REVISTA CONTRIBUCIONES A LAS CIENCIAS SOCIALES

Practicing requirements elicitation in a serious game: comparison between non-immersive and immersive virtual reality

Praticando a elicitação de requisitos em um jogo sério: comparação entre realidade virtual imersiva e não imersiva

DOI: 10.55905/revconv.16n.12-134

Recebimento dos originais: 10/11/2023
Aceitação para publicação: 15/12/2023

Adilson Vahldick
PhD in Information Sciences and Technology by Universidade de Coimbra
Institution: Universidade do Estado de Santa Catarina
Address: Ibirama – SC, Brasil
E-mail: adilson.vahldick@udesc.br

Luis Felipe Kuster
Graduated in Software Engineering
Institution: Universidade do Estado de Santa Catarina
Address: Ibirama – SC, Brasil
E-mail: luis.felipe.kuster17@gmail.com

Ana Fábia Coelho dos Santos
Graduated in Software Engineering
Institution: Universidade do Estado de Santa Catarina
Address: Ibirama – SC, Brasil
E-mail: afcoelho44@gmail.com

Gustavo Vargas de Andrade
Graduated in Software Engineering
Institution: Universidade do Estado de Santa Catarina
Address: Ibirama – SC, Brasil
E-mail: gupontoandrade00@gmail.com

João Marcus Cardoso
Graduated in Software Engineering
Institution: Universidade do Estado de Santa Catarina
Address: Ibirama – SC, Brasil
E-mail: joaomarcuscardoso26@gmail.com

Marília Guterres Ferreira
PhD in Science by Pontifícia Universidade Católica do Rio de Janeiro (PUC - RIO)
Institution: Universidade do Estado de Santa Catarina
Address: Ibirama – SC, Brasil
E-mail: marilia.gf@udesc.br

REVISTA CONTRIBUCIONES A LAS CIENCIAS SOCIALES

Pablo Schoeffel
PhD in Computer Science by Universidade Federal de Santa Catarina
Institution: Universidade do Estado de Santa Catarina
Address: Ibirama – SC, Brasil
E-mail: pablo.schoeffel@udesc.br

Marcelo de Souza
PhD in Computing by Universidade Federal do Rio Grande do Sul
Institution: Universidade do Estado de Santa Catarina
Address: Ibirama – SC, Brasil
E-mail: marcelo.desouza@udesc.br

ABSTRACT
One of the biggest problems in teaching Requirements Engineering at universities is the lack of access to real projects, which could allow students to interact with stakeholders and practice communication skills while conducting requirements elicitation. This article presents a serious game for practicing requirements elicitation through guided interviews with stakeholders. Two versions of the game were developed. The Immersive VR game was designed to run on mobile phones with Google Cardboard. A Non-Immersive version was also developed to run on Windows. Systematic literature reviews point out a lack of immersive VR games for learning Software Engineering. Moreover, the game proposed here is a pioneer in the addressed subject, including the use of low-cost technologies, such as Google Cardboard. The game was tested for 20 minutes by 46 students in two classes of the Bachelor's Degree in Software Engineering. After the test in the first class, the game was improved, and another test was conducted. The purpose of the validation was to verify the sense of presence among the students through a presence questionnaire (PQ). The highest averages were identified in the non-immersive version. For the immersive version, the statistically significant lowest evaluated values pertained to movement within the environment and the feeling of disorientation at the end of the experience. This indicates the need for improvements regarding movement. The literature suggests two directions: the first is to reduce the duration of experimentation with immersive environments, and the second suggests exploring alternative mechanisms for movement within the environment. It is important to remember that the students experimented with their own mobile phones, that is, they did not use dedicated VR equipment. Therefore, it can be concluded that developing VR solutions for these devices requires more restrictive decisions.

Keywords: serious games, virtual reality, requirements engineering.

RESUMO
Um dos maiores problemas no ensino de Engenharia de Requisitos nas universidades é a falta de acesso a projetos reais, que poderiam permitir que os alunos interagissem com as partes interessadas e praticassem as habilidades de comunicação durante a elicitação de requisitos. Este artigo apresenta um jogo sério para praticar a elicitação de requisitos por meio de entrevistas guiadas com as partes interessadas. Foram desenvolvidas duas versões do jogo. O jogo de RV imersivo foi projetado para ser executado em telefones celulares com o Google Cardboard. Uma versão não imersiva também foi desenvolvida para ser executada no Windows. Revisões sistemáticas da literatura apontam a falta de jogos de RV imersivos para o aprendizado de Engenharia de Software. Além disso, o jogo proposto aqui é pioneiro no assunto abordado,

incluindo o uso de tecnologias de baixo custo, como o Google Cardboard. O jogo foi testado durante 20 minutos por 46 alunos em duas turmas do curso de Bacharelado em Engenharia de Software. Após o teste na primeira turma, o jogo foi aprimorado, e outro teste foi realizado. O objetivo da validação foi verificar o senso de presença entre os alunos por meio de um questionário de presença (PQ). As médias mais altas foram identificadas na versão não imersiva. Na versão imersiva, os valores mais baixos avaliados, estatisticamente significativos, referiram-se à movimentação dentro do ambiente e à sensação de desorientação no final da experiência. Isso indica a necessidade de melhorias em relação à movimentação. A literatura sugere duas direções: a primeira é reduzir a duração da experiência com ambientes imersivos, e a segunda sugere explorar mecanismos alternativos para a movimentação dentro do ambiente. É importante lembrar que os alunos fizeram experiências com seus próprios telefones celulares, ou seja, não usaram equipamentos de RV dedicados. Portanto, pode-se concluir que o desenvolvimento de soluções de RV para esses dispositivos requer decisões mais restritivas.

Palavras-chave: jogos sérios, realidade virtual, engenharia de requisitos.

1 INTRODUCTION

The increasing demand for professionals in software development goes against the training of these professionals. In higher education courses, the subjects lack real-world practices, due to the difficulty in simulating real companies and business environments (STRIUK et al., 2022). Commonly, students develop their work by reading and interpreting case studies provided by professors. In real life scenarios, the company process information is not organized, and formatted like in case study texts. However, it is very difficult for the professor to find available companies to undergo frequent visits during the entire academic period, and even more challenging to get people from these companies to be available to be interviewed by multiple students (DAUN et al., 2022).

One of the first steps in the software development process is the Requirements Engineering (RE) phase, when professionals need to gather as much information as possible from the target environment (SOMMERVILLE, 2016). This requires intensive use of interpersonal skills to communicate with process participants. RE is a multidisciplinary phase, people-centred, and therefore highly dependent on communication aspects, and as Ernst and Murphy (2012) reaffirm, information is not centralized in a single place for reading and interpreting the text, as it happens in classrooms.

Virtual Reality (VR) is a technology focused on visualization in three-dimensional environments, where the user can move or interact with other elements in that environment

(TORI et al., 2020). There are different definitions for VR. One of them presents an approach in which its scope includes any software that takes the user to a simulated environment where they feel as if they are living in the virtual world (ROBERTSON et al., 1993). Another definition, immersive VR, considers that, complementary to the previous definition, headset is the main hardware between the user and the application (FREINA; OTT, 2015).

The use of VR in education can be considered a natural evolution in the use of technology to support learning (PANTELIDIS, 2009). VR allows for new forms and methods of visualization and interaction, providing students with closer and deeper observation and examination of objects and processes. Students feel motivated and challenged to walk and interact in a 3D environment. VR may require specialized equipment such as headsets and controls, for instance Oculus Quest and HTC Vive. However, these devices have a high cost and are not widely used in schools. A low-cost alternative is to use smartphones as 3D viewers, developing applications with Google Cardboard technology (GOOGLE, 2022), which simply requires fitting a compatible phone into a cardboard device or VR Box, that can be purchased for less than 1/10 the price of a traditional VR headset.

Supporting what has been previously described about the traditional methodology of teaching Software Engineering (SE), Garcia et al. (2020) suggest the use of games to support learning, as they not only allow for individualized development in the evolution of the student, but also offer practice-based teaching. Serious (or educational) games illustrate theory through practice, as they can simulate business environments, allowing the student to learn through continuous interaction between successes and mistakes (ROSA et al., 2017), making students aware of the importance of the requirements phase and thus reducing project failure rates.

The goal of this work is to present a game that, through practice, enhances the skills necessary for eliciting requirements through guided interviews with stakeholders. The game is developed with VR to allow the student to be immersed in the business environment, using the Google Cardboard technology, which enables them to use the game on their own smartphones. A Windows version was also developed so that effectiveness in performing tasks in non-immersive and immersive VR can be compared. To evaluate the game project, it was tested by two classes of the Bachelor's degree in SE at State University of Santa Catarina.

The research questions that guided this investigation, considering the two versions of the game, were:

RQ1. How proficient are students able to perform tasks in the game?

RQ2. How much can the game keep students immersed in performing requirements elicitation tasks?

With research question RQ1, we are interested in verifying the behaviour of the students when performing the tasks of requirement identification and classification. Through research question RQ2, we aim to identify the students' opinions regarding the quality and usefulness of the game.

2 BACKGROUND AND RELATED WORK

One of the most complex tasks in software development is understanding the solution requirements of the problem being addressed. Usually, customers and end-users of this solution are not explicit about the information universe, and even if they were, requirements constantly change (PRESSMAN; MAXIM, 2021). In an ideal software development environment, requirements engineers work in constant contact with customers and end-users. When people collaborate on a software project, other skills are necessary, such as communication and interaction with other team members and these customers (MATURRO et al., 2019).

As mentioned by Andrade et al. (2022), between 2015 and 2019, there were almost 50 studies involving VR in SE, however, the researchers often use specialized immersive VR equipment, such as Oculus Quest and HTC Vive, which are not available to the public due to their high price. An alternative is the VR Box headset (Figure 1), which is a device that allows smartphones to be attached. Users see the content with each eye on a different image, creating the stereoscopic effect. There are versions that come with a simple joystick.

Figure 1: VR Box headset.

Source: https://www.magaluempresas.com.br/oculos-vr-box-realidade-virtual-3d-android-ios-controle-bluetooth-bbg/p/edag4f7ba8/te/orvc

One way to evaluate the effectiveness of immersive VR environments is to assess the sense of presence reported by users. According to Witmer and Singer (1998), presence is defined as the subjective experience of being in a location or environment, without physically being there. For users to feel present, immersive VR systems heavily rely on immersion and engagement. Engagement depends on focusing the user's energy and attention on performing tasks in the environment, while immersion relies on the user perceiving themselves as part of the environment, being stimulated to interact in the environment without consciously noticing how they performed the interaction.

Witmer and Singer (1998) developed a presence questionnaire (PQ) based on Likert-scale questions for users to respond, addressing various aspects of their feeling of control, substantially related to the feeling of immersion in the environment. They mentioned that systems with the lowest PQ scores were those in which users felt more nauseous. Lawson and Stanney (2021) define that this form of nausea, occurring due because of exposure to immersive environments, is known as cybersickness.

Despite the existence of VR games for various disciplines in SE, there is still a lack of specific options for RE. In related works such as Gulec et al. (2021), an environment is presented where the student experiences the role of a software engineer in a company specialized in developing software (softhouse), to practice some fundamental SE activities. Regarding RE, the player is provided with a screen showing all the project requirements and needs to identify which ones are correct. Concerning VR, the game is compatible with HTC Vive, but the only VR functionality is the player's movement in an office where there are NPC (Non-Playable Characters) employees. Upon finding them, the employees provide tasks through texts.

In Mayor and López-Fernández, students learn the concepts of the Scrum framework and experiences some artifacts (Sprint Backlog, Sprint Review, Retrospective, Burnout Chart, and Poker Planning). The game was developed using the Google Cardboard library, with VR resources for interactions with Scrum artifacts and dialogues with NPC colleagues, in addition to player movement.

From the first project, the task of requirements classification was considered as inspiration for the final phase of the game proposed in this work. We have developed a mini game at the end of the game, allowing students to classify the collected requirements into functional, non-functional, and domain categories. The second project was developed using the same technology as the game presented in this work, providing several inspirations for improving the user-application interaction. For instance, how to conduct the interaction with the NPC.

3 RESEARCH METHODOLOGY

For the development of this research, two teachers with professional experience in the RE phase participated. They provided the case study developed in the game and later allowed the validation of the game with their classes.

The game validation was carried out at the end of the semester with two groups, at different times but in the same way. Initially, a link to the application was sent to the students to test on their own mobile phones, to verify if the game was compatible with their smartphones, as each one would use their own device for validation.

On the validation day, students started by filling out a form with some data to identify themselves, such as gender, age, experience with VR, whether they have vision problems, and whether the test app worked. They also signed a document consenting to make all data available for analysis and publication, respecting anonymity. With this data, the class was stratified into two equally distributed groups. The formation of these groups will be discussed later. Each student brought their own mobile phone and headphones. The University provided the VR Box headset. For 20 minutes, one experimental group used the VR version, and the control group used the Windows version.

Then, the students were invited to answer a presence questionnaire adapted from Witmer and Singer (1998) with 28 Likert-scale questions ranging from strongly disagree (1) to strongly agree (7). Three adaptations were made in this presence questionnaire. First, we removed six questions related to moving and examining objects, as well as a question to localize sounds in the environment. This was motivated by the fact that in the game. Interactions are only with NPC (Non-Playable Characters), and there are no sounds in the environment to be localized. Second, we converted the questions into statements. According to Arnold et al. (1967), questionnaire consists of a series of declarative statements that measures a certain attitude. We added two statements to inquire about fun and three open questions to highlight positives, negatives, and provide suggestions.

Informally, as students finished the validation protocol, they were questioned about their impressions. This conversation allowed for some improvements regarding the resolution of game errors that were not evidenced by the students in last questionnaire about impressions. After the first class had experienced the game, in the following two weeks, the research team adjusted to the game according to the class's observations and suggestions.

4 DEVELOPED GAME

The VR Box headset was chosen to be used as the main hardware for the application due to their low cost and ease of acquisition. Through preliminary tests with the development team, it was found that users with vision problems would have difficulties in visualizing the texts, so all dialogues were dubbed. Thus, the student would have the possibility to read and listen to the characters' lines. The game was developed using the Unity game engine and the C# programming language, with the Google Cardboard XR Plugin framework used to distribute the game for Google Cardboard (referred to as the VR version). The 3D models were developed using MagicaVoxel, with the art style inspired by games like Minecraft and Roblox. The choice of this style was made to ensure better performance of the game, as it should be running on students' smartphones, which may have varying levels of processing power.

As the research questions RQ1 and RQ2 involved comparing the two types of VR, a second version of the game was created for Windows 64 bits, undergoing minor changes to accept interaction using a mouse.

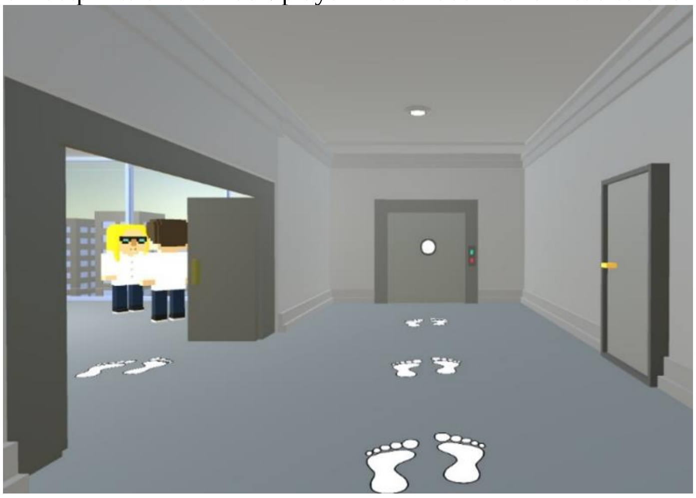
The protagonist, assumed by the student, comes from the future, and throughout the story, it is explained how and for what purpose they are at this point in our time. This main storyline was thought out following Malone (1980), which implies that one of the ways to make a game fun is the stimulation of curiosity. The student starts in a bathroom with a Voice in their Mind (VM) questioning what happened and encouraging them to explore the environment. The student takes on the role of a requirements engineer in a software house. Their task is to elicit requirements by interacting with the game's characters. The presented case study refers to developing a system to assist in controlling a volleyball match. The VM is always with the student, guiding the story, and providing specific tips on actions that can be taken at certain times. If the student remains inactive for a certain period, the VM encourages them to explore and investigate a location or interact with another character. This assistant works, as suggested by Schell (2008), as a mentor who maintains the student's pace by guiding them whenever they become idle or have experienced failure. After collecting all the information from the characters, the student is sent to a final stage where they need to classify the information as functional, non-functional, domain requirement, or discard the information.

The actions that the student can perform are moving around and talking to other characters. All the game experience data were stored in a database on a dedicated server. It is worth noting that, to analyse the game action data, students needed to authenticate themselves in the application, associating their interactions with the stored data.

One of the specified non-functional requirements for the game is the lack of a need for a controller to move or interact, as this could make the target hardware (VR headset) more expensive. Therefore, interactions were developed based on a fixed pointer in the centre of the user's screen. This pointer moves along with the user's head movements but always remains in the centre of the screen. When pointing to an object or character that allows interaction, a loading animation starts at the pointer, and the click is registered where the pointer is pointing.

The movement through scenario follows the on-rails guided principle (BISHOP; ABID, 2018), which means that the path where the student can walk is represented by a trail of footprints on the floor (Figure 2). Thus, when the user positions their head to point at a footprint on the ground, a loading animation will occur, and at its end, the user will be moved to a position immediately above the footprint from where they can move to other footprints or perform other interactions. Therefore, all interactions in the game are performed with the user's head movements, without the need for external hardware such as controllers or keyboards, for example. In the Windows version, the character's movement and interaction are based solely on mouse movement (while holding down the [Alt] key) without the need for a click action.

Figure 2: Trail of footprints on the floor: player must move his/her head to click on a footprint.



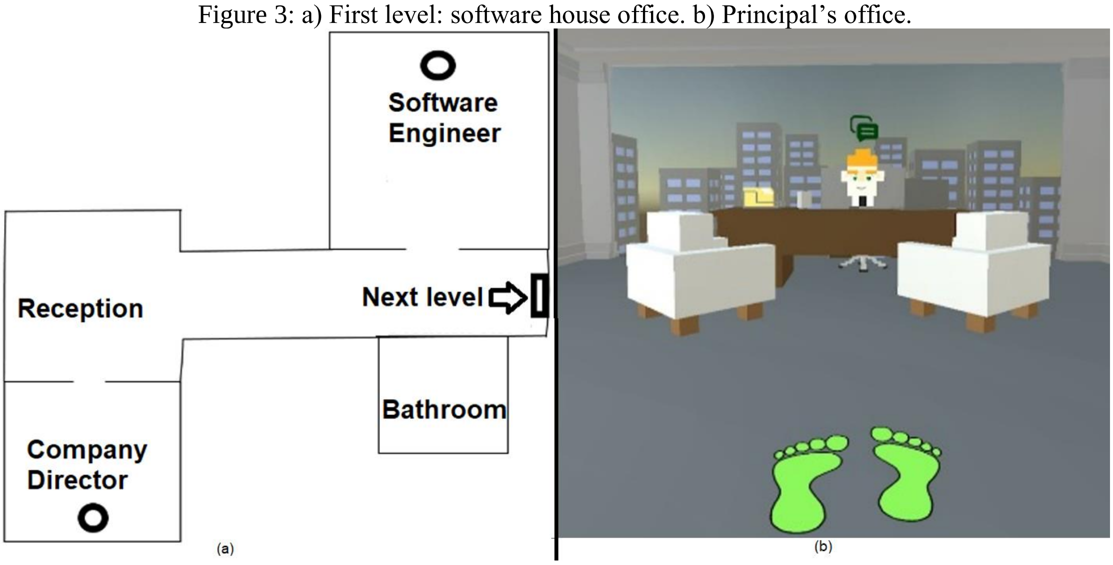
Source: Authors (2023).

After the student logs in, they start in the bathroom, where the protagonist wakes up and meets the VM. In the bathroom, there are some interactions so that the student can understand

REVISTA
CONTRIBUCIONES
A LAS CIENCIAS
SOCIALES

how the user interface works through head movements. They can also make some initial configurations, such as the speed of movement and the volume of dialogues and ambient sounds. Leaving the bathroom, the student is taken to the software house office (layout in Figure 3a). Although the office is divided into three environments: development sector, reception, and the director's room, the goal in this location is to talk to the company's director (Figure 3b), who explains concepts of elicitation of requirements and provides the first task to the student, that is, to proceed to the volleyball gym. The student needs to follow to the elevator door to be transported to the next location, which is the gymnasium. When it is possible to interact with the character, a balloon appears over their head. Figure 3b shows this feature when the player goes to principal's office.

Figure 3: a) First level: software house office. b) Principal's office.



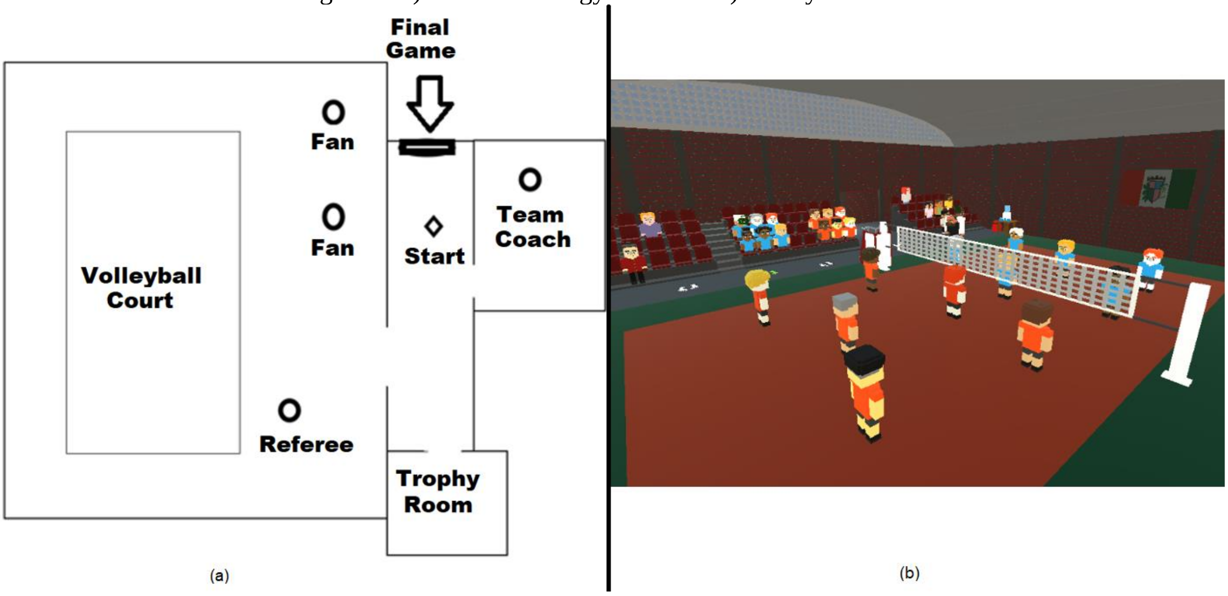
Source: Authors (2023).

Figure 4a shows the layout of the gymnasium, and Figure 4b illustrates an internal view of the gymnasium. Players need to talk to four characters (team coach, referee, and two fans) to collect the requirements for a volleyball match management application. In these interactions, the student will learn various aspects of requirements gathering and communication. Some dialogues are not useful as requirements, and others are redundant. As for communication: at the beginning of the conversation, there will be options in the menus where the phrases are rude or not very polite, resulting in responses that do not contribute to the task or even interrupting the conversation with the NPC. In Table 1, the ten required items are listed. When the student believes they have collected enough information, they must go to the elevator door to proceed to

REVISTA CONTRIBUCIONES A LAS CIENCIAS SOCIALES

the final game of requirements classification. To enter the requirements classification phase, the student must have collected at least half of all the available requirements in the gymnasium or have tried to pass through the door more than once (even if they haven't collected more than half of the requirements). In addition to these actions, to prevent the student from getting stuck in a phase for not being able to collect all the requirements, after 10 minutes, a click on the door will transport them to the requirements classification game.

Figure 4: a) Second level: gymnasium. b) Volleyball court.



Source: Authors (2023).

Table 1: Requirements in the volleyball use case (F: functional; NF: non-functional; D: domain).

ID	Description
F01	The system should keep track of the game scores.
F02	The system should maintain the game scoreboard with a record of the match using only the names of the teams.
F03	The system should indicate when the set or the match time has ended with the winning team.
D01	The volleyball match is divided into 5 sets. The team can only win the match if it wins 3 sets.
D02	To score a point within a set, the team must attempt to touch the ball on the ground of the opposing team's side.
D03	The scoreboard should start at zero for both teams. Each set has a specific point requirement: the first three or four sets have 25 points, and the last set has 15 points. The team that reaches the required score within the set earns one point on the scoreboard.
D04	In case of a tie during a set, the team will win the set if they manage to achieve a 2-point advantage, or a 5-point advantage depending on the scoreboard.
NF01	The system should display the game score on the gym's television screen.
NF02	The system should present a simple visual display.

NF03	The system should be compatible with Android systems.

Source: Authors.

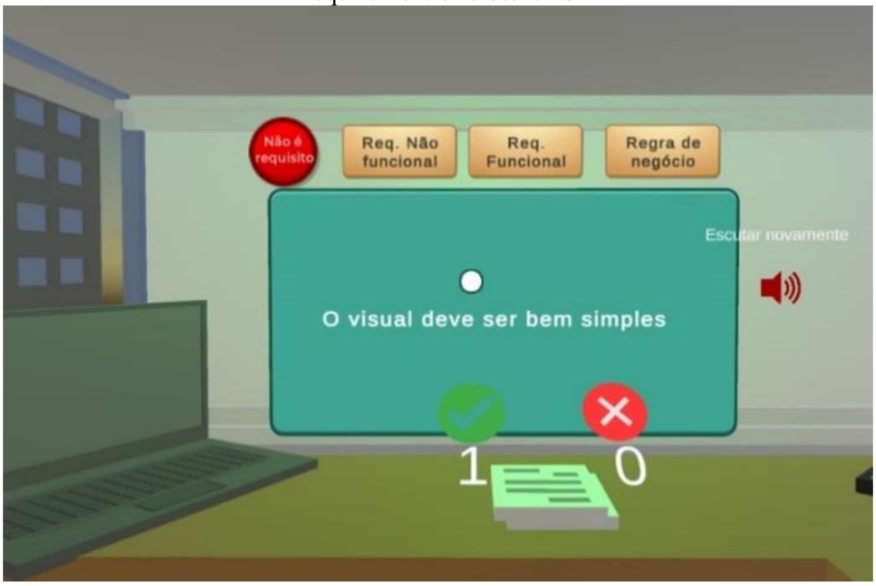
Figure 5 demonstrate what happens when the student finds a character and points at it. The interaction begins, and the character starts the conversation, presenting menus with possible answers, where the student clicks by moving their head and pointing as described earlier. This is the interaction method adopted in the game: a dialogue system which can be navigated through menu clicks. At the beginning, the VM explains that the software engineer needs to constantly take notes on useful information during the conversation with stakeholders. This information will be classified in the final game between functional requirement, non-functional requirement, domain requirement or discard it. In the same figure, part of the character's dialogue can be seen highlighted in yellow. If the engineer finds it pertinent to save this information, there is a [Save Speech] button so that they can classify it later. This form of game tutorial, presenting this part of the sentence in yellow, is highlighted in the first character to serve as an initial guide to the student, providing primary learning and making them feel more comfortable to proceed with the game. In the next characters, this highlighting of sentences no longer exists.

Figure 5: NPC Interaction.

Source: Authors (2023).

In this final stage, each saved speech is presented to the student, who must classify it as functional requirement, non-functional requirement, and domain requirement or discard it (Figure 6). The location is a meeting room with a panel displaying news about the future. Both the room and the panel undergo changes as the student classifies the requirements. If the classification is wrong, a catastrophic news is presented, and the room is destroyed with each mistake. If the student classifies correctly, the room is arranged, and a hopeful news is displayed on the panel. At the end of the mini game, the student is shown which requirements were missed to be collected and which ones were classified incorrectly. This way, the student can analyse which aspects he should pay attention to during the requirements elicitation process. The student is transported to another location, “The Final”, where they listen to a narration by the teachers according to their performance in the requirement classification.

Figure 6: Final game: classify texts saved during the gameplay as functional, non-functional or domain requirement or discard it.



Source: Authors

5 RESULTS AND DISCUSSION

The first group that tested the game belonged to the Software Requirements (SR) discipline, in the third phase, where 13 students tried the VR version, and 11 tried the Windows version. The average age was 20.9 years, with a population of 18 men and 6 women. Out of these students, 11 had prior experience with VR applications, and 12 students required glasses for vision correction. Fourteen students responded that the test application worked. Those who reported that it didn't work used iOS devices or mobile phones with incompatible hardware or software. The stratification's result considered only those students for whom the test application worked.

During the following week, the development team made improvements to the application. The most significant changes in the second version were as follows:

1. A new scenario at the beginning of the game before starting in the bathroom, so that the student could get used to the movements (clicking on footprints) and properly adjust the mobile phone in VR headset;
2. A tablet was added to the student's field of view to improve the visibility of the VM subtitles, which were previously displayed at the bottom of the screen. This tablet was located in the waistline of the user in the virtual world;
3. Sometimes there were so many menu boxes that the player had to turn their head to see all the options. It was revised to only have enough menu boxes for one screen;
4. Dialogues start automatically when the player comes close to the NPC and closes by itself when the player leaves a certain range. This took away the need for a click to open and close the dialogues;
5. No longer shows [Save Speech] button when texts are already saved as requirements;
6. The entire scenario was full of footprints, tending the player to click on more of them. Now there are fewer footprints on the way, prioritizing the main path;
7. Interactions and dialogues with NPC were revised to ensure a clear message with realism and proper context was being conveyed;
8. More VM tips and supports were added;
9. In the final game, the player was free to move his head around the scene, but not able to walk into the room. Now, the focus is into classifying the collected texts;
10. The final game room now starts in a messy environment and gradually becomes tidy as the student correctly classifies items, which is a positive reinforcement proposal.

After the improvements, the new versions of the game were tried by the second group, the Software Engineering Fundamentals (SEF) group, which is a subject for freshmen. In this first phase class, 9 students used the VR version, and 13 used the Windows version. The average age was 18.7 years, with a population of 15 men and 7 women. Out of these students, 8 had prior experience with VR applications, and 7 students required glasses for vision correction. Additionally, eight students reported having iOS devices, and four others had mobile phones with hardware that did not support the installation and use of VR applications.

Table 2 summarizes descriptive statistics of the game experience of the groups. The first two columns present data from the Software Requirements group. It can be observed that using VR, the average completion time was higher than the dedicated time (20 minutes) for validation, and consequently, fewer students completed the game. The “Requirements collected” row represents the number of requirements that were collected by those who finished the game. Regarding clicks, this also refers to those who completed the game. It is observed that the Windows version required a greater amount number of clicks, but the variability (standard deviation) between the two versions is practically the same. After interviewed the students and analysed the log records, it was found that those who used VR spent more time exploring the world and enjoying the game. Additionally, the investigators noticed that in the final game, students wanted to notice the difference between the tidy and the messy room and purposely made mistakes in classification.

Table 2: Descriptive statistics of the gaming experience of the two classes.

	Software Requirements (SR)		**Software Engineering Fundamentals (SEF)**	
	VR Version	Windows Version	VR Version	Windows Version
Students	13	11	9	13
Finished the game	12 (92.3%)	11 (100%)	4 (50%)	13 (100%)
Average completion time	25'04"	19'30"	15'15"	16'35"
Requirements collected	75.6%	82.6%	31.8%	61.5%
Average clicks	192.3 (± 57.0)	270.3 (± 58.4)	98 (± 25.4)	153.2 (± 24.02)

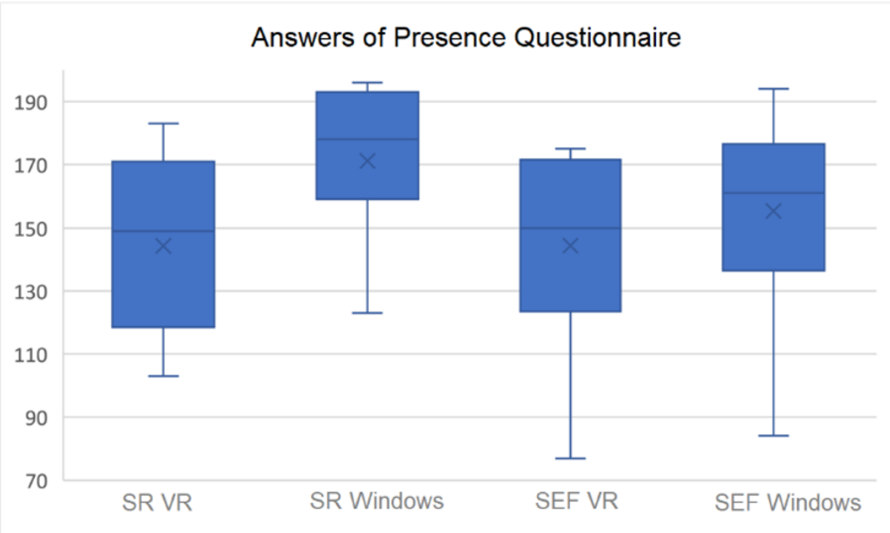
Source: Authors (2023).

The last two columns present the results of the experiment with the freshmen. It can be observed that the completion average was lower in the VR version compared to the previous group. Regarding the collected requirements, it was also significantly lower than the previous group. The main hypothesis to be considered concerns the maturity of the students compared to the phase in which they are located, as compared to the third phase. Furthermore, the subject of the third phase was about Software Requirements. Since the experiment was conducted at the end of the semester, third-semester students already had some knowledge about requirements. Only half of the students completed the VR version. One student stopped playing due to motion sickness. The others were interrupted when they reached 20 minutes. When questioned, these students responded that they either couldn't find the exit of the gym or were exploring the scenario. However, a significantly lower number of clicks was required compared to the previous version due to the improvement done in this second version.

Figure 7 illustrates the results of the presence questionnaire between the two versions in both classes. The maximum possible value is 196 (28 questions with 7 values each). From the minimum values in the SEF class, there were dissatisfied students, although the average was the same between SR and SF classes on the VR version. The size of each box also indicates a large variability in the sense of presence among the students.

The Cronbach's alpha value of the questionnaire in the SR class was 0.948, and in the SEF class was 0.957. A Shapiro-Wilk test of normality distribution was performed PQ in each of the two classes. The test results indicate that the data are not from a normally distributed population. The Mann-Whitney U test was used to compare opinions about presence between the two groups in each class. Statistical significance was set at $p \leq 0.1$.

Figure 7: Box plots of class responses on the Presence Questionnaire.



Source: Authors (2023).

Table 3 presents the individual means of the Presence Questionnaire items and highlights the items that have shown significant differences between the two groups in each class. The averages were all above 3.5, which is the median of the scale. There is no statistically significant difference between the preference for one version or another. However, it can be observed, corroborating with Figure 7, that the highest average was for the Windows version. Students agreed that they felt like they were in a real-world experience (items 2 and 11), got involved with the game (4, 8, 17, and 26), learned new things (24 and 25), and had fun (27 and 28).

In the SR class, there were more items with statistically significant differences, always in favour of the Windows version. For item 13, all groups from both classes agreed (≥5) that they managed to explore the world with vision. The visual quality (item 22) was more difficult for the VR group in the SR class. However, the improvements made for the second version must have worked well to improve the average in the second class.

It is interesting to observe item 16, which deals with the feeling of confusion or disorientation after the experience. In the SEF class, where only half of the students completed it. It had the lowest average in the entire questionnaire. Probably, they didn't pay attention on negative sentence.

To support the understanding of the results from the presence questionnaire, opinion forms were analysed and classified. Positive evaluations were similar in both versions, with highlights for gameplay, fun elements (art and dialogues), and feelings of learning about requirements.

Table 3: Averages of Presence Questionnaire (values statistically different, with $p \leq 0.1$, are presented in boldface).

#	Description	Req VR	Req Win	SEF VR	SEF Win
1	I felt comfortable using the controls in the game.	**4.00**	**6.00**	4.56	5.62
2	My actions were well interpreted in the game.	5.44	5.64	6.22	6.00
3	Interactions with the environment felt realistic.	5.00	5.91	4.89	5.00
4	I had the feeling that my vision and hearing were involved.	5,11	6.27	5.78	5.38
5	I felt involved with the visual aspects (scenes, characters, etc.) of the game.	5.11	5.73	4.67	5.08
6	I felt involved with the auditory aspects (dialogue, environment music, etc.) of the game.	**4.67**	**6.18**	5.67	5.54
7	Moving around the environment was a natural process.	**3.67**	**5.82**	4.56	3.85
8	I disconnected from what was going on in the real world around me.	5.56	5.82	6.00	5.23
9	Devices did NOT detract from my experience.	**4.11**	**6.45**	4.89	5.77
10	The information obtained by visual and auditory means were coherent and connected.	5.33	5.91	5.44	5.92
11	The experience provided felt realistic compared to the real world.	4.33	5.64	4.11	4.15
12	I could anticipate responses to my actions.	6.11	6.45	**5.11**	**6.23**
13	I was able to satisfactorily explore the entire environment using vision.	**5.00**	**6.73**	5.78	5.62
14	I recognized the sounds.	5.67	6.73	6.56	6.15
15	I found the feeling of moving around the game's environment convincing.	5.67	5.82	4.89	5.38
16	I did NOT feel confused or disoriented after finishing the experience.	6.11	6.00	**3.78**	**5.46**
17	I felt involved with the experience in the virtual world.	5.56	5.82	5.78	5.38

18	The devices did NOT distract me.	4.78	6.00	4.67	5.31
19	I did NOT experience delays between my actions and their effects on the game environment.	6.33	5.27	5.22	5.46
20	I quickly adapted to the experience in the game environment.	5.67	5.91	5.67	6.31
21	I felt mastery moving around and interacting with the characters in the game.	5.89	6.27	5.33	5.38
22	The visual quality did NOT distract me or interfere with performing tasks in the game.	**4.00**	**6.27**	4.78	6.00
23	The device did NOT distract me or interfere with performing tasks in the game.	**4.22**	**6.18**	**5.00**	**6.00**
24	I was able to focus on the tasks rather than the mechanics used to accomplish the tasks.	4.67	6.09	4.67	5.54
25	I consider that I have learned new things that will improve my performance on tasks.	5.56	6.36	5.11	5.85
26	I became involved in activities to the point of losing track of time.	5.11	5.73	5.33	5.62
27	I had fun with the game.	5.78	6.18	5.89	6.15
28	Something happened during the game (game elements, challenges, etc.) that made me smile.	5.89	6.09	6.00	5.85
Total average		**5.15**	**6.05**	**5.22**	**5.54**

Source: Authors (2023).

The main negative point for the Windows version was the way the character moved and turned his head. Since this version was an adaptation of the VR version, it was necessary for the user to press the [Alt] key to turn their head, and to move, they had to use the timing feature with the dot in the middle of the screen, which are not natural behaviours in first-person games for Windows and consoles.

On the other hand, negative points for the VR version included battery consumption (even overheating the phone), reports of motion sickness, and difficulties in reading the text. Regarding consumption, it is a known fact that the experience should not exceed 20 minutes. As for motion sickness (or cybersickness), this is the main problem when using this technology.

In fact, Riecke and Zielasko (2021) suggest using the teleportation technique to move around the scenario to reduce motion sickness. The difficulty of reading is related to the amount of information on the screen and the quality of the headset themselves. For this reason, voiceover dialogue was added so that students are not so dependent on the text.

Returning to the research questions, and based on the validation results, it can be noted that:

RQ1. How proficient are students able to perform tasks in the game?

In the first group, students with VR took longer than the expected time of 20 minutes to complete the game, which is also the time suggested by the literature to avoid motion sickness. The game was improved in aspects that led the students to finish the game in less time and significantly with fewer clicks. In the positive aspects, nine students found it important to mention the game's connection with education and considered that the interactions in the environment were like real ones. This was the primary intention of developing a 3D game with VR: to allow students to feel immersed in a business environment. Improvements in the game are still necessary, as will be discussed in the next research question, but here we provide a pathway for students to enhance aspects of communication and requirements discovery.

RQ2. How much can the game keep students immersed in performing requirements elicitation tasks?

This question is answered based on the presence questionnaire. Table 3 presents averages higher than 5 (1 to 7). The most found words in the positive feedback reported by the students were cute, fun, intuitive, engaging, voice acting, and the mechanics easy to understand. Analysing these aspects, we can conjecture that the game caught their attention and can be used as a supplementary resource in the learning of requirements elicitation.

However, there are aspects that need improvement since the ideal goal was to achieve averages above 6. We allowed them to also point out negative aspects and make suggestions for improvements.

In the VR version, in two groups, five students reported issues with the use of the headset, both due to dizziness and vision problems. The VR Box does not allow the user to wear their corrective glasses simultaneously, unlike other more expensive glasses like Oculus Quest and HTC Vive. Additionally, the ten improvements described at the start of this section were extracted from negative aspects reported by the first group.

In the Windows version, the most relevant negative aspect was the movement. The students didn't find it natural to walk by clicking on footprints. To move in Windows games, it is natural to use arrow keys or equivalent controls. We only recompiled the game for the Windows version and did not adjust in this regard. Now, at the end of the experiment, we concluded that both versions can be useful, and we can make these adaptations.

6 CONCLUSION

This paper presented two versions of a VR game developed to support requirements elicitation practices: one immersive version for use with Android and another non-immersive version to run on Windows. Firstly, validation was performed with 24 students from the third phase of the Software Requirements (SR) class. After improving the game, a new validation was conducted with another 22 students from the first phase of the Software Engineering Fundamentals (SEF) class. We divided both classes, and some students from each of them tested different versions. Twenty-two students used the VR version, and 24 students used the Windows version.

Our initial focus in this investigation was to develop a game with VR as a teaching application. In this validation, we were more interested in investigating the sense of presence than learning. We assumed that a type of VR interface that is not yet common among the public or developers would require improvements in its functionality, and its evaluation in terms of learning would be compromised.

In the SEF class, some students felt some dizziness, but all of them finished the game. In the SR class, only one student experienced motion sickness. In the evaluation of the sense of presence, both groups identified gaps in terms of movement and reported feelings of discomfort at the end of the experience, which allowed for further reflections on improving the immersive game to broaden its target audience. However, both groups enjoyed the game and had a positive overall experience.

The Windows version of the game showed better results both in terms of the feeling of presence and task performance. This demonstrates that the game has potential, but some improvements are necessary to test the learning content thoroughly. As mentioned by Porcino et al. (2021), the movement problem reduces the comfortable time limit for using VR. In fact, in the open-ended questions of students who used the VR version, them evaluated that they enjoyed the game and perceived its usefulness as a tool for learning requirements, although they declared to have a feeling of discomfort. The Windows version is an alternative to provide an opportunity to practice the skills addressed in the game for those who have experienced cybersickness.

The use of VR technology in educational software engineering games can be a viable alternative to assist in the teaching and learning process. Results indicate that the use of VR can provide a more engaging and immersive gaming experience, which can lead to greater student interest and engagement. Additionally, students who played the VR version spent more time exploring the game world and enjoying the experience, which can make learning more meaningful.

Some ideas for applying this game in disciplines: (i) it can be used before the subject of requirements elicitation, aiming to foster students' curiosity in understanding the process; (ii) as applied in this experience, it was used at the end of the semester to reinforce the concepts learned; (iii) in conjunction with a discipline where Unity can be used to implement VR applications, where students develop the case study model, describing the scenario, identifying the stakeholders and their relationship with the requirements, including writing the dialogues. The three examples depict significant learning in which the student is the active subject within the process.

The next steps, following the experience reported here, aim to study alternatives between reducing the space to be covered or changing the movement method and to develop a second case study for requirements elicitation based on a real company in the game. The technology used allows students to experience the game in any space, not necessarily in the classroom. As the university can lend the equipment, just as it lends library books, a new validation is planned to measure the effect of students using the game in an uncontrolled environment.

Anticipating the expansion of the application, the game was developed in an extensible manner, so that new case studies can be easily implemented. Therefore, as future work, organizing the implementation structure as a framework, to incorporate new case studies into the game. Another future work is a monitoring system of each student's progress, so that the teacher can track and intervene in the classes.

It is available for download (https://www.udesc.br/ceavi/gamelab/portfolio/rvnaer, all in Portuguese): the game in Android and Windows versions; a video demonstrating the usage of the game; and a document specifying the entire script of conversations for all NPCs.

REFERENCES

Andrade, G. V., Gomes, A. L. C., Hoinoski, F. R., Ferreira, M. G., Schoeffel, P., and Vahldick, A. (2022). Virtual Reality Applications in Software Engineering Education: A Systematic Review.

Arnold, W. E., McCroskey, J. C., and Prichard, S. V. O. (1967). The Likert-type scale. Today's Speech, 15(2): 31–33.

Bishop, I. and Abid, M. R. (2018). Survey of Locomotion Systems in Virtual Reality. In Proceedings of the 2nd International Conference on Information System and Data Mining – ICISDM '18, pages 151–154, New York, New York, USA. ACM Press.

Daun, M., Grubb, A. M., Stenkova, V., and Tenbergen, B. (2022). A systematic literature review of requirements engineering education. Requirements Engineering.

Ernst, N. A. and Murphy, G. C. (2012). Case studies in just-in-time requirements analysis.In 2012 2nd IEEE International Workshop on Empirical Requirements Engineering, EmpiRE 2012 - Proceedings, pages 25–32.

Freina, L. and Ott, M. (2015). A literature review on immersive virtual reality in education: State of the art and perspectives. In Proceedings of eLearning and Software for Education (eLSE), Bucharest, Romania.

Garcia, I., Pacheco, C., Méndez, F., and Calvo-Manzano, J. A. (2020). The effects of game-based learning in the acquisition of "soft skills" on undergraduate software engineering courses: A systematic literature review. Computer Applications in Engineering Education, 28(5):1327–1354.

Google (2022). Google Cardboard - Google VR. [Acesso em 18/10/2022].

Gulec, U., Yilmaz, M., Isler, V., and Clarke, P. M. (2021). Applying virtual reality to teach the software development process to novice software engineers. IET Software, 15(6): 464–483.

Lawson, B. D. and Stanney, K. M. (2021). Editorial: Cybersickness in Virtual Reality and Augmented Reality. Frontiers in Virtual Reality, 2.

Malone, T. W. (1980). What makes things fun to learn? Heuristics for designing instructional computer games. In 3rd ACM SIGSMALL Symposium, volume 162, pages 162–169, Palo Alto, California, USA.

Maturro, G., Raschetti, F., and Fontán, C. (2019). A Systematic Mapping Study on Soft Skills in Software Engineering. The Journal of Universal Computer Sciences, 25(1): 16–41.

Mayor, J. and López-Fernández, D. (2021). Scrum VR: Virtual Reality Serious Video Game to Learn Scrum. Applied Sciences, 11(19): 9015.

Pantelidis, V. S. (2009). Reasons to use virtual reality in education and training courses and a model to determine when to use virtual reality. Themes in Science and Technology Education, 2(1-2): 59–70.

Porcino, T., Trevisan, D., and Clua, E. (2021). A cybersickness review: causes, strategies, and classification methods. Journal on Interactive Systems, 12(1):269–282.

Pressman, R. S. and Maxim, B. R. (2021). Engenharia de software: uma abordagem profissional. AMGH, Porto Alegre, 9 edition.

Riecke, B. E. and Zielasko, D. (2021). Continuous vs. Discontinuous (Teleport) Locomotion in VR: How Implications can Provide both Benefits and Disadvantages. In 2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW), pages 373–374. IEEE.

Robertson, G., Card, S., and Mackinlay, J. (1993). Three views of virtual reality: nonimmersive virtual reality. Computer, 26(2): 81.

Rosa, L. H. C., Lucca, L. P., Lemos, E. L., Bernardi, G., and Medina, R. D. (2017). Jogos para Ensino de Levantamento de Requisitos de Software: uma Revisão Sistemática de Literatura. RENOTE, 15(2).

Schell, J. (2008). The art of game design: A book of lenses. Elsevier Inc., Burlington, MA.

Sommerville, I. (2016). Software Engineering. Addison-Wesley, Harlow, England, 10 edition.

Striuk, A. M., Semerikov, S. O., Shalatska, H. M., and Holiver, V. P. (2022). Software requirements engineering training: problematic questions. In CEUR Workshop Proceedings, volume 3077.

Tori, R., Hounsell, M. d. S., and Kirner, C. (2020). Realidade Virtual. In Tori, R. and Hounsell, M. d. S., editors, Introdução a Realidade Virtual e Aumentada, chapter 1, p. 11–29. Editora SBC, Porto Alegre, 3a edition.

Witmer, B. G. and Singer, M. J. (1998). Measuring Presence in Virtual Environments: A Presence Questionnaire. Presence: Teleoperators and Virtual Environments, 7(3): 225–240.