*Artificial Intelligence and Human-Computer Interaction*
*Y. Ye and P. Siarry (Eds.)*


*doi:10.3233/FAIA240160*

# A Unity3D Engine-Based Virtual Reality Game for Improving Cervical Mobility Using Gyroscope Controls

Shirong SONG[1], Yongsheng WANG, Shike WANG, Chengze WANG and Yuting YAO
*School of Art and Design, Lanzhou Jiaotong University, Lanzhou, China*
ORCiD ID: Shirong SONG https://orcid.org/0009-0009-7867-675X

**Abstract.** This paper introduces a virtual reality game based on the Unity3D engine, designed to enhance cervical mobility through gyroscope control in conjunction with the low-cost VR display device, Google Cardboard. The game requires players to use head movements to control in-game objects, reaching specific locations to complete levels. The paper provides a detailed account of the game's development process, including the implementation of gyroscope control and game logic using the C# programming language. This interactive approach allows users to simultaneously enjoy entertainment while exercising their neck, thus improving cervical health. Through this research, the aim is to showcase the innovative applications of virtual reality technology in related industries and demonstrate how gamification elements can be integrated with the field of health to provide users with a more engaging and effective experience.

**Keywords.** Virtual reality technology, gyroscope, cervical mobility enhancement game, unity3D engine

## 1. Introduction

The rapid development of Virtual Reality (VR) technology has generated widespread interest in the field of medical rehabilitation. VR's immersive experience and interactive nature make it a potential tool for improving patients' physical mobility. For instance, a Swiss company is dedicated to assisting post-stroke patients in overcoming the loss of mobility in their left hand. Using VR technology, they provide patients with a virtual left hand that is controlled through the movement of their real right hand. Although this approach doesn't directly resolve the paralysis of the left hand, it can alter the brain's perception of the paralyzed limb [1]. However, in comparison to the widespread application of Virtual Reality (VR) in other fields, the potential of VR technology in the context of cervical issues remains largely untapped. According to data from the American Academy of Orthopaedic Surgeons, millions of individuals seek medical consultation for neck-related ailments annually, which significantly impacts their quality of life. Medical research indicates that enhancing cervical mobility can improve neck health by increasing flexibility, alleviating muscle tension, mitigating neck pain, and enhancing posture. These factors are crucial for preventing cervical problems, yet the

[1] Corresponding Author: Shirong SONG, 1390146728@qq.com.

current methods for enhancing cervical mobility still face challenges in providing effective and personalized neck exercise solutions.

In this context, the primary objective of this study is to develop a virtual reality-based game that aims to enhance cervical mobility through the use of smartphone gyroscopes and Google Cardboard. In the game, users can immerse themselves in cervical exercises by controlling the movement of in-game objects through head movements to accomplish specific tasks. Within this paper, we will provide a detailed exposition of the feasibility of developing a VR-based game for enhancing cervical mobility, along with the programming implementation and testing of the game. Furthermore, we will explore the potential for VR technology in related industries.

## 2. Feasibility of VR cervical spine mobility enhancement game development

### *2.1. Causes of cervical spine issues and assessment of cervical mobility*

The main cause of cervical spine issues refers to the degenerative changes in cervical intervertebral disc tissues and their subsequent pathological alterations that affect surrounding tissue structures and manifest corresponding radiological changes. The primary symptoms are observed in three areas: the neck, shoulders, and arms. As people's living standards and material resources continue to improve, societal competition intensifies, and fast-paced lifestyles leave people with less time for physical activity, leading to an increased amount of time spent sitting. This significantly exacerbates the occurrence of cervical spine problems.

Cervical mobility refers to the measurement of the range of motion of the cervical spine by observing the angles during forward flexion, backward extension, lateral bending to the left and right, and rotation to the left and right. With the rest of the body below the neck held stationary, and the face directed forward with the neck in an upright position and the eyes looking straight ahead, parallel to the ground, in a neutral position, the range of motion for cervical flexion (lowering the head) and extension (raising the head) is approximately between 35-45 degrees. The range for lateral bending of the neck to the left and right is approximately 45 degrees, while the range for rotating the neck to the left and right is approximately between 60-80 degrees [2].

The measurement of cervical mobility holds significance in early diagnosis of cervical spine issues, assessing their severity, and determining the functionality of various segments of the cervical spinal cord.

### *2.2. The feasibility of using smartphone gyroscope for neck exercise*

The smartphone gyroscope is a sensor embedded in modern smartphones, playing a crucial role in virtual reality, augmented reality, gaming, and various other applications. Its operation is based on the law of conservation of angular momentum, similar to traditional mechanical gyroscopes. It consists of one or more small rotating masses, which generate corresponding changes in angular momentum when the phone undergoes rotation or tilting. The sensor measures these changes to determine the phone's rotational speed, angle, and direction [3]. When combined with the cervical spine in a fixed relative position, cervical mobility can drive the gyroscope's movement, allowing for the creation of corresponding games aimed at exercising the neck.

### 2.3. Google cardboard

Google Cardboard is designed specifically for entry-level VR experiences and 360-degree videos, and it is compatible with nearly all smartphones on both the iOS and Android platforms [4]. It is one of the most accessible mobile VR solutions available, allowing users to easily experience VR with just an iOS or Android smartphone. For instance, in November 2015, The New York Times released a VR documentary titled "The Displaced" that utilized VR technology in conjunction with Google Cardboard. This documentary allowed viewers to immerse themselves in refugee camps around the world, presenting the real-life stories of displaced children. The profound content, combined with the immersive presentation, sparked the interest of many people in VR for the first time [5].

In 2019, Google Cardboard was officially open-sourced, allowing users to manually craft a simple and cost-effective VR headset. The materials required for this DIY project include thick cardboard or cardstock, two identical biconvex lenses, head straps (for securing it on the head), and the necessary adhesive tools [6]. During the crafting process, attention should be given to reasonable measurements and cutting of the cardboard, ensuring its transparency, and accommodating the smartphone's spatial requirements. Once assembled, users have the flexibility to adjust the lens placement and focus according to their individual preferences, thereby achieving the optimal virtual reality experience tailored to their needs. Google Cardboard, as illustrated in Figure 1.

**Figure 1.** Google cardboard.

### 2.4. Technical circuits

Integrating Normal Cervical Mobility Values into the Game, where the range of neck movement corresponds to the gyroscope's movement range. However, individuals typically cannot achieve the full range of gyroscope rotation, so it is necessary to limit this range.

The maximum angle for gyroscope forward and backward flips is set to the midpoint of the range of cervical flexion (lowering the head) and extension (raising the head), with a deviation of ±40°. The maximum angle for gyroscope left and right flips is set to the standard range of lateral bending of the cervical spine, with a deviation of ±45°. As for cervical rotation while shaking the head left and right, this will be determined by the game's rules. Users will achieve the purpose of exercising the neck during the entertainment process, as illustrated in Figure 2.

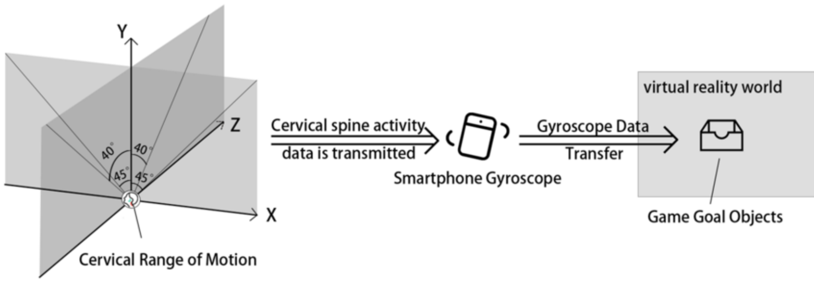


**Figure 2.** Technical circuit.

## 3. VR cervical spine mobility enhancement game implementation

### *3.1. Game content configuration*

Within a designated area, users search for randomly dropped coins using their binocular vision. Upon spotting a coin, users move their cervical spine to control a receptacle to collect the coin. Additionally, the game includes a scoring system to fuel players' competitiveness and enhance replayability. Throughout the gameplay, players' cervical spines receive ample exercise. Players need to set a playing time, and after reaching the set time, they pause the game for a break to avoid overuse and prevent cervical injuries.

### *3.2. Unity version and development environment setup*

The development of the VR Cervical Mobility Enhancement Game for this project was carried out using Unity version 2019.3.0f6. The game is designed to be compatible with Android smartphones and is intended to be experienced through Google Cardboard.

Therefore, it is necessary to download Unity version 2019.3.0f6 in Unity Hub and install the Android Build Support module, which includes OpenJDK and Android & NDK Tools. Additionally, for developing on Google Cardboard, the GoogleVRForUnity plugin needs to be installed in the Unity project, with the specific plugin version used being 1.200.1 [7].After the installation of the Unity Editor, Android modules, and the GoogleVRForUnity plugin is completed, the next step is to configure the project for the Android platform and set it to use Cardboard as the VR presentation mode [8].In the Unity Editor panel, select "File," open the "Build Settings" panel, add the scenes to "Scenes In Build," and change the platform to Android. Next, on this page, select "Player Settings." Check "Virtual Reality Supported" under "Player's XR Settings." Click the plus sign next to "Virtual Reality SDKs," and select "Cardboard." In the "Graphics APIs" under "Other Settings," remove Vulkan and leave only OpenGLES3 [9].

After enabling the VR development environment, Unity will set the orientation of the phone to landscape left by default. With this step, the development environment setup is complete.

### 3.3. Camera (binocular display) configuration

In the Unity scene, drag the plugin's provided prefab "GvrEditorEmulator" into the scene. This prefab offers editor-based simulation of head movement and controller input. Create an empty GameObject and name it "CameraRig," then reset its position. Reset the position of the existing main camera ("Main Camera") in the scene and drag it as a child of "CameraRig." Once their positions overlap, moving the "CameraRig" is equivalent to moving the scene's main camera. Position the "CameraRig" appropriately. Run the scene, and you can use the ALT key along with the mouse to achieve perspective rotation [10]. With that, the camera setup is complete, and the binocular display for Android smartphones is also configured.

### 3.4. Gyroscope control of target objects

In the game design, it's necessary to use the smartphone gyroscope (provided the device supports it) to control the receptacle for collecting coins. This involves mapping the values of gyroscope movement in the three-dimensional real world to two-dimensional plane movement values for target objects in the three-dimensional virtual world. In Unity, create a plane and place it in the scene. Simultaneously, drag the receptacle model into the Hierarchy panel and position it on top of the plane. Create a C# script named "GyroscopeControl" and attach it to the receptacle.

When the script is running, it allows the use of the gyroscope to control the receptacle's movement on the horizontal plane (XZ plane). The logic of the code is as follows: When the game starts running, it checks if the mobile device supports the gyroscope. If not, the game won't start. If supported, the gyroscope is enabled. After enabling the gyroscope, the script retrieves the current gyroscope's rotation speed information. When a left or right deviation occurs, the script controls the object to obtain the offset angle on the Z-axis. It then checks if the offset angle falls within the standard range of ±45° for cervical lateral flexion and rotation. If it exceeds the set offset angle, it adjusts the offset angle to be within this range and updates the position of the receptacle on the Z-axis. If it doesn't exceed the range, it accumulates the offset angle, which can be zero, and updates the position on the Z-axis. Similarly, when there is an up or down deviation in the gyroscope, the script controls the object to get the offset angle on the X-axis. It then checks if the offset angle falls within the middle range of ±40° for cervical flexion and extension. If it exceeds the set offset angle, it adjusts the offset angle to be within this range and updates the position of the receptacle on the X-axis. If it doesn't exceed the range, it accumulates the offset angle, which can be zero, and updates the position on the X-axis. When there is movement on both the Z-axis and X-axis (movement values can be zero), the script updates the object's movement on the XZ plane. This process repeats continuously to update the object's movement position. When the user exits, the game ends. The gyroscope control logic for the target object is shown in Figure 3.

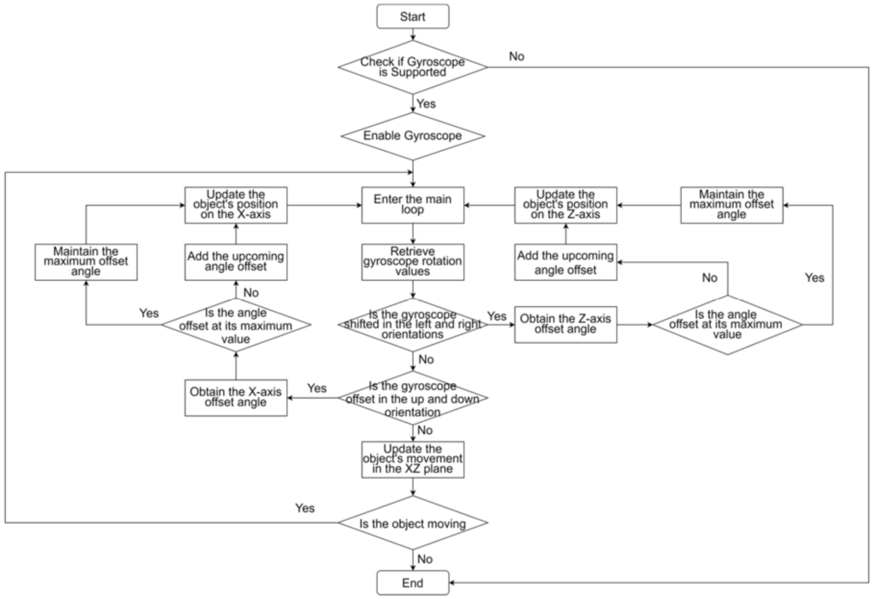


**Figure 3.** Gyroscope-controlled object logic diagram.

### *3.5. Random gold drop and score system implementation*

In this game, it is necessary to implement the effect of random coin dropping and calculate the score for the target object controlled by the gyroscope when it collects coins. Create a coin prefab named "Coin," change the tag of the coin prefab to "Coin," and attach a "Rigidbody" component to it. This will make the coin prefab subject to gravity and simulate the dropping effect. Check the "Is Trigger" checkbox to ensure that the coin doesn't generate physical collisions when entering a trigger area but triggers an event instead. Create a canvas and add a text element to the canvas to display the score.

Create a C# script named "CoinGenerator" and attach it to the target object controlled by the gyroscope. In the script, add the "UnityEngine.UI" namespace to respond to and pass parameters (numerical values) to the Text component in the scene to display the received number of coins in the UI layer each frame.

At the beginning of the game, coins are randomly generated within a certain range, for example, within a 10x10x10 area. Firstly, in the "Start" method, the "GenerateCoins" function is activated. Inside the "GenerateCoins" coroutine, the "Random.Range" function is used to randomly generate the positions of the coins. This function generates random X, Y, and Z coordinates in three dimensions, and these random values are stored in a Vector3 variable named "spawnPosition." This variable represents the position where the coins will be generated. Secondly, the "Instantiate" function is used to create instances of the coin objects. This function takes three types of parameters: GameObject (the object to be instantiated), Position (the spatial position of the object), and Quaternion (the rotation of the object). The X, Y, and Z values of the "spawnPosition" variable are assigned to the instantiated coin's Transform component. The coins are automatically destroyed after 10 seconds.

When a coin comes into contact with the receptacle, the script calculates the number of coins received by the receptacle and displays it as text on the screen. Create an integer variable named "coinNumber" and a Text variable named "coinNumberText." "coinNumber" will be used to keep track of the received coin count and should be initialized to 0. "coinNumberText" will reference the Text component in the scene. When an object collides with the collider detector, check the tag of the colliding object. If the tag is "Coin," increment the "coinNumber" by 1. Then, use the ToString() function to convert the value of "coinNumber" into a string, and assign this converted string to the "coinNumberText" component to display it. The logic diagram for coin random spawning and the scoring system is shown in Figure 4.

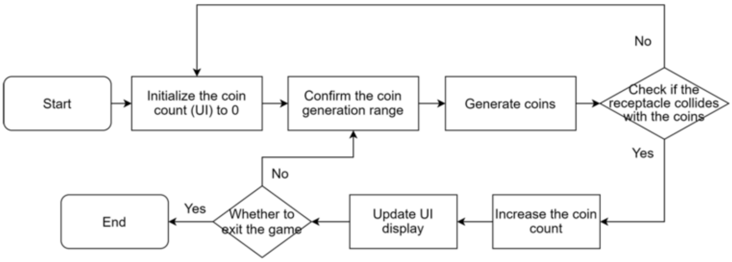


**Figure 4.** Logic diagram of random gold drop and scoring system.

## 4. Execution and testing

Enable developer mode on your Android smartphone and connect it to your computer. In the Unity editor, navigate to the "File" menu, then select "Build Settings." Add the game scene you've created, choose the model of your Android device in the "Run Device" section, and click "Build And Run." Specify the storage location and install the game on your device. If the software runs smoothly, it indicates that the program is functioning correctly. After the APK installation is complete, place your Android smartphone into a Google Cardboard headset, put it on, and enjoy the game in moderation to enhance cervical spine mobility.

After ensuring that the game runs correctly, it is necessary to conduct testing on various aspects of the game. This testing should include gyroscopic control testing, verification of whether the UI layer displays the correct coin count when the container receives coins, and conducting relevant latency testing. For this testing, we will use an Android smartphone, specifically the IQOO 10 PRO model, as an example.

**Gyroscopic Control Testing:** lace the smartphone screen directly in front of your eyes, and move the smartphone. The movement of the gyroscope should be equivalent to the movement of the smartphone. The gyroscope's movement in the real-world three-dimensional space should correspond to the game's movement in the virtual two-dimensional plane. The testing results are shown in Table 1.

**Table 1.** Gyroscopic control testing table.

| Frequency | Gyroscope Movement Direction | Corresponding In-Game Direction | Correct |
|---|---|---|---|
| 1 | Forward tilt | Forward | Yes |
| 2 | Backward tilt | Backward | Yes |
| 3 | Left tilt | Leftward | Yes |
| 4 | Right tilt | Rightward | Yes |

**Container Received Coins vs. UI Displayed Quantity Consistency Test:** In this test, the consistency between the number of coins received by the container and the quantity displayed on the UI layer is evaluated. When the container detects a collision with a coin, the UI layer's data should increment based on the collision count. The test results are presented in Table 2.

**Table 2.** UI layer data display test table. (Unit: pieces)

| Frequency | Received Coin Quantity | UI Displayed Quantity |
|---|---|---|
| 1 | 0 | 0 |
| 2 | 5 | 5 |
| 3 | 10 | 10 |
| 4 | 15 | 15 |
| 5 | 20 | 20 |

**Delay Test:** Testing the delay in controlling the target object with the gyroscope and the delay in UI layer display after receiving coins, the test results are shown in Table 3.

**Table 3.** Delay test table. (Unit: milliseconds)

| Frequency | Delay of gyroscope-controlled object | Delay of UI layer display after receiving coins |
|---|---|---|
| 1 | 20 | 402 |
| 2 | 28 | 385 |
| 3 | 25 | 370 |
| 4 | 30 | 360 |
| 5 | 26 | 396 |

Through the three tests, it can be observed that the orientation of gyroscope movement corresponds to the corresponding orientation of the controlled object in the game. The number of coins received by the container matches the UI layer display. The delay in the gyroscope-controlled object is within 30 milliseconds, and the UI layer display delay after receiving coins should be around 400 milliseconds. The UI layer display delay is higher than the gyroscope-controlled object delay because the gyroscope-controlled object delay involves real-time tracking of head movement, which requires fast response. On the other hand, the UI layer display after receiving coins is typically non-real-time feedback or information display and is not as critical as head movement control in terms of responsiveness. However, overall, it can provide users with an immersive virtual reality experience and a satisfactory cervical mobility experience.

## 5. Conclusion

This article primarily discusses the development of a Virtual Reality (VR) game for enhancing cervical spine mobility, utilizing the Unity3D engine and gyroscope control technology. The main objective of this game is to help users improve their cervical spine's range of motion, enhance neck flexibility, all within an engaging and interactive VR experience. Within the paper, we begin by emphasizing the potential applications of VR technology in the context of neck mobility exercises. We analyze the factors contributing to neck issues and how to assess cervical spine mobility, laying the theoretical groundwork for game development. Furthermore, we explore the feasibility of using smartphone gyroscopes for cervical exercises and successfully integrate the normal range of cervical mobility into the game mechanics. Subsequently, we provide a detailed description of the game's design and implementation process. This includes configuring a stereoscopic view through camera setup, programming the gyroscope-controlled object, and designing the coin randomization and scoring system. These elements are thoughtfully designed to ensure that the game not only promotes cervical exercise but also offers an enjoyable and challenging experience. Finally, we conduct runtime testing to validate the functionality and performance of the game. These tests are crucial in ensuring the game's playability and user experience.

In summary, this paper demonstrates the creation of a VR cervical spine mobility-enhancing game using Unity3D engine and gyroscope control technology. This game not only serves to address cervical spine issues but also serves as a showcase for the wide-ranging applications of VR technology in related fields. However, due to the low cost and simplicity of Google Cardboard, the user's VR experience is not without limitations. In the future, considering further improvements and expansions of this game, including adding new gameplay elements to enhance its entertainment value, or using this as a foundation to create other games for enhancing cervical mobility to meet the needs of different user groups. This will allow for continued exploration of the potential of VR technology in related industries.

## References


[1] Tilley Aaron. Mindmaze raises $100 million at a $1 billion valuation for 'neural virtual reality.Forbes.com. 2016;18-18.

[2] Wang Shizhong, Zhang Hong. Rehabilitation assessment studies. People's Medical Publishing House. 2012.

[3] Zhao Yanling. Exploration based on smartphone gyroscope. Wenyuan (Secondary School Edition). 2021; (12): 4316-4317. doi: 10.12252/j.issn.2096-627X.2021.12.2233.

[4] Fokides E, Polydorou E, Mazarakis P. Using Google cardboard compatible HMDs and spherical videos for teaching history to high school students. Int. J. Smart Educ. Urban Soc. 2020; 11:18-34, doi:10.4018/IJSEUS.2020100102.

[5] Perla R, Hebbalaguppe R. Google cardboard dates augmented reality: Issues, challenges and future opportunities. 2017. doi:10.48550/arXiv.1706.03851.

[6] Google. Get your Cardboard [EB/OL]. [2023-09-17]. https://arvr.google.com/cardboard/get-cardboard/.

[7] Google VR. Google VR SDK for Unity[EB/OL]. [2023-09-17]. https://github.com/googlevr/gvr-unity-sdk.

[8] US Vaissiere Vaissiere, Michael, US Pasamonik Pasamonik, Mike, US Bickhofe Bickhofe, Peter, et al. Google Cardboard App Development Guide [M]. People's Posts and Telecommunications Publishing House,2018.

[9] Google. Quickstart for Google Cardboard for Unity[EB/OL]. [2023-09-17]. https://developers.google.com/cardboard/develop/unity/quickstart#configuring_android_project_settings.

[10] Google. Quickstart for Google Cardboard for Android NDK[EB/OL]. [2023-09-17]. https://developers.google.com/cardboard/develop/c/quickstart.