HoloHealth - A Holographic Game Application for Promoting Physical Exercise

Elvira Forslund Widenroth
Luleå University of Technology
Luleå, Sweden
elvfor-0@student.ltu.se

Peggy Khialie
Luleå University of Technology
Luleå, Sweden
pegkhi-0@student.ltu.se

Jenny Sundström
Luleå University of Technology
Luleå, Sweden
ejyuso-2@student.ltu.se

Daniel Thungren
Luleå University of Technology
Luleå, Sweden
danthu-0@student.ltu.se

Abstract—**Physical inactivity is declining in modern society, necessitating strategies to promote everyday movement and exercise. Augmented Reality (AR) technology offers promising avenues for addressing this challenge, with potential applications in healthcare and lifestyle interventions. This paper explores the intersection of AR and exercise promotion by developing a HoloLens-based game application.**

The result is an implementation of the HoloHealth application, incorporating real-time tracking of movement metrics, facilitating user engagement and interaction. Despite challenges in integrating AR technology with accurate measurement capabilities, the project underscores the potential of AR-based interventions in promoting physical activity.

I. Introduction

The lack of physical activity and exercise is a growing problem in modern day society and can increase the risk to a number of different health related diseases. Therefore everyday movement and exercise plays a pivotal role in maintaining our overall health and well-being.

In article [1] the author discusses the potential of AR by emphasizing its capacity to not only redefine interactions and experiences, but the role that it will play in the future as the next universal computing platform. If AR tools are to be as dominating as the paper points out it magnifies the understanding of how these tools may be more accessible in the everyday lives of many people in the future.

Gamification of AR tools is an example that has already presented promising results for increasing measures of health. This type of motivation from gamification of AR technologies has been shown in the research article [2], where the goal was to prevent declines in physical and cognitive abilities due to aging. These findings underline the advantage of exploring innovative technological solutions, such as gamified AR tools, to address the pervasive challenge of exercise commitment and promote healthier lifestyles.

A challenge for introducing AR into clinical settings could be not having the accuracy of such systems evaluated which is something article [3] investigates. An assessment of the HoloLens ability to gather information about a trial gait compared to standard motion capture systems was made. The study was aimed to validate HoloLens accuracy in quantifying gait across various speeds, crucial for healthcare applications. The study contributes to integrating AR technology into clinical practice for more comprehensive and valid assessments of human movement.

The study [4] further investigates and validates the accuracy of the HoloLens 2 sensors and technology. Features such as Hand Movement capturing via the AHAT Depth Camera, Spatial Mapping of surroundings, Speech Recognition and much more were validated to have high accuracy. These are features that could be used when developing clinical applications on the HoloLens, and this paper verifies their accuracy.

The objective of this project is to use AR Technology to build a game application on the HoloLens with the purpose of promoting physical exercise. By using the AR technology in combination with the sensors and technology of the HoloLens to motivate users to play the game, the users will be promoted to more physical exercise. The paper presents methodologies used and results derived from the findings. A discussion about the limitations and future work is made followed by a conclusion about the research.

II. Methodology

The Microsoft HoloLens 2 device was used to create and display a holographic application. Figure 1 displays the system architecture in a high level and in regards to the user.
The architecture describes the users interaction with AR and the application via the HoloLens. The HoloLens will capture hand movements from the user along with a depiction of the real world and send these to the Holographic Game Application. The application will then feed holographic cubes and information of the AR world back to the user.

A. Tools

The application was developed using Unity together with Visual Studio. The Mixed Reality Toolkit (MRTK) [5] created

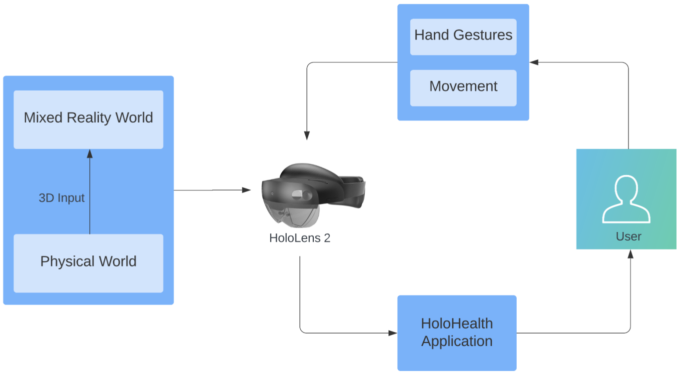

Fig. 1. A System Diagram

by microsoft, has been used to develop the Holographic application in Unity to create scenes and movable objects that can be moved with the hand movements detection from the HoloLens. A solution file of the Unity Project was built in Unity which was later built and deployed to the HoloLens using Visual Studio.

B. Application Details

The HoloHealth application is a game in which the user gets an easy task to promote moving. Data about the user's activity such as velocity and distance traveled is simultaneously displayed during game play.

In Unity, scenes serve as containers for organizing and managing game content, such as environments and objects. Within scenes, game objects represent individual entities with specific behaviors and properties. Scripts, written in C# define the logic and functionality of these game objects. Through Unity's scripting API, scripts can access and manipulate game object properties, respond to user input, and interact with other game objects. This interaction enables dynamic behavior and gameplay mechanics, facilitating the creation of immersive and interactive experiences within Unity projects.

There are three different scenes in the HoloHealth application. Firstly the Main Menu Scene, then the actual Game Scene and lastly a Final Score Scene containing information about the performance of the game played. All scenes have a cavnas with the SceneManager.cs script attached to them which allows for a smooth transition between them.

The Main Menu Scene is the start of the game application. It consists of game content such as a title, rules for the game, a play button that takes the user to the game and an exit button for exiting the application.

The Final Score Scene presents information about the current score, high-score and stats about the amount of movements made during the game. The scene also contains a button to restart the game and a button to take the user back to the menu.

Due to the simplicity of the first and last scene, the focus will be on the Game Scene and its components are what will be explained in more depth. Figure 2 displays the structure of the Game Scene and how its scripts interact with the different game objects.

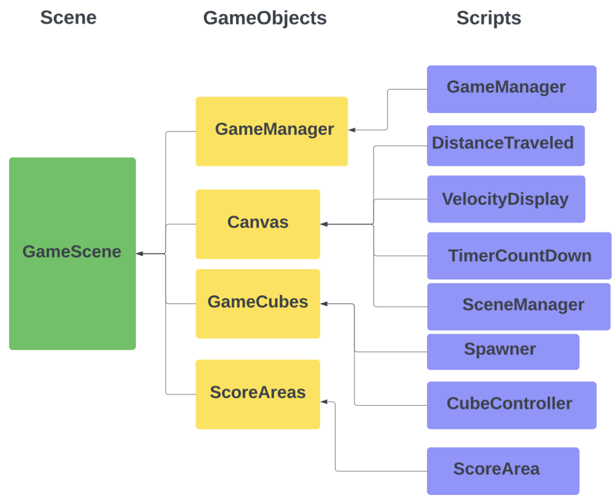

Fig. 2. Game Scene, Game Objects and their scripts

The **GameManager** is what manages the game through its own GameManager.cs script by creating an instance of the game. It also have functions for setting and getting both score and hi-score which is needed in order for the player to be able to save the hi-score variable and share it across new games. It is also needed in order for the GameCubes to access the score variable every game.

The **Canvas** is the UI and display information of the game such as a the timer, score, hi-score, velocity and distance travelled. There also exist a quit game button which quits the game and display the score the player had before quitting the game. A game is limited by the TimerCountDown.cs script which is displayed as a timer in the canvas and set to play for 60 seconds. When the timer reach zero the game transition to the Final Score Scene where the final scores are displayed for the player along with the total distance traveled during a game.

Also displayed in the Canvas are data about velocity and distance traveled by the player. The script DistanceTraveled.cs calculates and displays the total distance traveled by the main camera in the Unity environment. It tracks the movement of the camera frame by frame, computing the distance covered between the current and previous frames using a built in distance-function. The cumulative distance is updated and displayed in real-time on the screen by the text game object.

To compute and display the velocity the script VelocityDisplay.cs was made. It calculates the velocity vector of the camera using the camera's velocity property, which represents the speed and direction of its movement.

The magnitude of this velocity vector is then displayed in real-time on the screen, this also by updating the text game objects in the canvas.

The **GameCubes** are controlled in Unity by having collision and interaction added as components. They are then controlled by the CubeController.cs script which checks if they have collided with a score area, if so then the player will be awarded points. The Spawner.cs script is added to spawn a set amount of clones randomly across the game view. These clones are children to the main cubes and every component added to the parents will also be added to the clones when the game starts and they are generated. The clones are spawned randomly across the plane where the game is held.

The **ScoreAreas** are point holders which have 10 points attached to them, so if there is a collision with a cube of the same color, the score area will return 10 points and add them to the current score through the game manager instance.

III. Results

The result is a holographic game application on the HoloLens 2 headset that consists of three Unity scenes. A flow diagram of the game is displayed in Figure 3.

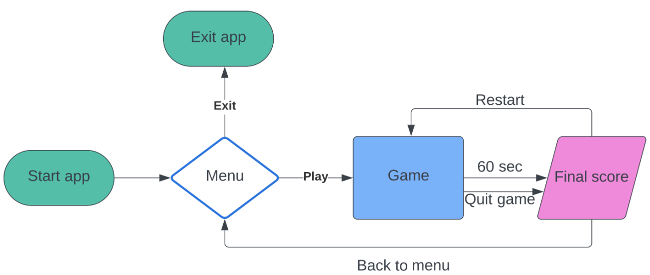

Fig. 3. Flow-diagram of Game

The application opens up to a Main Menu Scene that is seen in Figure 4. The Main Menu Scene then takes the user out of the application or further to the Game Scene.

In the Game Scene shown in Figure 5 the user of the game sorts the cubes of different colors by moving them across the room. Cubes will spawn randomly across the room and points will be given to the user when the cubes are moved to the correct score area. Game stats such as score, high score, velocity and distance moved will be displayed during the game for users to see.

At the end of a game the user will be transitioned to the Final Score and have the ability to restart the game or move to the menu, which is presented in Figure 6. The game also contains a quit game button for finishing the game early, and transitions the player to the Final Score Scene.

Fig. 4. The Main Menu Scene

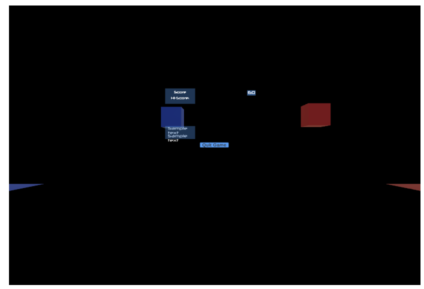

Fig. 5. The Game Scene

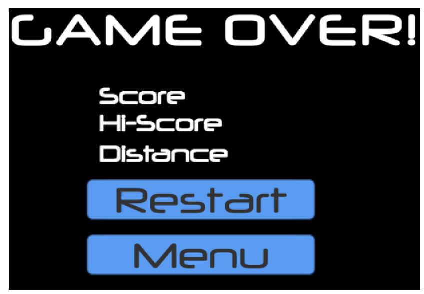

Fig. 6. The Final Score Scene

IV. Discussion

A. Limitations

The time was the largest limitation which limited what could be accomplished quite severely. Some things regarding the project had to be excluded from the project, such as: user study where different users tested the game, study to see improvements of motivation and amount movement over time to see the long-term effects of the application. The project was therefore limited to only developing the application.

B. Difficulties Encountered & Reflection on Results

At the projects current state, the velocity and the distance moved values which are displayed in the game, are calculated by tracking the main camera object in the Unity application. The accuracy of these estimated values by the inbuilt functionality from Unity are not validated and by simply testing the game application it is possible to conclude the estimations are not accurate. The aim was instead to use the HoloLens Research Mode [6] in order to get more accurate measurements of this.

However, the poorly documented API in combination with a language difference between the API and Unity, made this process difficult. The API is written in C whereas Unity uses C# Scripts. The API therefore had to be transformed to C++ and packaged as a Dynamic Linked Library (DLL) which was later imported to use in C# scripts in Unity. Some OpenSource projects where such a DLL plugin had been created was tested out, however, none of the tested applications worked when importing them to the HoloHealth application.

Getting accustomed to Unity can be challenging at first, the instructions about how to integrate the MRTK with Unity was not direct and took a lot of time. Although some insightful tutorials made by other Unity users proved resourceful, as they provide a quicker learning experience. Through these tutorials the intricate work of settings adjustments, how the components such as the scripts and game objects interplay with each other and the environment gets broken down into smaller parts, giving a clearer understanding of the work flow in Unity. These tutorials also helped with the challenge of not having prior experience coding in C#.

The resulting canvas and its components such as buttons and game rules are not aligned perfectly within the screen for the player as in a two dimensional game. Instead, it has a tag-along component that roughly follows the direction of which the player is facing. This component comes with some occasional rendering inconsistencies, but serves its purpose for the time being.

The choice of which version of Unity to use impacts the handling of materials, assets and project compatibility overall. An issue encountered involved the project being developed using a newer version of Unity, while the created git repository used an older version. This mismatch resulted in numerous errors involving the MRTK extension and project assets. The issue was resolved by overwriting the git repository with the original version and then rebuilding the project using the newer Unity version. This turned into a valuable lesson of the importance in consistency of using the same/latest version, and sticking to it for next time. By adopting this approach, such incompatibility issues might be avoided altogether, ensuring a smoother project development.

C. *Future Work*

Some future works that would improve not only the game experience but also the aspect of physical exercise lies both in how data is handled and in the game itself.

Getting streaming values from the Hololens Research Mode [6] would improve the accuracy of movement values. Furthermore, setting up a Database that can store such values for evaluation of movements over time would both expand the usage of the application, since user experiments could be performed and documented, and improve the overall completeness of the application.

For future improvements of the game, better spawning of the cubes would result in a more unpredictable pattern in placements of the cubes. Varied distances of were the cubes spawn would also mean more movement for the user to be able to reach them. The score areas could also be made more visible for the players sake of finding them with less difficulty. Multiple game modes where various body movements would be needed to reach the cubes would further enhance the experience of a physically challenging workout, targeting different user needs.

V. Conclusion

In conclusion, the HoloHealth project successfully aimed to leverage AR technology, specifically the HoloLens 2 headset, with the result of a game application designed to promote physical exercise. The project was motivated by the increasing challenge of motivating individuals to engage in regular physical activity, considering the potential health implications.

References

[1] M. Abrash, "Creating the future: Augmented reality, the next human-machine interface," in *2021 IEEE International Electron Devices Meeting (IEDM)*, 2021, pp. 1–11.

[2] T. Park and M.-J. Shin, "Effectiveness of an exercise program for older adults using an augmented reality exercise platform: A pilot study," *Annals of geriatric medicine and research*, vol. 27, 03 2023.

[3] M. M. Koop, A. B. Rosenfeldt, J. D. Johnston, M. C. Streicher, J. Qu, and J. L. Alberts, "The hololens augmented reality system provides valid measures of gait performance in healthy adults," *IEEE Transactions on Human-Machine Systems*, vol. 50, no. 6, pp. 584–592, 2020.

[4] H.-J. Guo and B. Prabhakaran, "Hololens 2 technical evaluation as mixed reality guide," 2022.

[5] "Mixedrealitytoolkit (mrtk)," https://github.com/microsoft/MixedRealityToolkit, 2024, accessed: March 20, 2024.

[6] "Hololens2forcv," https://github.com/microsoft/HoloLens2ForCV, 2022, accessed: March 20, 2024.