS '24 Companion.

Cagliari, Italy: Association for Computing Machinery, pp. 108–110. isbn: 9798400706516. doi: 10.1145/3660515.3662838. url: https://doi.org/10.1145/3660515.3662838.

Vázquez-Ingelmo, Andrea, Alicia García-Holgado, and Francisco J. García-Peñalvo (2020). "C4 model in a Software Engineering subject to ease the comprehension of UML and the software". In: *2020 IEEE Global Engineering Education Conference (EDUCON)*, pp. 919–924. doi: 10.1109/EDUCON45650.2020.9125335.

Vergari, Maurizio et al. (2024). "Digital Eyes: Social Implications of XR EyeSight". In: *Proceedings of the 30th ACM Symposium on Virtual Reality Software and Technology*. VRST '24. Trier, Germany: Association for Computing Machinery. isbn: 9798400705359. doi: 10.1145/3641825.3689526. url: https://doi.org/10.1145/3641825.3689526.

Vervoorn, M. T. et al. (2022). "Mixed Reality in Modern Surgical and Interventional Practice: Narrative Review of the Literature". In: *JMIR Serious Games* 11. doi: 10.2196/41297.

Vilar, João, Armanda Rodrigues, and Nuno Correia (Nov. 2024). "An Extended Reality Platform for Cultural Gaming: Enabling Interactive Narratives in Spatial Contexts". In: *J. Comput. Cult. Herit.* Just Accepted. issn: 1556-4673. doi: 10.1145/3702484. url: https://doi.org/10.1145/3702484.

Vivekanandam, B. (2023). "Recent development in Extended Reality technologies". In: *Recent Research Reviews Journal*. doi: 10.36548/rrrj.2023.1.11.

Wang, Hanqiu et al. (2024). "GAZEploit: Remote Keystroke Inference Attack by Gaze Estimation from Avatar Views in VR/MR Devices". In: *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*. CCS '24. Salt Lake City, UT, USA: Association for Computing Machinery, pp. 1731–1745. isbn: 9798400706363. doi: 10.1145/3658644.3690285. url: https://doi.org/10.1145/3658644.3690285.

Wells, G. (2015). "The Future of iOS Development: Evaluating the Swift Programming Language". In.

Xu, Jiangnan et al. (2024). "Spatial Computing: Defining the Vision for the Future". In: *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. CHI EA '24. Honolulu, HI, USA: Association for Computing Machinery. isbn: 9798400703317. doi: 10.1145/3613905.3643978. url: https://doi.org/10.1145/3613905.3643978.

xxxTruthxxx (2022). *Hi, how are you doing (COMPUTER)*. url: https://pixabay.com/sound-effects/hi-how-are-you-doing-computer-106055 (visited on 07/20/2025).

Yanez, Anthony (2018). *52-Card Deck*. Licensed under Creative Commons Attribution 4.0 International (CC BY 4.0). url: https://sketchfab.com/3d-models/52-card-deck-dc2e1196295e45649d4471791ed23f5b (visited on 07/08/2025).

Yang, Chaowei et al. (2011). "Using spatial principles to optimize distributed computing for enabling the physical science discoveries". In: *Proceedings of the National Academy of Sciences* 108, pp. 5498–5503. doi: 10.1073/pnas.0909315108.

Zhao, Shengdong et al. (2024). "Heads-Up Computing: Opportunities and Challenges of the Next Interaction Paradigm with Wearable Intelligent Assistants". In: *Companion of the 2024 on ACM International Joint Conference on Pervasive and Ubiquitous Computing*. UbiComp '24. Melbourne VIC, Australia: Association for Computing Machinery, pp. 960–963. isbn: 9798400710582. doi: 10.1145/3675094.3677563. url: https://doi.org/10.1145/3675094.3677563.

Appendix A

AceR Screenshots

This appendix presents additional screenshots of the AceR's prototype visionOS implementation that were not included in the main body of the document.

Figure A.1: Splash Screen

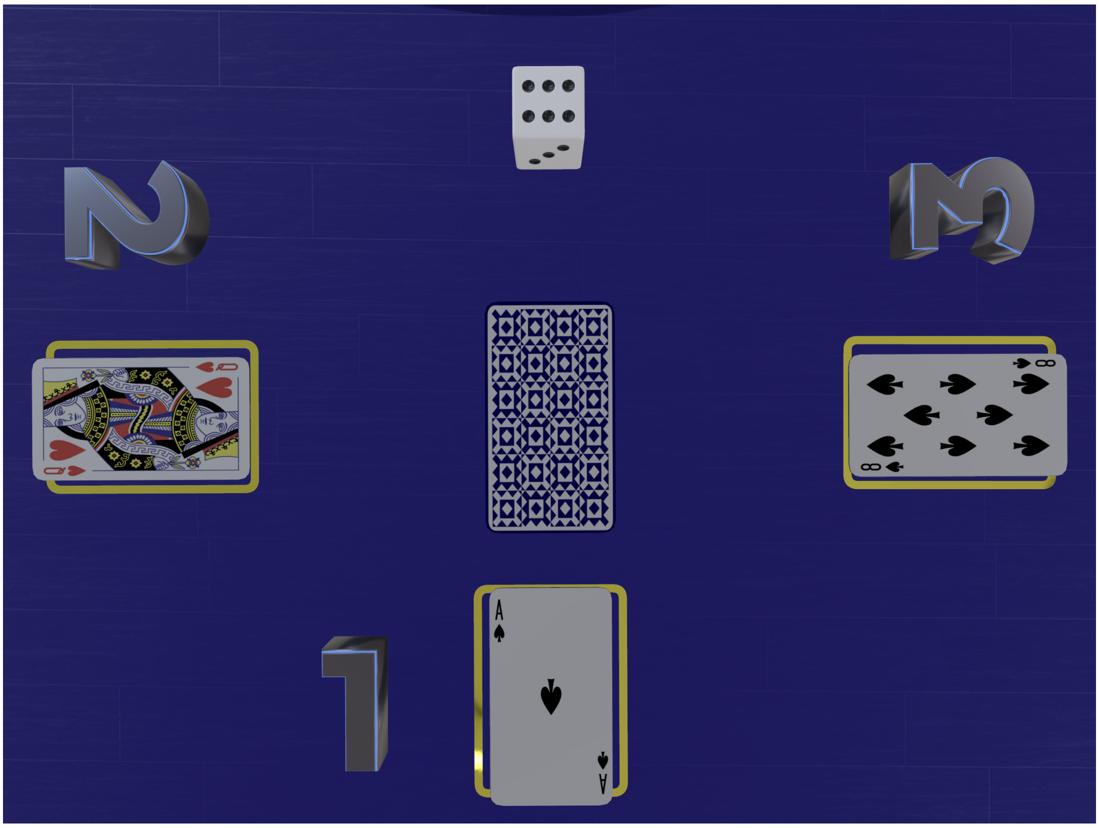


Figure A.2: Play Card Close-Up

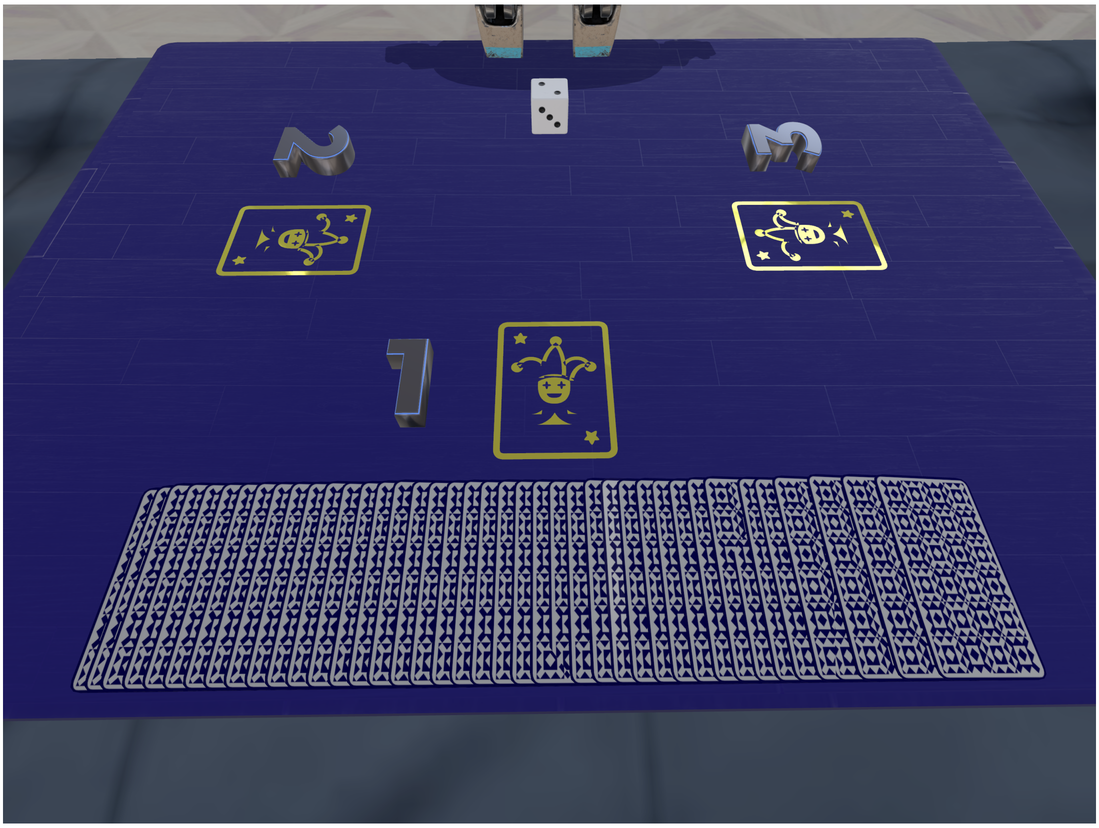


Figure A.3: Player's Hand

Figure A.4: Another Angle of the Robot Reacting to a Round Win

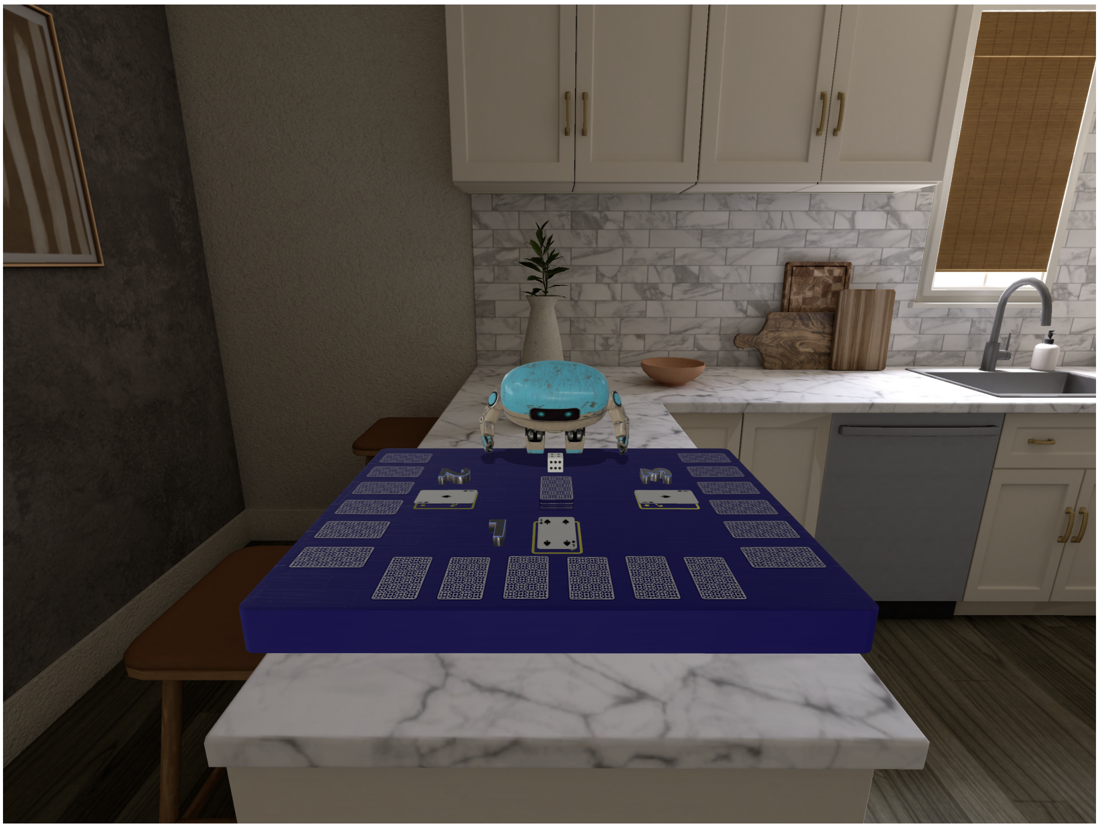


Figure A.5: Another Angle of the Robot Reacting to a Round Draw

Figure A.6: Multitasking: Browsing the Internet

Figure A.7: Another Example of the Board Anchored

Appendix B

Expert Semi-Structured Interviews

This appendix will present some aspects of the expert semi-structured interviews, such as the invitation email, consent form, scheduling via Doodle and post-interview questionnaire.

B.1 Invitation via Email

Below is a template of the formal invite sent via email to each expert before the interviews took place. The email is in portuguese as all experts and the author speak portuguese as their main language.

Subject: Convite para Participar numa Avaliação de Protótipo XR (Apple Vision Pro)

Bom dia,

Espero que esta mensagem o/a encontre bem.

Estou a entrar em contacto para o/a convidar a participar numa sessão remota onde irá ser apresentado um protótipo de um jogo XR para Apple Vision Pro e visionOS, desenvolvido no âmbito da minha dissertação de Mestrado.

A sessão incluirá uma demonstração do protótipo, seguida de uma breve entrevista semi-estruturada para recolher algumas impressões sobre o mesmo. Esta entrevista terá uma duração entre 30 minutos a 1 hora e será realizada remotamente através da plataforma Microsoft Teams, sendo gravada e transcrita para posteriormente ser utilizada no documento da dissertação. A transcrição utiliza inteligência artificial. Todas as citações e transcrições utilizadas na tese serão anonimizadas.

Se estiver disponível e interessado/a, agradecia que lesse e preenchesse o seguinte formulário de consentimento: `[Google Form Consent Form Link]`

Para agendar a sessão, agradeço que indicasse a sua disponibilidade através dos seguintes links da ferramenta Doodle:

- `[Doodle Scheduling Links]`

...

Fico à disposição para esclarecer qualquer dúvida. Agradeço desde já a sua atenção e disponibilidade. Muito obrigado.

Com os melhores cumprimentos,
Pedro Santos
Mestrado em Engenharia Informática
Instituto Superior de Engenharia do Porto

B.2 Consent Form

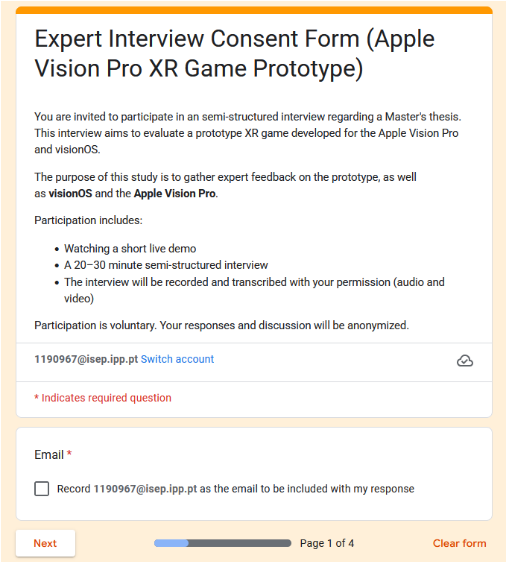


Figure B.1: Consent Form: Email

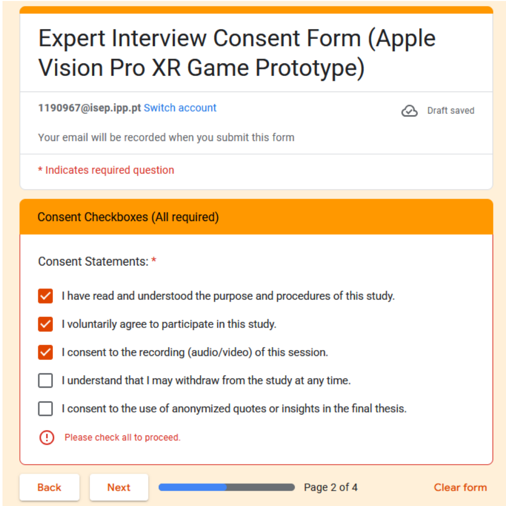


Figure B.2: Consent Form: Consent

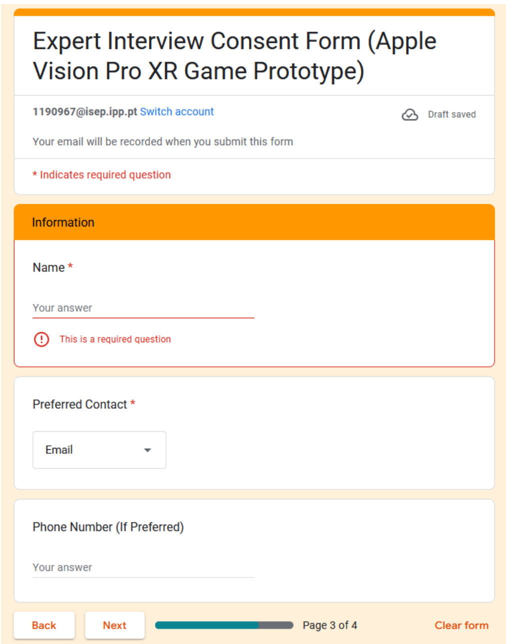


Figure B.3: Consent Form: Information

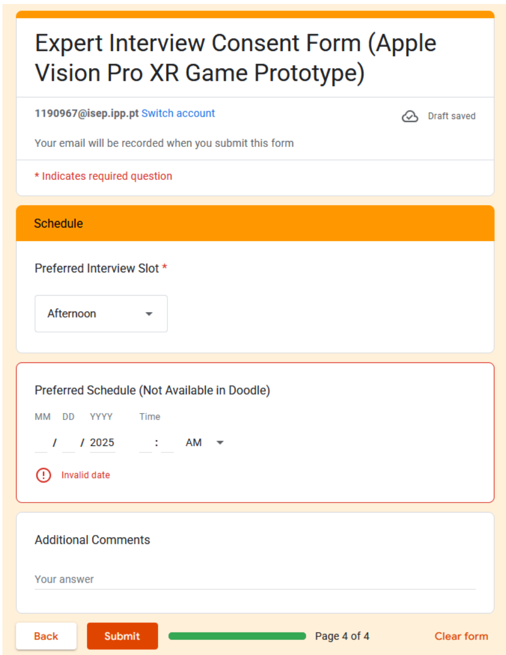


Figure B.4: Consent Form: Scheduling

B.3 Scheduling via Doodle

Figure B.5 represents one of the Doodle links sent to experts in order to schedule the interview. As Doodle has some restrictions for free users, many links were made available to each expert in order for them to choose their preferred schedules.

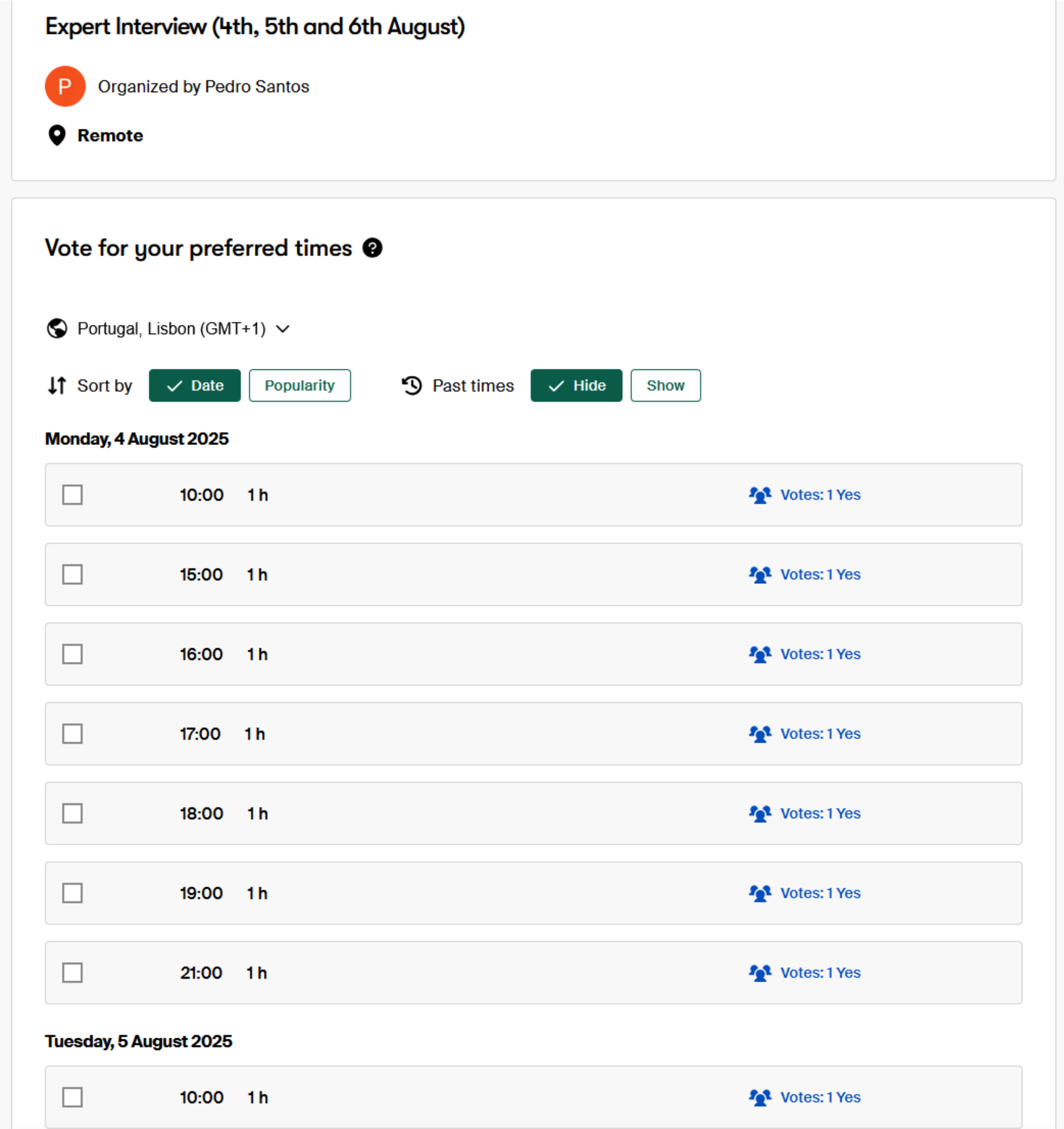
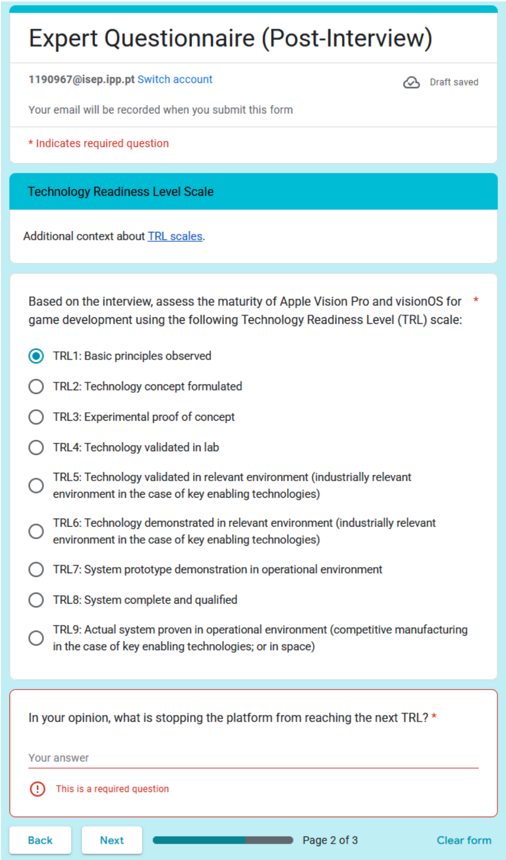


Figure B.5: Scheduling via Doodle

B.4 Post-Interview Questionnaire

This questionnaire was made available to experts after the interview was concluded in order to gather feedback regarding the platform's maturity and the prototype itself. The questionnaire was filled soon after the interview concluded so that expert still had the interview's content fresh in their minds.



Expert Questionnaire (Post-Interview)

1190967@isep.ipp.pt Switch account

Draft saved

Your email will be recorded when you submit this form

* Indicates required question

Technology Readiness Level Scale

Additional context about TRL scales.

Based on the interview, assess the maturity of Apple Vision Pro and visionOS for game development using the following Technology Readiness Level (TRL) scale: *

- TRL1: Basic principles observed
- TRL2: Technology concept formulated
- TRL3: Experimental proof of concept
- TRL4: Technology validated in lab
- TRL5: Technology validated in relevant environment (industrially relevant environment in the case of key enabling technologies)
- TRL6: Technology demonstrated in relevant environment (industrially relevant environment in the case of key enabling technologies)
- TRL7: System prototype demonstration in operational environment
- TRL8: System complete and qualified
- TRL9: Actual system proven in operational environment (competitive manufacturing in the case of key enabling technologies; or in space)

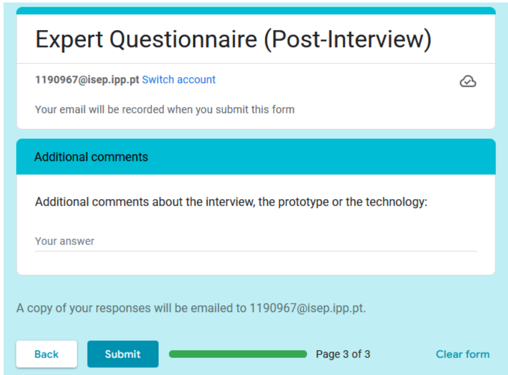
In your opinion, what is stopping the platform from reaching the next TRL? *

Your answer

This is a required question

Back Next Page 2 of 3 Clear form

Figure B.6: Questionnaire: TRL Scale



Expert Questionnaire (Post-Interview)

1190967@isep.ipp.pt Switch account

Your email will be recorded when you submit this form

Additional comments

Additional comments about the interview, the prototype or the technology:

Your answer

A copy of your responses will be emailed to 1190967@isep.ipp.pt.

Back Submit Page 3 of 3 Clear form

Figure B.7: Questionnaire: Additional Comments

Appendix C

Expert Interview Transcripts

In this appendix, all raw transcripts for each expert semi-structured interview will be presented in order to provide a clear image of what happened in each interview. Since all experts and the author have Portuguese as their main language, all interviews were made in that language, thus all transcripts are in the Portuguese language. The built-in Microsoft Teams recording and transcription features were used to gather these transcripts. The transcription feature uses artificial intelligence for its purpose, so it is not completely accurate. In order to mitigate it, all transcripts were reviewed and edited by the author. Despite that effort, some instances may still be inaccurate.

C.1 Expert A

[Introduction and asking for consent before starting the recording and transcription.]

Pedro Santos started the transcription.

Pedro Santos 0:04

A gravação já começou. Eu estou a explorar o Apple Vision Pro que é uma plataforma um bocado nova, numa perspetiva de game development. Um dos frameworks que a Apple tem first party relacionados com a área de gaming é o TabletopKit que permite a implementação de jogos de tabuleiro. Têm alguma estrutura que já permite fazer jogos de tabuleiro. Eu vou fazer uma demonstração do jogo já a seguir.

Expert A 0:50

Antes disso.

Pedro Santos 0:51

E foi um bocado por aí.

Expert A 0:54

E foste ver algum jogo tabletop migrado para a versão digital antes disto?

Pedro Santos 1:03

Vi alguns exemplos mas não me inspirei em nenhum. O jogo em si, o conceito fui eu que criei com algumas inspirações de jogos existentes. É um tabletop de cartas.

Expert A 1:11

Hmm.

Pedro Santos 1:19

Com alguns elementos extra. As regras fui eu que as tive a fazer, etc. Mas tem algumas inspirações de alguns jogos tradicionais de cartas, mas não é um jogo específico.

Expert A 1:35

Como que jogos tradicionais de cartas?

Pedro Santos 1:39

Há, por exemplo, a guerra, com algumas diferenças, por exemplo. Tem a parte de ser implementado num contexto mais de realidade estendida.

Expert A 1:53

Então. Portanto, estamos num jogo de jogador contra jogador, certo?

Pedro Santos 1:59

Não percebi.

Expert A 2:01

É um jogo jogador contra jogador, portanto, para ser jogado por 2 pessoas.

Pedro Santos 2:04

3 jogadores. A minha implementação permite single player e multiplayer. Uma das coisas que eu queria, que eu queria explorar era a integração do do Apple Vision Pro, também com a parte do ecossistema da Apple. Neste caso, eu vou demonstrar na parte da demonstração.

Expert A 2:12

Sim.

Pedro Santos 2:29

Ter uma vertente multiplayer embutida, por exemplo, neste caso através do FaceTime. Isto depois abre bastantes portas. Não sei o quão contextualizado está com a parte do Apple Vision Pro, mas existem spatial personas. Que são modelos 3 da cara da pessoa partilhadas no FaceTime. Com a integração do FaceTime, as pessoas conseguem estar a jogar o meu jogo neste caso. As pessoas estão a jogar com outras pessoas, mas a ver a cara das pessoas, esse Modelo 3D, as pessoas mexem-se e reagem.

Expert A 3:09

Certo.

Pedro Santos 3:12

Aparecerem nos lados da board.

Ou seja, explorei um bocadinho esta vertente.

Expert A 3:21

Hmm.

Pedro Santos 3:24

A plataforma ainda não saiu para todos os mercados, nomeadamente Portugal, daí esta limitação enorme de usar o simulador. Na parte do meu desenvolvimento e até da demonstração que faço hoje. Porque demonstrações não poder ser feitas num device físico. Tem que ser tudo através de simulador que tem certas limitações na integração do facetime etc. Eu também demonstro na parte da demonstração.

Expert A 3:57

OK, vamos ver então.

Pedro Santos 4:03

Pronto, então vou partilhar o meu ecrã. Pronto acho que estou a partilhar direito.

Expert A 4:13

Sim, estás.

Pedro Santos 4:14

Pronto isto é o jogo num dos ambientes do simulador. É uma sala neste caso. O jogo é esta board azul. Aqui tem cartas em cima e este Robot. Tenho uma toolbar com algumas ações que o utilizador pode fazer e está sempre disponível para o jogador.

Expert A 4:39

Que são essas onde está o rato, certo?

Pedro Santos 4:42

Sim, sim. Exato isto onde eu passo rato, num contexto real e num universo físico, seria o olho do utilizador, seja olhar e com os gestos. Que pode ser à frente do utilizador que eu estou a fazer agora.

[Makes a pinching gesture]

Ou então em baixo, porque tem câmaras para baixo. O seleccionar no meu caso é um clique do rato. O Jogo em si fica anchored, ou seja, utilizador metê-lo numa mesa e vai à cozinha. Quando volta e o jogo está lá à sua espera.

Expert A 5:21

Hmm.

Pedro Santos 5:24

Portanto, limitações do simulator que não é um bocado intuito de mover. Pronto, vou falar um bocadinho das regras, cada jogador tem uma hand. Com cartas viradas ao contrário.

Expert A 5:43

Quantas mãos, quantas cartas na mão?

Pedro Santos 5:44

E escolhe. Depende do modo, eu tenho 2 modos, tenho um modo rápido e tenho este modo normal. As regras são parecidas, mas eu, eu já explico. Há, por exemplo, agora uma das coisas relacionadas com as regras seria aqui. Tem um rule set onde o utilizador pode colocar onde quiser. E pode estar jogar a olhar para as regras ao mesmo tempo, sem ter que estar a parar. Ajuda no onboarding, nos primeiros jogos. Pode meter aqui. Fica ancorado. Sobre os outros jogadores em multiplayer. Cada um pode ter o seu próprio rule set onde quiser e também podem partilhar as regras. Eu também já mostro essa parte na parte multiplayer. Como estava a dizer, eu acho que estou a partilhar o som, consegue ouvir?

Expert A 6:37

Sim, estás.

Pedro Santos 6:38

Portanto, são 7 cartas para cada jogador num jogo regular e tem um stack com as restantes cartas. Isto é um baralho normal, ou seja, também permite depois a cada jogador implementar house rules. Tem aqui depois um modo para desativar as regras, porque uma coisa do TabletopKit é que posso restringir movimentos e interações, por exemplo. Neste caso, esta é a minha hand. Eu não posso interagir com as cartas das hands de outros jogadores. Posso determinar onde é que o utilizador pode largar cada elemento. Neste caso, pode deixar nestas aqui, onde o cada jogador vai jogar a sua carta. Então como é que funciona o Jogo? Cada jogador tem cartas na hand viradas para baixo.

Expert A 7:32

Sim.

Pedro Santos 7:39

E seleciona uma carta aleatória sem saber o valor da hand e joga a carta virada ao contrário. Utilizador depois não pode tirar a carta. Jogou aquela carta e ponto. Cada jogador faz o mesmo. Depois de todos os jogadores jogarem uma carta, a carta com valor menor ganha e fica com as cartas do adversário. Por exemplo, neste caso o jogador tem uma Dama e ganha as cartas dos outros jogadores. A Dama vale 1 e o Ás vale zero, ou seja, o Ás ganha a uma Dama. Quem ganha fica com as cartas e se houver um empate, é resolvido por este dado que está aqui. É lançado entre os jogadores que empataram e quem lançar o menor valor fica com as cartas. Se houver um empate nos dados, vão ser lançado outra vez até alguém ter o menor valor e assim ganhar as 3 cartas. Ai depois de cada ronda, cada jogador vai buscar uma carta ao Stack do meio. Até não haver basicamente cartas. As rondas continuam. Um jogador é eliminado quando não tiver cartas para jogar e não houver cartas no stack. Um jogador ganha quando os outros 2 foram eliminados. As regras são bastante simples. Mas posso mostrar aqui um bocadinho em ação e neste caso em single player. Quando um jogador local joga, os restantes jogadores funcionam como bots, jogando automaticamente. Por exemplo, neste caso, houve um empate com 2 ases. Seria resolvido pelo lançamento do dado. Também permito aqui aos jogadores decidirem outras maneiras de fazer o desempate e não automaticamente mover as cartas. Tem que ser movidas manualmente também para deixar alguma interação entre os jogadores, não restringir muito as ajudas da jogada. Como disse, também tem um outro modo mais rápido. Neste caso, são só 3 cartas para cada um e não há stack.

Expert A 9:59

Certo.

Pedro Santos 10:00

Este modo é mais fácil de demonstrar. Neste caso, para apresentar dá mais jeito consigo mostrar. Por exemplo, este jogador foi eliminado. Tem um efeito de som diferente. E eventualmente alguém irá a ganhar o jogo. Mesmo com 3 cartas, às vezes poderá demorar. Penso que vou perder.

Expert A 10:44

Há alguma hand ao lado do espaço da mão, tipo um outro deck.

Pedro Santos 10:52

Não, não, não, este é o único. Mas poderia ser implementado.

Expert A 10:55

E, ou seja, se o jogador 3 agora tivesse 20 cartas, como é que farias isso?

Pedro Santos 11:01

Eu posso demonstrar, na implementação do layout elas metem-se em cima das outras. No espaço que tem para a hand, as cartas dão stack. De forma a ajudar a pegar em qualquer uma das cartas.

Expert A 11:24

Ou seja, no modo rápido. Eu já percebi porque não, na verdade, nós vamos buscar mais cartas ao baralho, certo? No modo normal é que isso pode ser um problema, porque para haver um vencedor estamos a falar de uma altura em que um jogador chega a ter 30 cartas e outro a ter só 10, por exemplo, não é?

Pedro Santos 11:34

Exato. É possível definir como é que as cartas vão ser posicionadas no parent neste caso. E vai-se pelo código que eu implementei, vai ser tudo por Estado, ou seja, fica na no lado da Mesa do utilizador. Neste caso, como só tem 3 cartas, eles são separados, espalhados pelo área total. Quanto mais cartas, por exemplo, aqui consegue saber quais as 3 cartas são separadas, estas estão mais juntas e eu já mostro queria só mostrar um vencedor. Eventualmente há um vencedor. E para demonstrar essa questão.

Expert A 12:51

Winner winner, chicken dinner.

Pedro Santos 12:56

Sim. [Laughs]

Expert A 13:07

Hmm.

Pedro Santos 13:09

Desativa as regras de restrições, por exemplo, eu com a toggle ativa, não consigo mexer nas cartas de outros jogadores, etc. Se desativar, agora já permite os jogadores mexerem nas outras cartas e para também permitir a parte. As house rules que eu já tinha falado, como definir alguma maneira diferente de empate. Pedra, papel, tesoura por exemplo.

Expert A 13:21

Sim.

Pedro Santos 13:32

Dar essa liberdade, porque muitas das vezes os jogos definem as regras, mas consenso da população geral até arranjam outras regras que sejam acordadas por Toda a gente a jogar. Mas fácil no simulador, mas acho que dá para ver mais ou menos. Não sei se isso responde à questão.

Expert A 14:04

Hmm, sim. OK, mais comum geralmente. Por norma neste tipo de jogos, o que se faz é bloqueia essas slots daquilo que representa a nossa mão, não é? E haveria um sitio onde eu tenho o resto das minhas cartas e à medida que eu vou usando, eu simplesmente vou tirando de lá e vou pondo na hand. Ou seja, permite uma maior escalabilidade. Compreendo que visualmente a tua implementação seja mais atrativa.

Pedro Santos 14:25

OK.

Expert A 14:45

Alguma dificuldade em termos de experiência, porque se são aleatórios é indiferente, qual é escolho. Mas eu posso querer clicar numa específica. Depois, a pequenez que se principalmente se for para para jogar num telemóvel, não é dada pequeninas do interface e principalmente, dar precisam se a gente for usar eye-tracking para movimentos, eu quero escolher a quarta carta e acabo por escolher outra.

Pedro Santos 14:51

Sim. Exato, exato.

Expert A 15:20

OK, isso pode eventualmente levar a questões de frustração do jogador, principalmente se ele perder não é? É que vai simplificar, embora sejam aleatórias. Eu percebo a ideia da guerra, porque na guerra a gente nem sabe o que é que tira e pronto, não é? Pensaste bem, gostei do teu trabalho, está fixe, funciona. Em termos de usabilidade por parte dos jogadores, é que eu levantaria algumas questões por causa do tal min max, numa altura em que o jogador 1 está.

Pedro Santos 15:33

Exato.

Expert A 15:51

Para vencer o Jogo, ou seja, vamos imaginar que o jogador já foram, já foram embora e isto acaba quando também acabarem as cartas do baralho. O jogador, há de ter 39 cartas na mesa né? Eu queria ver o que é que onde é que vamos pôr? E 39 cartas com um mínimo de legibilidade. Hoje para o jogador escolher qual é a carta que percebes?

Pedro Santos 16:03

Exato sim.

Expert A 16:14

Mas isso pormenores em termos de UX. Portanto, mas está fixe a ideia. A ideia do jogo está fixe. Eu queria te perguntar bem. Porque a limitação para 3 jogadores, porque O Jogo parece-me perfeitamente escalável para mais.

Pedro Santos 16:29

Sim, foi uma limitação que foi definida por mim. Não foi nada de limitação, de número de utilizadores etc. Como isto é um protótipo, queria demonstrar o multiplayer com spectators. Mas é perfeitamente possível escalar para mais jogadores que o TabletopKit não limita nada a possibilidade. Ter uma boards retangular foi uma decisão minha neste caso.

Expert A 17:08

Certo. E outra questão é, e pronto, mas isso lá está. Já são manias minhas.

Pedro Santos 17:22

Força.

Expert A 17:22

Eu sei o que é que ele faz. Mas porque o dado, em vez de utilizarmos elementos que nós já temos? E neste caso, em caso de empate, imagina o jogador 2 perdeu, o jogador 1 e jogador 3 empatam. Para vencer, eventualmente porque não simplesmente tirar mais uma carta de cada um e eventualmente quem vencer e ganha todas.

Pedro Santos 17:44

Exato, exato foi uma decisão deliberada por causa de introduzir aqui um novo elemento de tabletop diferente para poder explorar também a parte dado que o TabletopKit fornece.

Expert A 17:49

OK, faz sentido, foi a resposta certa.

Pedro Santos 17:58

O TabletopKit fornece estados de cada equipamento. Estados específicos, por exemplo, para cartas. Estas têm a face up e face down. Outro estado para dados e também um estado custom para outros elementos.

Cá está, era mais para explorar estas diferentes vertentes.

Expert A 18:22

É a resposta certa, OK, tipo tinhas curiosidade. Queria ver como é que isto funcionava. Estava a falar de tabletop puro, ou seja, se isto fosse apenas para a mesa física, quanto menos elementos a gente tiver, melhor. Não é neste caso pronto. Lá está seria provavelmente mais entusiasmante eu entrar numa sub competição e em vez de eu ganhar 3 cartas, eu poder ganhar 4 não é, ou seja, eu empatei com com o jogador 3.

Pedro Santos 18:29

Exato.

Certo.

Expert A 18:53

OK fixe, ganhei mais uma e o método que continua a ser absolutamente igual. Portanto não há, não há vantagens nem desvantagens. Se a justificação é essa, parece-me perfeitamente plausível. Quiseste experimentar, ver como é que funcionava e testar os limites da máquina, okay.

Pedro Santos 19:01

Exato.

Expert A 19:15

Portanto, e mais.

Pedro Santos 19:18

Também tenho esta possibilidade. Neste caso, está numa vertente mais mixed reality. Neste caso os elementos são virtuais porque é um simulator, mas seriam os elementos reais do contexto do utilizador com o jogo em si, mas têm a possibilidade de virar para um ambiente totalmente virtual. Neste caso, existem várias possibilidades. Eu escolhi fazer um ambiente que ainda tem alguns elementos do ambiente do utilizador. Por exemplo, para ter cuidado com as partes físicas mais perto, mas ter o um ambiente geral também pode ser completamente virtual. É permitido pelo framework e pela plataforma em si. Agora sobre esta parte de multiplayer com FaceTime, que é uma parte também que queria explorar, especialmente esta integração com o ecossistema da Apple. Um multiplayer tem a integrado com o FaceTime e com a feature de Shareplay. E eu vou começar agora uma chamada FaceTime com 2 utilizadores que também têm um Apple Vision Pro e então eles aparecem assim. Lá está uma limitação do simulador. Num contexto real com utilizadores com a devices físicos em vez destes emojis de gato e robô seria mesmo a cara da pessoa 3D. Atualizava conforme as expressões, movimentos etc. Ou seja, daria uma imersão maior pois parece que as pessoas estão a jogar. Essas pessoas iam buscar a buscar as cartas e mexer-se conforme esse movimento. Lá está uma limitação do simulador. Tem a limitação de não haver device físico para poder testá-la e apresentá-la de forma.

Expert A 21:30

Hmm.

Pedro Santos 21:32

Pronto, de forma completa. A aplicação identifica que uma chamada está ativa. Através deste botão aqui, onde tem algumas opções da parte do share play. Este aqui, serve para partilhar o jogo com os participantes da chamada FaceTime. Aqui está outra limitação, como é um ambiente simulado não tenho acesso ao ponto de vista e ao conteúdo de cada um dos participantes. Não consigo fazê-los a entrar no jogo e ir para um dos seats, mas dá para demonstrar um bocadinho o multiplayer se algum jogador tiver numa chamada com 4 pessoas. O quarto jogador que não tem um sítio pode espectar o jogo.

Expert A 23:02

Certo.

Pedro Santos 23:08

O Single Player, ou seja, já não. Já joga automaticamente, por exemplo, posso mostrar aqui. Está a ser partilhado. Tem duas destas janelinhas em que facilmente utilizador pode, se eu conseguir rodar. Uma coisa é pode adicionar pessoas e partilhar o link e através de mensagens. Também partilhar com pessoas dos contactos, por exemplo. Um contacto chamado John e essa pessoa é convidada rapidamente para entrar no jogo.

Expert A 23:41

As salas criadas são públicas?

Pedro Santos 23:46

Apenas com contactos. É através da aplicação de Facetime da Apple.

Expert A 23:47

OK. Ou seja, Eu não consigo chegar aí e deixa-me ver se eu consigo entrar em algum jogo.

Pedro Santos 23:56

Não teria que ser com amigos e é vocacionado mais para amigos. Isso poderia ser uma melhoria futura num caso real deste deste jogo. Neste caso através do Facetime seria através de um link.

Expert A 23:56

OK.

Pedro Santos 24:17

Mas tem que ser partilhado sempre por alguém. Não há um sistema de public rooms, por exemplo.

Expert A 24:19

Sim.

Pedro Santos 24:22

Há também outra outra vertente é interessante. Está relacionada mais com a plataforma em si, não com a minha implementação do jogo, mas é, mas permite também multitasking. Alguém está a jogar pode ver um vídeo ao mesmo tempo, pode muito bem-estar a ver um filme. Neste caso eu pus música. Acho que está muito alto, se calhar. Pode estar a ouvir música, pode dar browse na Internet enquanto está a jogar.

Expert A 24:58

Hmm.

Pedro Santos 24:59

A chamada também pode ser terminada a qualquer momento e o jogo volta a ser o local. O game state é sincronizado entre todos os jogadores e as animações são calculadas localmente. Em termos de demonstração, penso que seja um bocadinho por aí.

Expert A 25:38

Hmm.

Pedro Santos 25:44

Trabalho futuro seria quando houvesse disponibilidade em Portugal do device físico para testar com várias pessoas. E penso seria isto da demonstração. Alguma questão? Alguma coisa que eu possa demonstrar?

Expert A 26:12

Eu não tenho. Bom é assim que me parece pronto, construíste uma bom protótipo e investigação. É uma, é uma investigação exploratória daquilo que são as capacidades do Apple Vision Pro. Sendo que o jogo acaba por uma tech demo. A linha orientadora daquilo que é que a investigação: okay, deixa-me ver se eu consigo dominar esta ferramenta, deixa-me ver a riqueza daquilo que eu consigo criar ali e até onde é que eu consigo ir. Eu não sei exatamente que tipo de comentários ou feedback esperarias da minha parte.

Pedro Santos 27:08

Eu tenho algumas questões.

Expert A 27:10

Força.

Pedro Santos 27:12

Primeiro começar com uma pergunta mais opinião: a implementação de realidade estendida deste jogo oferece algum valor único comparativo, como por exemplo uma implementação do jogo físico. Mesmo em comparação entre esta implementação, por exemplo, alguém com um baralho de cartas e um dado. Qual seria o valor único da implementação XR?

Expert A 27:40

OK é assim. São coisas radicalmente diferentes em termos de utilização de uma tecnologia emergente e a exploração dessa tecnologia para criar um produto vendável, essa parte parece me excelente. Aquilo que me parece claudicar um pouco ainda é a questão do desenvolvimento, do produto ou do jogo enquanto produto. Na medida que não é, não é de maneira nenhuma uma crítica ao game design no sentido porque isto eventualmente também pode se defender, mas em termos de tabletop e em termos de mercado de videojogos, este este acaba por ficar ali no ultra casual, porque o jogador não tem grande agência, não é? E o facto de eu poder escolher uma de 7 cartas que estão ali viradas para baixo. Na verdade é uma, é uma ilusão da escolha, porque eu não sei que carta que eu estou a escolher. Eu não sei que a Carta e a que eu não estou a escolher. Eu estou a escolher uma posição e nem sequer é transparente para mim. Se ali é um 2 ou um 3.

Pedro Santos 28:43

Certo.

Expert A 28:55

Se houvesse a intenção disso não é isso poderia ser aldrabado. Ou seja, isto em termos de jogo da casa, por assim dizer, se fosse um jogo de Casino, isto dificilmente passaria, porque Eu Não tenho clareza de que de que Carta está ali e o jogador não tem grande agência, na medida em que é tudo totalmente aleatório. E tiro, não tenho, não tenho escolha. Ou seja, se vamos a falar em termos de desenvolvimento de um produto e que depois seria interessante no para passar uma outra coisa que é okay, se eu vejo isto a ser utilizado. Por jovens, por jogadores. Se o produto for bom se o jogo forem bons o suficiente, eventualmente isto acabará por ser utilizada. É um bocadinho, a dada altura é um bocadinho tipo um dos tabletop famosos, mas que depois é conhecido também por causa disso, em que não há agência do jogador que é o endgame e o mid game do Monopoly. Em que a gente basicamente não pode fazer absolutamente nada. Ou pago renda, ou passos dados a outro jogador. Não há qualquer agência. A minha sugestão não foi pedida, mas se eu dou um baralho de 8 cartas, tivesse uma mão de 3 e eu e cada jogador selecionasse a Carta que quer colocar em jogo. Ou seja, imagina, eu tenho um Ás, um 2, um 7 e uma dama e eu é que escolho. Qual é que coloco sem ver o seu? Depois elas são todas apresentadas ao mesmo tempo. Eu tenho escolha e permite-me trabalhar uma componente estratégica. Ai, eu agora posso perder. Isto é uma dama, alguém vai ganhar, eu posso as cartas altas todas e deixa ficar aqui. Eu quero ter aqui um 2, quero ter aqui um e eu sei que eu vou ganhar estas 3 jogadas Se eu quiser, OK, portanto, isso permite-nos um outro tipo de engagement que depois aí sim gera um uso mais aprofundado da ferramenta. Está muito bem criado, querem termos de layout. Pronto lá está eu, eu eu mudaria aquela questão ali do segundo baralho de cartas, mas isso é uma decisão tua e o que tens funciona, portanto, mas há o facto de poder ter coisas ali à frente, ou seja, consigo estar a jogar e estar a ver um vídeo, estar a estar a ler um livro, se me apetecer aí é essa vertente está muito bem trabalhada, está muito bem implementada.

Pedro Santos 32:04

Obrigado.

Expert A 32:28

E lá está com um tipo de jogo, mais apelo. O apelo dos jogos em sempre uma coisa muito subjetiva, um jogo pode me agradar muito a mim e não agradar outra pessoa, mas com um tipo de jogo mais apelativo eu já ia usufruir da ferramenta, gostei imenso do robozinho, não porque está de robô intrinsecamente, mas porque em termos de desenvolvimento de produto, isto é uma coisa que poderia depois ser fonte de rendimento, tens um robozinho como forma de engagement. Está, OK. Ganhaste 3 jogos um robô passa a ser prateado ou passa a ser dourado. Ganhaste a 3 amigos, ele passou a ganhar um chapéu. Queres meter aqui 10 euros no jogo e deixas de ter um robô, passa a ter um pastor alemão. Principalmente se for visível para os outros, não é? Se for um avatar e cada jogador tem a sua personagem, se for visível para outros, entra no...

Pedro Santos 33:16

Exato.

Expert A 33:37

...no mercado de transações. Isto, sim, pode ser um ferramenta bem interessante. De resto aquilo que me parece que funciona e é mesmo a questão do do jogo que para mim está simplificado demais para no fundo se tirar partido do engagement agora tudo o resto construíste está bastante bom.

Pedro Santos 34:07

O Jogo é casual mas também foi uma ideia desde o início não ser muito complexo, porque também é apenas um protótipo. As cartas são aleatórias, sim, mas também há uma componente de memória, as cartas ganhas são colocadas em determinados sítios.

Expert A 34:36

Ah, sim. Isso ajuda.

Pedro Santos 34:41

Assim, por exemplo, a eu posso se calhar, demonstrar até.

Expert A 34:45

Ah, as cartas são colocadas.

Pedro Santos 34:50

Sim.

Expert A 34:54

Sempre pela mesma ordem? A minha carta, depois a carta do jogador 2 e depois a carta de jogo do jogador 3? São colocadas sempre na mesma ordem?

Pedro Santos 35:05

Sim.

Expert A 35:14

Ou seja, eu tenho a minha carta e é a primeira. Eu joguei um 4 e tenho aqui um 4.

Pedro Santos 35:16

Exato.

Expert A 35:29

Isso poderia ser um ponto de partida muito interessante.

Pedro Santos 35:32

Exato, com essas 3 cartas já não é aleatório.

Expert A 35:45

Certo, ou seja, mas em termos de engagement, nós procuramos tudo aquilo que faça sentir que o jogador tem agência, tem um poder de decisão.

Pedro Santos 35:47

Concordo.

Expert A 35:58

Por exmplo, eu como jogador sou mais esperto do que tu ou eu joguei bem ou eu joguei mal. Efetivamente, podia ter jogado diferente.

Pedro Santos 36:03

Sim.

Expert A 36:05

A batalha, eu chamava lhe batalha, mas também se conhecia por guerra. A gente jogava quando estava a ver televisão e olhava para as cartas, OK, a ganhaste. É um jogo que não envolve, não envolve a minha atenção.

Pedro Santos 36:13

Hmm.

Expert A 36:34

Há uma coisa que a gente tratou no curso que eu fiz de board games, que é ser à prova de telemóveis. Um jogo que exige a minha atenção o suficiente e que eu não vou pegar no telemóvel. Claro que aqui é diferente de um jogo físico.

Pedro Santos 36:52

Pois.

Expert A 37:14

Mesmo a questão de tirar o dado. Eu sei que o dado está ali para um propósito específico e eu apoio essa essa decisão. Mas em caso de empate, fosse um objetivo que eventualmente me permitisse ganhar mais, eu até poderia tentar jogar para o empate. Deixa-me ver. Eu sei que ele apanhou um 2 ao bocado. Será que ele vai jogar 2 agora e eu poderia tentar jogar para o empate e depois o permitir-me a mim guardar, por exemplo, um ás para que sempre que houvesse um empate, eu jogar o Ás para ganhar.

Pedro Santos 37:40

Compreendo.

Expert A 37:55

Era tipo Uno com os +2, onde posso jogar mas depois pode correr mal.

Pedro Santos 37:56

Sim sim.

Expert A 38:09

Portanto, é isso, mas agora parece-me que em termos de desenvolvimento e de projeto, parece uma cena muito bem feita, muito bem feito mesmo.

Pedro Santos 38:19

Eu concordo. Até gostei da parte do do robô. Poderia cada jogador ter o seu próprio um robô. E aí funcionaria como se fosse um avatar.

Expert A 38:34

Exatamente.

Pedro Santos 38:38

Como uma segunda pergunta: este protótipo demonstra eficazmente que esta plataforma é capaz de suportar desenvolvimento de jogos num contexto real?

Expert A 38:58

Sim sim. Parece-me que sim.

Pedro Santos 39:01

Hmm.

Expert A 39:40

É um mercado ingrato. Tu podes ter um grande jogo e se não tiveres o poder financeiro para alimentar o marketing ou sorte ou altura certa de lançamento, o teu jogo pode cair por entre as malhas e nunca mais é descoberto. E aquilo que é muitas vezes utilizado nas minhas aulas chamam-se ou o modelo chinês de desenvolvimento. Por exemplo, ter um estúdio e desenvolver simultaneamente vários projetos ao mesmo tempo. Lanço-os para o mercado e depois em avalia-se a performance deles. Quais deles cortar? Quais deles continuar a desenvolver? Dependendo de como corre. Um jogo pode ser fantástico, mas se não teve adesão do público, eu simplesmente deixo-o cair. E arranco com outro projeto. É um bocadinho, como vai correndo. A gente aposta e às vezes não me parece uma grande ideia. Mas se tem adesão do público.

Pedro Santos 41:35

Exato.

Expert A 41:36

Portanto, e ao mesmo tempo o público alvo é extremamente difícil de identificar no meio disto se não temos a barreira do dinheiro. Não temos a barreira de acessibilidade, não temos a barreira às vezes ou em muitos casos, sequer de leitura de tradução. As ferramentas do Apple Vision Pro podem ser utilizadas para fazer jogos em maior escala e mesmo este jogo pode creser para isso. Pode crescer? Sim, pode crescer exponencialmente. Se tiver um conjunto de pessoas que passem a adotar isto como forma de jogar e de conversar enquanto estão a jogar alguma coisa? Claro que sim.

Pedro Santos 42:35

Exato, então passo a outra questão. Com base neste protótipo e também no conhecimento geral, as funcionalidades únicas do Apple Vision Pro e do ecossistema da Apple apresentam alguma oportunidade para game development neste setor de XR. Comparando também com outras plataformas de realidade estendida. Meta Quest, HTC Vive, Microsoft HoloLens, etc.

Expert A 43:27

Aquilo que me saltou à vista.

Pedro Santos 43:31

Exato.

Expert A 43:33

Gostei do eye-tracking, gostei. Pode ser dificil de calibrar. Eu trabalhei um pouco com o Tobii na altura. Pode ser bastante difícil de calibrar e tirar partido, mas ao mesmo tempo gostei muito de ver essa implementação no teu protótipo. Gostei bastante disso. Foi aquilo que me chamou mais a atenção. Depois a parte de mudar os fundos, de AR para VR.

Pedro Santos 44:08

Exato.

Expert A 44:16

Sim, é interessante. Acho alguma piada ao HoloLens mas considero aquilo muito, muito gimmicky. É aquela coisa que okay, experimento uma vez e já experimentei. Já tirei aquilo que eu poderia retirar desta experiência. Não, não desenvolve muito mais.

Pedro Santos 44:38

Okay e por último, que oportunidades e desafios vê no design e implementação de jogos para visioOS? Algo que salta à vista, algum desafio, alguma oportunidade no desenvolvimento de jogos para a plataforma com base no protótipo.

Expert A 45:03

Para utilizar esta tecnologia em específico?

Pedro Santos 45:07

Sim.

Expert A 45:09

Muita coisa mesmo. Se calhar não ia para jogos muito densos. Mais para jogos em torno de cartas que não tenham muitos elementos no ecrã. Não diria Magic, por exemplo. Lembro-me de um que poderia ser muito interessante, que também era do Richard Garfield, o Spectromancer, ou seja, jogos com um interface mais balizado. Magic chegou a fazer isso com Duels of the Planeswalkers, em que só há aquelas cartas normais e a meta está confinada de certa maneira, para limitar aquilo que a gente tem em jogo. Mas sim. Isto seria, por exemplo, muito interessante fazer um espécie de um Ticket to Ride ou um micro Ticket to Ride para passar para este tipo de meio. Pode ser, sem dúvida, muito interessante. Agora, também está-me a ser difícil sair um pouco do contexto de board games ou similares. Eu acho que teria que usar mesmo para ver, OK, será que isto se aplicaria para um Baldur's Gate.

Pedro Santos 47:03

Sim.

Expert A 47:07

Se calhar um Baldur's Gate não mas um dos que saiu aqui este ano. O Clair Obscur: Expedition 33 em que eu consigo selecionar esta vai ser a magia que eu vou fazer. E agora tenho que clicar aqui na altura em que eu tive que reagir. Pode eventualmente ou poderia eventualmente ser utilizado para isso porque são jogos lá está, eu só tenho aquilo, eu só tenho uma luta. E escolho as minhas habilidades, está aqui. É isto que eu vou fazer. Isso poderia ser eventualmente aplicado, que é o, aliás, o que o que alguns estão a tentar recriar. Não, não com esses IPs, não com esse tipo de desenvolvimento, mas temos temos várias empresas a fazer esse tipo de jogos não especificamente para o Apple Vision Pro. Teria esse potencial. Em termos de em termos de desafios. É aquilo que eu já disse, eu vejo um risco enorme de uma empresa depositar os ovos, ou um estúdio, depositar os ovos todos no só para o desenvolvimento em Apple Vision Pro. Porque ou há algo que mobilize o apoio do desenvolvimento de um número grande de jogadores ou corre-se o risco de termos um produto que pode ser fenomenal e simplesmente não atinge o público não é? E depois ainda temos uma outra questão que é, como é que eu vou monetizar isto? E para a gente entrar em micro-transações. Nós precisamos de uma massa crítica muito grande. Nós precisamos do número de jogadores muito grande, OK? Portanto, sei lá, se PUBG vai buscar milhares de milhões por ano em micro-transações com o PUBG Mobile, por exemplo. E com o PUBG: New State, mas estamos a falar de um número de jogadores absolutamente gigantesco, difícil de de quantificar.

Pedro Santos 49:40

Claro.

Expert A 49:45

Portanto, e nesses casos é possível porque eu através de meio cêntimo em meio cêntimo, conseguimos efetivamente, construir uma fortuna porque eu tenho essa base de jogadores. Portanto, o grande desafio é isso. É conseguido atingir a massa crítica e uma estratégia de monetização que não afaste os novos jogadores. Ter conteúdo que mantenha os jogadores empenhados e motivados para continuar a jogar, a querer ver o que aí vem a seguir. Isso é muito, muito, muito difícil de se construir. E estamos a falar de uma coisa que depois é um processo que vai ser iterado ao longo dos anos. Eu acho portanto, que o PUBG saiu em 2017 e o PUBG Mobile em 2019. E continua a render. Fruto também daquilo que, aí sim poderemos falar de oportunidades, que é uma coisa que vai um bocadinho contra nossa cultura, nossa cultura Europeia, em que para nós o quarto típico de um jovem. Por exemplo, no quarto dos meus filhos. Cada um tem o seu quarto, tem uma secretária, dá para ter um portátil ou um computador lá em cima. E se a gente for olhar para a China, para o Japão, para a Índia, não é tanto assim. Eles viram-se muito para isto, viram-se para os telemóveis também. OK, portanto, para nós, a toda a gente tem computador.

Pedro Santos 51:06

Certo.

Expert A 51:11

Mas nem toda a gente tem computador. Espero ter conseguido responder a tua pergunta.

Pedro Santos 51:18

Já foi bastante contexto, bastante informação.

Expert A 51:25

Olha mas todos os meus parabéns. Estás a fazer qual mestrado?

Pedro Santos 51:30

Engenharia informática, engenharia de software.

Expert A 51:34

Informática, programação. Nota-se que tem muito, muito trabalho de programação, muita coisa interessante do ponto de vista tecnológico por trás.

Pedro Santos 51:34

Obrigado.

Expert A 51:42

E gostei, gostei muito desse trabalho de programação por trás. E que linguagem utilizaste para isso?

Pedro Santos 51:49

Foi maioritariamente first party. Usei Swift como linguagem e SwiftUI como framework de UI. Depois outros frameworks como o TabletopKit, RealityKit e ARKit. Tudo frameworks da Apple. Foi um bocadinho para tentar explorar a plataforma e as vantagens que eles tentam trazer com os seus próprios frameworks. A tecnologia é nova. Apesar da linguagem ser usada para iOS, visionOS tem formas de interagir completamente diferentes.

Expert A 52:37

É isso, foi muito giro. Estranhei, lá está limitação para 3 jogadores, não é um modo muito comum, mas tudo o resto, partes de princípios muito interessantes.

Pedro Santos 52:58

Obrigado.

Expert A 53:17

Em termos de design está simples, mas está muito eficiente naquilo que busca fazer e tu tens algumas noções que gostei de ver. O design do jogo é escalável, daí eu ter estranhado porque só 3, isto dá para 10 ou 20 em simultâneo.

Pedro Santos 53:20

Sim, o jogo é escalável. O número de cartas é virtual e escalável. Escolhi fazer com cartas normais para permitir o jogador ter house rules. Por exemplo, jogar com um deck físicos e um deck virtual ao mesmo tempo. Como tem o pass-thorugh dá para ver os objetos a volta. Se quiserem, por exemplo, estarem localmente a jogar com a parte visual e física ao mesmo tempo.

Expert A 54:04

Ou seja, para demonstrar as funcionalidades da platforma.

Pedro Santos 54:48

Exato.

Expert A 55:04

Tipo, eu posso fazer isto, eu posso fazer isso, eu posso fazer aquilo. Okay, deixa-me ver como é que eu consigo demonstrar isto em funcionamento.

Pedro Santos 55:20

Exato, a minha ideia era um bocado explorar a tecnologia.

Expert A 55:46

Hmm.

Pedro Santos 55:50

Para ver a viabilidade em game development. Também algumas decisões foram tomadas nesse aspecto.

Expert A 56:10

Pronto, é válido e funciona, Ficou fixe.

Pedro Santos 56:15

Da minha parte eram, eram só essas questões, mais alguma questão?

Expert A 56:20

Olha dou-te os parabéns, dou-te os parabéns. E muita sorte no mestrado.

Pedro Santos 56:20

Hoje, mais tarde, amanhã envio um pequeno questionário, Não é nada de demais, é só para recolher algumas dados sobre a tecnologia e a entrevista.

Expert A 56:29

Claro, muito bem.

Pedro Santos 56:34

Portanto, agradeço.

Expert A 56:39

Valeu, felicidades.

Pedro Santos 56:39

Agradeço imenso a disponibilidade e entrevista. Obrigado.

Expert A 56:49

Um abraço, até logo.

Pedro Santos 56:54

Um abraço.

Expert A 56:58

Tchau.

Pedro Santos 56:59

Obrigado.

Pedro Santos stopped transcription

C.2 Expert B

[Introduction and asking for consent before starting the recording and transcription.]

Pedro Santos started the transcription.

Pedro Santos 0:04

A gravação começou.

Esta entrevista relacionada com a minha tese, como mencionei, que está relacionada com a exploração do Apple Vision Pro e do visionOS.

Expert B 0:04

Força.

Pedro Santos 0:17

São plataformas novas de extended reality e spatial computing. Explorei numa vertente de game development. No âmbito do estudo, desenvolvi um protótipo para avaliar um bocado as features e algumas limitações encontradas no desenvolvimento para esta plataforma. Com isso, ia fazer uma demonstração desse protótipo e algumas questões depois. Alguma questão antes de começar?

Expert B 0:51

Não, não.

Pedro Santos 0:52

Então, passo a partilhar o meu ecrã.

Expert B 0:53

Força.

Pedro Santos 0:57

Dá para para ver?

Expert B 1:00

Sim, sim, consigo ver tudo.

Pedro Santos 1:01

É exato há aqui uma limitação que está relacionada com esta plataforma ainda não ter sido lançada em Portugal. Eu mencionei isso no email que enviei. Então a demonstração tem que ser a base de simulador, o que restringe um bocadinho a demonstração.

Expert B 1:18

Claro.

Pedro Santos 1:19

Há que possa fazer.

Expert B 1:19

Portanto, isto que nós vamos ver na simulação, veríamos com os nossos óculos de XR?

Pedro Santos 1:30

Sim. Mencionar também que o jogo que eu fiz foi um tabletop e explorando um bocadinho o framework TabletopKit da Apple. Eu tentei explorar ao máximo tudo aquilo que a Apple lança com visionOS. Para avaliar a maturidade da plataforma.

Expert B 1:38

OK, então é mesmo especificamente para o Vision Pro.

Pedro Santos 1:54

Exato, é escrito em Swift como linguagem de programação e SwiftUI, tudo first party da Apple.

Expert B 1:54

OKOK. Sim, sim.

Pedro Santos 2:02

E o TabletopKit é mesmo para jogos de tabuleiro, é vocacionado mesmo para isso. Ou seja, num contexto real, este ambiente é do simulador. Ou seja, seria o contexto real da pessoa, as cadeiras da mesa, tudo da casa, exceto esta board azul. Tudo está em cima da borda, o robozinho e as cartas ,etc. é mesmo do Apple Vision Pro.

Expert B 2:09

Sim, certo. OKay, estou a perceber.

Pedro Santos 2:33

Aqui está também uma limitação de mostrar num simulador.

Expert B 2:37

Claro, claro, mas pronto.

Pedro Santos 2:42

Não não sendo lançado em Portugal, também não havia outra outra hipótese. Falando aqui um bocadinho do jogo. Como eu disse, é um jogo de tabuleiro à base de cartas, também têm um dado aqui, tem uma tab bar que tem algumas ações rápidas que o jogador pode fazer. Mudar algumas settings, etc. Eu também vou demonstrar.

Expert B 3:07

OK.

Pedro Santos 3:09

O Jogo em si é um jogo simples, é um jogo de cartas. O conceito e as regras específicas fui eu que estive a criar com base em jogos já existentes de jogos tradicionais, de cartas, etc. E mas o jogo não é um transição de um para um de um jogo de tradicional e tem algumas algumas coisas específicas também para para tirar partido um bocadinho desta desta tecnologia.

Expert B 3:31

Hmm.

Pedro Santos 3:38

O Apple Vision Pro usa eye-tracking e reconhecimento de gestos. No simulador usa-se o rato para interagir com as cartas mas num contexto real seria o utilizador a olhar para a carta. Depois para ir buscar seria com a sua mão e ir lá ao local.

Expert B 3:57

Sim.

Pedro Santos 4:02

A aplicação também permite gestos indiretos com o UI, ou seja, o jogador pode estar com a mão para baixo. Não precisa de ir exatamente ao local. As regras do jogo são simples, cada jogador tem uma hand à frente com várias cartas, neste caso 7. A frente, tem um sítio para jogar uma carta e tem um stack aqui com várias cartas. Em cada ronda. cada jogador joga uma carta sem ver o valor, elas estão viradas para baixo. Quando o jogador joga a carta, ela vira-se ao contrário. Uma vez que os jogadores joguem a sua carta, não podem retirá-la. Depois de todos os jogadores jogarem, um vencedor é escolhido. A carta com menor valor ganha. Neste caso, é um baralho de cartas normal. Cartas numeradas, o valor é o número. Também tem Damas que vale 1 e o Ás é a melhor carta, que vale zero. Ou seja, o número mais baixo fica com as cartas dos adversários e adiciona à sua hand, que vai crescendo.

Expert B 5:02

OK, então não tem, não tem valetes e reis.

Pedro Santos 5:07

Não, não. Foi uma decisão minha.

Expert B 5:08

OK.

Pedro Santos 5:10

Não é que tivesse alguma limitação da plataforma, foi mesmo uma decisão de design.

Expert B 5:17

Era para haver menos número de cartas com valor de um.

Pedro Santos 5:21

Exato, eu considerei ter todas as cartas de imagens com o mesmo valor ou arranjar valores diferentes, mas depois também podia levar a confusão ao jogar. Não em termos de implementação, mas em termos de onboarding, regras e assim. Depois de cada jogada, o vencedor recebe as cartas. Pode haver um empate, duas ou três pessoas com cartas iguais, nesse caso será resolvido por este dado.

Expert B 5:29

Sim.

Pedro Santos 5:49

Cada jogador que empatou, lança o dado. Quem tiver menor valor, recebe as cartas. Se empatarem outra vez, repetem o processo até alguém eventualmente terem valores diferentes. Depois de cada ronda, vão buscar uma carta ao stack no centro da Mesa. O jogo continua, jogadores são eliminados quando já não há cartas para ir buscar ao baralho e um jogador ganha quando eventualmente os outros 2 jogadores forem eliminados. É assim, as regras são bastante simples, como eu disse.

Expert B 6:40

OK.

Pedro Santos 6:41

Também aqui tenho uma ação na toolbar que é uma maneira de cada jogador ver as regras a qualquer momento. Pode posicionar a view enquanto ele está a jogar, está a ver as regras, o que ajuda um bocadinho onboarding, penso.

Expert B 6:49

Sim sim.

Pedro Santos 6:50

Depois este jogo tem 2 vertentes, uma vertente single player e outra mais de multiplayer. Também depois permite partilhar as regras no multiplayer, se quiserem, em conjunto várias pessoas ao mesmo tempo.

O jogo é anchored. Ou seja, se o jogador o posicionar num sítio e for embora, neste caso, se for à cozinha e voltar, o Jogo permanece no mesmo sítio.

Expert B 7:18

Hmm.

Pedro Santos 7:19

No simulador não é tão graceful a movimentação.

Expert B 7:26

Pois.

Pedro Santos 7:29

Em termos do jogo em si, tem este robô que está aqui à frente. Ele funciona um bocadinho como um DJ ou helper que vai reagindo. Quando há alguém que ganha reage diferente de quando alguém empata, com animações e sound effects. Eu acho que estou a partilhar o som.

Expert B 7:49

Ah, deixa ver se consigo ouvir o som.

Pedro Santos 7:52

Dá?

Expert B 7:54

Ok, estou a ver, estou a ouvir. Sim, sim.

Pedro Santos 7:54

Exato eu não pus música mesmo porque este jogo seria vocacionado para a parte multiplayer, para conversas, etc.

Expert B 8:05

Portanto, ele é uma aplicação. No fundo, nós conseguimos abrir depois ao lado ter outra a tocar música não é?

Pedro Santos 8:13

Exato exato. É isso mesmo. Isso era uma parte que depois ia demonstrar. Eu posso mostrar agora, mas sim, tem multitasking. Ou seja, indo por exemplo, ao Safari. Eu acho que já tenho aqui aberto.

Expert B 8:14

OK.

Pedro Santos 8:26

Portanto, pode-se ouvir música ou ver um filme.

Expert B 8:34

Oh.

Pedro Santos 8:41

Metendo em full screen, ele baixa a luminosidade do resto da sala, mas mantém o jogo a funcionar.

Expert B 8:46

Muito bem. E dá para partilhar?

Pedro Santos 8:47

Como é uma aplicação da Apple e tem integração com o FaceTime e SharePlay, dá para partilhar a música. O safari, Apple Music e outras apps podem ser partilhadas, incluindo o jogo. Mas eu já mostro isso. Permite fazer multitasking de séries se for através da aplicações de streaming. Se for esse o uso que se aqui quiser.

Expert B 9:14

Hmm.

Pedro Santos 9:28

Eu acho que mostrei aqui um bocadinho a jogar e o Robot reage com um sons, não sei se são está a passar corretamente.

Expert B 9:30

O som está perfeito, sim.

Pedro Santos 9:36

Como o jogo é aleatório, não tenho muito controlo para forçar um empate. Bem, aqui está um. Até correu bem.

Então, ele reage ao empate. Os jogadores que empatam interagem com o dado com gestos e quando lançam, o dado mostra o valor.

Expert B 10:08

Okay.

Pedro Santos 10:09

O fast mode passa a ser um bocadinho mais rápido e melhor para demonstrar. Vou jogando... Tirar cartas do stack. E pronto, fui eliminado. Portanto, também tem um som de efeito de eliminação. O jogo não é completamente aleatório e tem aqui um bocadinho de skill. O jogador tem que ter um bocadinho de memória. Tem aqui uma componente de jogo de Memória, porque elas são viradas para baixo mas o jogador que ganha recebe as cartas numa determinada ordem. O jogador pensa que uma carta está num sítio mas pode não ser assim.

Expert B 10:46

Oh, percebo. Pronto. Realmente há uma coisa que eu acho que ainda não percebi, ou seja, tu tens que jogar a carta mais baixa, mas quando ganhas uma ronda tu agregas a carta dos outros.

Pedro Santos 11:06

Sim.

Expert B 11:07

OK.

Pedro Santos 11:07

Por exemplo, ao jogar outra vez, aqui o jogador 3 vai ganhar e fica com as cartas dos outros. Depois, essas cartas vão numa ordem para a hand dele. Primeiro a dele edepois as dos outros.

Expert B 11:17

Esse fica com as cartas que foram jogadas, ou seja, tu depois já tens informação sobre a mão do adversário.

Pedro Santos 11:19

Com as cartas dos outros 2, sim.

Exato e depois começa um bocadinho um jogo mental de memória, porque não se consegue ver exatamente quais são mas pode-se ir seguindo as cartas ganhas e decorar.

Expert B 11:27

Sim, muito bem.

Pedro Santos 11:34

É para não ser completamente random essencialmente.

Expert B 11:37

Claro, mas depois, quando é que se ia buscar ao stack?

Pedro Santos 11:38

Era depois de cada ronda. Eu neste caso não, não tem stack porque está no jogo rápido.

Expert B 11:46

Neste jogo mais rápido, ia se ao dado?

Pedro Santos 11:48

Isso seria jogado em caso de um empate.

Expert B 11:52

E quem ganhar depois vai ao stack?

Pedro Santos 11:56

Não, depois de cada ronda, cada um vai buscar ao stack.

Expert B 12:00

Ah, toda a gente ganha uma carta no fim.

Pedro Santos 12:03

Exato e depois continuam as rondas. Continuam até não haver cartas no deck e aí sim começam a ser eliminados.

Expert B 12:04

OK, certo.

Pedro Santos 12:09

Estava no jogo mais rápido. Com 3 cartas parece que é muito rápido, mas às vezes há ali um momento em que se ganha e perde que pode ainda demorar.

Expert B 12:14

OK.

Pedro Santos 12:23

Foi uma decisão minha que achei interessante para o protótipo em si.

Expert B 12:31

Sim.

Pedro Santos 12:31

Neste caso mostrei até alguém ganhar. O Table top permite restringir a interação de cada utilizador com os elementos do jogo. Regras específicas, por exemplo. No meu caso, se eu sou este jogador, não consigo interagir nem pegarem nas cartas do jogador 3 para não haver batota. E pegando numa carta, é restringido onde pode colocar a carta. Portanto, neste caso, tirando aqui uma carta não a posso pôr no baralho de outro jogador mas posso meter no meu próprio, etc. Permite fazer esta restrição, mas para permito os jogadores implementarem algumas house rules. Se as pessoas estiverem a jogar e quiserem definir certas regras ou até a parte do empate, podem. Por exemplo, não querer usar dados para o empate, usar outra forma de empate, etc. Tem aqui um toggle na toolbar que permite desativar as regras e agora já consigo pegar no que quiser. Ou seja, aqui os jogadores jogam, pegam manualmente, é o modo mais manual que permite algumas house rules e as regras do jogo não são forçadas.

Expert B 13:33

Ah OK. Sim.

Pedro Santos 13:48

E assim fica com um jogo mais genérico.

Apesar de ainda ter alguma alguma lógica e assim. Mas fica mais genérico e as pessoas têm liberdade.

Expert B 13:54

Muito bem.

Pedro Santos 14:00

A toolbar também permite aos utilizadores de mudar o background para um background virtual em vez de estar mais AR como está agora. Eu deixei neste modo mais progressivo que ainda mostra o ambiente perto do utilizador por uma questão de normalmente ser jogado perto de uma mesa. Para os jogadores não bater, etc. Pode ser completamente virtual. Pode ser completamente VR. E pode ser desligado a qualquer momento. Por último, isto seria o modo single player. Como falei, tem uma componente de multiplayer através do SharePlay do Facetime. Ou seja, se eu começar aqui uma chamada com 2 outros participantes no FaceTime que tenham os seus próprios Apple Vision Pro, irão aparecer a flutuar. A ideia seria fazer esta integração com first party tools para explorar aqui um bocadinho a integração. Porque uma das vantagens do Apple Vision Pro é uma coisa que chamam de spatial personas, que são basicamente modelos 3D da pessoa.

Expert B 15:38

Interessante.

Pedro Santos 15:38

São completamente 3D, seguem as expressões das pessoas, movimentos da cabeça e neste caso, como é um simulador, mais uma limitação, aparecem estes test users com emojis, mas com esta implementação e 2 ou 3 devices reais, aparecia mesmo a cara da pessoa e os gestos movimentando as cartas, etc. Neste caso, eu comecei a chamada e aparece um elemento na aplicação a dizer que não está shared. Com algumas opções como fazer share automático. Clicando aqui ativa o multiplayer do jogo. Tinha um alerta antes de partilhar. Ele começa um jogo em multiplayer, ou seja, as pessoas que estão que estão na chamada FaceTime irão entrar em cada um dos seats. Nesse caso, a limitação é que eu não tenho acesso a estes utilizadores para aceitar o convite e entrar. Então eles ficam como espetadores.

Expert B 16:42

Hmm.

Pedro Santos 16:45

Ou seja, isto permite ter 5 pessoas na chamada, com 3 pessoas a jogar e 2 a espectar. Os espetadores apareciam aqui como estava antes. O game state é sincronizado e as animações são realizadas localmente. Por exemplo, aqui ao utilizador mostra que está a ser partilhado esta aplicação. A aplicação neste caso chama-se AceR.

Expert B 17:11

Sim.

Pedro Santos 17:12

Permite ver alguma informação sobre a call, adicionar pessoas, convidar pessoas, por exemplo. E convidando uma pessoa, ela é convidada para a call FaceTime e, consequentemente para o Jogo. Também permite partilhar um link por mensagem, etc. Ou seja, usa o facetime como forma de criar uma vertente multiplayer através do SharePlay, com convites, esta parte das spatial personas e assim. A qualquer momento, o utilizador pode parar de partilhar o jogo. Também já mostrei a parte Multitasking, isso também é permito em multiplayer como mencionei. E até se mostrar aqui as regras outra vez. Neste caso, as regras seriam locais para cada jogador, ou seja, se algum jogador meter num sítio específico. Fica só para ele localmente. Mas também partilhar a window numa chamada facetime. Da parte da demonstração é basicamente isto. Alguma questão, alguma coisa que eu possa demonstrar?

Expert B 18:45

Não, eu acho que percebi o conceito todo, sim.

Pedro Santos 18:50

Hmm.

Expert B 18:52

Parece me que está a fazer bom uso das ferramentas do Vision Pro.

Pedro Santos 18:53

Obrigado.

Expert B 18:56

Está interessante.

Pedro Santos 18:59

Ok, então eu se calhar passo aqui para uma vertente com algumas questões.

Expert B 19:04

Hmm.

Pedro Santos 19:06

Para recolher alguns dados.

Expert B 19:09

Certo.

Pedro Santos 19:19

A primeira questão era se a implementação deste jogo específico em realidade estendida oferecia algum valor único em comparação com o mesmo jogo num ambiente físico, um jogo com cartas e dados reais. Este jogo é possível jogar fisicamente com as mesmas regras.

Expert B 19:41

Ah sim, claro que oferece. Primeiro que tudo, não tens que fazer um setup. Este jogo em si até não é um jogo muito complexo. Alguns board games são, mas mesmo assim, neste também precisas de um dado, precisas de um baralho de cartas. Há sempre algum setup, não é? E poder fazê-lo automaticamente só com o Apple Vision.

Pedro Santos 20:05

Exato.

Expert B 20:12

Parece-me vantajoso. Para além disso, também o próprio jogo sendo automático também acaba por ser mais rápido, não é? E consegue o próprio jogo verificar as regras por si só. Não tenho que passar por um ser humano, apesar que tu dás a opção também de jogar manualmente, o que é bastante interessante. Acho que é uma boa adição ao jogo. Mas sim, eu não sei quanto é que me devo estender nas respostas, porque se calhar já posso dar respostas a outras perguntas ao estender muito nesta primeira resposta.

Pedro Santos 20:51

À vontade, à vontade eu depois também faço tratamento das respostas.

Expert B 20:55

OK mas sim, acho que foi foi um bom, um bom uso das ferramentas. Outra coisa que tu tens também é o robozinho. É algo que consegues fazer aqui num ambiente virtual que não conseguirias num ambiente físico tão facilmente. E pronto, assim como poderias adicionar efeitos visuais extra. Efeitos sonoros também passam. São coisas que só podemos ter um jogo digital. E não num jogo físico.

Pedro Santos 21:44

Exato, passo então para minha segunda pergunta. Este protótipo demonstra eficazmente que o Apple Vision Pro é capaz de suportar desenvolvimento de jogos no mundo real?

Expert B 22:01

Hmm, mas o que quer dizer mundo real aqui nesta pergunta?

Pedro Santos 22:06

Num contexto de desenvolvimento para o mercado.

Expert B 22:13

Ah sem dúvida, sem dúvida.

Pedro Santos 22:16

Com base em tudo da plataforma.

Expert B 22:19

Com base no que eu vi agora, mas também com base no conhecimento.

Pedro Santos 22:22

Exato.

Expert B 22:27

Pronto, tenho alguma familiaridade com equipamentos de VR e claro que tem imenso potencial.

Pedro Santos 22:33

OK, obrigado. A terceira pergunta seria: com base no protótipo ou talvez em algum conhecimento existente sobre a plataforma.

Expert B 22:52

Hmm.

Pedro Santos 22:52

As funcionalidades únicas do Apple Vision Pro e, por exemplo, da sua integração com sistemas Apple, com o ecossistema Apple. Como o FaceTime que mostrei, por exemplo. Se representam uma oportunidade para desenvolvimento de jogos em comparação com outras plataformas XR?

Expert B 23:18

Ou seja, se a integração dos frameworks da Apple e do Apple Vision Pro tem mais valias em comparação a outras frameworks de desenvolvimento de jogos, é isso?

Pedro Santos 23:31

Sim, um bocadinho uma comparação entre esta plataforma com as suas funcionalidades únicas com outras plataformas existentes de realidade estendida.

Expert B 23:45

Sim, eu penso que sim, acho que é algo que pode estar ainda um pouco explorado. No exemplo que tu mostraste o teu protótipo., dá para entender que talvez uma das melhores mais valias é mesmo esta fluidez de sistemas, fluidez de aplicações, portanto tu estás no jogo, mas ao mesmo tempo estás no YouTube. Não é comum num dispositivo mais tradicional em que só conseguimos realmente ter uma aplicação aberta ao mesmo tempo, não é? Conseguimos ter ali tudo a acompanhar um bocado. Quer dizer, eu posso estar aqui a jogar este jogo e por um vídeo atrás com uma lareira a arder e fazer companhia. Portanto, acho que isso aí tem tem bastante mais valias e pode ser explorado de uma forma muito interessante.

Pedro Santos 24:55

Exato, perfeito. Última pergunta da minha parte. Que oportunidades e desafios podem ser identificados no design e implementação de jogos para esta plataforma?

Expert B 25:19

Há oportunidades e desafios.

Pedro Santos 25:20

No design ou implementação.

Expert B 25:30

Sim realmente podemos ir em muitas, muitas direções. Mesmo desafios, se calhar é mais fácil. Acho que em termos de usar multiplayer, tentar simular em multiplayer a presença de um outro user. Tudo nesta integração, parece que é realista o suficiente que não haja uncanny valley. Porque aqui, quando estamos a falar de dispositivos de VR ou AR, já há uma aproximação ao real. Não é a mesma coisa que estar, por exemplo, no Discord, jogar com amigos ou assim e sei que eles estão noutra casa e tudo bem. Acho que o nosso cérebro aceita isso muito facilmente, mas depois, quando começamos a colocar as pessoas já com avatares 3D ao nosso lado e assim, já entra neste território.

Pedro Santos 26:13

Claro.

Expert B 26:34

E pode estragar aí um bocado a fantasia, portanto, a simulação. E eu falo aqui no multiplayer porque eu sei que multiplayer tem historicamente muitos problemas, mesmo em qualquer tipo de dispositivo.

Pedro Santos 26:49

Sim.

Expert B 26:53

Acho que pode ser um desafio, pronto. É a mesma adopção aqui de um dispositivo que funciona diferente de todos aqueles que nós estamos habituados e pode trazer alguma dificuldade para alguns tipos de utilizadores. Mas em termos de oportunidades, eu acho que são tão vastas que é que é difícil de definir. Mas o teu exemplo de um jogo board game, acho que é bastante interessante, com tudo que eu estava mencionar de não ter que fazer setup. Porque alguns bordo games tem um setup muito grande, de 30 ou 40 minutos. É bom. Daqueles que pronto, também são supostos durarem 6 horas, mas nunca é divertido. Portanto, acho que passar essa parte à frente pode ser muito fixe, pronto. E depois também tem o arrumar, não é? Depois, ninguém quer arrumar e assim já fica arrumado automaticamente.

Pedro Santos 27:58

Exato.

Expert B 28:02

E é isso, acho que nós já tínhamos falado antes, o facto de ter uma personagem ao lado, ter um cenário a acompanhar. Poderia ser até mais. Menos intrusivo do que o cenário que tinhas colocado ali como opcional. Portanto, teríamos na nossa sala, mas estamos a acrescentar alguns detalhes. Por exemplo, as paredes. Imagina que é um jogo medieval assim de board game, mas com tema medieval. Depois de repente começam a aparecer paredes de masmorra e assim, mas estão no mesmo sítio.

Pedro Santos 28:32

OK, sim.

Expert B 28:58

E se calhar assim haver uns monstrinhos assim ao longe, pode ser engraçado em vez de mudar completamente de cenário à volta, percebes? Acho que isso pode ser uma boa evolução.

Pedro Santos 29:48

Sim sim.

Expert B 29:56

Aquele misto entre VR e AR, tipo, acho que pode ser fixe.

Pedro Santos 30:15

Da minha parte, em termos de questões, seria só isso. Mais alguma coisa a apontar? Mais algum observação ou alguma questão?

Expert B 30:47

Acho que pode ser aqui uma boa opurtunidade porque há sempre óculos a aparecer. Há alguns populares e assim, mas nenhum se tornou indiscutível. Aconteceu com outras áreas e quem sabe, pode ser o da Apple. Pronto, e quem forem os primeiros a ter experiência, já tem uma vantagem sobre os outros.

Pedro Santos 35:00

Claro, é um bocado por isso que este estudo existe. Para avaliar a viabilidade da platforma e ver se faz sentido ser early adopter.

Expert B 35:15

Isso. Sim, mas é muito interessante, muito. Espero que tudo corra bem com o teu mestrado.

Pedro Santos 35:34

Obrigado igualmente e agradeço a disponibilidade. Muito obrigado. Eu também vou mandar hoje ou amanhã um pequeno questionário. São 3 perguntas para recolher alguns dados da entrevista.

Expert B 35:36

É isso, olha muito bem.

Pedro Santos 35:54

E depois envio por e-mail.

Expert B 35:57

Obrigado, abraço.

Pedro Santos 35:57

Obrigado com licença. Igualmente, abraço, abraço.

Pedro Santos stopped transcription

C.3 Expert C

[Introduction and asking for consent before starting the recording and transcription.]

Pedro Santos started the transcription.

Pedro Santos 0:03

Já está a gravar. Então, dando um bocadinho contexto, esta entrevista é no âmbito da minha tese de mestrado. Que estou a desenvolver também em parceria com a Mindera. Em que estou a explorar e avaliar o Apple Vision Pro numa perspetiva de game development.

Então, fiz um protótipo de um jogo utilizando first party tools disponibilizadas pela Apple.

Expert C 0:36

Muito bem.

Pedro Santos 0:36

Também usei um dos frameworks que eles têm para o desenvolvimento de jogos de tabuleiro, o TabletopKit.

Expert C 0:44

Hmm.

Pedro Santos 0:44

Se calhar, vou começar com a demo e partilhar o ecrã. Dá para ver?

Expert C 0:59

Sim, dá.

Pedro Santos 1:01

Está fluido?

Expert C 1:06

Está aceitável para o que é uma partilha de ecrã, portanto, está tudo bem.

Pedro Santos 1:10

OK, então vou explicar as regras do jogo. Não é muito complexo, são bastante simples. É um jogo de cartas e também têm um dado. É um jogo para 3 jogadores e cada jogador tem uma mão com cartas viradas para baixo, que são estas cartas que tenho aqui.

Há um sítio para jogar uma carta por cada ronda, um baralho central, o dado e este robô vai reagindo ao que se vai passando. Funciona como um DJ também com efeitos sonoros. Cada jogador escolhe uma carta sem saber o seu valor e joga neste sítio aqui.

Expert C 1:41

Certo.

Pedro Santos 1:47

Depois de um jogador jogar a sua carta, esta vira-se ao contrário e dá para ver o seu valor. Quando a carta é revelada, o jogador não pode retirar a carta.

Expert C 1:54

Hmm.

Pedro Santos 1:57

Depois de todos os jogadores jogarem as suas cartas, a carta de menor valor ganha. O jogador que ganha fica com as cartas que foram jogadas e adiciona-as à sua mão. Ou seja, a sua mão vai crescendo.

Expert C 2:08

Diz-me uma coisa?

Pedro Santos 2:08

Força.

Expert C 2:10

Desculpa, só porque cortou um bocadinho mas disseste que o jogador com a carta com o menor valor ou maior?

Pedro Santos 2:16

Menor valor.

Expert C 2:17

OK.

Pedro Santos 2:18

Ou seja, é um baralho de cartas normal com as cartas numeradas, um Ás e uma Dama. As cartas numeradas é o valor do número, por exemplo, 3 ganha a um 4, etc. A melhor carta é o Ás com zero de valor e a Dama tem 1 como valor.

Expert C 2:21

Ok, certo.

Pedro Santos 2:40

Depois de cada ronda, todos os jogadores retiram cartas do stack central.

Expert C 2:50

Certo.

Pedro Santos 2:50

As rondas vão continuando e os jogadores são eliminados quando já não tem cartas na mão nem no stack.

Um jogador ganha quando os outros 2 jogadores forem eliminados.

Expert C 3:02

Sim.

Pedro Santos 3:02

E o jogo é basicamente algo deste género.

Tenho um modo mais rápido, que são apenas 3 cartas por jogador. E aí o baralho central não está disponível. É para ser mais situações mais rápidas. Vou demonstrar agora.

Expert C 3:12

Certo.

Pedro Santos 3:23

Neste caso, esqueci-me de mencionar ao início, mas há uma limitação da plataforma ainda não estar disponível comercialmente em Portugal, ou seja, não há maneira de testar fisicamente.

Expert C 3:33

A sério? Coitado de ti para testar isso.

Pedro Santos 3:36

Pois é, inicialmente, o plano era ter um device físico, só que os planos da Apple mudaram e eles não chegaram a lançar em maioria dos países. Só nos principais. Acho que aqui na Europa, só em França, Alemanha e Reino Unido tiveram, mas o resto já não.

Expert C 3:49

Hmm.

Pedro Santos 3:56

Ou seja, é uma limitação, porque tenho que demonstrar num simulador e não posso fazer user testing.

Expert C 3:56

Pois.

Pedro Santos 4:02

Não sei se o meu rato está a ser partilhado.

Expert C 4:06

Sim está, está, eu consigo ver consigo.

Pedro Santos 4:07

O ambiente aqui à volta é o ambiente real da pessoa.

E este esta board azul e tudo está em cima é o meu jogo digital no Apple Vision Pro.

Expert C 4:18

Certo.

Pedro Santos 4:21

Através do pss-trhough do headset.

Expert C 4:21

Sim sim.

Pedro Santos 4:26

A interação neste caso, eu estou a fazer com rato mas num contexto real, o jogador olhava para cada carta. Acho que dá para perceber que a carta está a ficar mais clara, certo?.

Expert C 4:40

Sim, dá.

Pedro Santos 4:42

Pronto, isso seria apenas como o eye-tracking do Apple Vision Pro. Depois com os gestos, por exemplo, um gesto como eu fiz agora. [Pinch gesture]

Expert C 4:44

OK, sim.

Pedro Santos 4:51

Pegavam na carta e moviam-na para o local indicado.

Expert C 4:56

Hmm.

Pedro Santos 4:56

Ou seja, ao pegar na carta, podia mudá-la e jogar. Neste caso, o jogo está em single player, não há outros jogadores, ou seja, joga automaticamente. Também pode haver um caso de empate.

Expert C 5:08

Certo.

Pedro Santos 5:11

Alguma pergunta?

Expert C 5:12

Sim ia perguntar-te, portanto, nós não vemos a nossa mão, portanto, isto é sorte no fundo.

Pedro Santos 5:19

Temos um bocadinho de jogo de memória porque, eu até vou tentar demonstrar com um bocadinho de sorte.

Expert C 5:23

OK. Ah, as cartas vêm para sítios específicos quando ganho. Pronto era essa a minha pergunta.

Pedro Santos 5:27

Exato, ao ganhar.

Expert C 5:30

Certo.

Pedro Santos 5:30

Ao ganhar, elas vêm para um sítio específico, ou seja, dá para decorar onde as cartas ficam na mão. Também dá para decorar as cartas dos outros e depois aí já há alguma complexidade. A primeira, as primeiras vezes a poderá ser através de sorte, mas tem esta vertente de estratégia. Foi como eu disse, o jogo é simples, mas também tem algumas vertentes de estratégia.

Expert C 5:34

OK. Certo, certo.

Pedro Santos 5:58

Também tem uma window com as regras que o jogador pode colocar onde quiser. Pode estar a jogar e a ver as regras ao mesmo tempo. Ajuda um bocado no onboarding.

Expert C 5:58

Pois.

Pedro Santos 6:09

O Jogo também é arnchored, ou seja, se o jogador se mexer e for até à cozinha. O jogo está fica lá parado.

Expert C 6:13

Sim. Certo.

Pedro Santos 6:19

Vou tentar voltar. Pronto, isto é um simulador, o movimento do câmara não é muito fluído. Então, se houver um empate, aí é que entra o dado.

Expert C 6:23

OK.

Pedro Santos 6:31

Os 2 ou 3 jogadores com a mesma carta mais baixa, empatam. As cartas ficam paradas e cada jogador lança o dado até descobrir alguém que tenha o valor menor do dado para ganhar as 3 cartas da ronda.

Expert C 6:47

OK.

Pedro Santos 6:47

E o robô neste caso vai reagindo. Se for uma vitória, faz esta dança. Se for um empate, já faz outra outra animação. Deixo também alguma liberdade para house rules, por exemplo. O jogo tem uma componente de sorte, então não sei se vou conseguir reproduzir um empate.

Expert C 7:10

Certo.

Pedro Santos 7:10

Mas, se houver um empate, as cartas não vão automaticamente para lado nenhum. Os jogadores fazem o desempate com o dado. Neste caso pegam no dado, lançam o dado e chegam à conclusão, mas também deixo liberdade para house rules. Ou seja, se quiserem fazer alguma coisa física, alguma maneira de desempate ou se quiserem mudar as regras de alguma forma, depois as cartas são manualmente postas na hand parte da pessoa que ganha.

Expert C 7:18

OK. Só nessa questão de desempate é que temos aí as house rules se os jogadores quiserem, é?

Pedro Santos 7:43

Não. Num jogo normal, implementei restrições de interação através do framework. Por exemplo, eu não consigo mexer nas cartas dos outros, só consigo consigo mexer nas minhas que estão na minha hand ou no stack. E também consigo definir um sítio específico onde o jogador pode largar os elementos com base na lógica do jogo. Por exemplo, a jogar uma carta.

Expert C 7:49

Certo. Hmm.

Pedro Santos 8:08

Mas depois tenho aqui esta toggle em que os jogadores podem desativar as regras do jogo.

Expert C 8:13

Ah.

Pedro Santos 8:15

Agora, os jogadores já têm liberdade para fazer o que quiserem e jogar como quiserem. Assim já permite house rules. E jogar de diferentes formas.

Expert C 8:22

Hmm. OK, entendi.

Pedro Santos 8:24

Também tenho aqui um modo de jogo mais rápido, só para mostrar as eliminações e assim. Porque o outro ainda pode demorar um bocadinho com um baralho grande. Pronto, este jogador vai ser eliminado.

Expert C 8:32

Sim.

Pedro Santos 8:41

E depois, eventualmente, alguém há de ganhar. Um dos jogadores, neste caso penso que vai ser este jogador.

Expert C 8:47

Está.

Pedro Santos 8:51

Até posso desligar as regras também. E, neste caso, foi um empate.

Expert C 8:59

Sim.

Pedro Santos 8:59

Mas eventualmente ganha alguém e depois aparece uma mensagem de vitória com música e afins. Há outros aspetos. Já mostrei as regras, já mostrei isto também.

Expert C 9:18

Hmm, okay.

Pedro Santos 9:19

Permito uma mudança para um modo mais virtual, mudando o ambiente. Neste caso, o ambiente já seria todo digital. Este.

Expert C 9:25

Hmm.

Pedro Santos 9:25

Nesta parte, deixo passar um bocadinho do ambiente à volta do utilizador, na parte mais perto dele. Ainda passa um bocadinho, é uma escolha, podia ser completamente virtual. Neste caso, como poderá ser jogado numa mesa ou assim, achei melhor e foi uma decisão minha. Mas também a plataforma permite completamente VR, ou seja, dá para fazer esta mudança entre um contexto mais AR, misto ou mais virtual.

Expert C 9:48

Certo. Certo.

Pedro Santos 9:57

E dá para voltar a desligar. No meu caso, é só apenas um ambiente diferente, mas noutros tipos de jogos poderá ser envolvido, por exemplo, um portal do mundo real para o mundo virtual.

Expert C 10:12

Certo.

Pedro Santos 10:13

O jogo também tem um aspeto de multiplayer. Tenho uma integração com SharePlay e FaceTime, a aplicação da Apple. Isso permite-me fazer multiplayer, ter multiplayer com jogadores cada um no seu sítio. A jogar com o game state sincronizado, etc. Uma feature interessante do Apple Vision Pro são as spatial personas que são um Modelo 3D da pessoa.

Expert C 10:21

OK. Hmm. Certo.

Pedro Santos 10:48

Em que estão disponíveis através do FaceTime. Ao fazer esta integração, o jogo em si também mostra estas spatial personas. Neste caso, só consigo mostrar a parte do simulador.

Expert C 10:59

Hmm.

Pedro Santos 10:59

Mas com devices reais, era mesmo uma pessoa real e as pessoas ficam no nos seus sítios, ou seja, como se fosse um bocadinho um jogo real com elementos digitais. Mas as pessoas ficam no sítio, têm as suas interações e mostram os gestos. Vou criar aqui uma chamada FaceTime para demonstrar.

Expert C 11:03

Certo. Certo.

Pedro Santos 11:21

Lá está, como é um simulador, o cão e o gato são utilizadores mocked.

Expert C 11:32

É interessante permitirem também simular isso por acaso. É bom.

Pedro Santos 11:37

Sim, dá. Tem algumas limitações, como disse.

Expert C 11:41

Hmm.

Pedro Santos 11:42

Ou seja, neste caso aparecem emojis a flutuar, mas a implementação está pronta para num contexto real a aparecer mesmo a imagem 3D da pessoa, a mexer-se, as expressões pessoas faciais, movimentos, etc.

Expert C 11:46

Certo, faltava o headset.

Pedro Santos 12:00

E lá está, é aquela limitação. Também não tendo device, não consigo testar ao máximo, mas a seria experimentar. A implementação está pronta para isso.

Expert C 12:15

Hmm.

Pedro Santos 12:15

Ao criar uma chamada, a aplicação identifica imediatamente e aparecem estas opções do FaceTime. Dá para criar aqui um jogo multiplayer instantaneamente. Vou só chegar o jogo um bocadinho para trás. Começando o jogo de novo. OK, já fica partilhado.

Lá está a limitação de ser mocked users. Não consigo controlar nem mostrar perspetivas dos outros utilizadores. Há algumas alternativas para este ambiente multiplayer, mas não são compatíveis com TabletopKit.

Expert C 12:47

Certo.

Pedro Santos 12:57

Mas pronto, neste caso, os jogadores aceitam no lado deles. Aparece o pop-up que me apareceu a mim. Entrei no jogo automaticamente, foi me atribuído um seat e começaram a jogar com o game state igual para todos. As animações são feitas localmente para cada jogador e o game state é sincronizado.

Expert C 13:05

Hmm. Sim.

Pedro Santos 13:16

É uma coisa assim.

Expert C 13:16

Uma coisa. Queria colocar uma questão em relação a... claro que o teu foco foi usar o Apple Vision Pro, mas por acaso isto permite multiplayer com jogadores que não estejam em VR mode? Ou haverá um bocadinho como temos aqui, esta simulação, metade o computador?

Pedro Santos 13:43

Um ponto bom do Facetime é que permite fazer chamadas para todo o ecossistema da Apple. Não permite cross-play ou cross-platform com outros headsets XR mas permite cross-play com iPhone e iPad se forem implementadas aplicações iOS e iPadOS.

Expert C 13:54

Sim sim. Certo.

Pedro Santos 14:07

Jogadores com iPhone poderiam ter o jogo usando RealityKit e ARKit. Também usa a mesma linguagem de programação Swift e SwiftUI para o UI.

Expert C 14:12

Ora isso mesmo.

Pedro Santos 14:21

Claro que existem diferenças entre a implementação visionOS e iOS ou iPadOS, não seria um para um.

Expert C 14:28

Sim.

Pedro Santos 14:29

Mas desenvolver uma aplicação iOS podia aproveitar algumas partes da aplicação do Vision Pro, ou seja, não seria uma aplicação completamente de raiz e poderá haver aqui uma partilha de código e assim. Neste caso, não implementei a aplicação iOS porque não era o foco da tese mas é perfeitamente possível.

Expert C 14:40

Teoricamente daria. OK, interessante, sim.

Pedro Santos 14:50

Sim sim, sim.

Expert C 14:51

Depois.

Pedro Santos 14:52

Até porque neste caso eu criei uma chamadas com jogadores que estão a usar Apple Vision Pro, os chamados spatial users. Mas posso criar com, por exemplo, 3 spatial users e 2 regular users.

Expert C 15:00

Isso. Grátis.

Pedro Santos 15:11

E eles permitiam estar na chamada, mas no seu telemóvel ou tablet.

Expert C 15:18

Okay.

Pedro Santos 15:19

Outra coisa boa dentro da integração com o FaceTime é que aqui já está aparece o jogo partilhado e permite convidar jogadores facilmente. Tudo integrado na parte do do FaceTime. O jogador pode convidar amigos dos seus contactos ou enviar um link para a chamada.

Expert C 15:45

Hmm, certo.

Pedro Santos 15:45

É, por exemplo, aqui, neste caso tem a chamada com jogadores. E assim permite partilhar o link para adicionar pessoas dos contactos, por exemplo.

E as pessoas entrariam logo no jogo como são 3 jogadores. Foi uma escolha minha serem 3, é escalável. Mas neste caso, portanto, se for uma chamada de 4 jogadores, por exemplo. 2 deles podiam entrar com pessoa. E o terceiro ou quarto. Neste caso ficavam de fora, ficava com este aqui a espectar o jogo.

Expert C 16:16

Certo.

Pedro Santos 16:18

Tinha liberdade de andar à volta do jogo, ver como é que está a acontecer as coisas, etc.

Ou seja, não limita também a jogabilidade e depois trocar de seats com quem perder se quiser, etc.

Expert C 16:27

Sim.

Pedro Santos 16:29

A outra parte é interessante, nisto é que o sistema operativo tem outras aplicações. Então, permite multi-tasking. Por exemplo, pode ouvir música. Eu tenho aqui um video preparado.

Expert C 16:42

OK. Interessante.

Pedro Santos 16:46

Ou, neste caso, ver um filme, por exemplo.

Expert C 16:50

Engraçado.

Pedro Santos 16:50

Pronto, dá para ver um vídeo, ouvir música, etc. Estou a tentar parar. Okay, dando full screen, até baixa a luminosidade do ambiente real da pessoa. Ou seja, tem esta esta vertente mais de multitasking que permite jogar ao mesmo tempo, enquanto o jogador da browse na web.

Expert C 17:11

Engraçado. Sim.

Pedro Santos 17:13

Exato. Também permite, por exemplo, a parte das regras. Se eu, por exemplo, colocar as regras aqui. Estou a jogar, estou a ver as minhas próprias regras. As outras pessoas podem ter as suas próprias regras noutro sítio, mas se quiserem partilhar, também podem chegar aqui e partilhar.

Expert C 17:23

Certo. Hmm.

Pedro Santos 17:29

Tem esta vertente social.

Expert C 17:29

OK. Sim, isso é fixe.

Pedro Santos 17:36

Sim, claro que está sempre limitado. Agora a Apple normalmente é uma empresa em que a integração entre as coisas do seu ecossistema é bastante fluida e com bastante qualidade.

Expert C 17:42

Sim. Sim.

Pedro Santos 17:52

Permite assim esta integração e ter estas features todas embutidas na aplicação.

Expert C 17:59

Diz-me uma coisa, por exemplo, quando puseste o browser com o YouTube a dar, também dá para partilhar aquilo? Teoricamente as outras pessoas estariam a ouvir também o vídeo. É isso? Isto seria a ser replicado para os outros?

Pedro Santos 17:59

Sim. Mas não dá para fazer tudo ao mesmo tempo. Com o meu jogo por exemplo.

Expert C 18:26

Hmm.

Pedro Santos 18:27

Isto tudo utiliza o SharePlay do FaceTime. Não testei mas penso que não permita partilhar duas aplicações diferentes ao mesmo tempo.

Expert C 18:39

Certo, OK, portanto OK.

Pedro Santos 18:40

No meu caso o jogo é uma aplicação e o Safari outra.

Expert C 18:48

Hmm.

Pedro Santos 18:49

Por exemplo, se eu partilhar aqui.

Expert C 18:53

Certo.

Pedro Santos 18:58

Eu acho que o Jogo já não está partilhado.

Expert C 19:00

OK, portanto o browser é uma aplicação separada tecnicamente, portanto não pode ser shared ao mesmo tempo. OK, já entendi. Sim. OK.

Pedro Santos 19:02

Exato. Abrir outra vez o jogo. Guardou state. Eu acho que também da parte da demo, se não me esqueço de nada, acho que era um bocadinho por aí. Eu agora tinha eram algumas perguntas. Antes disso, alguma questão sobre a implementação ou sobre a plataforma?

Expert C 19:19

Não acho que acho que não. Já fui colocando algumas das questões que me iam surgindo. Não, portanto, acho que podemos avançar para as perguntas e se calhar também vão surgindo outras coisas.

Pedro Santos 19:54

Sim, a entrevista também é semiestruturada mesmo para permitir esta flexibilidade. Vou parar aqui por partilhar e vou só fazer algumas perguntas. Começando por uma mais de opinião. A implementação desta versão de realidade estendida deste jogo que eu acabei de apresentar...

Expert C 20:04

Sim.

Pedro Santos 20:18

Oferece algum valor único em comparação, por exemplo, com a versão física do jogo?

Expert C 20:23

Hmm.

Pedro Santos 20:24

Com cartas e dados reais.

Expert C 20:27

Certo.

Pedro Santos 20:31

Alguma coisa única?

Expert C 20:33

Eu, eu até de certa forma penso ao contrário pelo seguinte, porque, por exemplo, durante a pandemia e não só depois também joguei bastante jogos online deste género. Um exemplo é Jackbox Games, etc. Portanto esses jogos de salas em que estamos a partilhar a experiência.

Pedro Santos 20:47

Sim.

Expert C 20:55

Ou seja, há aqui espaço para este tipo de jogos em situações em que as pessoas não conseguem partilhar o mesmo espaço físico, não é? Eu sei que há muita gente que ainda mantém o uso deste tipo de jogos com pessoal que conheceu online, o pessoal que está longe, etc. Portanto, acho que tem valor nesse sentido. Até faço outra pergunta, será que tem valor único versus VR ou mesmo sem ser VR, no meu computador. Suponho que tenha a questão de ser mais imersivo. Aí fica levanta a questão, o que será ser imersivo num contexto de Apple tabletop games? Será que traz muito valor?

Pedro Santos 21:28

Hmm.

Expert C 21:42

Suponho que eu estou a fazer perguntas enquanto estou a responder, desculpa, mas isto é uma questão que surge muito, porque eu, pelo menos em termos de aplicações de realidade aumentada, muitas vezes, quando aparecem projetos de clientes. Acho que é sempre essa a minha pergunta interna de porquê AR? Para que é que está a ser usado neste caso?

Pedro Santos 21:47

Certo.

Expert C 22:03

Suponho que se tivesses a partilhar o espaço físico com mais alguém que também tivesse no jogo e depois tivesse outra pessoa que estivesse fisicamente afastada, poderias querer ter essa representação, ou seja, conseguir ver a pessoa que está ao teu lado a jogar. Ao mesmo tempo, estar a ver a outra pessoa que está afastada ou eventualmente se estás numa situação em que não queres estar completamente isolada do teu espaço, não é? Estou imaginar, por exemplo, alguém que tem crianças pequenas, andar por aí. Se calhar, queres manter o olho. Ter certeza que não se tenha alhadas, ou seja, não está completamente isolada. Portanto, acho que a minha resposta considere nessas opções e eu acho que em alguns casos pode ter o seu interesse. Consigo imaginar alguns casos de famílias que estão distantes. Terem aquele ritual de fazerem um FaceTime e terem assim o momentozinho de jogar. Isto permite-nos não ficar muito isolada do que está à volta deles. E estando na mesma a sentir que estão com as pessoas à sua volta. Há, portanto, acho que consigo imaginar valor em algumas coisas, sim.

Pedro Santos 23:14

Okay, obrigado. A minha segunda questão estava relacionada se este protótipo demonstra eficazmente que esta plataforma é capaz de suportar desenvolvimento de jogos num contexto real, do mundo real?

Expert C 23:28

Diria que sim. Embora para o desenvolvimento tabletop games, como é um caso muito específico. Fica a questão o quão bem isto se estende para outro tipo de jogos, não é? Jogos em que é preciso lidar com states mais complexos do jogador, em ambiente 3D etc.

Pedro Santos 23:49

Sim.

Expert C 23:57

Portanto, diria do que tu mostraste parece ser bastante intuitivo. E tem suporte para este contexto de top games. Fico ainda com a incógnita de como é que funcionaria para outro tipo de jogos.

Pedro Santos 24:10

Sim. Pronto, agora também, com base neste protótipo, as funcionalidades únicas do Apple Vision Pro e do ecossistema da Apple representam alguma oportunidade no contexto de desenvolvimento de jogos? Em comparação com outras plataformas XR?

Expert C 24:32

Pois.

Pedro Santos 24:33

Plataformas de realidade estendida.

Expert C 24:36

É complicado, especialmente de um ponto de vista de game development, temos sempre alguns problemas com a Apple. Esta questão de se fecharem muito bloqueia todo um público alvo, não é? E, portanto, eu acho que vai sempre ter esse desafio agora.

Pedro Santos 24:54

Exato.

Expert C 25:00

Eu se entendo bem, eu sei que, por exemplo, nos Estados Unidos há toda aquela coisa de tens que ter Apple. A Apple é que domina lá. Se imaginarmos a situação do FaceTime acaba por ser, se calhar, a aplicação mais usada para videochamadas. Tipo neste contexto mais casual, entre famílias e amigos. Se calhar, abra uma oportunidade e o market nesse sentido.

Pedro Santos 25:38

Sim. Sem dúvida.

Expert C 25:49

Acho que é essa a minha resposta. Espero que tenha resumido.

Pedro Santos 25:52

É sim, faz total sentido. A Apple, mais conhecida pelo chamado walled garden. Em que tudo lá dentro funciona muito bem, mas se for preciso alguma coisa fora disso.

Expert C 25:59

É isso. É eu sei que o pessoal que usa adora e nunca quero largar, porque já estão também tão habituados que nunca pensariam nisso, estão tão confortáveis.

Pedro Santos 26:05

É.

Expert C 26:15

Depois são sempre incentivados a continuar a comprar outros devices dentro disso. Porque é tudo super compatível e fluído. Mas sim, acho que para quem está a querer publicar um jogo e quer chegar a mais pessoas, que é sempre esse nosso objetivo. É algo a considerar porque afeta tem sempre menos pessoas numa plataforma específica.

Pedro Santos 26:37

Pronto, só tenho mais uma pergunta. Que oportunidades e desafios podem ser identificados no design e implementação de jogos para visionOS? Design, implementação.

Expert C 26:57

Bem, eu não tendo estado com visibilidade direta durante o desenvolvimento, claramente o facto de não poder ter um device físico é um problema tão grande que acho que isso é um automático, não?

Pedro Santos 27:15

Sim.

Expert C 27:32

Mas eu sei que, por outro lado, é bem possível que o processo de desenvolvimento se viveres num dos sítios que te permite utilizar ou testar. As coisas devem ser relativamente fluidas. Imagino no desenvolvimento, não é? Já te dão as frameworks, tudo está tudo já bastante plug and play e é claro que isso é sempre apreciado. A realidade é olhando para outros outras ferramentas, tipo implementar a VR em Unreal ou Unity. Nós também muitas vezes temos a frameworks. A Meta também fornece muita coisa. E não sei se o que eles fornecem-te, dá um balanço suficiente para compensar o que os outros também dão. Mais a toda a facilidade de desenvolvimento, etc.

Pedro Santos 28:35

Sim.

Expert C 28:43

Espero ter respondido a sua pergunta sempre, muitas coisas.

Pedro Santos 28:52

Sim sim, sim.

Expert C 28:54

Foi?.

Pedro Santos 28:57

Foi, foi respondida. Não tinha mais nenhuma pergunta. Alguma questão extra?

Expert C 29:05

Bem, eu gostava de perceber a tua experiência com o desenvolvimento. Quanto foi ligar coisas que já existiam? Quanto é que foi tipo código? Esse tipo de coisas?

Pedro Santos 29:23

Teve bastante código. Depende, eles têm os frameworks que eu usei para este caso específico, tem o TabletopKit, que já traz algumas ferramentas para tabletop games, como states para o equipamento.

Expert C 29:34

Hmm.

Pedro Santos 29:40

Escrever código é mais focado onde restringir as interações, o que fazer o código mais de game logic também.

Expert C 29:52

Certo, certo, sim, sim.

Pedro Santos 29:52

Não tanto a interação entre o jogador e o objeto. Aí também também permite algumas coisas, os custom gestures e assim.

Expert C 30:04

Certo, certo.

Pedro Santos 30:17

É um novo mundo do que Swift em iOS, mas familiar em algumas partes. Houve algumas limitações. Eles têm uma tool, o Reality Composer Pro para fazer asset editting e assim. Encontrei algumas limitações nessa parte e tive que usar Blender para fazer tapar algumas limitações.

Expert C 30:26

Certo. Era isso que ia perguntar a seguir a questão dos assets. Como é que é feito? Eles têm tipo uma biblioteca com coisas gratuitas da comunidade?

Pedro Santos 30:45

Alguns assets deles.

Expert C 30:55

É pouquita?

Pedro Santos 30:55

Sim. Eu usei maioritariamente third party. Outra coisa, como tive que ir buscar assets third party, há algumas condições de formatos que se pode usar. O que introduz algumas limitações também.

Expert C 31:02

Faz sentido.

Pedro Santos 31:09

Na parte do Reality Composer Pro.

Expert C 31:15

Certo, certo. OK, exato. O que achas que foi... Lá está aquelas questões que se faz numa entrevista. O que é que foi o teu maior desafio neste neste projeto? Estou curiosa.

Pedro Santos 31:27

Eu acho que é fácil, não ter device físico.

Expert C 31:31

Verdade, OK vamos, vamos dizer tirando essa. O que é que dirias?

Pedro Santos 31:31

Pronto, tirando... Não, não diria que houve assim grandes limitações. É uma tecnologia muito nova, há sempre limitações, ainda está muito verde.

Expert C 31:51

Sim sim, sim, sim. Isso.

Pedro Santos 31:56

O que diria seria que apesar da Apple ter documentação boa, bastantes exemplos e sample projects, não há exemplos práticos do mundo real.

Expert C 32:00

Certo.

Pedro Santos 32:06

Se houver algum bug específico é difícil. Não há muitas discussões da comunidade e assim. É muito niche. Exato, exato.

Expert C 32:26

Quando comunidade desenvolve, pois é certo.

Pedro Santos 32:50

É, eu diria que, se calhar era, era mais por aí, porque a interação com o TabletopKit até foi bastante boa.

Expert C 32:57

Hmm.

Pedro Santos 33:03

Que também é um framework novo e muito específico.

Expert C 33:07

Certo.

Pedro Santos 33:09

Há a possibilidade de fazer jogos, também considerei fazer jogos não tabletop, mas foi para explorar um bocadinho este framework novo que eles lançaram, mas diria que era que era isso.

Expert C 33:17

Sim.

Pedro Santos 33:25

A documentação da parte da Apple está boa, mas fora disso...

Expert C 33:31

Sim. Estás um bocadinho a andar sozinho pelos mares. Exato.

Pedro Santos 33:33

Exato, por exemplo, a questão do...

Expert C 33:36

Sim.

Pedro Santos 33:39

Do FaceTime através do simulador. Existe, por exemplo, um projeto que faz mock ao FaceTime num servidor local e dá para ligar com vários simuladores. Só que não é compatível com TabletopKit. Ou seja, existe algum suporte, mas nada comparado com outras plataformas já existentes.

Expert C 33:51

Certo, pois é, acaba por ser normal quando até mesmo noutras ferramentas, quando temos um lançamento de um novo headset. Estamos mesmo muito dependente da documentação, da própria empresa que fez o headset e nessa situação, a Apple costuma ter documentação com qualidade, mas depois também precisas da comunidade, não é? É uma coisa super importante, pessoal a partilhar as experiências.

Pedro Santos 34:13

Exato.

Expert C 34:35

Partilhar problemas que têm tido e puderes ver solução que outra pessoa que já bateu com a cabeça na parede nesta situação. Sim, mas acho que isso não é um problema específico do Apple Vision Pro. Parece-me um problema de um headset novo em geral.

Pedro Santos 34:37

Exato. Sim sim.

Expert C 34:49

Realmente, é pena. É pena essa questão de não teres device físico. Mas ainda conseguiste fazer o desenvolvimento todo. Chegando a parte de simulação, pronto, pelo menos isso dá para se retirar conclusões. Pode ser que lancem no futuro e dê para testar isso num device real.

Pedro Santos 35:14

Sim. Exato.

Expert C 35:25

Sim sim.

Pedro Santos 35:34

Tirando essa parte de testar a parte multiplayer, de resto, acho que o simulador permite bastantes coisas.

Expert C 35:44

Também essa parte, concordo.

Pedro Santos 35:51

É só mesmo esta parte multiplayer que já não deixa.

Expert C 35:55

Já nisso. Devo dizer que o simulador até funciona bastante bem, comparando com alguns que já vi para VR. Até parece ser bastante fluido, portanto, isso é um plus. Olha Pedro que desafio grande que tiveste e pronto conseguiste. Conseguiste concluir aqui, está interessante, tenho mesmo pena que não tenha dado para usar o device físico.

Pedro Santos 36:06

Obrigado.

Expert C 36:29

Pronto, da minha parte, mais alguma coisa?

Pedro Santos 36:29

Obrigado pela disponibilidade.

Expert C 36:37

Claro que sim, sem problema nenhum.

Pedro Santos 36:37

Não, não, é só isso.

Expert C 36:39

Olha, foi um gosto também poder contribuir de alguma forma.

Pedro Santos 37:11

Obrigado. Eu depois também mando hoje ou amanhã um pequeno questionário para poder recolher alguns dados.

Expert C 37:19

Sim sim. Força.

Pedro Santos 37:24

E eventualmente também partilho a transcrição da chamada.

Expert C 37:30

Sim.

Pedro Santos 37:31

Está, obrigado.

Expert C 37:31

Pronto. Sem problema. Boa sorte, então, com o resto, Pedro corra tudo bem. Adeus e até uma próxima.

Pedro Santos 37:34

Obrigado. Adeus.

Pedro Santos stopped transcription