

Received 2 December 2022, accepted 11 December 2022, date of publication 14 December 2022,
date of current version 21 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3229210

RESEARCH ARTICLE

Authoring Tools for Procedural Modeling of Virtual Reality-Based Rehabilitation Exercises

DAVID QUINTANA¹, ANTONIO RODRÍGUEZ, AND IMMA BOADA¹

Graphics and Imaging Laboratory, Universitat de Girona, Girona, 17003 Catalonia, Spain

Corresponding author: Imma Boada (imma.boada@udg.edu)

This work was supported in part by the MCIN/AEI/10.13039/501100011033 under Grant PID2019-106426RB C31, Grant PDC2021-120997-C32 and Grant PRE2020-095947, and in part by the ESF Investing in Your Future.

ABSTRACT Patient compliance is one of the key factors for rehabilitation programs to be effective. To increase patients' engagement game technologies and virtual reality can be applied. However, the advantages provided by these techniques are not enough to be applied in real scenarios since experts do not always have the required technological skills. To overcome this limitation, a new virtual reality-based rehabilitation system that combines virtual reality, procedural modeling, and authoring tools, is proposed. The system provides easy-to-use editors for the experts to prepare rehabilitation exercises that will take place in an expanding and varied open virtual world created using procedural generation strategies. Patients using a virtual reality headset explore this immersive virtual scenario and solve challenges by applying actions according to their rehabilitation goals and receiving success/failure feedback. While the patient interacts with the exercises through immersive virtual reality, the expert supervises the patient's actions via a computer monitor where the exercise is displayed at the same time as in the headset. The paper presents the system and the evaluation that has been carried out to set the system configuration parameters that allow non-experts the easy creation of varied open virtual worlds for rehabilitation purposes. In addition, it collects the first impressions of rehabilitators which have been very satisfactory.

INDEX TERMS Virtual reality, procedural modelling, authoring tools, open worlds, rehabilitation.

I. INTRODUCTION

Rehabilitation is defined as a set of interventions designed to optimize functioning and reduce disability in individuals with health conditions in interaction with their environment. It is considered an essential part of universal health coverage that may be required at any point of our lives, following an injury, surgery, disease or illness, or because our functioning has declined with age [1]. Rehabilitation is a main focus of research and development with a large number of proposals and products aiming to improve patient's quality of life and make rehabilitation programs more efficient and effective [2], [3], [4], [5]. In this context, the proliferation of low-cost communication technologies, as well as gaming strategies, have opened up an array of opportunities for more efficient and cost-effective rehabilitation sessions which can be done remotely using telephone, video-conferencing, telemonitoring or web-based applications [6], [7]. In addition,

it has been seen that using game technologies and Virtual Reality (VR) techniques, patients' engagement increases and also the rehabilitation effectiveness [8], [9], [10], [11], [12], [13], [14]. However, the advantages provided by these techniques are not enough to be applied in real scenarios. The lack of computer skills of many therapists and experts has been pointed out as one of the main limiting factors for their use [15], [16]. To tackle this problem, strategies that make the creation of rehabilitation sessions easier while exploiting the advantages of current technologies are required. With this aim, in this paper, a new VR-based rehabilitation system that combines VR, procedural modeling, and authoring tools is proposed.

VR is an advanced form of human-computer interface that allows the user to interact with and become immersed in a computer-generated environment in a naturalistic fashion. Through the use of technical devices such as head-mounted displays, desktop computers, video capture systems, tracking systems, and motion-sensing gloves, virtual experiences that closely resemble everyday environments and

The associate editor coordinating the review of this manuscript and approving it for publication was Tai-Hoon Kim.

actions can be created and used for rehabilitation [17]. In recent years, the use of commercial and custom-made VR-based gaming systems for rehabilitation has significantly increased [18]. Proposed systems have been customized to target specific patients' disabilities including reduced motor function, mobility, postural control, or cognitive impairments, among others [13], [19]. The specificity of these solutions makes their application to more global scenarios challenging.

Regarding procedural modeling, although there is no unique definition of the concept, it can be described as a generative technique that, given a set of input parameters, is able to semi-automatically or automatically produce a specific type of content. This technique is used in movies, games, and simulations. For more details on the topic see [20]. In the context of rehabilitation, procedural modeling will allow the creation of variable scenarios that increase patient motivation and compliance by avoiding the monotony of scenarios that induce boredom [21], [22].

The advantages of VR and procedural modeling have led to their combination being a powerful tool for rehabilitation [23]. Since its application requires specific technical knowledge, our system will provide authoring tools (or easy-to-use editors) that reduce the complexity of use [24]. These editors will allow non-technology skilled professionals to create immersive VR scenarios that make rehabilitation routines challenging as well as enjoyable for patients, to enhance their adherence to the therapy. To create these editors the technical evaluation of the system will be necessary to define the default parameters that will ensure the correct creation of scenarios as well as real-time exploration. The aim of the paper is to present the rehabilitation system, the evaluation that has been carried out to configure the parameters required to create VR scenarios, using procedural modeling and satisfying the demands of the experts and also collecting their first impressions.

Besides this introduction, the paper has been structured as follows. In Section 2, Related Work is presented. In Section 3, Material and Methods, the requirements and main components of the proposed rehabilitation system are described as well as the different tests that have been designed to set its parameters and collect the experts' impressions. Results and discussion are given in Section 4. Finally, Conclusions and Future Work are presented in Section 5.

II. RELATED WORK

In recent years the use of VR has increased considerably, being applied in different areas such as health [25], tourism [26], training [27], or industry [28], just to name a few. Generally, these applications are created by experts, with advanced knowledge on programming, 3D modeling, geometry, etc., using specific software tools such as Unity [29], Unreal Engine [30], CryEngine [31], etc. These tools include visual programming editors that make VR content creation easier, faster, and more efficient, although they still require a high level of technological knowledge to use them. To help

programmers on the creation process, there are also immersive authoring tools that allow them to create VR content while immersed in the virtual environment. In the context of industry, for instance, Cassola et al. [32] proposed a novel immersive authoring tool for experiential and situated learning in virtual reality-based training. Focusing on academic content creation, Lopez et al. [33] reviewed VR tools for the creation of virtual learning environments analysing how these kinds of tools can be combined with authoring tools in intelligent tutoring systems that present to students realistic, interactive and immersive educational content. Zhang and Oney [34] introduced FlowMatic, an immersive authoring tool that allows programmers to specify behaviors that react to discrete events such as user actions, system timers, or collisions and also provides primitives to create and destroy objects, abstract and re-use functionalities, and import three-dimensional models. More recently, Coehlo et al. [35] surveyed the existing literature on authoring tools for immersive content creation analyzing its features and how they are evaluated.

Note that all these proposals have been designed focusing on the programmers and less attention has been given to the final users. In most of cases, final users have been considered during requirements gathering to detect the needs of the application area and also during testing. Moreover, due to the novelty of the approaches, the integration of authoring tools for these final users has not been considered. As a consequence, modifications to VR scenarios generally require continuous interaction with the programmers. To overcome this limitation, authoring tools for final users also have to be considered. The main proposals on this topic have been done in the education area, where different authoring tools to prepare educational content have been proposed. In the context of rehabilitation new tools are required to cover the needs of the sector.

III. MATERIAL AND METHODS

A. SYSTEM REQUIREMENTS

The requirements have been defined considering the needs of the rehabilitation team and the two profiles that have to be supported: (i) the rehabilitation team, composed of doctors, nurses, physiotherapists, etc., and responsible of rehabilitation sessions and (ii) the patients which have to perform the exercises defined by the experts in the context of a rehabilitation session. Since the key element of rehabilitation sessions are the rehabilitation exercises, we will start presenting their requirements.

1) REHABILITATION EXERCISES

Focusing on rehabilitation exercises, the proposed system should allow:

- Immersive VR: To avoid external distractors, the exercises will take place in a virtual scenario explored using VR with a consumer-grade headset.
- Avoid repetition: The exercise scenarios will have an expanding and varied virtual world that generates new

content on the fly in order to avoid repetition and monotony.

- City-focused procedural environment: Exercise scenarios will reproduce different cities connected by roads and surrounded by a different type of vegetation. Focusing on the cities, these will be composed of different types of buildings, houses and other constructions which will have different distribution depending on parameters such as density of population, type of neighbourhood, etc. Between buildings different type of roads will be considered. All these elements will be placed on a terrain that can have different heights. In addition, the sky will allow for day/night cycle, weather effects with clouds and rain, and miscellaneous elements such as bird flocks with different sizes and formations.
- Fun: The exercises will be presented as games to motivate patients to perform rehabilitation tasks. To achieve game challenges different type of actions related to the rehabilitation goals will be required. This first prototype will be centered on cognitive rehabilitation.
- Flexible Difficulty: Exercises will support different levels of difficulty to fit different patient profiles.

2) PATIENTS

Focusing on the patients, the proposed system should allow:

- Immersion: To explore an immersive virtual scenario and solve challenges with specific actions related to their rehabilitation goals.
- Clear Feedback: To receive feedback on the performance of the actions.
- Adaptive difficulty: To adapt the level of challenge to the patient's performance.

3) REHABILITATION TEAM

Focusing on the rehabilitation team, the proposed system should allow, for the exercise set-up:

- Configurable play style: Allow choosing between multiple game modes, such as flying over a city, riding a car, or cycling.
- Configurable environment: Provide options for configuring the terrain shape, the frequency and variability of the trees, the frequency and variability of the cities, the kinds of cities that will appear with the kinds of buildings and roads they will contain, and the kinds of obstacles that will be encountered.
- Rehabilitation exercise configuration: Provide an intuitive user interface to configure the values above.

And for the rehabilitation session:

- Configurable session length: Give the rehabilitator the choice of how long (seconds/minutes) the play session will run for.
- Configurable difficulty goals: Allow configuring the target difficulty, such as the time between subsequent obstacles, or the magnitude of the required change in position.

- On-line control: Allow seeing the state of the rehabilitation session, interact with the objects on behalf of the patient, and change some of the configuration values during the session.

4) NON-FUNCTIONAL REQUIREMENTS

In addition to previous requirements, we consider necessary for the function of the system:

- Wide range of supported devices: Support a variety of standard gaming headsets connected to a host computer.
- Multi-language display: Allow selecting between different user interface languages.
- Support Stationary VR: The patient may need to be sitting on a chair, and may not be able to walk around the environment themselves.

B. THE REHABILITATION SYSTEM

Taking into account defined requirements, a system that supports rehabilitation teams in the preparation and execution of rehabilitation sessions, and patients in the performance of rehabilitation sessions, is proposed. Exercises take place in an expanding and varied open virtual world created using procedural generation strategies. Patients using a VR headset explore this immersive virtual scenario and solve challenges by applying actions according to their rehabilitation goals and receiving success/failure feedback. While the patient interacts with the exercises through immersive VR, the expert supervises the patient's actions via a computer monitor where the exercise is displayed at the same time as in the headset. Both views have a shared exercise visualization component but differ in the methods of interaction, and controls. The main components of the system are presented in Figure 1 and described below.

1) PLAY AREA GENERATOR

The *Play Area Generator* generates and sets up the exercise scene. It consist of two modules:

- The *Virtual World Generator* procedurally models the open-world scene. Its components are: (i) The *City Generator* which calculates the cities that are within view distance of the player and makes sure they are generated. The generation process first calculates the road network for the city, and then adds buildings to the sides of the streets [36]; (ii) The *Terrain Generator* which calculates the shape of the terrain and the terrain properties at each point, such as the texture and smoothness, accounting for the generated cities. Then, it places forests and other decorations on the terrain, including city buildings, streets, and roads; (iii) The *Sky & Weather Manager*, which manages the elements related to the sky which are not tied to the terrain, such as the sky box texture (sky color, stars, sun, and moon), clouds, and weather and atmospheric effects (rain, snow, dust, smoke, ...). It also takes care of flying decorations such as birds; and (iv) The *Objective Generator* which places objectives on the world, such as obstacles to avoid. The generation

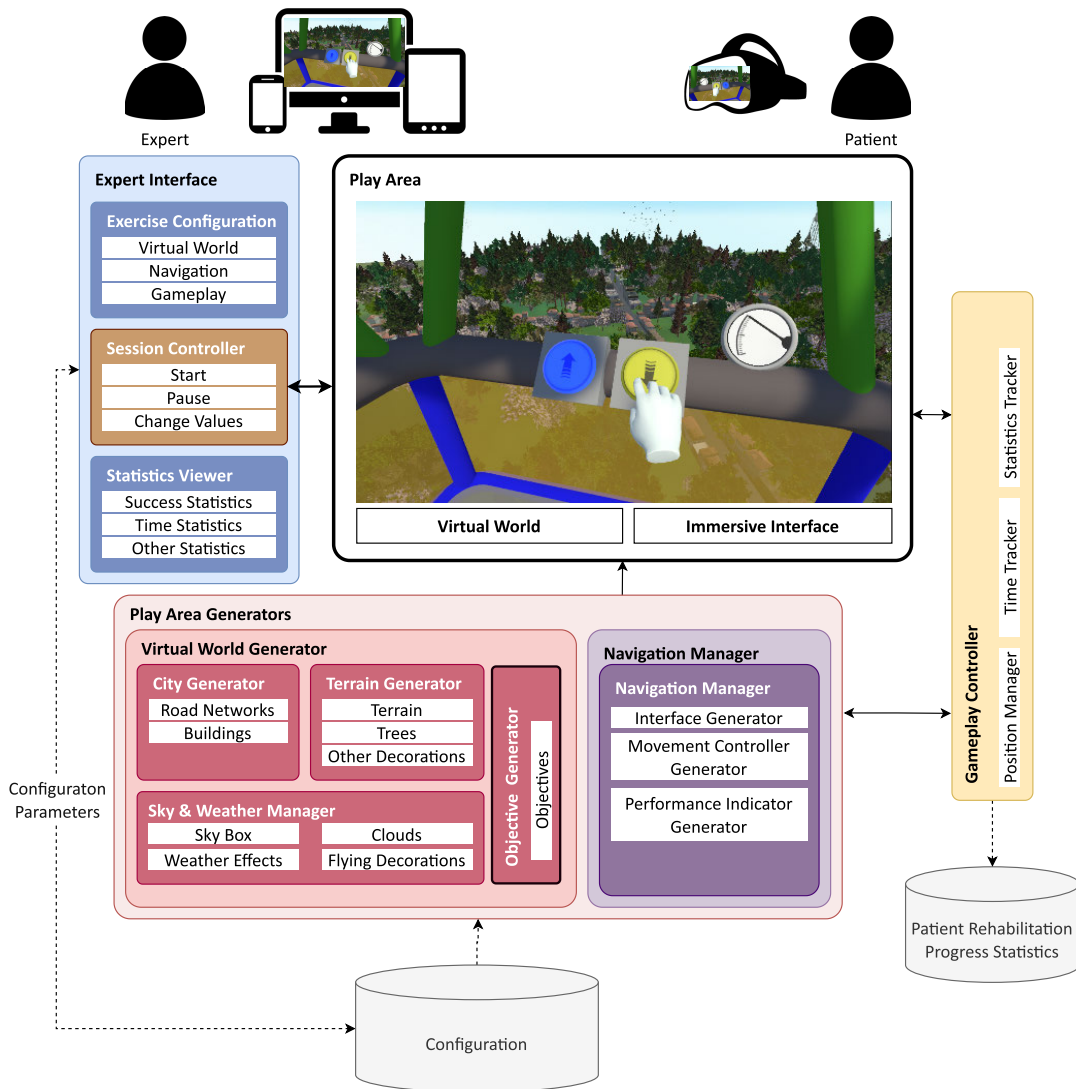


FIGURE 1. Main components of the proposed framework to support immersive VR rehabilitation exercises with scenarios procedurally generated.

depends on the type of rehabilitation objective and the configuration.

- The *Navigation Manager* controls the virtual scene navigation via a vehicle such as an air balloon, a car, etc. It has three components: (i) The *Interface Generator*, which sets up the immersive user interface, consists of a 3D model of the interior of the chosen vehicle, if applicable, on which appear the interactable elements, and the feedback indicators; (ii) The *Movement Controller Generator*, which sets up the logic for how the vehicle will move within the virtual world (flying, riding on roads, . . .), and if necessary, calculates the path the vehicle will follow; and (iii) The *Performance Indicator* which generates additional information, such as confetti for success conditions, or explosions for certain failures.

As shown in Figure 1 the Play Area Generator receives input configuration parameters introduced by the experts via Expert

Interface and generates the Play Area where the rehabilitation exercise will be carried out as output.

2) EXPERT INTERFACE

The *Expert Interface* is an authoring tool that allows non-experts to generate and control virtual immersive scenarios. It consists of three modules.

- The *Exercise Configuration* allows the expert to create, edit, and delete configuration presets for exercises, including the goals and challenges, the type of gameplay (such as raising or lowering a hot air balloon or steering a car), and the environment definition (virtual world and immersive interface). The environment configuration includes: the view distance, the distance between cities, the maximum radius of a city, the distance between trees, or the style of buildings in the city. A set of tests specifically designed have been used to set the parameter values.

- The *Session Controller* allows the expert to select an Exercise Preset, assign specific configuration values that have been declared in the preset to be configurable from this screen, and start a session with the chosen settings. While in the session, the expert is able to interact with the play area on behalf of the patient, pause the exercise, and change values for which the change does not require starting a new session.
- The *Statistics Viewer* gives the expert a place to see the stored information, and follow up on the patient's progress, comparing it against previous sessions and seeing evolution over time.

3) PLAY AREA

The *Play Area* contains all the elements generated by the *Play Area Generator*, including the Virtual World and the Immersive Interface. The Virtual World is made out of the results of the Virtual World Generator, including the terrain, the cities and roads, and the sky. The Immersive Interface is made out of the generated results of the Navigation Manager, specifically, the Interface Generator, which creates the vehicle panel model, and places on it the chosen Interactables such as the steering wheel or up/down buttons, and Indicators such as the speedometer, altimeter, or if an object has been loaded onto the vehicle.

4) GAMEPLAY CONTROLLER

The *Gameplay Controller* manages the state of the exercise, and gathers statistics. It has three components: (i) The *Gameplay module*, which connects the Immersive Interface with the Virtual World Generator. It keeps track of the current position in real-time, detects success and failure conditions, and triggers the appropriate feedback; (ii) The *Time Tracker*, which keeps track of the run time of the exercise, and can automatically terminate the exercise when the time limit has expired. This time is also available to be queried by other statistics to timestamp them and calculate time-related information such as reaction times, and (iii) The *Statistics Tracker* which keeps track of the relevant statistics, and records them on the patient statistics database. The statistics will then be accessible by the experts to evaluate the patient's performance. These statistics will also be used to automatically adjust the difficulty parameters of the exercise in response to the patient's performance.

In Figure 2 a schema of the user interactions and the system control flow is illustrated.

C. IMPLEMENTATION

The system has been implemented with Unity game engine [29] due to our existing expertise and out-of-the-box support for VR headsets. Although this engine supports a number of VR back-ends through their API, the OpenXR [37] backend has been selected since this supports all of the headsets we have access to. This library has been used via the Unity XR API [38]. This can be used via the legacy input system [39] or via the new input system package [40]. This

last one has been selected due to its flexibility and its ability to process headset controller inputs transparently. On top of that, we will be relying on the XR Interaction Toolkit [41] package, which handles a lot of the complexity of controller tracking and interactions.

The generation of the virtual world happens in a series of steps. The most important ones are: 1) City graphs are generated by selecting start points from a hexagonal grid with density and variance. A macrostructure consisting of major streets is grown from this start point, zones are derived from the macrostructure, and minor streets are generated within these zones. Finally, neighboring cities on the hexagonal grid are randomly connected with roads. 2) Objectives are selected based on the city structure and terrain properties, and can replace parts of the city structure if necessary. 3) The terrain is generated using a square grid of tiles, using Perlin noise [42] sources for the different attributes of the terrain (elevation, forest density, etc.). The visuals for the streets, buildings and objectives are generated for the parts of the city graphs that overlap each piece, generating the street models, and adding buildings on the sides of the streets with plot sizes and height limits decided based on the parameters at each point. After filling the piece with the city data, trees and other decorations are laid out by selecting points on hexagonal grids that are not overlapping the existing models. The main calculations are performed in a background thread to avoid slowing down the gameplay and rendering, but due to the limitations of the engine, the steps which create the 3D geometry and place it in the virtual world happen in the main process thread.

D. USABILITY TESTS

To evaluate the usability of the system, three rehabilitators were asked to prepare rehabilitation exercises. No information was given to them but only the access to exercise creations interface illustrated in Figure 3. From top to down, this interface has the *Gameplay Settings* section with three parameters: (1) the *Gameplay Type* assigns a type of vehicle and interaction mode and has three options: (i) balloon with up/down controls; (ii) car with left-right controls and (iii) bicycle controlled by activating pedals in rhythm; (2) the *Objective Type* select the goals for the exercise and has three options: (i) Obstacle: use the gameplay controls to avoid obstacles; (ii) Collect: use the gameplay controls to collect the prizes; (iii) Spot: identify and point toward the targets; (3) the *Feedback Type* to select how the exercise will display success and failure situations. It has three options: (i) Confetti (success) and Explosion (failure), (ii) Confetti (success) and Sad faces (failure), (iii) Smile faces (success) and Sad faces (failure). In addition, a set of sliders are provided to modify the default values of the *Movement Speed* (the default is a moderately slow speed adequate for patients with mild impairment.) and the *Virtual World Settings*. These last are: the *Terrain Elevation Variance* (the default has a mixed terrain with a combination of flatter areas and hills), the *Forest Distribution* (the default is a mix of forest areas

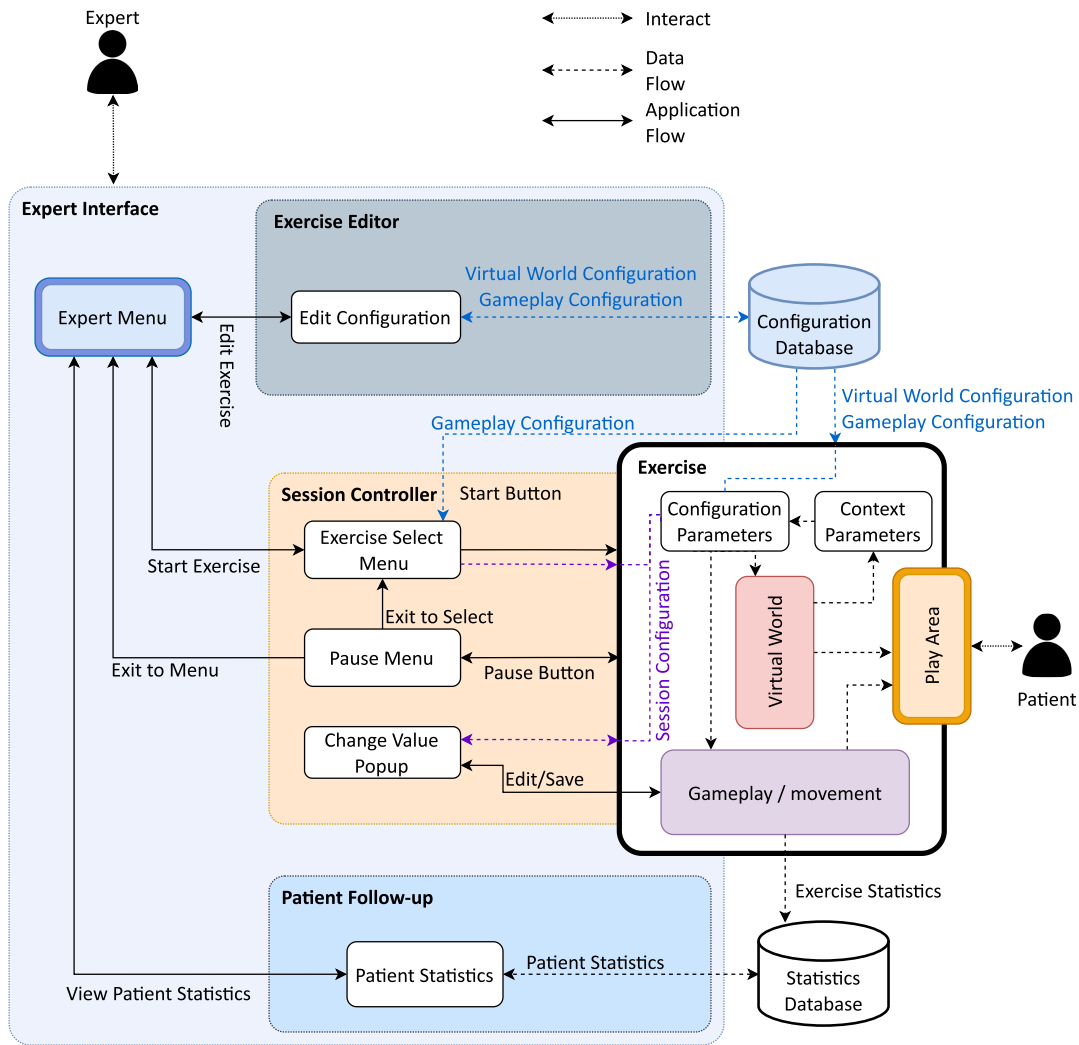


FIGURE 2. System schema considering expert and patient interactions and the application and data flow.



FIGURE 3. Exercise creation interface.

and grass areas), the *Distance Between Trees* (the default is a medium setting with a substantial number of trees), the *Distance Between Cities* (the default is a terrain where new cities appear in the distance, but there is green space in between), and the *Maximum City Size* (the default is a size that generates cities that feel large and expansive, but not endless). In the *City Generator* settings, the distance between

cities, and the radius of the cities, can be selected. Finally, the *Display Settings* changes how the virtual world is presented, by changing the view distance of the camera. The menu provides a set of default values for each option determined by the different tests presented in the next section. This interface has also a help option to access a document with detailed information about each option.

E. PARAMETERS SETTING

The creation of virtual scenarios requires a set of parameters to be configured. To make this process as simple as possible different tests to evaluate the influence of each parameter on the system performance have been designed. The goal is to ensure the best immersion in the procedural-generated virtual world while graphical quality is preserved. A stable target of 90 frames per second (fps) has been considered [43].

1) TEST1(VIEW DISTANCE)

Test 1 will evaluate the view distance parameter to find a range of View Distance that makes the virtual world large

enough to be immersive, but not so large that the performance drops below our target. The view distance also changes the generation distance, which needs to remain large enough to avoid generating content inside already visible areas. It will be adjusted automatically based on the view distance and the sizes of the cities. The view distance will start at 1000 meters, and increase by 100 meters every 5 seconds, until the 5 second fps average drops below 45 fps for 10 frames in a row.

2) TEST2(DISTANCE BETWEEN CITIES, MAXIMUM CITY RADIUS)

Test2 will evaluate the distance between cities and the maximum city radius to find how much the presence of cities affects the performance. It will start with small cities that are far apart (10000 meters with a city limit of 500 meters) and test up to very large cities very close together (3000 meters distance with a limit of 2000 meters). It will use the view distances at 50% and 90% of the range from Test 1 and see how the different view distances affect the results.

3) TEST3(CITY STYLE)

Test3 will evaluate the city style to find how much the number of buildings and streets affects the performance. It will use the settings for 90% view distance and the maximum amount of cities on screen (big cities close together, as per Test 2). For this, two extra configuration sets have been prepared, one for historic city, with low density of buildings, which favors long streets and large plots, and one for high-rise city, which favors shorter streets with smaller plots, and has a larger density of buildings.

4) TEST4(TREE DENSITY)

Test4 will evaluate the tree density to find how much the number of trees affects the performance. It will use a configuration of the city generation which uses a small radius, large distance, and low density, to maximize the area available for trees to appear in. The view distance will be set to the 50% point of the results of Test 1. The distance between trees will be set to 10, 20, 50, 80, and 100 meters.

5) TEST5(MOVEMENT SPEED)

Test5 will evaluate the movement speed to know how the movement speed affects the performance. Higher movement speeds will cause more of the virtual world to be visible during the exercise, so the total amount of procedural generation is expected to be higher.

F. TESTING DEVICES

The tests will be carried out on a High-end Desktop PC from 2018 with an Intel Core i7-9700K CPU, an NVidia RTX 2080 GPU, and 16GB of DDR4 RAM. Two headsets will be used. The first is a VIVE PRO EYE with 2-room position trackers, which relies on the PC to generate the graphics, and has a display update rate fixed to 90 frames per second (fps). The second is an Oculus Quest 2 device, which can work connected to a PC (Oculus Link), or standalone

(as an Android device). The Oculus Quest 2 device has a configurable update rate when connected to a PC, which will be set to 90 fps for consistency. In standalone mode, where the hardware is much more limited, the update rate is always limited to 72 fps.

For the performance tests, the time between iterations of the engine's update loop, which includes the time spent on processing logic and the time waiting for the graphics card to finish drawing and updating the headset display, will be measured. The vsync feature works such that after a frame is completed the graphics device waits until the display output is about to start sending the next frame to swap in the contents of the newly rendered frame. This means that we expect the loop to take a multiple of the frame interval, which is 11.1 milliseconds for the 90 fps sync interval of the headset.

G. USE CASE

Since our interest is in cognitive rehabilitation exercises, the tests will consider a use case of an open-world exercise where the patient sees themselves on a hot air balloon and has to avoid obstacles by raising and lowering the balloon as it moves along a fixed path (see Figure 4). This case has been chosen because of the open-world nature, which requires larger view distances and as such, will have a much bigger load on the hardware than it would be required using an interior scene, or view from the ground level.

IV. RESULTS AND DISCUSSION

As a primary use case for the tests, the VR headset connected to a PC has been considered. In addition, the performance limitations of the Quest 2 headset in its standalone mode have also been tested.

A. VIEW DISTANCE EFFECT ON PERFORMANCE

The results of Test 1, where view distance is changed, and by extension also the generation distance, are presented in Figure 5(a-c). The view distance affects the quantity of objects on screen, increasing the time required to prepare and render each frame, while the generation distance similarly increases the procedural generation work. From Figure 5(a) it can be seen that as the view distance (orange) grows linearly, both the render times (blue) and the script overhead (gray) grow, but the majority of spikes leading to slowdowns in frame rate appear to come from the rendering, while procedural generation is spread over multiple frames. From Figure 5(b) it can be seen that when including the v-sync delays, the slowdowns do not show through until well past the 6000 meter mark. From Figure 5(c) this observation can be confirmed, showing a straight line of stable frame rate until the view distance is over 6000 meters. Visually, distances between 3000 and 6000 meters feel open and expansive, while remaining within the performance target. This provides a wide range of distances in which there is acceptable performance, while maintaining the sense of immersion. In Figure 6(a) it can be seen that the virtual world ends abruptly, while Figure 6(b) and (c) give a sense of horizon.

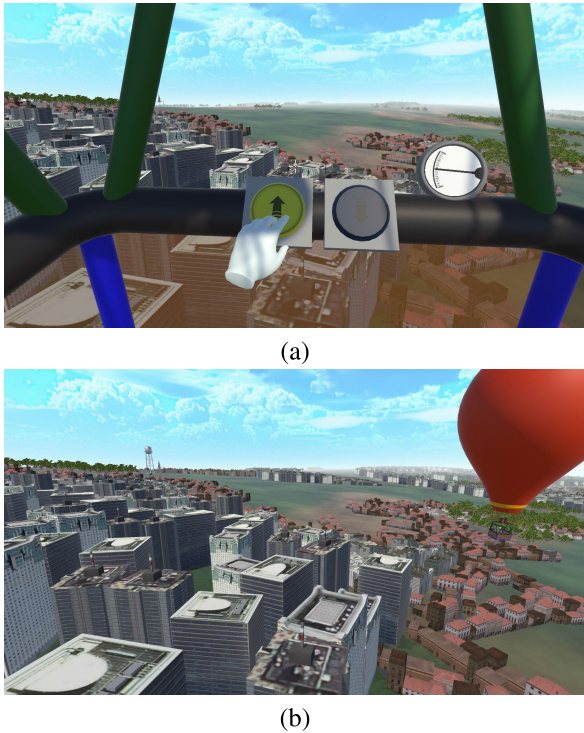


FIGURE 4. Exercise use case. A hot air balloon flying over a procedural terrain with cities.

Therefore, the supported values for these parameters range from 3000 to 6000 meters.

B. DISTANCE BETWEEN CITIES AND CITIES RADIUS EFFECT ON PERFORMANCE

The results of Test 2, where the distance between cities and the maximum radius of the cities are changed, are presented in Figure 7(a). Based on the previous results, a 8800 meters setting was used as the upper limit, with the 50% and 90% view distances being set to 4400 and 8000 meters, respectively. When generating small cities far away (distance 10000 meters and radius 500 meters), the performance was good with both view distances, as was with big cities close together (distance 3000 meters and radius 2000 meters) when the view distance was low. However with big cities close together, due to the quantity of buildings and roads on screen at the same time, if the view distance is large the performance quickly degrades to 45fps as the world is progressively generated, and afterward it remains stable at 45 fps. Changing the distance too far from the default means that the proportion of urban areas to natural landscapes is too skewed toward cities or forests, and a value of 4500 meters between cities with a city radius of 2000 meters will be used as the optimal trade off. In Figure 6(d) it can be seen that making the distance between cities too large leads to big expanses of monotonous terrain. In the settings, the range for distance between cities will be between 3000 and 10000 meters and default to 4500 meters, and the range for radius of the cities will go between 500 meters and 5000 meters, and default to 2000 meters.

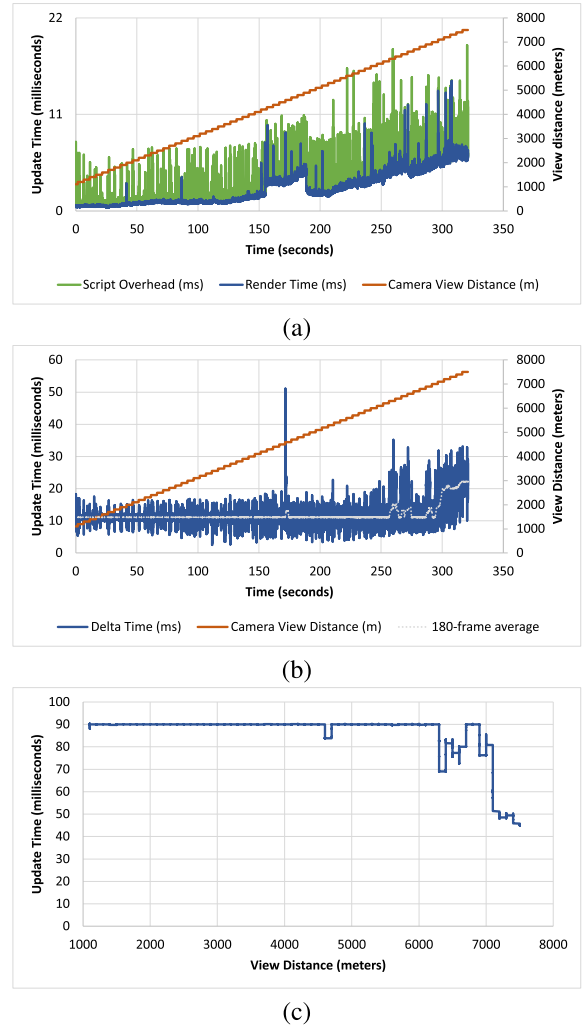


FIGURE 5. Results of Test 1 where (a) is the influence of procedural generation on cpu usage (b) is the influence of procedural generation on the time between frames and (c) is the relation of view distance on the average measured frames per second.

Having both low distance and large radius at the same time is discouraged.

C. CITY STYLE EFFECT ON PERFORMANCE

The results of Test 3, where the style of the cities is changed, are presented in Figure 7(b). This test was performed using a view distance of 8000 meters, at the point where the default configuration set already caused performance to drop. This number was chosen to ensure that the results would have a definite influence in performance. Despite that, this change did not have the expected effect, and instead showed that the performance remains the same when using city styles with less density of buildings, and the styles with higher density. Figure 6(e) and (f) show, in contrast with Figure 6(c), that the best experience is obtained with a variety of zone types, and limiting to low or high density zones leads to lower visual quality. Therefore, the choice of city style will not play a factor in the conclusions, but presets for mixed, high rise residential, and historic are provided.

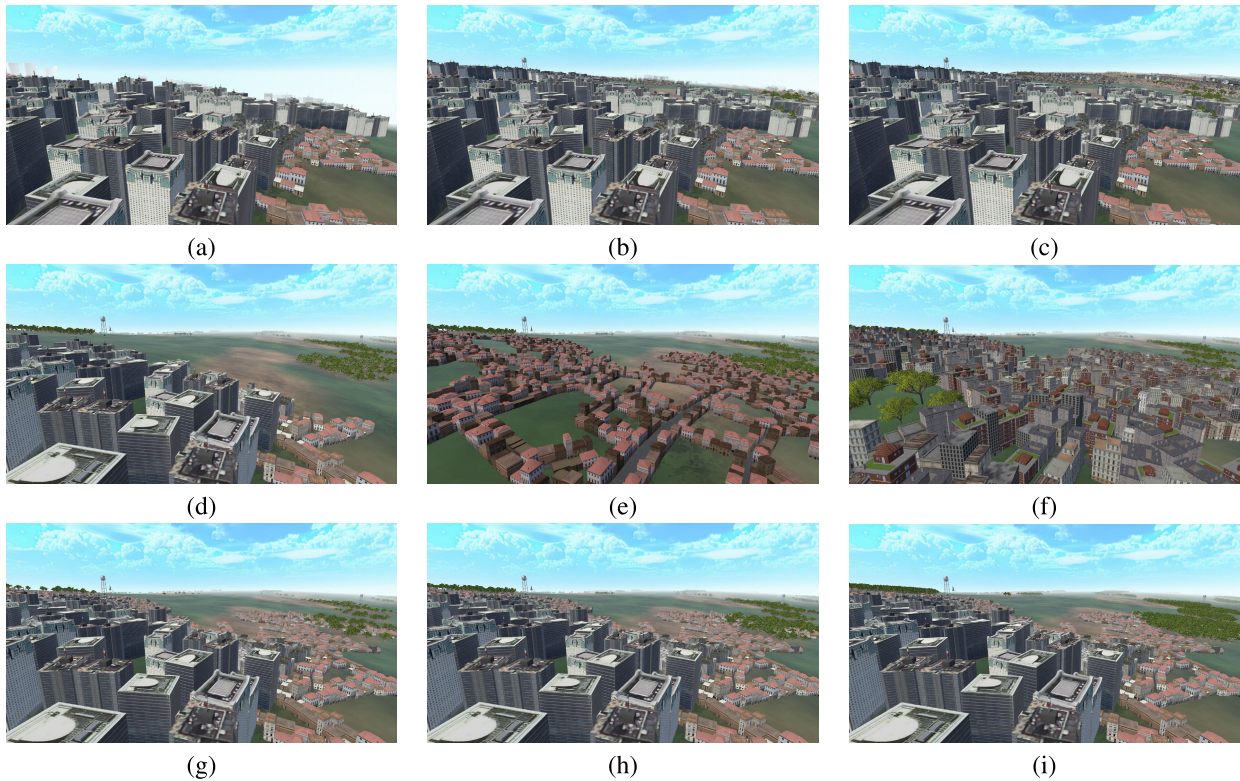


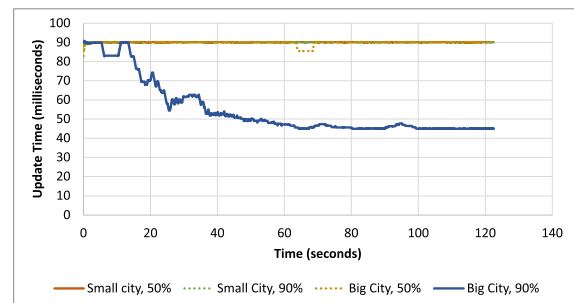
FIGURE 6. Effect of configuration parameters in visual quality corresponding to: (a) short view distance (1000 meters), (b) medium view distance (3000 meters) (c) long view distance (6600 meters), (d) smaller cities, (e) historic city style, (f) high-rise city style, (g) low tree density (50 meters average distance), (h) medium tree density (50 meters average), and (i) high tree density (20 meters average).

D. DISTANCE BETWEEN TREES EFFECT ON PERFORMANCE

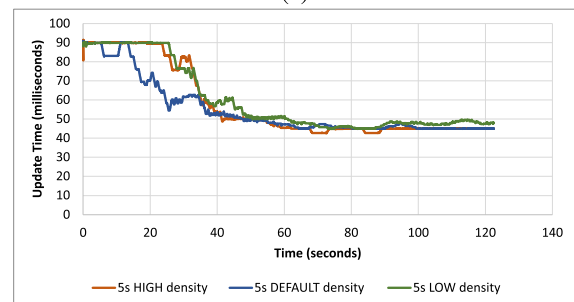
The results of Test 4, where the distance between trees is changed, are presented in Figure 8. Trees generate on a virtual grid that extends infinitely through the terrain. The nodes in this grid are selected to have a tree if they are in an area that should have trees. Increasing this number means that the nodes are more spaced out, and the trees will be further apart on average. It can be seen that 10 meter and 20 meter distances have heavy loads, due to the number of total trees on screen at the same time, while 50 meter and up, the performance remains within the target. With regard to the forest density, the density of trees directly affects the rendering times, but it has been observed that, at least for the use case of an open world seen from the sky, the density can be kept low by making the trees a bit larger than they would be in the real world without it breaking the immersion. Visually, Figure 6(g) shows that a distance too large leads to visible gaps between trees, and both Figure 6(h) and (i) show good coverage. Therefore, the range for the distance between trees will be between 30 and 80 meters.

E. MOVEMENT SPEED EFFECT ON PERFORMANCE

The results of Test 5, where the movement speed is changed, are presented in Figure 9. Higher movement speed will cause more of the terrain to be explored in the same amount of time, requiring more of the virtual world to be generated. It can be



(a)



(b)

FIGURE 7. Results of Test2 and Test3, influence on the performance of: (a) city distance and radius, for 50% and 90% view distances, and (b) city style.

seen that we did not observe any difference in performance caused by the choice of movement speed, even for values

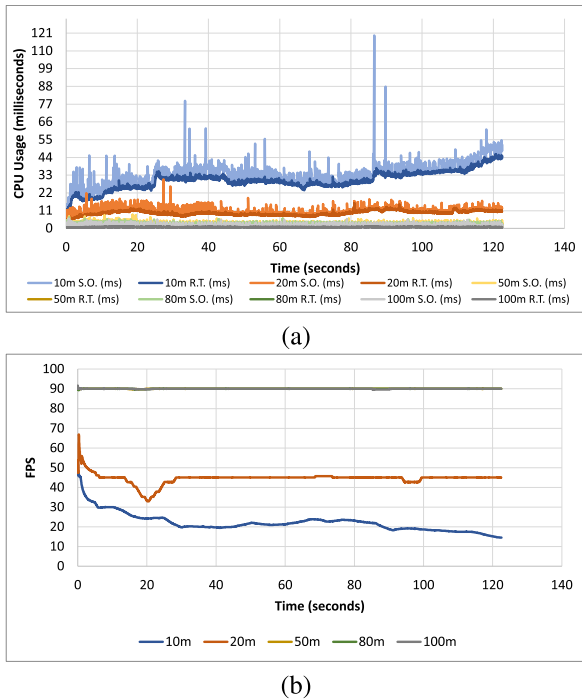


FIGURE 8. Results of Test4 (a) effect of distance between trees, on the cpu usage and (b) effect of the distance between trees on the FPS.

far beyond what would be reasonable in the context of a rehabilitation exercise. The settings for movement speed will have a range between 20m/s and 100m/s.

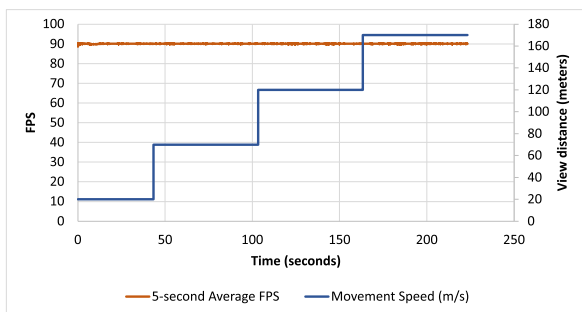


FIGURE 9. Results of Test5, effect of movement speed on performance.

F. REHABILITATORS IMPRESSIONS

The test with rehabilitators was very satisfactory. The three experts were able to prepare the exercises with no difficulties. They said that menu options were very clear and there were no doubts on how to proceed. It has to be said that before the creation of the exercise they accessed the help guide where a brief description of the creation process is provided. These first impressions were collected in personal interviews and a more rigorous evaluation needs to be done. In addition, it is also necessary to evaluate the patients’ impressions and acceptance of created scenarios since this has been considered out of the scope of this paper.

G. FINAL REMARKS

Performance experiments have been defined to set the range of values that can take the configuration parameters. These are collected in the configuration interface to make the exercise creation process easier (see Figure 3).

Many of the results show frame rates stabilizing around numbers like 90 fps, 45 fps, or 30 fps. These figures come from the v-sync, where the software waits for the next available display refresh to present the rendered image; when the update time is bigger than the 11.1 ms time between refreshes at the 90 hz display frequency, those updates consistently reach every other refresh, leading to the 45 fps number, and in cases where they fail to make the 22.2 ms interval, the actual frame rate would drop to 30 fps.

Our testing has focused on a single use case of an open-world environment, where due to the necessity of a wider view distance, more content needs to be generated in order to fill the terrain and give a sense of the world being endless until the horizon. The results indicate that we can safely use high view distances of up to 6 kilometers without sacrificing visual quality. This means there is room to increase the visual quality, while still having an expansive open world.

However, this is only true when using a PC as the host. If we use the Oculus Quest 2 headset in standalone mode, due to the limitations of mobile-class hardware, only 72 fps are possible. While keeping the recommended settings for PC, it can be seen in Figure 10 that it is possible to reach 3000 meter view distance, but if we increase the number of trees or the number of cities, the performance remains too low even at 1000 meters view distance. For this reason, the range of values that will be allowed in the configuration will be between 1000 and 3000 meters, instead of the 3000 to 6000 meters used for PC, and it is strongly discouraged to use options beyond the default, for the other settings that affect the performance.



FIGURE 10. Performance of the exercise running in the headset standalone mode.

In closed environments, such as interiors of buildings, caves, or mines, the performance could increase to be usable in the mobile headset.

Focusing on usability tests the impressions of the users have been very satisfactory. However, new tests with more users are required. In addition, we are planning new tests to collect the patients’ impressions.

V. CONCLUSION

In this paper, an authoring tool to allow non-technological skilled professionals to exploit the advantages of VR and procedural modeling strategies, in the context of rehabilitation scenarios, has been proposed. A detailed description of the system and the testing that has been carried out to ensure the variability of scenarios while preserving their graphical quality has been given. Testing has considered all the parameters that influence the generation of rehabilitation scenarios. The experiments have been carried out considering two of the state-of-the-art devices, the VIVE PRO EYE with 2 room position trackers, which relies on the PC to generate the graphics, and the Oculus Quest 2. As a result, from our testing, a set of configuration parameters has been proposed. These have been integrated into an easy-to-use interface that allows the preparation of rehabilitation exercises. Our future work will be centered on the testing of the system with more rehabilitators and also considering patients.

REFERENCES

- [1] WHO. *World Health Organization*. Accessed: May 2022. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/rehabilitation>
- [2] B. Noronha and D. Accoto, "Exoskeletal devices for hand assistance and rehabilitation: A comprehensive analysis of state-of-the-art technologies," *IEEE Trans. Med. Robot. Bionics*, vol. 3, no. 2, pp. 525–538, May 2021.
- [3] S. Anwer, A. Waris, S. O. Gilani, J. Iqbal, N. Shaikh, A. N. Pujari, and I. K. Niazi, "Rehabilitation of upper limb motor impairment in stroke: A narrative review on the prevalence, risk factors, and economic statistics of stroke and state of the art therapies," *Healthcare*, vol. 10, no. 2, p. 190, Jan. 2022.
- [4] A. de Sire and M. Invernizzi, "Musculoskeletal rehabilitation: State-of-the-art," *Appl. Sci.*, vol. 11, no. 14, p. 6243, Jul. 2021.
- [5] M. Fekete, V. Fazekas-Pongor, P. Balazs, S. Tarantini, A. N. Nemeth, and J. T. Varga, "Role of new digital technologies and telemedicine in pulmonary rehabilitation," *Wiener Klinische Wochenschrift*, vol. 133, nos. 21–22, pp. 1201–1207, Nov. 2021.
- [6] Y. Morimoto, T. Takahashi, R. Sawa, M. Saitoh, T. Morisawa, N. Kagiyaama, T. Kasai, B. Dinesen, M. Hollingdal, J. Refsgaard, and H. Daida, "Web portals for patients with chronic diseases: Scoping review of the functional features and theoretical frameworks of telerehabilitation platforms," *J. Med. Internet Res.*, vol. 24, no. 1, p. e27759, 2022, doi: 10.2196/27759.
- [7] I. Gaboury, M. Tousignant, H. Corriveau, M. Menear, G. L. Dorze, C. Rochefort, B. Vachon, A. Rochette, S. Gosselin, F. Michaud, J. Bollen, and S. Dean, "Effects of telerehabilitation on patient adherence to a rehabilitation plan: Protocol for a mixed methods trial," *JMIR Res. Protocols*, vol. 10, no. 10, Oct. 2021, Art. no. e32134.
- [8] M. Gustavsson, E. K. Kjörk, M. Erhardsson, and M. A. Murphy, "Virtual reality gaming in rehabilitation after stroke—user experiences and perceptions," *Disability Rehabil.*, vol. 44, no. 22, pp. 1–8, Dec. 2021.
- [9] A. Borrego, J. Latorre, M. Alcañiz, and R. Llorens, "Comparison of oculus rift and HTC Vive: Feasibility for virtual reality-based exploration, navigation, exergaming, and rehabilitation," *Games Health J.*, vol. 7, pp. 151–156, Jan. 2018.
- [10] S. H. Lee, H. Jung, S. J. Yun, B. Oh, and H. G. Seo, "Upper extremity rehabilitation using fully immersive virtual reality games with a head mount display: A feasibility study," *PM R*, vol. 12, no. 3, pp. 257–262, Mar. 2020.
- [11] L. Weber, D. Nilsen, G. Gillen, and, "Immersive virtual reality mirror therapy for upper limb recovery following stroke: A pilot study," *Amer. J. Phys. Med. Rehabil.*, vol. 98, no. 9, pp. 783–788, 2019.
- [12] D. A. Tran, M. Pajaro-Blazquez, J.-F. Daneault, J. G. Gallegos, J. Pons, F. Fregni, P. Bonato, and R. Zafonte, "Combining dopaminergic facilitation with robot-assisted upper limb therapy in stroke survivors: A focused review," *Amer. J. Phys. Med. Rehabil.*, vol. 95, no. 6, pp. 459–474, 2016.
- [13] K. Törnborn and A. Danielsson, "Experiences of treadmill walking with non-immersive virtual reality after stroke or acquired brain injury—A qualitative study," *PLoS ONE*, vol. 13, no. 12, Dec. 2018, Art. no. e0209214.
- [14] K. Crebbin, T. Grisbrook, C. Elliott, and A. Thornton, "The use of serious gaming to improve sensorimotor function and motivation in people with cerebral palsy: A systematic review," *Games Health J.*, Sep. 2022, doi: 10.1089/g4h.2022.0112.
- [15] S. Glegg and D. Levac, "Barriers, facilitators and interventions to support virtual reality implementation in rehabilitation: A scoping review," *PM R*, vol. 10, pp. 1237–1251, Sep. 2018.
- [16] C. Kushniruk and A. Nøhr, "Participatory design, user involvement and health it evaluation," *Stud. Health Technol. Informat.*, vol. 222, pp. 139–151, Sep. 2016.
- [17] C. M. Hayre, D. J. Müller, and M. J. Scherer, *Virtual Reality in Health and Rehabilitation*. Boca Raton, FL, USA: CRC Press, 2021.
- [18] M. C. Howard, "A meta-analysis and systematic literature review of virtual reality rehabilitation programs," *Comput. Hum. Behav.*, vol. 70, pp. 317–327, May 2017.
- [19] K. R. Lohse, C. G. E. Hilderman, K. L. Cheung, S. Tatla, and H. F. M. Van der Loos, "Virtual reality therapy for adults post-stroke: A systematic review and meta-analysis exploring virtual environments and commercial games in therapy," *PLoS ONE*, vol. 9, no. 3, Mar. 2014, Art. no. e93318.
- [20] R. M. Smelik, T. TuteneI, R. Bidarra, and B. Benes, "A survey on procedural modelling for virtual worlds," *Comput. Graph. Forum*, vol. 33, no. 6, pp. 31–50, Sep. 2014.
- [21] M. M. Danzl, N. M. Etter, R. D. Andreatta, and P. H. Kitzman, "Facilitating neurorehabilitation through principles of engagement," *J. Allied Health*, vol. 41, no. 1, pp. 35–41, 2012.
- [22] D. Dimovska, P. Jarnfelt, S. Selvig, and G. N. Yannakakis, "Towards procedural level generation for rehabilitation," in *Proc. Workshop Procedural Content Gener. Games (PCGames)*, Jun. 2010, pp. 1–4.
- [23] F. Kern, C. Winter, D. Gall, I. Kathner, P. Pauli, and M. E. Latoschik, "Immersive virtual reality and gamification within procedurally generated environments to increase motivation during gait rehabilitation," in *Proc. IEEE Conf. Virtual Reality 3D User Interfaces (VR)*, Mar. 2019, pp. 500–509.
- [24] S. Vert and D. Andone, "Virtual reality authoring tools for educators," in *Proc. ITM Web Conf.*, vol. 29, 2019, pp. 896–900.
- [25] H. Mäkinen, E. Haavisto, S. Havola, and J.-M. Koivisto, "User experiences of virtual reality technologies for healthcare in learning: An integrative review," *Behav. Inf. Technol.*, vol. 41, no. 1, pp. 1–17, Jan. 2022.
- [26] U. Radhakrishnan, K. Koumaditis, and F. Chinello, "A systematic review of immersive virtual reality for industrial skills training," *Behav. Inf. Technol.*, vol. 40, no. 12, pp. 1310–1339, Sep. 2021.
- [27] M. R. Paro, D. S. Hersh, and K. R. Bulsara, "History of virtual reality and augmented reality in neurosurgical training," *World Neurosurgery*, vol. 167, pp. 37–43, Nov. 2022.
- [28] L. Bao, S. V.-T. Tran, T. L. Nguyen, H. C. Pham, D. Lee, and C. Park, "Cross-platform virtual reality for real-time construction safety training using immersive web and industry foundation classes," *Autom. Construction*, vol. 143, Nov. 2022, Art. no. 104565.
- [29] Unity Technologies. *Unity Real-Time Development Platform*. Accessed: Nov. 29, 2022. [Online]. Available: <https://unity.com/>
- [30] Epic Games. *Unreal Engine*. Accessed: Nov. 29, 2022. [Online]. Available: <https://www.unrealengine.com/en-US>
- [31] Crytek GmbH. *CRYENGINE*. Accessed: Nov. 29, 2022. [Online]. Available: <https://www.cryengine.com/>
- [32] F. Cassola, M. Pinto, D. Mendes, L. Morgado, A. Coelho, and H. Paredes, "A novel tool for immersive authoring of experiential learning in virtual reality," in *Proc. IEEE Conf. Virtual Reality 3D User Interfaces Abstr. Workshops (VRW)*, Mar. 2021, pp. 44–49.
- [33] O. L. Chávez, L.-F. Rodríguez, G. S. Lugo, and L. A. Castro, "Authoring tools and virtual environments in intelligent tutoring systems: Challenges and opportunities," *Res. Comput. Sci.*, vol. 146, no. 1, pp. 9–16, Dec. 2017.
- [34] L. Zhang and S. Oney, "FlowMatic: An immersive authoring tool for creating interactive scenes in virtual reality," in *Proc. 33rd Annu. ACM Symp. User Interface Softw. Technol.*, Oct. 2020, pp. 35–342.
- [35] H. Coelho, P. Monteiro, G. Gonçalves, M. Melo, and M. Bessa, "Authoring tools for virtual reality experiences: A systematic review," *Multimedia Tools Appl.*, vol. 81, no. 19, pp. 28037–28060, Aug. 2022.
- [36] J.-S. Kim, H. Kavak, and A. Crooks, "Procedural city generation beyond game development," *SIGSPATIAL Special*, vol. 10, no. 2, pp. 34–41, Nov. 2018.
- [37] The Khronos Group. *OpenXR Overview*. Accessed: Nov. 29, 2022. [Online]. Available: <https://www.khronos.org/openxr/>
- [38] Unity Technologies. *Unity Manual: XR*. Accessed: Nov. 29, 2022. [Online]. Available: <https://docs.unity3d.com/Manual/XR.html>

- [39] Unity Technologies. *Unity Manual: Input Manager*. Accessed: Nov. 29, 2022. [Online]. Available: <https://docs.unity3d.com/Manual/class-InputManager.html>
- [40] Unity Technologies. *Input System*. Accessed: Nov. 29, 2022. [Online]. Available: <https://docs.unity3d.com/Packages/com.unity.inputsystem/1.4/manual/index.html>
- [41] Unity Technologies. *XR Interaction Toolkit*. Accessed: Nov. 29, 2022. [Online]. Available: <https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit/2.2/manual/index.html>
- [42] K. Perlin, "An image synthesizer," *SIGGRAPH Comput. Graph.*, vol. 19, no. 3, pp. 287–296, 1985.
- [43] S. Bhattacharya, E. Lin, G. Sajith, L. Munroe, K. Pushparajah, J. Schnabel, J. Simpson, A. Gomez, A. De Vecchi, S. Deng, and G. Wheeler, "Immersive visualisation of intracardiac blood flow in virtual reality on a patient with HLHS," *Eur. Heart J.-Cardiovascular Imag.*, vol. 22, no. 1, pp. 6759–6765, Feb. 2021.



DAVID QUINTANA received the M.Sc. degree in computer science from the Universitat de Girona, in 2015, where he is currently pursuing the Ph.D. degree in computer science. He is also a Researcher with the Graphics and Imaging Laboratory. His research interests include authoring tools, procedural modeling, virtual reality, and serious games for health.



ANTONIO RODRÍGUEZ received the M.Sc. and Ph.D. degrees in computer science from the Universitat de Girona, in 2016 and 2020, respectively. He is currently an Associate Professor with the Department of Computer Science and Applied Mathematics, Universitat de Girona. He is also a Researcher with the Graphics and Imaging Laboratory. His research interests include serious games, e-learning, and medical applications, among others.



IMMA BODA received the M.Sc. degree in computer science from the Universitat Autònoma de Barcelona, in 1992, and the Ph.D. degree in computer science from the Universitat Politècnica de Catalunya, in 2001. She is currently a Professor with the Department of Computer Science and Applied Mathematics, Universitat de Girona, and also a Researcher with the Graphics and Imaging Laboratory. Her research interests include research and development on visualization, image processing, medical applications, serious games, and e-learning, among others.

...