Developing a Virtual Escape Room Game: Existing Challenges and How to Overcome Them

Calíope Corrêa de Araújo*, Jaqueline Ferreira Holanda de Melo *,
João Marcelo Xavier Natário Teixeira *, Fabiana Frata Furlan Peres†
*Universidade Federal de Pernambuco, Recife, Brasil
†Universidade Estadual do Oeste do Paraná, Foz do Iguaçu, Brasil
Email: caliope.araujo@ufpe.br, jaqueline.fhmelo@ufrpe.br, jmxnt@cin.ufpe.br, fabiana.peres@unioeste.br

***Abstract*—This paper presents the challenges faced during the conception and development of a virtual reality game, of the Escape Room type, for simulation on the Meta Quest 2 mobile device. Based on the experience obtained, some important insights are provided, which are necessary to overcome the challenges faced and to promote a greater immersion of the player in the virtual environment. Related works demonstrate the importance of evaluating and continuing the studies presented, seeking to identify and promote significant purposes for virtual games, and promoting virtual reality teaching and learning based on the creation of proposals dedicated to immersive devices in a 3D virtual environment.**

***Keywords*—Virtual Reality Game, Escape Room, Meta Quest 2**

I. Introduction

Despite the technological advances of virtual simulations in the process of collaborative design, little is said about the parameters needed to ensure adequate performance of immersive virtual simulations and, thus, provide a comfortable, safe environment that encourages the participation of future users [1].

This article presents some of the problems inherent to the development of virtual environments for execution on mobile devices like Meta Quest 2, observed from the development of an Escape Room game project.

The purpose of the project to which this work refers consisted of developing an immersive environment where it would be possible to experience some of the interactions presented in the Meta Quest 2 demo application, an autonomous device for immersion in Virtual Reality (VR) manufactured by Meta.

For this, an Escape Room-type game, called Vertical Maze, was planned, where the player needs to solve puzzles and meet challenges while trying to escape (by exploring a place) a room. Escape Room is a subgenre of point-and-click adventure games, in which the player needs to escape from a certain place by exploring its surroundings. The room usually consists of a locked door, objects to manipulate, and clues or secret compartments. The player must use the objects to interact with other items in the room and reveal the way to escape [2].

This project was part of the Virtual Reality course offered by the Graduate Program in Design at the Federal University of Pernambuco, and was developed as a result of experiments in the teaching context, so that theory and practice could be correlated, in a collaborative perspective and autonomy, having its development limited to the scope of the subject in the program. Therefore, several challenges were found, some proposed solutions will be explained throughout this article, as well as the difficulties that still need to be overcome to improve the game.

The project was developed using the Unity real-time development platform, and the tests were carried out with Meta Quest 2, an immersive device in a 3D virtual environment, in the form of a headset with a VR platform that operates with an operating system based on Android, where it is possible to use its resources to control all the actions proposed for this game. The functionalities applied to the game were understood from the tutorial available on the Unity Learn platform, with the focus of the proposal being the best use of locomotion and teleportation within the virtual environment, seeking the spatial precision of the user to achieve the sensation of immersion and reality during the gaming experience.

Some solutions were found for challenges throughout the development, such as creating small snippets of code in C# to add functionality to the project, in addition to using the features of the Unity Editor, such as creating time counters and score. One of the development challenges involved the player's spatial issue while using the Meta Quest 2, which generated the need for a prior study during creation, from the game's plot and history, to the implementation phase. This article addresses these challenges, as well as describes the solutions found for some of the problems inherent to development, and concludes by highlighting what needs to be deepened and improved.

The remainder of this article is organized as follows. Section 2 briefly describes the development process for the Meta Quest

2 device. Section 3 describes some techniques on how to constrain the physical space required for immersive navigation. Section 4 presents some related works, while Section 5 presents the game proposal and the project requirements are listed in Section 6. In Section 7 the development is approached, the proposed solutions for the challenges encountered throughout the development, followed by Section 8, that addresses the testing phase, how they were performed and the players' difficulties. In Section 9 the results are discussed and future work possibilities are presented, as well as the important applications for this project.

II. Developing games for the Meta Quest 2

The Meta Quest 2 is a standalone and wireless device in the form of a technological head-mounted display that supports positional tracking with six degrees of freedom using sensors and a series of cameras, positional audio, and anatomical design. It also features controls developed with ring design and tracking technology [3]. As it is a widely commercialized device for VR gaming, currently sold by Meta for 299 dollars, it enables the popularization of both VR experience and game development for virtual environments. It is now easy to find information, resources, and tutorials online for amateur and professional developers, focusing on game creation using Unity and Unreal platforms.

To illustrate this scenario, we conducted a search for courses with the main keywords: "Unity" and "Unreal", which are engines for 3D game development. We found paid and free courses/tutorials on platforms such as Udemy, Image Campus, SAGA, EBAC, as well as on the Developer Community - DEV COMMUNITY (from Epic Games). These courses are described in Table I.

There are still many video tutorials made available by YouTubers, whether for simple commands and specific operations, such as the timer used in the development of this game, the score counting, applications developed based on tips and adaptations of widely available codes on the internet, as well as complete tutorials for game creation. For instance, a recent channel called "TDG VR" [4] was found through a free search on the YouTube platform with the search query "create game for Quest 2".

For the development of the project at hand, we used the standard tutorial for virtual environment development offered by the Unity platform more extensively, which was one of the reasons for choosing it as the engine for creating the game. Unity has a platform dedicated to learning and developing various applications, Unity Learn, where a step-by-step tutorial with simple challenges for creating a scenario with locomotion, teleportation, interaction, activation, and other commands was found. Based on the learning of these commands, the knowledge obtained in the course was applied to the creation and development of the project.

Another possibility for game development would be the Unreal Engine platform. Similar to Unity, they are popular development platforms widely used in the gaming industry, allowing the creation of 2D and 3D games without the need for advanced programming knowledge and exporting them to various platforms, from mobile phones to VR devices.

Unreal aims to be the open and advanced real-time 3D creation tool that enables creators in all sectors to generate cutting-edge content, interactive experiences and immersive virtual worlds. It is currently in its version Unreal Engine 5, it offers full access to the source code, a robust C++ API, visual scripts and customizable plugins [5]. Unreal was the engine for creating famous games like Fortnite, Final Fantasy VII Remake, Minecraft Dungeons, Mortal Kombat 11, among other famous games today.

Unity, on the other hand, claims to be much more than just a leading platform for real-time development, enabling rapid prototyping and scalable asset pipelines with a customizable Editor and high-performance C# code. It has a collection of complete courses that guide the developer throughout the creation process, allowing editing at runtime and testing, making development more dynamic [6]. Unity is behind the development of games like Pokemon Go, Super Mario Run, Among Us, Monument Valley II, and others. It also promises to be the best option for multi-platform game development. The Unity Editor interface can be seen in Figure 1.

Both platforms have user-friendly interfaces, allowing game development without advanced programming knowledge, offering a free version for amateur game development, and have full compatibility with various devices, including Meta Quest 2, used for testing in this project. Unity has a large community of developers, availability of tutorials, and libraries with assets available for free download, while Unreal allows character modeling within the interface itself and provides high-fidelity visual elements.

From a simple Google search, it was possible to observe that there is a discussion about which platform should be used. This debate or even the indication of the best platform for game creation can be found in blog posts, forums, and websites. There is opposition between Unity and Unreal, presenting positive and negative factors for each. However, this information comes from primary sources, and apparently, scientific studies are still being conducted to determine the most recommended or even most used platform. Some academic works that aimed to test the different platforms apparently made choices according to the project to be executed and the conditions available to game developers. Therefore, based on the conducted search, it was

Table I
LIST OF REPRESENTATIVE UNITY AND UNREAL COURSES.

Website/Institution	Link	Creator Engine
Udemy	https://www.udemy.com/course/o-guia-definitivo-para-desenvolvimento-de-jogos-com-unity/	Unity
Unity Learn	https://learn.unity.com/course/create-with-vr?uv=2021.3	Unity
EABC	https://ebaconline.com.br/desenvolvedor-unity	Unity
SAGA	https://www.saga.com.br/cursos/criacao-de-jogos	Unreal
DEV COMMUNITY	https://dev.epicgames.com/community/unreal-engine/getting-started/games	Unreal

Figure 1. Unity Editor interface.

not possible to identify precise data on the reasons for choosing a particular platform.

Nevertheless, despite the difficulty in defining the best platform, some criteria can be used for such a choice. Silva [7] analyzes the main available tools, bringing previous studies that also compared these creation engines, and concludes that despite the scope of these studies, many of their results need to be revisited due to updates of the analyzed game engines and the emergence of new ones. However, the methodologies and a large part of the parameters used for comparisons still remain valid, including scripting system, learning curve, licensing and cost, documentation, support, heterogeneity, compatibility, visual quality, asset store, and ecosystem.

After comparing all these parameters between Unity, Unreal, and CryEngine Lumberyard platforms in the previously mentioned work, Unity stood out due to the quality of its documentation, ease of finding materials on the internet, and the size of its community, making it more accessible and a safer choice for the development of the project, which required a short duration [7].

Regarding the game developed for the Vertical Maze project discussed in this article, both Unity and Unreal could have been used since the project has simple functionalities. However, we opted for Unity by choice and feasibility of execution, which allowed us to use the C# programming language to create the score counter and the timer. Thus, throughout the development of this project, we experimented with resources that go beyond the basic functionalities quickly offered in the interface of the mentioned platforms, giving more freedom for development and learning of the concepts studied during the project.

III. CONSTRAINING USER'S PHYSICAL SPACE

In VR, it is possible to provide users wearing head-mounted displays with the illusion of walking in any direction for infinite distance while, in reality, they are walking a curvilinear path in physical space [8]. This is a technique called "Redirected Walking (RW) ", and is basically accomplished by introducing unnoticeable rotations to the Virtual Environment (VE).

Ideally, the walker's movement within a VE should be constrained only by the virtual topography and architecture, not by the size of the available physical space. Most RW approaches are based on one of two forms of manipulation: techniques that manipulate the mapping between the user's real and virtual translation and rotation and thereby steer the user away from

the edges of the tracking space and physical obstacles, and techniques that manipulate the architectural properties of the VE, e.g., manipulate the location of rooms, hallways, and doors, to produce self-overlapping virtual spaces that make it possible to compress comparatively large VEs into smaller tracking spaces. The later was the approach adopted by the Vertical Maze project.

Redirection techniques can be applied without the user's knowledge (subtle techniques) or can be detectable by the user (overt techniques). While the former category fulfills the criteria of being imperceptible and is thus preferable for most applications, since we want to provide the user with the feeling of changing between building floors, the later was also chosen for this project.

The ideal RW technique has at least four criteria:

- It is imperceptible, the user is unaware that redirection is taking place;
- It is safe, the walker is prevented from leaving the tracking space and colliding with physical obstacles and other users;
- It is generalizable, it is applicable within any VE and with any number of users;
- It is devoid of unwanted side effects, it does not introduce cybersickness or interfere with primary and secondary tasks.

The implemented technique through teleportation fulfills three of the four previous criteria. In the case of the first one, the project chose to provide the sensation to the user of changing its location when interaction to one of the three doors inside the room. Regarding safety, despite the floor number, the user is always kept inside the floor area, which is the same for all of them. It can be applied to any virtual building, having to consider the limitation of teleporting to a place already occupied by another user, in case of a multiuser VE. At last, from the small amount of tests performed, we were not able to notice any discomfort e.g. cybersickness from users.

IV. Related work

Different virtual Escape Room options were previously implemented. For example, Bejarano et al.'s Escape Room in VR [9] was used for the diagnosis and treatment of a mental disorder. Similarly, Carnaúba et al.'s project [10] aimed to create a simulator for combat training and fire prevention through virtual reality, while Saint Martin's project [11] developed an VR-based Escape Room game for hospital evacuation training in case of fire. Furthermore, other works like those by Araújo et al. [12] and Guedes et al. [13] demonstrated proposed solutions and challenges to be overcome in creating Escape Rooms in VR for educational contexts.

From the "realityremake" website, it is possible to see a list of the 7 best VR Escape Room games for Meta Quest 2[1], as shown in Table II. The last two columns, current rating and interaction space, vary from 1 to 5. While current rating reflects the rating from the Meta Quest store, interaction space means the size of the virtual space used for interaction, where 1 indicates a very small interaction space (for instance, the user is seated on a chair in the virtual world), while 5 means having to walk along a big building or different rooms. As a comparison, the Vertical Maze would score 3 in this metric.

Table II
7 best VR Escape Room games for Meta Quest 2 from realityremake website.

Game name	Current rating	Interaction space
A Fisherman's Tale	4.51	2
Red Matter	4.61	5
Red Matter 2	4.79	5
Keep Talking and Nobody Explodes	4.56	1
The Room VR: A Dark Matter	4.89	4
I Expect You to Die	4.81	2
I Expect You to Die 2	4.82	2

V. The Vertical Maze project

The game developed for this project is an Escape Room, but it can also be considered an "Escape Building" since the game's story takes place within a residential building. It could even be referred to as a Vertical Maze, as we will call the game, as the player can not only escape but also get lost and return to the same location repeatedly during the gameplay. The main scenario consists mainly of an apartment complex, and in the proposed story, the player would be a jewel thief who invaded a residential building while fleeing from the police. Upon realizing that the building contained valuable objects, the thief decides to collect as many valuable items as possible before making their escape. The challenge given to the player is to collect all the jewels they find along their journey until they successfully evade the police. The jewels are playfully presented in the form of floating diamonds, placed in distinct locations chosen for each section of the scenario, and they may be easily visible or hidden behind furniture or objects. In addition to the challenge of collecting jewels, the player will also need to solve puzzles, which are also one of the premises of Escape Room-type games.

The building where the game takes place has 10 floors, as shown in Figure 2. At the start of the game, the player will be on the first floor, where they entered. The player will

[1]https://www.realityremake.com/articles/7-best-vr-escape-room-games-on-the-oculus-quest-2

have to traverse the floors above until they reach the rooftop of the building, where their final escape will take place. The puzzle to be solved involves finding the correct exits for each floor. On each floor, the player has three exit options from that room. There are three identical doors with no identification, and each one will lead the player to another floor of the building. Depending on their choice, the player may either go up or down. The choice of exit may lead the player to their destination faster or end up causing them to return to a floor they have already been on before.

The challenge lies in choosing the correct doors so that the player can move up from floor to floor, collecting as many jewels as possible until they reach the highest floor, defined as the exit of the game, as quickly as possible. Upon reaching the rooftop above the building, the player wins the game, and their score will be calculated based on how many diamonds they have collected. The initial proposal of the game also included a rescue of the player from the rooftop of the building before the police arrived and captured them, hence the need to be as agile as possible by making the right choices. Since the game's initial proposal included a timer in which the player would need to complete their journey within the allotted time, if the time runs out, the police would capture the player, and they would lose the game.

VI. Game requirements

During the conception and development of the game, several important requirements were considered, such as defining the available space for game development, creation, and testing. It was necessary to conduct a study on user movement during the game and design a scenario and a script that suited this space. The level of difficulty needed to be compatible with the knowledge covered in the scope of the VR discipline offered by the program, as well as with the commands presented in the followed tutorial. It could go beyond in seeking diverse solutions, but it had to remain realistic in relation to the overall content, as there was a deadline and a project evaluation for the completion of the discipline. Additionally, considering technological limitations, the game couldn't be too heavy since slow processing could hinder progress within the time of each class, and it would require highly robust technological resources that were not available in the used laboratory. Another fundamental requirement was that the game needed to be fully compatible with the Meta Quest 2, making the best use of its resources and functionalities, as it was the device used for the development and testing of the project. Therefore, the creation of the game was carefully planned to comply with these predefined criteria.

One of the biggest challenges in developing the game was related to the spatial proportion of the virtual environment based on the player's movement. The available space for testing and playing the game was limited to an area of approximately 30 square meters, in which the player could freely move during the game. To create a game with multiple spaces that went beyond this limitation, the solution was to stack the scenarios, forming the idea of a building. This way, the player could navigate through a much larger area virtually, while their physical space in the real world would be limited to the size of just one floor within the scenarios. The teleportation feature allowed the player to switch between scenarios with a command, avoiding the need to physically traverse the distance between spaces, especially when the scenarios had the same size and teleportation occurred to the same local coordinates. This also contributed to the sense of reality, as each apartment had a similar-sized room to the actual physical space used.

As teleportation was used to change floors, once the player was in a certain position of the scenario, he would necessarily need to be teleported to the same position on another floor of the building, due to the delimitation of space being the same for all floors. If the player walked to the boundary region of one apartment, for example, when teleported to another floor, if their positioning was not the same, they would lose their reference to the game area, and they might traverse paths beyond the predefined limits if they moved in an unexpected direction. To ensure that the player's movement always respected this limitation, teleportation areas were strategically placed in front of each floor's doors. Since the player would necessarily pass through the door to access another part of the scenario, they would already be positioned on one of these teleportation areas whenever they activated the teleportation feature. Each of these areas was related to the same position on another floor, ensuring that the player did not lose their reference to the scenario's boundaries.

VII. Development

As mentioned earlier, for the creation and development of the game, the tutorial available on the Unity Learn platform called "Create with VR" was used [14]. It is a 30-hour course aimed at beginner developers, providing a didactic presentation of the main functionalities of the Unity interface for developing a virtual environment with locomotion and interaction, compatible with various virtual reality devices such as the Meta Quest 2. Before the testing phase, the project was initially developed using the Unity simulator, without the need for an external device. The mentioned tutorial also provides a step-by-step guide for necessary software installations, configurations for using the simulator on a conventional computer, and compatibility with the Meta Quest 2. It further offers a library of objects called assets that can be used to create scenarios and interactions in the virtual environment.

The majority of objects used in both the scenario and for interaction with the game were obtained from the Course Library, a package of assets available for download in the tutorial. Other objects, such as the diamonds used in the scenes, were obtained from the Unity Asset Store, and all materials were provided for free. In the first part of the Unity tutorial, you can add all the chosen objects to the project's library and position each object in the scenario using the concept of exploration presented in the tutorial. The scenarios were created with spacious apartment rooms, furniture, and decorations, each floor having its distinct characteristics. The rooftop was designed with a pergola and a garden, open to the landscape and for the player's rescue, as seen in Figure 3.

As Guedes et al. [13] recommend, the development of an Escape Room demands careful attention to details to ensure users have a good experience. Therefore, a significant portion of the development time was dedicated to obtaining objects and carefully positioning them, as well as structuring a storyline and defining puzzles. In Unity, it is possible to freely position elements by moving them with the mouse to the desired location, but some care was taken to ensure that the elements were on the corresponding scene plane and properly positioned. The diamonds were strategically placed to be found by the player. To facilitate the development of games with many elements, Unity allows organizing them into hierarchical folders, as shown in Figure 4, which was crucial for editing and modifying the positions throughout the project, considering the game includes a large number of small elements.

Inside Unity, the element called XR Rig, by the tutorial, present in the basic project hierarchy, determines the player's positioning, and it is possible to define the player's starting position and orientation at the beginning of the game. This positioning is crucial for correct locomotion within the predetermined environment. It is also necessary to define the positioning of the cameras so that when the game starts, the user is correctly positioned and viewing the scene. In figure 5 it is possible to see the initial position of the player, configured to be in front of the first door of the labyrinth. In addition, other basic resources can be defined at this stage, such as the lighting of the environment.

It was necessary to define objects that represented the player's hands, which would remain visible throughout the game and be used to trigger and interact throughout the gameplay. All these definitions were made through the Unity interface, where it is possible to assign objects to the controllers and determine their triggering power based on the distance from the wands, trigger bars, and player's interaction, wich can also be seen in Figure 5. It is also the hands that the player will move during the game while holding the controllers of the Meta

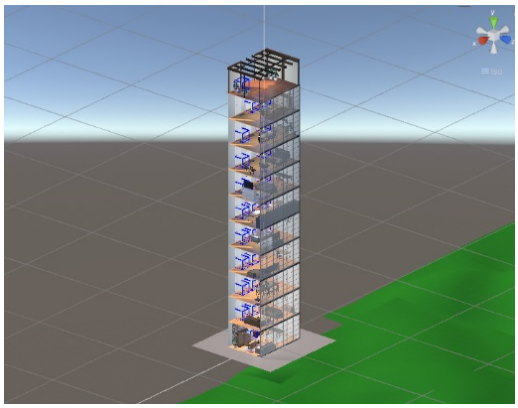

Figure 2. The 11 floors of the Vertical Maze implemented on Unity.

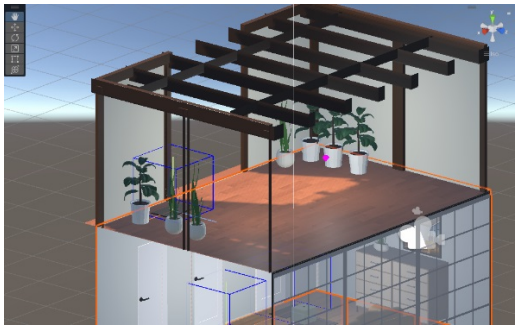

Figure 3. Roof terrace over the building and player's escape point.

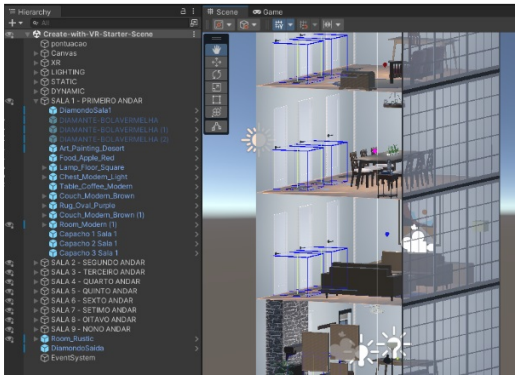

Figure 4. Objects hierarchy on Unity Editor.

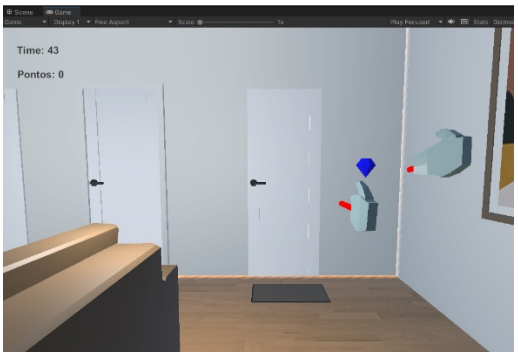

Figure 5. Game scene, user interface and wands.

Quest 2, which have buttons for the necessary triggering and interaction.

After these important predefinitions, it was possible to develop the system of locomotion and interaction with objects. By using the Teleportation Area and Teleportation Anchor resources, the player can move between floors. A teleportation anchor is positioned on the doorknobs, and when the player interacts with the doorknob, they are teleported to a predetermined area. The teleportation areas were positioned on a rug in front of each door. Another Unity resource used for interaction with objects was Grabbable Objects, where we initially defined how the player's hand would appear in the virtual environment and how they could collect an object, a feature that was especially used for collecting diamonds in the scene.

From Unity's UI (User Interface) resource, it is possible to define a visualization space, where the user can get messages throughout the game. A user interface area was then created, where the timer is displayed, which determines the time the player has to complete the mission, messages such as the game over message at the end of the uncompleted mission, and a point marker, which increments the every diamond collected. This information was positioned in the top left corner of the screen, as seen in Figure 5.

In addition to the functionalities offered by the Unity interface, it was necessary to add some pieces of code called scripts, which are explicitly described below.

Through code, a timer was implemented, which, when the time runs out, ends the game, displaying the game over message. The initial time set for the player's escape was 300 seconds, or 5 minutes, as shown in Listing 1.

Listing 1. Code for stopwach and game over

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;


public class cronometro : MonoBehaviour
{
    public Text countDisplay;
    float count = 300;
    // Start is called before the
    // first frame update
    void Start(){}
    // Update is called once per frame
    void Update()
    {
        if(count>0)
        {
            count = count - Time.deltaTime;
            countDisplay.text = "Time:"
            + ((int)count).ToString();
        }
        else
            countDisplay.text = "GAME OVER";
    }
}
```

At the moment when the player interacts with the diamond, it disappears from the scene, effectively being collected, and one more point is added to the game's score counter. In the same code, an effect was added to the diamonds, making them float and rotate, giving the impression of a valuable object and highlighting the shine present in the 3D image, as shown in Listing 2.

Listing 2. Code for collecting diamonds and creating an effect

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Diamante : MonoBehaviour
{
    Vector3 lastPos;
    public GameObject obj;
    // Start is called before the
    // first frame update
    void Start()
    {
        lastPos = obj.transform.position;
    }
    // Update is called once per frame
    void Update()
    {
        transform.Rotate(0*Time.deltaTime,
```

```
            0*Time.deltaTime,10*Time.deltaTime);

            if(obj.transform.position != lastPos){
                lastPos = obj.transform.position;
                obj.SetActive(false);
                pontos.valor++;
            }
        }
}
```

A script was also created to display the points on the user interface and to reset the score to zero at the beginning of each game, as shown in Listing 3.

Listing 3. Code for displaying points

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class pontos : MonoBehaviour
{
    public Text displayPontos;
    public static int valor;

    // Start is called before the
    // first frame update
    void Start(){valor = 0;}

    // Update is called once per frame
    void Update()
    {
        displayPontos.text =
        "Pontos:" + valor.ToString();
    }
}
```

VIII. Tests

The usability tests of the game, using the Meta Quest 2 device, were conducted with a few individuals due to the time limitation of the project scope. Specifically, the tests were carried out by the three authors of this article and two other users present during the presentation. Figure 6 shows one user testing the game. The frames where captures in different interaction situations along the gameplay. A video containing one of the tests performed can be accessed using the following url: LINK. It illustrated both virtual and real world views.

During the tests, it was possible to initially perceive the enchantment of the users with the teleportation system, especially when arriving at the highest floors of the scenario, where it was possible to have a vertical spatial notion similar to reality, since the walls of the apartments were composed of a glass film, causing even a slight fear in relation to perceived height. With Quest's controls, the user could collect diamonds and activate teleportation from the door handles, which made the player interact with the environment constantly throughout the game, also contributing to the user's immersion. Despite being interesting that the image of the environment was realistic, Thomaz et al. [15] confirm in their study that immersion is not necessarily related to realism and the existence of images, but other factors also play a strong role in this issue, such as the creation of a plot, the possibility of interacting with other users and with objects in the environment. In this way, the controls act on the player's autonomy, bringing that sense of realistic interaction expected for the project.

One challenge concerning user control was related to the size of the wands, the hand-reaching bars of the player. Short wands were defined to ensure that when the player triggered teleportation, they would be positioned in the pre-determined area in front of the door. Thus, upon arriving on another floor, they would be correctly positioned at the defined location, staying within the game space. This prevented the player from activating the doorknob from a distance too far from the teleportation spot. Consequently, collecting diamonds and interacting with objects were only possible at a very close distance between the player and the object.

The distance from the player's arm to the doorknob also affected their positioning when teleporting. If the player triggered the doorknob while being exactly in the designated spot, they would teleport correctly to the desired location in the scenario. However, if the player extended their arms to trigger the doorknob from a farther position than the one studied, when teleporting to the pre-defined location, the player could lose some spatial awareness and end up outside the space defined by the Meta Quest 2 device as the game area. Being beyond this space might result in losing the ability to activate teleportation, limiting the player's autonomy, as they always needed to be at the pre-studied average and comfortable distance for activation and interaction.

IX. Conclusion

The development of this project provided the learning of various VR concepts and game creation for virtual environments, and demonstrated that there is still much to be explored within this context. This includes the implementation of resources for virtual environments, device testing, platform comparisons for creation, and device mapping for VR games. Concerning

Figure 6. User testing the game developed. The upper left frame indicates the start position of the user inside the real space. Using measures to make the user stay inside an specific area is important due to the limited space, as showed here.

the development of the described game, some solutions were explained in this article, but there are still challenges to be overcome and possible improvements that can contribute to the continuation of this work.

Despite the development engines offering interfaces that allow the development of virtual environments without advanced programming knowledge, for some seemingly simple applications, it was necessary to develop codes, albeit basic ones, for implementation. However, Unity proved to be a good alternative for VR game development. Even in a project with a short time frame and developers still possessing basic knowledge, it was possible to implement an environment that provides a complete gaming experience for users, including challenges, puzzles, scene changes, locomotion, and visual resources.

Although this work has been developed based on the study of the Unity and Unreal game engines, it is important to consider implementation in other open-source 3D game development engines, as there are several options available, such as the Godot Engine, Flax Engine, FNAF Engine, and Open 3D Engine. Evaluating the possibilities and limitations of these tools in the construction of an escape room with multiple teleportation features proves to be a relevant suggestion for future research.

Regarding the development specifically aimed at Meta Quest or similar wireless mobile devices, there is still a field to be explored in the academic context. Throughout the project, it was possible to observe a lack of research that delves deeper into studies for specific devices, perhaps due to their constant change and updates or the various possibilities of media for experimenting with virtual environments today. This leaves a gap to be explored concerning these devices.

To continue the Vertical Maze project, it is still necessary to evaluate other possible solutions for spatial issues and player positioning, enabling the game to expand into larger spaces or horizontal environments without compromising the user experience. Additionally, it is essential to develop solutions for

alternative ways of interaction, such as allowing the player to physically open doors, providing a realistic experience to access other areas of the game, or improving the accuracy of hand interactions with collectible items and doorknob interactions.

Furthermore, improvements can still be implemented, such as auditory features that will add more realism to the game, the completion of the rescue story that could enhance the emotional experience of the game, and a more user-friendly interface with icons, images, and options throughout the entire gameplay, from the beginning to the end of the session.

It is also recommended to conduct more tests with users, as the tests were performed with few individuals in a limited space. To implement the improvements and potential solutions mentioned earlier, it is necessary not only to expand the number of testers but also to explore other possibilities of real spaces, including smaller ones than those previously used. This will lead to finding a solution that allows users to play even with minimal available space.

Even though the proposed project and the game's creation did not initially have any educational or social intent, solely aiming at learning about RV development, various purposes become possible regarding the game's objective. The aforementioned related works demonstrate the importance of expanding studies around this project. An example is the VR Escape Room developed by Bejarano et al. [9], which was used for the diagnosis and treatment of a mental disorder. Meanwhile, the project by Carnaúba et al. [10] aimed to create a simulator for combat training and fire prevention through virtual reality, as well as the project by Saint Martin [11], which developed a VR Escape Room game for hospital evacuation training in case of fire. In addition to other works such as those by Araújo et al. [12] and Guedes et al. [13], which demonstrated proposed solutions and challenges to be overcome in the creation of VR Escape Rooms for educational contexts.

Therefore, the importance of evaluating and continuing the studies presented becomes evident, specifically concerning VR games, such as Escape Room types, developed for use on Meta Quest. This is in order to enhance improvements and solutions, identify and promote significant purposes in various fields, and foster the teaching and learning of Virtual Reality through game development, involving individuals from diverse areas of expertise.

References

[1] J. B. Mandola, F. G. Graton, and C. Imai, "Simulações virtuais imersivas em projetos arquitetônicos: os motores de jogos como prática no projeto colaborativo," *Simpósio Brasileiro de Qualidade de Projeto do Ambiente Construído*, vol. 7, pp. 1–10, 2021.

[2] L. T. Pereira, F. H. C. Silva, P. T. Palomino, and S. Isotani, "A abordagem construtivista no desenvolvimento de um serious game do gênero escape room," *Proceedings of SBGames*, vol. 2018, pp. 1011–1018, 2018.

[3] M. Quest. (2023) Oculus Quest 2. [Online]. Available: https://www.meta.com/quest/products/quest-2/

[4] (2023) Tdg vr. [Online]. Available: https://www.youtube.com/@TDGVR/

[5] I. Epic Games. (2023) Unreal Engine. [Online]. Available: https://www.unrealengine.com/pt-BR

[6] U. Technologies. (2023) Unity Engine. [Online]. Available: https://unity.com/pt/products/unity-engine

[7] D. L. da Silva, "Análise de ferramentas e desenvolvimento de jogo para treinamento de paratletas."

[8] N. C. Nilsson, T. Peck, G. Bruder, E. Hodgson, S. Serafin, M. Whitton, F. Steinicke, and E. S. Rosenberg, "15 years of research on redirected walking in immersive virtual environments," *IEEE computer graphics and applications*, vol. 38, no. 2, pp. 44–56, 2018.

[9] A. F. Bejarano, J. D. Correa, and P. Figueroa, "Escape room virtual reality: a tool for diagnosis and treatment of attention deficit disorder," in *2020 22nd Symposium on Virtual and Augmented Reality (SVR)*. IEEE, 2020, pp. 331–338.

[10] B. M. Carnauba, R. B. R. d. L. Alexandre, and C. A. C. L. Lessa, "Criacao de um simulador para o treinamento de combate e prevenção a incendios atraves da realidade virtual," in *Proceedings of SBGames 2020*. IEEE, 2020.

[11] V. B. Saint Martin *et al.*, "Hospital escape vr: desenvolvimento de um jogo sério para treinamento de evacuação hospitalar diante de um incêndio," 2022.

[12] R. F. S. Araujo, C. A. C. Bogarim, and J. A. Felberg, "Enigma da sala: um jogo gratuito de realidade virtual para criança," in *Anais III JOIN: Edição Brasil, 2017*. Campina Grande: Realize Editora, 2017.

[13] G. N. Guedes, B. H. P. da Silva, and R. Tori, "Escape room em realidade aumentada: Proposta de um serious game para o ensino fundamental i."

[14] U. Technologies. (2023) Unity learn. create with vr. [Online]. Available: https://learn.unity.com/course/create-with-vr?uv=2020.3

[15] A. R. Thomaz, C. Maraschin, E. Zaniol, and E. M. M. Schuch, "Dispositivos de imersão em ambientes de realidade virtual," *Revista brasileira de informática na educação. Florianópolis. Vol. 13, n. 2 (maio/ago. 2005), p. 21-32.*, 2005.