Final Project Report

Kaiwen Guo
kg3354
kg3354@nyu.edu

Abstract

This project focuses on using Unity to create an augmented reality (AR) visualization of network connections surrounding users. Augmented reality (AR) enhances real-world environments through interactive digital visual elements, sounds, and other sensory inputs, often facilitated by holographic technology [1]. Wireless networking, particularly network protocols based on IEEE 802.11 (Wi-Fi), surrounds each AR user. With this AR application, users can easily visualize network performance, including Wi-Fi upload and download speeds, in their immediate vicinity.

1. Introduction

AR combines the "real world" and the virtual world. Essentially, AR takes the real world and enhances it through the use of computer-generated perceptual information or virtual overlays. [2] An interpretation of the virtual world can offer visible demonstrations of the invisible elements surrounding individuals, such as network connections. This project aims to extend the real-world experience to an AR environment by enabling users to visualize wireless network connections. This project enables the users to self-diagnose wireless network connectivity issues and move the device to locate optimal spots for the best wireless performance. This application has broad potential for AR applications, especially AR games that requires a stable network connection.

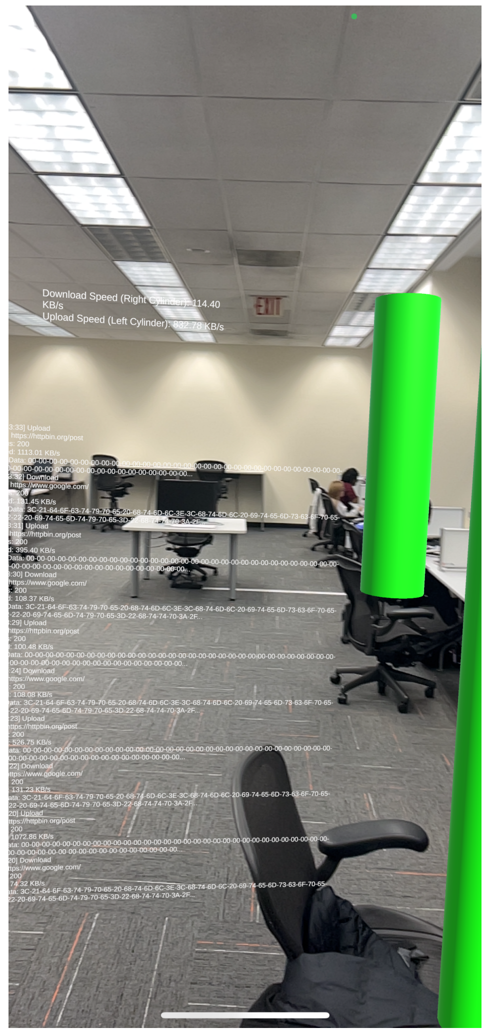

Figure 1. Sample screen shot of the project.

2. Results and Demonstration

There are three core functionalities of the project. First of all, it shows the Uplink and Downlink Speed in real-time. The textmeshpro object containing the real-time speed information will relocate itself to the center of the camera when the end user tabs the screen. Secondly, when the end user holds the screen for over 3 seconds, the textmeshpro object with the real-time network logging information will also relocate itself to the center of the camera when the end user tabs the screen. Lastly, the project contains 2 cone objects whose texture and height changes also in real time. Their texture color (red, yellow, green) represents the range of byte per second, and the height represents the relative speed.

Thus, end users will have two ways of iterating with the application: holding down the screen over 3 seconds, or tapping the screen. Each will centralize different objects to the center of the camera.

3. Implementation

The implementation of the project is relatively straightforward.

There are only four assets are involved in Unity AR, which are the two cones that uses color and height to present the relative uplink and downlink speed, and two TextMeshPro objects to display the network speed and network packet raw data logs in the scene.

The C sharp script contains 247 lines of code, where the key functions are Update, CenterTextInFrontOfCamera, UpdateNetworkInfo, UpdateConeHeights, GetSpeedColor, MeasureDownloadSpeedCoroutine, MeasureUploadSpeedCoroutine, and LogPacketData. The two functions MeasureDownloadSpeedCoroutine and MeasureUploadSpeedCoroutine conputes the real-time network speeds by posting information to https://httpbin.org and sending get request to https://www.google.com. The LogPacketData function will listen to all the network packets that goes through user's device, thus can print out the raw data of both what they send and what they receive in real-time. The rest of the functions are helper functions to display the data in a more professional and user-friendly manner.

The C sharp code is open source, and can be find on GitHub under github.com/kg3354/UnityARNetworkVisualizer

4. Discussion

The current system has a limitation: the target build is iOS, which restricts many built-in Unity network functionalities compared to Android. As a result, all network speeds in this project are calculated manually using various functions. This approach is inefficient and can be problematic, especially if not all network data is transparent to developers or users by default.

An extension of this project could involve improving the interaction model. Instead of requiring the user to tap the screen at various locations to move the TextMeshPro object, the objects should remain at their initial spawn points and retain the speed data from when they were first created. When the user moves a certain distance or requests another text object, a new TextMeshPro object containing the current data could be spawned at the user's new location. This enhancement would provide a more intuitive way for users to visualize how network performance changes as they move between different locations.

References

[1] Microsoft. What is augmented reality or AR?, 2024. https://www.microsoft.com/en-us/dynamics-365/topics/augmented-reality/what-is-augmented-reality. 1

[2] Pluralsight. Augmented Reality (AR): What is it, and how is it being used? www.pluralsight.com, Accessed: Oct. 28, 2024. https://www.pluralsight.com/blog/creative/what-is-augmented-reality-ar. 1