

# Mobile Multiuser AR/VR for Training

Syltinsy P. Jenkins  
Department of Computer Science  
Bowie State University  
Bowie, USA  
jenkinss0218@students.bowiestate.edu

**Abstract** — Augmented reality (AR) is a growing technology for building immersive and interactive applications for anyone to use. Integrated development environments such as Unity when enhanced with additional plugins like Unity’s AR Foundation and Photon Network, enable the rapid development of multiuser, mobile, augmented, and mixed reality (XR) applications that can be both entertaining and useful. This report is about a mobile AR and virtual reality (VR), or XR multiuser classroom prototype application. This project was developed on Unity with Unity’s AR Foundation Kit, Photon Network, and Android AR Core plugins to implement an Android based mobile phone application and a desktop application. The final multiuser applications include a real-time multiuser AR/VR environment for an emulated AR/VR Classroom, and an AR Spinning Top Demonstration.

**Keywords**— augmented reality (AR), virtual reality (VR), mixed reality (XR), Unity, Photon Network, AR Foundation, AR Core.

## I. INTRODUCTION

Since mid-March of 2020 to current date, late December 2020, much of the United States has been affected by the COVID-19 pandemic.

Due to COVID-19 related quarantines and shutdowns, there has been a push towards remote education and learning. Traditional content is less engaging. Both parents and teachers complain about students being less attentive and engaged in online class. At the same time, parents and teachers also complain about the inordinate amounts of time students choose to spend engaged in social media and gaming online.

Prior studies demonstrate an improvement in attention for users engaged in immersive experiences. With the long hours of working and studying from home, AR and VR may be able to provide more engaging experiences and platforms for learning. But which is better for learning, AR or VR? Some could immediately conclude that “*it depends*,” however few studies perform a direct comparison.

The purpose of this project is to implement a mobile 3D multiuser interactive augmented reality and virtual reality environment for training as a starting point for further research. Some related research questions include:

- Which is more immersive, AR or VR?
- Which is more engaging, AR or VR?
- Does AR or VR provide better learning results?

This project is a work in progress where a virtual classroom has been created. There is a space for learning in a

traditional sense with a classroom setting with a presentation screen. This AR/VR technique is a way that can create a sense of telepresence in a virtual or augmented space. There is also a game called *The Spinning Top Demo* that was implemented as a common area for game play for stress relief and entertainment that was created during the research and learning phases. It also exercises many of the desired attributes for an AR implementation. The spinning tops have physics, collisions, and sound effects.

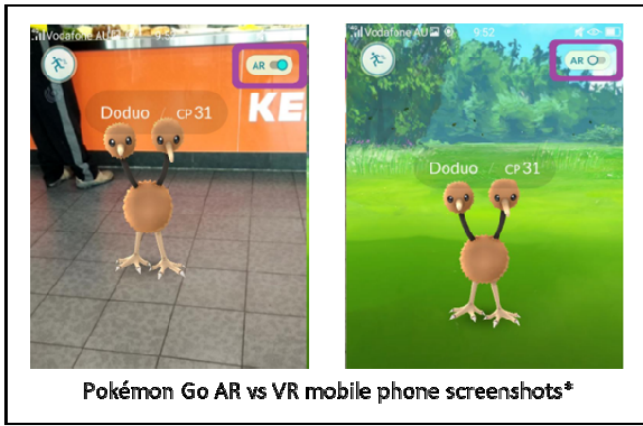
The objectives for this project include creation of a unified approach for development of AR and VR experiences, implementation of a seamless transition between AR and VR, and to support more than three simultaneous players.

## II. BACKGROUND

Augmented and virtual reality technologies can be considered relatively mature as these systems have been used in real world applications for many years. For example, since the late 1990’s, the Boeing Company has used augmented reality in aircraft production to aid assembly workers route complex electrical wiring resulting in 20-50% faster assembly when using the AR system[1].

Extended reality (XR) is a term referring to all real and virtual combined environments and human machine interactions generated by computer technology and wearables, from the completely real, through augmented and mixed reality to the completely virtual as in virtual reality[2]. In this sense virtual reality is considered fully synthetic. Virtual reality is defined as “a high-end user-computer interface that involves real-time time simulation and interactions through multiple sensorial channels” whereas “augmented reality (AR) is a user-computer interface whereby “graphics are over-imposed on the visual image of the user’s immediate surrounding area[1].”

Although the AR/VR field of research is relatively mature, many people have not used or experienced full immersive VR or AR due to lack of infrastructure[3], cost and limited availability of special equipment like head mounted displays (HMDs). Today, many peoples first experiences using AR and VR technologies have been in the entertainment space in video games and on mobile devices for example “Pokémon Go” shown in the following figure.



Images from: user106385 <https://gaming.stackexchange.com/questions/77643/there-is-a-pok%C3%A9mon-in-the-kebab-store-can-i-turn-off-augmented-reality>

Figure 1. *Pokémon Go* AR vs. VR mobile screenshots [4].

The Unity development platform is used to create interactive 3D for AR and/or VR applications. Analogous AR and VR experiences for the same educational content can be created using a common code-base, application flow, and set of models.

Figure 1 includes AR and VR screenshots of gameplay from *Pokémon Go*; the left image is AR and the right image is VR. The user can switch between the two using the AR switch outlined with purple in upper right-hand corner of each screenshot. While the animated characters appear well placed in the above image, during gameplay, the characters are not stably positioned within the frame. In a more immersive implementation, features such as plane detection can be leveraged to enable the virtual character to be placed consistently “on the ground.”

Users notionally see and feel an immersive difference between AR and VR experiences. Few studies attempt to quantify and capture AR/VR differences with respect to learning. Previous research has found that, “in terms of presence, embodiment, and performance demonstrate that both first- and third-person perspectives seem consistent with the induction of a high spatial presence feeling. However, the first-person perspective remains the best condition to induce a sense of embodiment toward a virtual body [5].”

The Android platform was selected for mobile AR and VR because it is widely available [6] and supports multiple formfactors including: handheld mobile phones, tablets, and compatible head mounted display (HMD) for 360° immersion such as Samsung Gear Oculus and Google VR headsets.

### III. METHODOLOGY

#### A. Development Phases

An iterative design methodology was used for development. The application leverages a simple design using on 8 the 10 most commonly used heuristics in developing user interfaces[2] including:

1. Match between system and the real world

2. User control and freedom
3. Consistency and standards
4. Error prevention
5. Recognition rather than recall
6. Flexibility and efficiency of use
7. Aesthetic and minimalist design
8. Help and documentation

Large display font and button size were preferred over field of view based on findings by Arntz et.al. in *Navigating a Heavy Industry Environment Using Augmented Reality - A Comparison of Two Indoor Navigation Designs* [1].

The project was developed using an evolving iterative cycle as shown in Figure 2.

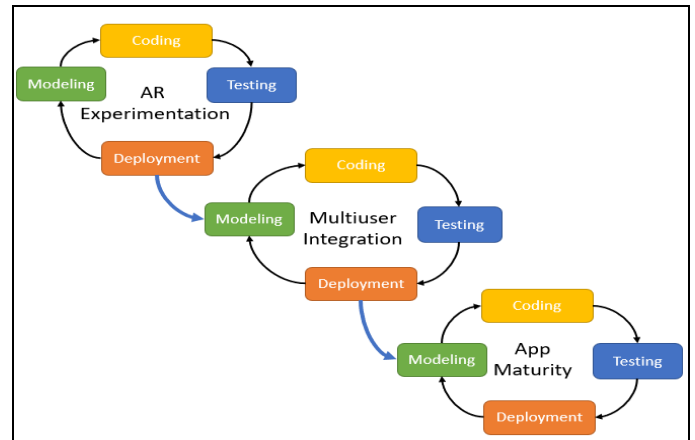


Figure 2. Development Phases.

There were three main phases: AR experimentation, multiuser integration, and application maturity. The basis of each phase was a cycle consisting of modeling, software development or “coding”, testing, and deployment. Modeling involved collecting modifying or creating 3D objects to be used in the 3D environment. Software development or coding is the use of C# programming in the Unity IDE leveraging plugin application programming interfaces (APIs). Unity’s development methodology exposes variables and methods created in code to be accessible via fields in the IDE application.

The first phase was AR experimentation using Vuforia versus AR Foundation. A variety of documentation and resources were evaluated and used. Basic avatar models were obtained, and a basic scene was constructed for AR experimentation. The first approach was the use of image targeting using Vuforia followed by AR Foundation. Image targeting involves the use of a physical marker in the user’s environment known as a “target [7].” The camera component of the AR system detects the target marker and then renders an AR image on top of the camera video feed relative to the target marker. It was found that when moving the handheld mobile device, that AR images were not stable relative to the AR target and had to frequently be “reacquired” if the user lost the target from the field-of-view (FoV). Vuforia had superior image target identification and mapping compared to AR Foundation. To establish AR object persistence outside the

FoV, plane detection and mapping was used [8]. Due to time and resource constraints, this was implemented only using AR Foundation. The plane detection approach provided greater AR object stability and permanence.

The second phase, multiuser integration involved the modeling and development of a series UI menus to manage and facilitate logins to support multiple simultaneous users. Exploration of the AR capability along with the exchange of data across multiple devices included the development of the Spinning Top Demonstration.

Finally, the third phase, application maturity, began with the modelling of the classroom environment and included avatar animation. 3D models and animations were collected and modified from various internet resources including SketchUp free 3D models, and Unity assets.

### B. Development Environment

The system was developed on a computer with an Intel Core i7-7700HQ CPU @ 2.80GHz, 32 GB DDR4 RAM, and NVIDIA GeForce GT1060 (6GB GDDR5) running Windows 10 Professional. Autodesk 3D Studio Max, and Trimble SketchUp, and Adobe Mixamo were used for modeling and animation.

The development environment consisted of the Unity 2019.4.12f1 configured with AR Foundation Kit, Photon Unity Networking (PUN) plugin for multiuser support, and Android AR Core for mobile AR support. The research and development phase included experimental development using the Vuforia plugin for AR and Player IO as an alternative for multiuser support.

The minimum Android API supported by the AR Foundation with AR Core for AR plane detection is Android 7.0 API 24[9]. The phone requires a multicore processor and AR Core installed on the phone. The implementations were deployed and tested on a Samsung Galaxy Note II using Android 7.0, Samsung Galaxy S7 Edge Android using Android 8.0, and a Google Pixel 4a using Android 10.0. AR launched but did not render correctly on the Note II. The applications were successfully deployed to the S7 Edge and Pixel 4a.

Ultimately PUN and AR Foundation with AR Core were selected for development due to the availability of resources. Plane detection was found to be more stable and useful than image targeting in AR Foundation. AR Image targets were found to be more stable in Vuforia but combining Photon with Vuforia with multiuser support was more difficult. Vuforia has a separate implementation for supporting multiuser AR. Vuforia also has a development environment that is different than Unity. AR Foundation is developed and maintained by Unity. Similarly, PUN was selected over Player IO due to the availability of resources.

### C. System Design

Figure 3 depicts the basic system design consisting of the Photon Unity Network (PUN) servers and multiple users supported simultaneously as individual users on the right, multiple users in the same physical environment, and multiple users across multiple physical environments. The glasses in the

single user block illustrates the potential for a 3D HMD formfactor.

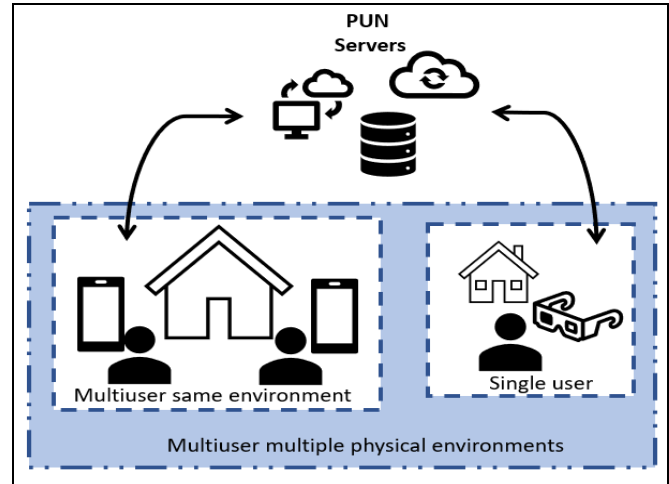


Figure 3. System Design.

Each mobile device runs the application independently. The PUN servers provide a host platform to identify and manage information exchange between each participant based on the developer’s application information and user sessions.

### D. Application Flow

Figure 4 depicts the basic application flow. Multiuser applications have several aspects in common. In Figure 4, the nodes which are rectangles with rounded edges in the diagram represent major scenes. The arrows represent flows and communication. The light blue box shows processes managed via the Photon Unity Network server. The green boxes represent AR Rendering by the mobile device and the bright blue scenes are where major gameplay and interactions occur. This is where players are instantiated and rendered locally to the device.

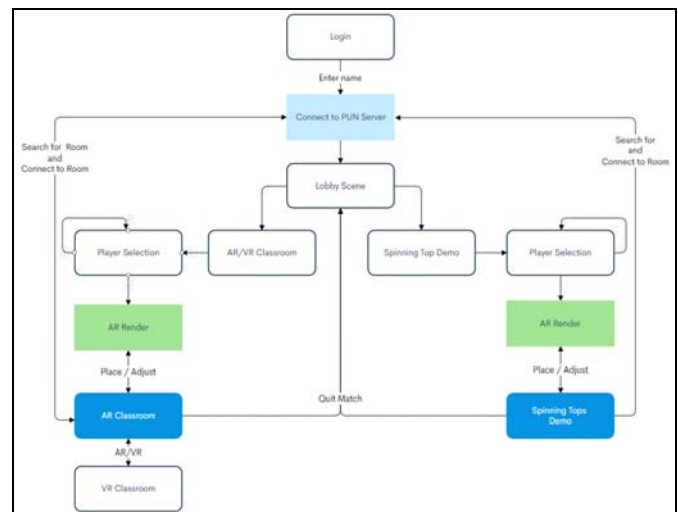


Figure 4. Multiuser Application Flowchart.

There is a common application flow used for the mobile multiuser AR/VR application. This involves establishing a session on the server and participant identification through login, character or player selection, placement and rendering of the AR scene, and gameplay. The following section provides a walkthrough and screenshots of the resulting application.

#### IV. RESULTS

##### A. Prototype Application

This application uses the Photon Unity Network (PUN) server and Unity's AR Foundation Kit with Android AR Core plugin.

Each user must connect to a common server which are identified by connection and Unique IDs [10]. This begins with a user login as shown in Figure 5. The login name is for the human users to know who-is-who across all the connected devices at play. Internally, the PUN Server creates a unique ID based on connection and login information[10].



Figure 5. Login.

After successful connection with the Master server, the application takes the user into the lobby as shown in Figure 6.



Figure 6. Mobile device screen capture of Lobby.

In this game, there are two options, AR/VR Classroom and the Spinning Top Demo. The application flow is similar whether the user selects AR VR classroom or the Spinning top demo. In both cases, there is a player selection.

The player selection screen is provided so that users can look at and choose between various characters. Figure 7 is a mobile phone screen capture of the player selection scene for the AR/VR classroom.



Figure 7. AR/VR Classroom User Selection.

Figure 8 is a mobile device screenshot of the player selection scene for the Spinning Tops Demo.



Figure 8. Player selection for Spinning Tops Demo.

Once a selection is made, the application goes into AR rendering created with AR Foundation and AR Core for Android. To establish environmental persistence, XR planes are used.

The mobile device detects a plane within the room and the user places the AR viewing item relative to the plane for object permanence in the environment. Figure 9 shows the plane detection and user placement of the battle arena in the Spinning Top Demo.

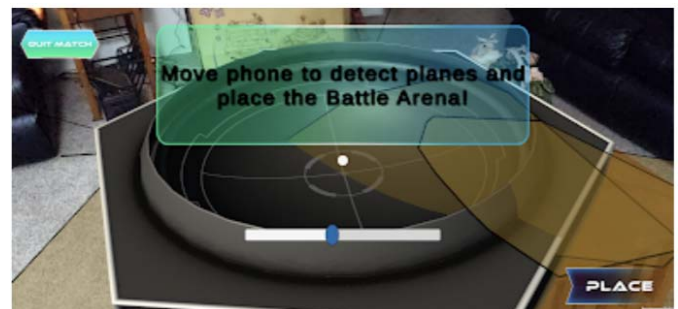
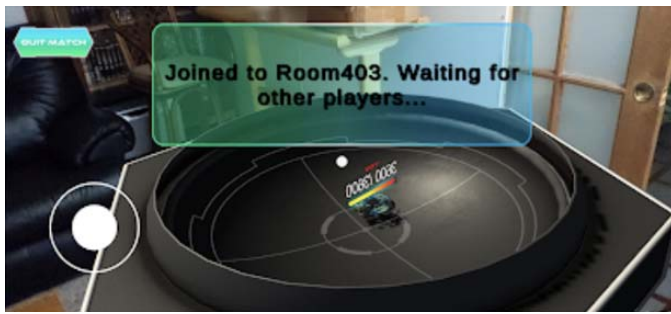


Figure 9. Spinning Top Demo plane detection and user placement of battle arena.

After placement, the user joins the application room on the PUN to join or await other players as shown in Figure 10.



**Figure 10. User joining PUN room.**

After at least two players join, the multiuser experience or game begins. Figure 11 shows five simultaneous players in the Spinning Top Demo.



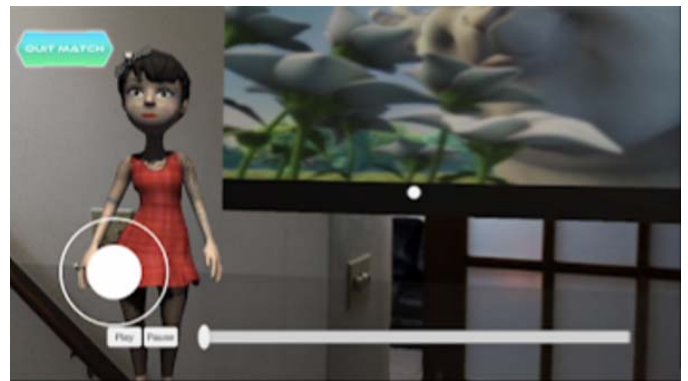
**Figure 11. Spinning Top Demo with multiple users.**

The Spinning Top Demo includes scoring, sound and visual effects, and spinning. The player with the faster velocity when striking another player reduces the hit points of the impacted player.

#### 1) AR/VR Classroom

The following section focuses on mobile device screen captures of the AR/VR Classroom.

Figure 12 depicts an emulated virtual instructor with a video playing as presentation material. In this scene, there are three main parts to the user interface (UI). The teal button in the upper left labeled “Quit Match” is clicked to exit the scene and return the user to the Lobby. The circle within a circle is a touchscreen joystick controller used for moving the user’s third person avatar. The third part of the UI is the pair of buttons next to a slider at the bottom of the screen. This is used for interactive control of the video. As the video is intended to serve as instructional material, these controls allow the user to play, pause, rewind, and fast-forward the video



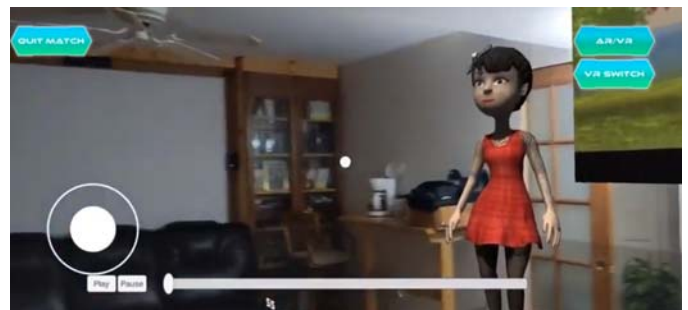
**Figure 12. Virtual instructor with video playing in AR classroom.**

Figure 13 shows multiple players simultaneously in the AR Classroom in third person perspective. The main player who controls the local device is labeled “You.” Player 1 and Player 4 are running on different devices. Player 4 and Player 1 have different AR environments.



**Figure 13. Multiple players entering AR Classroom.**

Users can switch seamlessly between AR and VR environments using the AR/VR button in the upper-right. Figure 14 shows the virtual instructor in the AR Classroom environment while Figure 15 shows the same environment in virtual reality.



**Figure 14. Virtual instructor in AR Classroom.**

This seamless transition, also known as mixed reality allows the application to maintain 3D object persistence between both

modes based on AR object placement and plane detection in the users environment.

Figure 15 depicts the VR analogue of the AR Classroom in Figure 14.



**Figure 15. Virtual instructor in VR Classroom analogue.**

The following figures illustrate the AR and VR analogue screen captures obtained when the user pans the room away from the instructor. Figure 16 shows two users in the AR Classroom.



**Figure 16. Two users in AR Classroom.**

Figure 17 shows the VR mixed reality analogue of Figure 16. Note that the positioning of the players, instructor, and viewscreen as seen in the AR environment is retained when the user transitions between AR and VR. Additional furniture and additional virtual character agents are also present in the VR scene.



**Figure 17. Two users in XR VR Classroom.**

Figure 18 shows the fully VR Classroom environment. In it, there is no relationship between stability and placement of the 3D objects when transitioning between AR and VR.



**Figure 18. Virtual instructor in VR Classroom.**

The transition between AR and VR was achieved in both a strictly AR to VR environment and an AR to mixed reality VR. In the mixed reality implementation, character avatar placement is appropriately mapped to the physical plane detected on the floor. However, the placement of the virtual instructor and screen needs to be adjusted to make this easier for the user. The AR Core adaptive lighting for the mixed reality VR needs to be adjusted. The adaptive lighting attempts to match the average lamination of the area. When the environment is rendered in VR, the result is too dark.

## V. CONCLUSION

The intent of this project was to create a consistent augmented presence where the people and objects can be stable when transitioning between the virtual and augmented modes. This application uses a common code baseline which facilitates the development of AR, VR, and XR analogue environments which can be used for future research.

There is a common application flow used for the mobile multiuser AR/VR application. This involves establishing a session on the server and participant identification through login, character selection, placement and rendering of the AR scene, and gameplay.

The project is mostly successful in that it achieves the implementation of having a seamless AR and VR transition for a virtual classroom within the same scene. However, the application requires a great deal of improvement from the perspective of human animation, lighting, and interaction.

This prototype demonstrates that multiuser AR and VR can be implemented with object stability between environments.

Several techniques were investigated for implementing multiuser AR including Vuforia and AR Foundation for augmented reality were both investigated and utilized during development. Photon Unity Network and Player IO were explored during development.

Immediate next steps include adding animations and behaviors to the agents and avatars, improve video player synchronization, and correcting AR lighting. With further refinements for controlling content, this can contribute to

developing an AR/VR platform to learn more about user interaction and learning in a mixed-reality environment.

#### ACKNOWLEDGMENTS

This paper was prepared as part of coursework for Bowie State University's Computer Science Department Advanced Virtual Reality Systems course under the direction and guidance of Professor Sharad Sharma, Ph.D.

#### REFERENCES

- [1] A. Arntz *et al.*, "Navigating a Heavy Industry Environment Using Augmented Reality - A Comparison of Two Indoor Navigation Designs," in *Virtual, Augmented and Mixed Reality. Industrial and Everyday Life Applications*, vol. 12191, J. Y. C. Chen and G. Fragomeni, Eds. Cham: Springer International Publishing, 2020, pp. 3–18.
- [2] J. Nielsen and R. Molich, "Heuristic evaluation of user interfaces," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, Mar. 1990, pp. 249–256, doi: 10.1145/97243.97281.
- [3] M. M. O. da Silva, R. A. Roberto, I. Radu, P. S. Cavalcante, and V. Teichrieb, "Why Don't We See More of Augmented Reality in Schools?," in *2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, Oct. 2019, pp. 138–143, doi: 10.1109/ISMAR-Adjunct.2019.00-61.
- [4] "pokemon go - There's a Pokémon in the kebab store! Can I turn off augmented reality?," *Arqade*. <https://gaming.stackexchange.com/questions/272643/there-s-a-pok%C3%A9mon-in-the-kebab-store-can-i-turn-off-augmented-reality> (accessed Dec. 15, 2020).
- [5] G. Gorisse, O. Christmann, E. A. Amato, and S. Richir, "First- and Third-Person Perspectives in Immersive Virtual Environments: Presence and Performance Analysis of Embodied Users," *Front. Robot. AI*, vol. 4, 2017, doi: 10.3389/frobt.2017.00033.
- [6] R. Singh, "An Overview of Android Operating System and Its Security Features," vol. 4, no. 2, p. 3, 2014.
- [7] "Getting Started with Vuforia Engine in Unity." <https://library.vuforia.com/articles/Training/getting-started-with-vuforia-in-unity.html> (accessed Oct. 27, 2020).
- [8] "AR Plane Manager | AR Foundation | 2.1.10." <https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@2.1/manual/plane-manager.html> (accessed Dec. 15, 2020).
- [9] "ARCore supported devices," *Google Developers*. <https://developers.google.com/ar/discover/supported-devices> (accessed Dec. 15, 2020).
- [10] "Glossary | Photon Engine." <https://doc.photonengine.com/en-US/realtime/current/reference/glossary> (accessed Dec. 06, 2020).