www.cadanda.com Computer-Aided Design AND APPLICATIONS

Aiming for Inclusion: Virtual Reality Archery for Athletes with Disabilities

Luca Andriolo[1] , Allegra Bianucci[1] , Alberto Pietrobon[1] , Sata Arlati[2] and Mario Covarrubias[1]

[1]Politecnico di Milano, Italy,
[2]STIIMA, CNR, Italy,

Corresponding author: Mario Covarrubias Rodriguez, mario.covarrubias@polimi.it

Abstract. Virtual Reality (VR) has emerged as a transformative tool for enhancing training and skill acquisition across various domains, including sports. This study introduces the development and evaluation of a VR archery simulator specifically designed to teach fundamental archery techniques within a realistic and structured training environment. Unlike existing VR archery applications focused primarily on entertainment, this project prioritizes skill development through a three-phase instructional model: a guided tutorial, targeted training exercises, and a semi-realistic simulation module.

Developed in Unity and deployed on the Meta Quest 3 headset, the simulator features a physics-based bow and arrow system, immersive environmental elements, and dynamic variables such as moving targets and wind effects to replicate real-world conditions. Notably, the system was co-developed by students with disabilities aged 18 to 25, underscoring a strong commitment to inclusivity and accessibility in VR-based sports training.

Usability was assessed with 18 participants using the System Usability Scale (SUS), resulting in an average score of 86, indicative of high usability and instructional effectiveness. An additional evaluation was conducted with 14 students from the ASPOC association, whose feedback informed further accessibility improvements.

The findings demonstrate that VR-based archery training can effectively support skill acquisition in a safe, engaging, and accessible environment. The system offers promising applications not only for beginners but also for athletes seeking off-season or rehabilitative training options. Future work will focus on enhancing haptic feedback, improving visual fidelity, and implementing adaptive difficulty features to further tailor the experience to individual users.

Keywords: hand exoskeleton, rehabilitation, CAD modeling, rheumatoid arthritis
DOI: https://doi.org/10.14733/cadaps.2026.750-763

1 INTRODUCTION

Virtual Reality (VR) and Augmented Reality (AR) are immersive technologies that leverage high-resolution displays and motion-tracking sensors to create dynamic, interactive experiences. VR fully immerses users in a computer-generated environment, while AR superimposes digital elements onto the physical world through semi-transparent displays. These experiences are typically delivered via head-mounted displays (HMDs), which enable users to engage with visual and auditory stimuli and in more advanced systems haptic and olfactory feedback, enhancing the sense of presence and realism.

The adoption of VR and AR spans a wide range of domains, including entertainment, education [12, 4, 1], rehabilitation [10, 18], and increasingly, sports [15, 8]. In the context of physical rehabilitation, VR and AR technologies offer controlled and engaging environments where users can perform exercises in a motivating and safe manner [6]. This has proven especially useful for maintaining patient engagement during repetitive training routines. Beyond rehabilitation, these immersive technologies are gaining traction in sports training as tools to support the acquisition of technical skills [13]. VR, in particular, has shown promise in disciplines such as fencing, where users can train specific movements, develop reaction times, and improve focus through targeted virtual scenarios.

Archery represents a sport that aligns naturally with virtual simulation due to its limited spatial requirements and emphasis on precision and control. The core techniques-nocking an arrow, drawing the bowstring, aiming, and releasing can be effectively replicated within a virtual setting without the need for expansive physical environments. Despite the availability of several archery-themed VR applications, many are designed primarily for entertainment and gamification purposes. While these arcade-style applications help increase awareness and interest in archery, they often lack the pedagogical structure needed for formal skill development.

In this context, the present project introduces a VR archery training application specifically designed to teach fundamental techniques in a calm and focused virtual environment. The system adopts a structured three-phase approach consisting of: (1) a Tutorial phase to introduce basic posture and mechanics; (2) a Training phase featuring targeted exercises to enhance shooting accuracy and control; and (3) a Simulation phase that incorporates semi-realistic challenges to test skill application under variable conditions. The application also aims to serve users who may benefit from training in non-traditional settings, such as beginners, athletes during recovery, or individuals with limited access to physical archery ranges.

The following sections detail the development process, technological framework, and evaluation methodology used to implement and assess the VR archery simulator.

2 State of Art

Related to VR archery experience, a lot of simulators have been developed in the last few years. However, the majority of them are thought as entertainment without the concrete goal to be used to specifically train the sport in consideration. Some examples that can be found are "Archery Pro" by SavySoda [16], "Arrowhead-Medieval Archery VR" by kilogrames [9], or "Backyard Virtual Archery" by MISSING GAME STUDIO [17], which by the way intend archery not as a sport to learn and progress into, but mainly as a game or arcade competition. A positive aspect of the diffusion of such games is the increasing interest in the development of items and tools that can make the virtual experience more realistic: an example is the "Artemis VR Game Bow" by Wonder Fitter [7], a physical bow equipped with compartments for VR controllers and a piston that simulates the shooting of arrows. This type of equipment could also be applied to more technical applications, like the one presented in this work, to enhance haptic sensations of the users and create a realistic experience. Concerning the state of art about archery in virtual reality, few articles and journals were found during the research, even though this topic is taking more importance as demonstrated with the previous examples. Silviu Butnariu [11] in his article developed a prototype of bow, composed by a simple frame, electrical motor, sensors and a system of pulleys, interconnected with the virtual reality through a channel, in order to strengthen the haptic interface during shooting. In the experiment the Oculus Rift DK2 was used and a simple medieval landscape was bulit in Unity. Depending on the drawing of the physical bow (rotation angle of the pulley due to the bow string) and on the direction of headset, the arrow was shot in the virtual world at different distances. Additionally, the prototype gave the possibility to change its weight and the length of the draw in order to simulate different categories of bow. A more insightful idea was presented by Deniz Bedir [5], who decided to compare the use of Virtual Reality Based Imagery (VRBI) with the Visual Motor Behavior Rehearsal and Video Modeling (VMBR + VM), which is the common technique adopted by elite athletes to gain concentration and remember the most important movements before a competition (also with the help of a 2D video). The pool of subjects that were tested included curling, bowling and also archery athletes. The result of the study showed that the use of VRBI, so displaying through a headset all the steps of the competition made rigorously by a professional athlete, helped the subjects to perform better in the test with higher score respect to the traditional imagery. Moreover, the speed of progress was faster with the VR set. Another example of how the virtual reality is adopted as a tool in the archery field can be found in the article of Nihal Dal [19]. In this study a comparison between VR archery and the real one was conducted to analyze the advantages and shortcomings of the first one. "Archery Kings VR" gaming application (Appnori Inc., Busan, Republic of Korea, 2018) was used. The test was based on retrieve biological parameters of the subjects like HR, HRV, breathing rate or rate of perceived exertion (RPE) during both the activities to understand if the level of stress, perceived difficulty, sympathetic responses and psychological aspects remained the same. The conclusions revealed that subjects suffered more the real archery in terms of perceived difficulty and exertion (higher HR and RPE) while the level of anxiety resulted to be the same between virtual and real practice. Furthermore, participants performed better in the simulation, scoring more points than with physical equipment. This results converged to the idea that the use of virtual reality is not able to replicate perfectly the effort required during a real competition because the tools and the perception are completely different, anyways it can be a strong help in other aspects of the archery, as learning basic techniques and movements. As it is possible to notice from the various topics of the articles just reported, a VR application completely dedicated to teach the fundamentals of archery sport in a simple but refined way is difficult to find. For this reason in the following paragraphs all the steps taken to develop a virtual reality app able to teach the basics of archery are presented.

3 Methods

The VR application was built using the software Unity (Version 2022.3.44f1) in collaboration with Visual Studio 2022 to write all the necessary scripts in C#. For the creation of the items present in the game several packages were downloaded from the Unity Asset Store, as reported in the following sections. The Meta Quest 3 was used as head-mounted display to try and develop the application. Concerning the testing of the VR app, a pool of 18 subjects was tested and the usability of the system was retrieved by the SUS protocol (System Usability Scale) ideated by John Brooke in 1986 [2]. The questionnaire was sent as a simple Google Form and all the responses were easily collected in a linked Excel file to better work on the final results.

4 Development

For the development of the application, the different components have been created separately and then integrated together. The main components are the bow and the arrow, the targets, and the user interface elements; their development will be described in the following paragraphs.

4.1 Bow

The development of the bow started with the download of the model from the internet, that must be without the string; in fact, to properly represent its bending behavior when pulled, the string is created by means of a line render each time a scene is uploaded, thanks to the "Bow_string" script. In order to guarantee the right interactions of the bow with the other elements of the game, different components have been assigned to it.

The first is the "Rigidbody" component, thanks to which the bow can behave in a realistic way, act under imposed forces and interact with other objects. To facilitate the handling from the users, the bow is set to not feel gravity. Another component is a "Capsule Collider" placed at the level of the handle of the bow, that permits to detect collision with other component of the game, mainly arrows, and to trigger events thanks to the "Is Trigger" feature selected (Figure 1). A fundamental characteristic for the game development is the possibility to grab the object with the controllers, thanks to the "XR Grab Interactable" component. The bow can be grabbed by the user by pressing the grip button of the controller with the middle finger. The pulling of the string is enabled by using an "XR Grab Interactable" cube without visibility, then called "Midpoint grab", positioned at the middle of the string (Figure 1). Also in this case it can be grabbed by the user by pressing the grip button. Another cube, called "Midpoint visual", is placed in the same position as the previous one, and it is used for the visualization of the string when pulled (Figure 1). However, this gameobject is constrained to move with the string, only in the symmetry plane of the bow and until a certain distance from the bow. The choice of constraining movements only in the sagittal plane of the bow has been made to facilitate the user in the shooting process. As a consequence, if the user moves the controller outside the feasible positions its representation will be distant from the string, leading to a non completely realistic behavior. The visualization of the string is managed by a dedicated script called "Bow_string_controller"; when the "Midpoint grab" is triggered, the position of the cube is continuously checked and the string is bent consequently. On the release of the cube, the string returns in its original unloaded shape. In the same script, a force factor that varies linearly between 0 and 1 is calculated. The value 0 corresponds to the unloaded condition, while 1 to the string completely pulled; the intermediate values correspond to intermediate positions. This factor will be used in the calculation of the force to apply to the arrow enabling its flight. To permit the right positioning of the arrow, a "XR Socket Interactor" on the "Midpoint visual" has been used, this component will be explained better in the following paragraph.

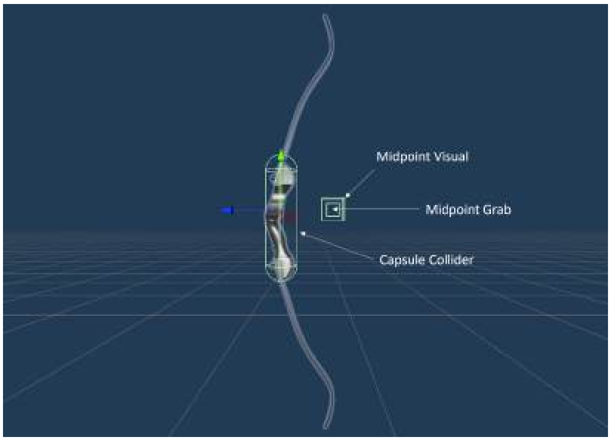

Figure 1: *Bow components.*

4.2 Arrow

The procedures followed for the development of the arrow are strictly related to the different phases of the shooting process; in fact two slightly different types of prefab were created: one used to manage the loading of the arrow on the bow ("Arrow_start"), the other to handle its release and flight ("Arrow_fly"). Concerning the loading of the arrow, the prefab appears on a gameobject, called "Support", as soon as the scene is loaded. Like the bow, the arrow has the "Rigidbody" component without being subjected to gravity. The "XR Grab

Interactable" component is required for grabbing the object and placing it on the bow; the arrow can be grabbed by the user by pressing the grip button of the controller (like for the bow). A "Capsule Collider", representing the arrow shaft, is used to detect the interaction with other objects. In order to place the arrow on the bow in the right position, the "Socket Interactor" present on the so called "Midpoint visual" is used. When the arrow comes into contact with the "Midpoint visual", it is automatically placed in a predefined position, and when the grip button is released the arrow starts moving with the bow. After the pulling of the string, when the "Midpoint grab" is released, an Event called "OnBowReleased" handles different actions. The first is the disappearance of the arrow loaded on the bow and the creation of a new arrow ("Arrow_start") on the "Support", managed by the "Arrow_destroy" script. At the same time, thanks to the "ReleaseArrow" function in the "Arrow_controller" script, the "Arrow_fly" prefab is created on the bow (in the same point where was placed the destroyed one) and a force impulse in the forward direction of the arrow is added in order to make it fly. The amount of force given to the arrow is calculated by multiplying the force factor with a maximum speed value, previously selected by trial and error. Since two different types of arrow are used, in the code controlling the releasing of the arrow, some conditions have been created in order to make possible the release of the arrow only if the "Socket Interactor" has been previously triggered by an arrow. When the arrow is released the condition is reset and no arrows can be created until another one is loaded on the bow. In order to prevent the interaction of the released arrow with the bow, the collision detection between these two objects has been disabled in the "layer collision matrix" by assigning the appropriate layers to the objects. This procedure has been done also to prevent the interaction between the arrow just released and the other already stuck into the target. In order to make the flight of the arrow more real, a rotation that follows the direction of the velocity vector is imposed to the arrow while flying. In this way, the arrow flies following a parabolic trajectory; this process is possible thanks to the "Arrow_rotation" script assigned only to the "Arrow_fly" prefab. To better clarify, the main difference between the types of arrow prefabs is that the first type isn't subjected to gravity and it is kinematic, meaning that it isn't affected by forces; in contrast, the second type feels the gravity and is affected by forces, both fundamental for the flight of the arrow. The last part of the development involved the sticking process of the arrow into the target. When the arrow collides with an object with a collider, in the "Arrow_sticking" script, assigned only to the second arrow prefab, the arrow is turned to kinematic and it is also set as a child of the target. This last step is fundamental when moving target are used, permitting to the arrow to follow the target movements. The collider of the arrow doesn't cover the entire length of the arrow shaft, but it is a little bit shorter; in this way it was possible to recreate the effect of penetration of the arrow in the target. Eventually, a third type of arrow prefab ("Arrow_fly_wind") was used in the development of the game. It was recalled inside a new script "Arrow_controller_wind", that was added instead of the previous "Arrow_controller" to the bow, in order to manage the interaction with the wind present in the 'simulation' part of the application, which will be analyzed more in detail in Paragraph 4.6.4.

4.3 Targets Mechanism

In the game two types of target have been chosen. One is a classical outdoor round bale target with three wooden legs, called "Target ground" (Figure 2). It is divided into ten different rings, colored two by two in white, black, blue, red, and yellow. The other one, called "Target fly", has the same traditional target face, but is unconventionally suspended, and it is able to move horizontally or vertically in the space, at different speeds. The motion of this last target type has been made possible thanks to two specific scripts, one responsible for the horizontal movement, called "MovingTarget", while the other one was responsible for the vertical movement, called "MovingVertical". Thanks to public variables the range of motion and the speed can be adjusted in the development phase. These targets are 3D models downloaded from Turbosquid and Free3D, specific websites that allow to download for free their models in the correct .obj file format and together with their .mtl files. Once imported on Unity, the .obj object is the raw 3D model, while the .mtl file is needed because, thanks to a mesh renderer, it makes possible to apply the correct color and appearance to the model. To attribute the target the capability to detect collisions, and so be able to keep track of the score, they have been equipped with ten different colliders, one for each score ring. These colliders have been created as separate 3D models on Autodesk Inventor. Nine of the models are hollow rings, while the central one is simply circular. Then they were placed concentrically inside the width of the targets, as children of the target game object, in order not to ruin their appearance. To transform these 3D models into 3D colliders, on each one a mesh collider has been applied. These colliders were then tagged with different names, from Target1 to Target10, since their size and position is strictly related to the score that each one represents. Indeed, in the Unity environment, a tag is used to identify a game object. This tag is at the base of the script called "Single_target", that assigns the partial score and updates the total score at each throw, depending on the hit collider. In order to make the experience more realistic, also the sound effects are fundamental. That's why every target has an audio source component linked with the script related to the score. Inside this script, specific code lines are used to find the audio associated to the target and reproduce it whenever it is hit.

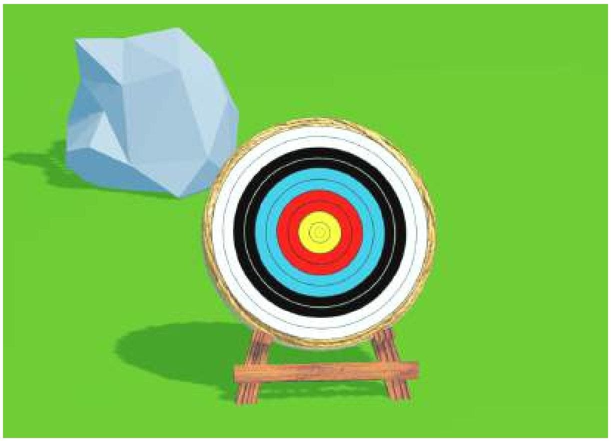

Figure 2: *Target example.*

4.4 UI Game Elements

The user interface of this game is simple, yet effective. In fact, the user can interact with it only by clicking buttons, which are specific UI elements that can be clicked in order to trigger an event. In this case buttons are used to change level, to go through the tutorial steps and the level explanations. Differently from other buttons, the level-change ones required a specific script to manage them. In fact, all the levels have been made navigable thanks to 'LoadScene', a static method of the scene manager, which is used inside the button scripts to load the desired scene by its name or, as in this case, by its build settings index. Another fundamental component needed to be able to interact with the buttons was the tracked device graphic raycaster, added directly to the canvas, otherwise the click inputs coming from the Meta Quest 3 controllers weren't detected. This component permitted to visualize a ray exiting from the controllers, allowing the user to understand immediately what he was pointing at. To place the buttons, a canvas was necessary, since it serves as the rendering space for all the UI elements, which is why all of them have been placed as child. The render mode of the canvas was world space, the most appropriate for the VR purpose. A fundamental UI element not mentioned yet are the texts, needed for the tutorial and levels description, but also placed as children of every buttons to describe their function. To ensure high quality text, the TextMeshPro element was the chosen one. For some applications, like the tutorial and the level explanation, both text and buttons were organized in a panel, which is a container whose purpose is to group UI elements together. After having built the basic structure, and having put all the elements together in the scenes, the work done was to obtain a good-looking result, changing the size, the disposition, the colors and the appearance of the UI elements. To do that, an appropriate font was chosen, as well as different colors for the selected and unselected condition of the buttons. Even their shape has been improved, thanks to a specific package downloaded from the Unity Asset Store. About the disposition of the UI elements in the scenes, all of them have been placed such in a way that the user could reach and interact with them without using the controllers to move in the VR environment. Moreover, the level-change buttons were placed always in the same position, so that it was more intuitive for the user to find them in the game.

4.5 Audio

Inside every scene there are different audio to make the game more engaging and realistic. As already said, every target is coupled with its hitting sound effect. Similarly to this concept, also other action-related audios are present in the game and related to a script, like the pulling string action and the arrow release. However, other sound effects have been used, which are not related to specific actions but left as background. These sounds have been applied to the scenes using the audio source component and they start as soon as the scene is loaded, thanks to the 'play on awake' option. Examples of sound effects used are the birds chirping, the crackling fire, the flowing water, the wind, the flag flapping and the background music of the menu scene. These audio have been downloaded from specific Unity Asset Store sound packages, or from the internet as mp3 files. No other modifications were necessary, apart from some audio trimming, to ensure that the starting timing of the audio matched the actual sound within it.

4.6 Scenes

4.6.1 Menu scene

The first scene that appears as soon as the game is opened is the menu scene. As its name says, its function is to let the user move into the different game levels. Three different buttons are present in front of the player at the beginning of the game: the 'tutorial', the 'training', and the 'simulation' one. If pressed, the scene changes depending on the choice made. A quiet and enjoyable landscape was chosen for the menu scene, composed by a tent, a campfire, and a river surrounded by trees and a mountain scenery (Figure 3). The same was used for the 'tutorial' scene, even though the location of the player is different. The landscape was downloaded from the Unity Asset Store and its characteristic is to be low poly, which means it has a polygonal mesh with a relatively small number of polygons, leading to better loading performances while still having acceptable quality. Moreover, background music was added in the menu scene to better involve and amuse the user in the first part of the VR experience.

4.6.2 Tutorial scene

The 'tutorial' scene was created as first step to follow to learn the basic technique of archery. When the user is brought into this part, the bow and an arrow are placed on two supports (trunk and rocks) near to his left and right hand respectively. The arrows are continuously generated by the spawner gameobject that was mentioned in the previous sections and that was positioned inside the support to give an effective visualization. In front of the player, instead, a fixed target is present as aim to shoot at. In addition a canvas showing all the steps for a correct posture and loading of the bow is displayed beside the target: the archer has to rotate the whole body perpendicularly to the target with only the head looking at it; the arrow has to be nocked on the bow, which must be brought at the level of the shoulders with the carrying arm straight; then, the bow string must be drawn till touching the end part of the arrow with the lips; during the drawing, the pulling arm has

Figure 3: *Menu scene.*

to be parallel to the arrow and at the same height of the shoulders; eventually, the release must be performed aiming just bit below the desired point. On the other side of the target, an avatar provided with bow and arrow was added to the scene in order to give to the user a visual reference of all the steps necessary to take. The prefab of the avatar was downloaded from the Unity Asset Store and an animation found on Mixamo was applied. Furthermore a script to control the position of the bow on the avatar and the setting on/off of the arrow visibility (to simulate the shooting) was created. As mentioned in the previous paragraph, the same landscape of the menu scene was used as background for all the objects just presented (Figure 4). The idea behind the 'tutorial' scene was to let the user face the most important fundamentals of shooting with a bow in an assisted environment, in order to prepare him for the 'training' levels, in which more advanced skills are required and taught, as explained in the following sections.

Figure 4: *Tutorial scene.*

4.6.3 Training scene

In the 'training' scenes the user is moved into a different space with respect to the menu and the 'tutorial'. The landscape was built manually piece-by-piece using prefabs of several 'low poly' packages found on the

Unity Asset Store. Multiple types of terrains and mountains were used as main part of the environment, while different trees, rocks, plants and other items were added to make the scene pleasant and peaceful. Moreover, a sky dome was applied with the possibility of changing its color and the one of the clouds depending on the preferred conditions (Figure 5). The disposition of the bow and the arrow spawner were built similarly to the 'tutorial' scene, in order to ease the grabbing for the user. Background sounds, like birds chirping, were also included in the scene to make it more realistic. The 'training' part of the application was divided in three levels of increasing difficulty. To develop them, several archery exercises were taken as input from the manual "Linee Guida - Livello Avanzato per l'istruttore di tiro con l'arco" of Filippo Clini [3], in which most part of the essential theory is explained properly. The first level is based on shooting at three different fixed targets from a short distance (Figure 5). The simplicity of the task was chosen to teach the user to grow confidence understanding the trajectories of the arrows and measuring the force to apply to the bow string. In other words, consistency at aiming for the best score is the real objective of this level. Indeed, a canvas displaying the partial (points of the last shot) and total (sum of the overall points) score was also placed above the targets to show the player his progresses. This was kept even for the other exercises and levels. The second level instead shows the same disposition of the targets of the previous level but with an additional difficulty for the user: the removal of the visibility during the drawing of the bow string. The idea came from one of the exercises cited in the manual of Filippo Clini [3], in which it is suggested to train archers with blind shots to improve kinaesthesia, and in particular the sensitivity to small displacements and loss of shooting line during the aiming phase. So, to simulate the closing of the eyes, a canvas with a black image as child was wrapped around the visibility of the main camera and the script "blindness" was applied to the "Midpoint grab" in order to switch on the canvas on the grabbing of the bow string and its switch off on the release through the called functions "OnGrab" and "OnRelease". The choice to make the user blind at the moment of drawing was taken in order to ease the process of nocking the arrow, that in the case of this VR application is difficult to perform without a visual reference. However there were many other possibilities for this type of exercise, for example to keep the eyes closed during the entire gesture, just before grabbing the bow string or even before nocking the arrow, that could be all included in a future development.

Figure 5: *Training level 1-2 scene.*

Eventually, the third level is composed by four moving targets. Two of them move vertically on parallel lines while the remaining others horizontally, with the possibility to change their speed and range of motion in phase of development (Figure 6). As the previous levels, a real exercise was used as inspiration and adapted to the application. In this case the directions on which the moving targets slide were chosen to represent the training based on shooting along vertical and horizontal lines. This type of practice is adopted to improve the comprehension of mistakes in different directions. As further step, not simple lines were chosen, but moving targets in order to increase the difficulty and make the exercise more competitive for the player. All three levels were considered independent of each other and without any restrictions in terms of reaching specific results or scores to unlock new exercises. This was chosen to let all users experience several aspects of archery training challenging themselves to perform better every time, but also for the application's testing purposes: to give the possibility to all the tested subjects to try the entire game within all its parts.

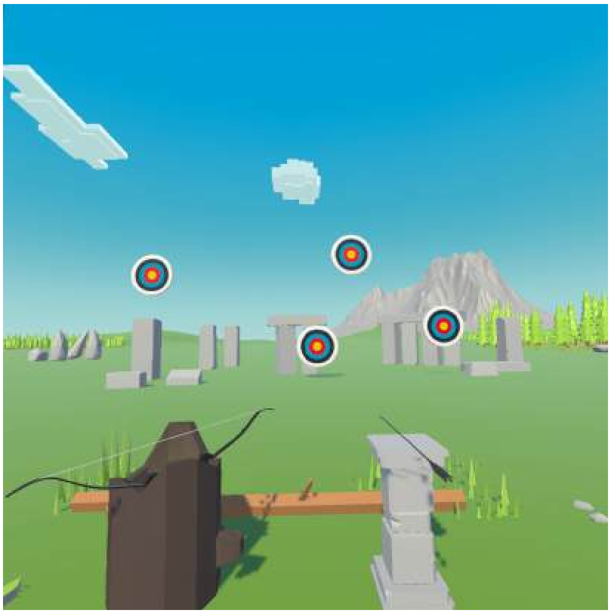

Figure 6: *Training level 3 scene.*

4.6.4 Simulation scene

In the 'simulation' scene the main goal was to let the user challenge himself in a test that resembled real shooting in a more complete way. To do so, the distance between the user and the single target present in the scene is increased respect to the 'training' and the effect of the wind is introduced in order to influence the behavior of the arrow during the flight. A directional wind zone of Unity was used to create the air moving with the addition of some particle systems to visualize it. To make it more realistic, the direction and the intensity of the wind were randomized through a script that each twenty seconds changed casually the force ("WindMain") and the direction ("Transform") of the zone within defined ranges. Moreover, it was necessary to replace the script "Arrow_controller" with a further one ("Arrow_controller_wind") to apply to the bow in order to make the flying arrows react to the wind, because only particle systems are affected by it in Unity. This script simply added in the function "ReleaseArrow" the detection of the intensity and direction of the wind at the shooting instant, and then applied a force with the same magnitude and direction to the component "Rigidbody" of the arrows just instantiated. Eventually, to let the user visualize easily the behavior of the wind, a flag was developed and put as child of the wind zone to follow the same rotation. The flag was built by simply applying a cube object with changed dimensions (cloth) to a slender cylinder (pole). The component "Cloth" was also added to the flag in order to animate it as moved by the air. Concerning the landscape, different items of the same packages used for the 'training' scenes were chosen: winter-themed terrains, trees and rocks were included in the scenery to trigger sense of quiet and concentration on the actions to perform and conditions to consider (Figure 7). Moreover, the particle system resembles the snow moved by the wind, giving stronger realism to the landscape. Also in this scene bow and arrows were placed as the cases cited before, and the partial and total score were displayed just above the target. The 'simulation' is thought as the last step to take in the experience of the archery application in order to apply all the former learning to a semi-realistic situation.

Figure 7: *Simulation scene.*

5 TESTING

To assess both the strengths and limitations of the developed VR archery application, a user study was conducted involving a sample of 18 participants. All participants were novices in archery, with no prior experience in the sport at a professional level. Their ages ranged from 18 to 56 years. Each subject engaged with all components of the application tutorial, training, and simulation and was allowed to ask for support or clarification from the development team throughout the experience. Following the session, participants completed a questionnaire designed to evaluate usability.

The questionnaire was based on the System Usability Scale (SUS), a standardized and widely adopted tool introduced by John Brooke in 1986 for assessing the usability of systems in a quick and cost-effective manner [2, 14]. The SUS consists of ten statements, with responses measured on a five-point Likert scale ranging from 1 (strongly disagree) to 5 (strongly agree). Scoring is calculated by subtracting 1 from the response for odd-numbered items (1, 3, 5, 7, 9) and subtracting the response from 5 for even-numbered items (2, 4, 6, 8, 10). The adjusted scores are then summed and multiplied by 2.5 to yield a final score out of 100, where higher values represent greater usability (Figure 8).

The average SUS score for the VR archery application was 86, a result well above the generally accepted usability threshold of 70. This indicates a high level of usability and aligns with the core objective of the project: to develop an intuitive and accessible tool for introducing users to archery fundamentals in a supportive virtual environment.

Notably, the most divergent responses were associated with statement 1 ("I think that I would like to use this system frequently") and statement 4 ("I think that I would need the support of a technical person to be able to use this system"). These variations suggest that while some users may have lacked intrinsic interest in the sport of archery, others found the initial learning curve or interaction with the VR controls somewhat challenging. These insights highlight the need to improve onboarding support and to enhance motivational elements that could foster user engagement and curiosity as can be seen from Figure 9.

It is important to acknowledge the limitations of the study, particularly the relatively small sample size, which may affect the generalizability of the results. Future evaluations involving a broader user based including trained archers could offer valuable additional perspectives. As noted earlier in the report, one intended use of the application is to support athletes who require continued training during periods of physical limitation or rehabilitation. Whether the current simulation complexity is sufficient to meet the training needs of more advanced users remains an open question for future research.

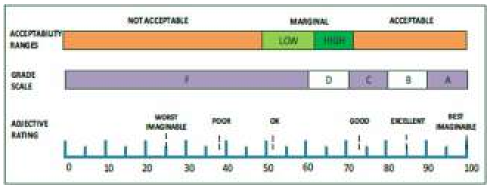

Figure 8: SUS *score scale evaluation.*

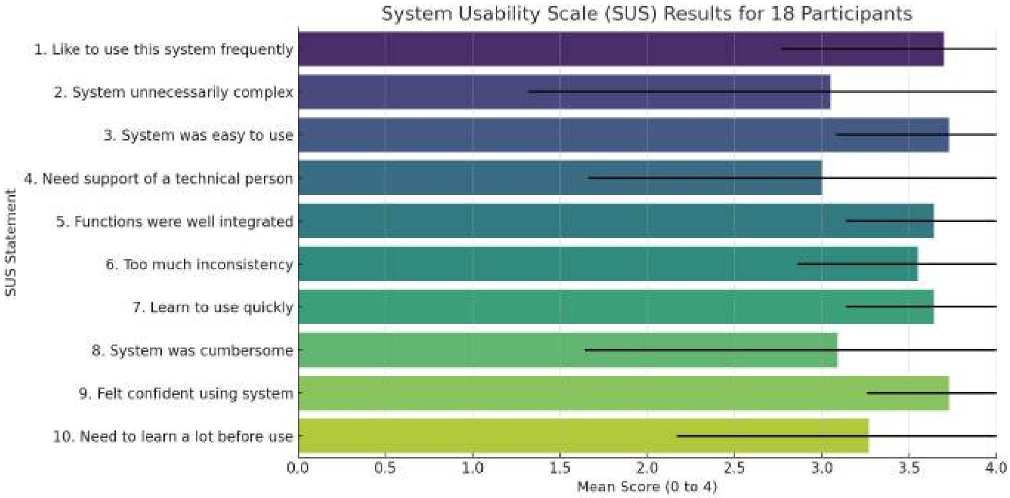

Figure 9: System Usability Scale (SUS) results for 18 participants. The bars show the average normalized score (0–4) for each SUS statement, with error bars indicating standard deviation.

The SUS protocol was also administered to a group of 14 students with disabilities from the ASPOC association (Figure 10), in order to evaluate the accessibility and inclusiveness of the VR archery application.

These participants, who had varying levels of familiarity with virtual reality technologies, completed the same usability testing procedure as the initial group. Their feedback provided valuable insights into how the application performs when used by individuals with specific cognitive or physical challenges. The responses indicated that, overall, the system was well-received and manageable, with participants highlighting the clarity of the instructions and the calming nature of the virtual environment. Nonetheless, some users expressed a need for improved onboarding and more customizable control options, which will be taken into account in future iterations to further enhance accessibility and engagement for all users.

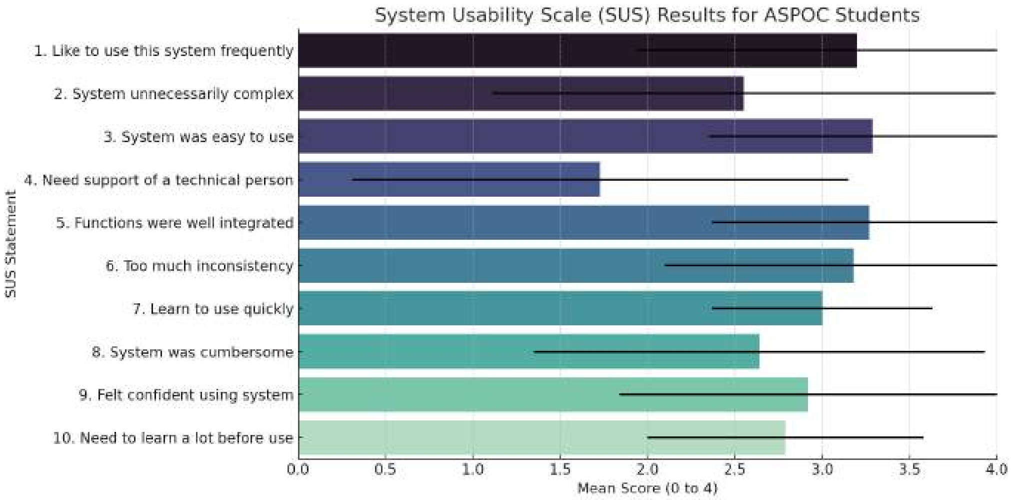

Figure 10: System Usability Scale (SUS) results for 14 students from the ASPOC association. The average normalized scores (0–4) for each SUS statement are shown, with standard deviation error bars.

The feedback from the pilot study indicated a high level of interest and engagement among students. Many were excited about the possibility of seeing and interacting with virtual entities, an activity not commonly encountered in their usual learning environment. The data analysis revealed significant differences in usability among the three configurations tested.

6 CONCLUSIONS

This study presented the development and evaluation of a Virtual Reality archery training application aimed at teaching fundamental techniques in an immersive, accessible, and structured environment. Unlike existing VR archery games that focus primarily on entertainment, this system was designed with a pedagogical approach, offering users a three-phase experience comprising a guided tutorial, targeted training exercises, and a semi-realistic simulation.

Usability testing conducted with 18 novice users yielded an average SUS score of 87, demonstrating strong usability and validating the effectiveness of the design in promoting ease of use and engagement. Furthermore, an additional evaluation involving 14 students with disabilities from the ASPOC association confirmed the application's potential for inclusive training, highlighting its accessibility and overall positive user experience. These findings support the use of VR as a promising alternative for skill acquisition, particularly in contexts where physical limitations, space constraints, or motivational challenges might prevent traditional practice.

While the results are encouraging, the study also identified areas for future improvement, including enhancing on boarding support, refining control intuitiveness, and increasing user customization. The next stages of development will focus on integrating adaptive difficulty settings, improving haptic feedback, and exploring applications for more advanced archers or athletes undergoing rehabilitation.

Ultimately, this project underscores the valuable role of VR in democratizing access to sport training and offers a foundation for future research into its broader applications in physical education, therapy, and inclusive recreation.

REFERENCES

[1] Becattini, N.; Cascini, G.; Covarrubias Rodriguez, M.; Dozio, N.; Ferrise, F.; Li, S.; Morosi, F.: Learning inextended reality a comprehensive framework fortechnology-related skill acquisition. Lecture Notes in Mechanical Engineering, 411 418, 2025. http://doi.org/10.1007/978-3-031-76594-0_47.

[2] Brooke, J.: SUS: A quick and dirty usability scale, 1995.

[3] Clini, F.: Linee Guida - Livello Avanzato per l'istruttore di tiro con l'arco.

[4] Covarrubias, M.; Arlati, S.; Andriolo, L.: Inclusive vr training applications for students with cognitive disabilities: Co-design and usability evaluation. Proceedings of the IEEE Conference on Extended Reality and Accessibility, 112–119, 2023.

[5] Erhan, D.B..S.E.: The effect of virtual reality technology on the imagery skills and performance of target-based sports athletes. frontiers in Psychology, 11, 2021.

[6] Faure, C.; Magnan, A.; Bordeleau, M.; Moffet, H.: Training using virtual reality improves balance in people with parkinson disease: a systematic review. Journal of Geriatric Physical Therapy, 37(3), 127–132, 2014.

[7] Fitter, W.: Artemis VR Game Bow, 2023. https://wonderfitter.com/.

[8] Hughes, A.; Weinberg, R.; Stranahan, A.: Virtual reality in sports training: A systematic review. International Journal of Sports Science & Coaching, 16(2), 283–297, 2021.

[9] kilogrames: Arrowhead - Medieval Archery VR, 2022. https://www.kilogrames.com/.

[10] Lindgren, R.; Johnson-Glenberg, M.C.: Embodied learning and mixed reality: How embodiment in virtual environments can enhance learning. Educational Research Review, 25, 1–14, 2018.

[11] M., B.S.D.M.B.R.G.F.P.C.C.: An Interactive Haptic System for Experiencing Traditional Archery. Acta Polytechnica Hungarica, 15(5), 185–208, 2018.

[12] Mashat, A.; Alharthi, H.; Alabdulqader, N.: Designing inclusive vr learning environments: Accessibility guidelines and case studies. Computers & Education, 179, 104411, 2022.

[13] Mencaci, E.; Covarrubias, M.; Roma, S.; Bodega, D.: Inclusive opportunities foryoung students withdisabilities through mixed reality. Lecture Notes in Networks and Systems, 1058 LNNS, 348 355, 2024. http://doi.org/10.1007/978-3-031-65522-7_31.

[14] Ritz, M.; Brewer, B.R.: Reliability and validity of the system usability scale (sus) in vr-based interventions. JMIR Serious Games, 5(3), e2496, 2017.

[15] Riva, G.; Wiederhold, B.K.; Mantovani, F.: Virtual reality in the assessment and treatment of weight-related disorders. Cyberpsychology, Behavior, and Social Networking, 19(2), 69–74, 2016.

[16] SavySoda: Archery Pro, 2024. https://www.archerypro.com/.

[17] STUDIO, M.G.: Backyard Virtual Archery, 2022. https://www.backyardarcherygame.com/.

[18] Wiederhold, B.K.; Riva, G.; Kim, S.H.: Virtual reality and rehabilitation: A review of current research. Annual Review of Cybertherapy and Telemedicine, 19, 40–47, 2021.

[19] lker Balk , Said Enes Ylmaz , Erdal Binboa, N.D..S.T..: Comparison of heart rate variability psychological responses and performance in virtual and real archery. Brain and Behavior, 14, 2024.