PAPER • OPEN ACCESS

A mobile game SDK for remote collaborative between two users in augmented and virtual reality

To cite this article: Y H Ong *et al* 2020 *IOP Conf. Ser.: Mater. Sci. Eng.* **979** 012003

IOP Conf. Series: Materials Science and Engineering **979** (2020) 012003 doi:10.1088/1757-899X/979/1/012003

A mobile game SDK for remote collaborative between two users in augmented and virtual reality

Y H Ong [1]*, A W Ismail [1], N A.Iahad [1], M R M Rameli [2], and R Dollah [2]

[1] School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, 81310 Johor, Malaysia
[2] School of Education, Faculty of Social Sciences and Humanities, Universiti Teknologi Malaysia, 81310 Johor, Malaysia

* ongyuehuei@gmail.com

Abstract. Remote collaboration is important in Augmented Reality (AR) and Virtual Reality (VR) that allow users to communicate and interact with each other. Mobile games are gaining its popularity due to its portability and lower in cost. Software Development Kit (SDK) allows developer to speed up the development process. However, most of the collaboration of these technologies only focus on either AR or VR independently. Besides, most AR and VR that had been developed are only available in expensive devices but not in mobile devices. Therefore, this research aims to create a remote collaboration between AR and VR with mobile games. There are four phases have been carried out, analysis phases, development phase, construction phase and evaluation phase. The evaluation has been performed and it is based on the usability and user acceptance. The usability results show the application is accepted by the user. User acceptance results show the application works accordingly. Based on the results, this research has successfully produced a working SDK for remote collaboration between AR and VR with mobile games.

1. Introduction

Augmented Reality (AR) and Virtual Reality (VR) technology are unique and powerful by itself. AR enhances the real world environment by sharing real and virtual world at the same time [1]. VR entirely replaces the real world by simulating a virtual environment [2]. Collaboration between AR and VR becomes an important aspect to exploit both technologies to the strength [3]. Development of technology and the decreasing market price of mobile devices added an advantage for these technologies to be supported in mobile devices [4]. Developing mobile game with technologies like AR and VR consumes time and required specific knowledge. The lack of interdisciplinary ideas of developing game, AR and VR increase the demand of related Software Development Kit (SDK) during implementation and design level [5]. SDK allows the developer to quickly develops the applications with ease [6]. Creating a SDK for remote collaboration between AR and VR with mobile games has several advantages: (1) leverage the advantages of both AR and VR together, (2) allow users to experience AR and VR in the mobile games with affordable price, and (3) able to develop AR and VR mobile games without programming knowledge with ease. Based on the current existing SDK has been produced for Unity, it does speed up the development process. The application such as mobile games are limited to a single platform such as AR alone and VR alone due to the hardware restriction and lack of collaborative plugin or modules. The collaboration of AR and VR technologies only focus on either AR or VR independently. Most AR or VR have been developed to only available with the particular and specific devices but very less in mobile devices. Therefore, this research aims to provide a SDK to enable a remote collaboration between AR and VR with mobile games.

2. Literature review

2.1. Augmented reality

Three main characteristics of AR [7] are: (1) combines both real and virtual entity, (2) interactive in real time, and (3) must be registered in 3-Dimensional (3D). AR superimposed virtual objects into the real world. Besides images, many other computer-generated information can be superimposed too [1]. There are several tracking techniques used in AR such as image feature detection, plane detection and object recognition. The development of AR in mobile devices continue to grow and received numbers of investments from top technology companies like Google, and Facebook. Some AR SDK that are compatible with the Unity3D game engine are Google ARCore, Vuforia, Wikitude, EasyAR, ARToolKit and ARKit. In this research, we implement the Vuforia SDK as it provides the natural feature tracking [8] that align with the tracking technique implemented in this research and it is compatible to works with mobile phone. Table 1 shows the comparison of AR SDKs.

Table 1. Comparison of AR SDKs.

	Vuforia	Wikitude	EasyAR	Kudan	ARToolKit	Maxst	Apple ARKit	XZIMG
Licence	Free, Commercial	Commercial	Free, Commercial	Free, Commercial	Free Open Source	Free, Commercial	Free	Free, Commercial
Supported platforms	Android, iOS, UWP	Android, iOS	Android, iOS, UWP, macOS	Android, iOS	Android, iOS, Linux, Windows, macOS	Android, iOS, Windows, macOS	iOS	Android, iOS, Windows
Smart glasses support	+	+	-	-	+	+	+	-
Unity support	+	+	+	+	+	+	+	+
Cloud recognition	+	+	+	-	-	-	+	-
3D recognition	+	+	+	+	-	+	+	-
Geolocation	+	+	-	-	+	-	+	-
SLAM	-	+	+	+	-	+	+	-

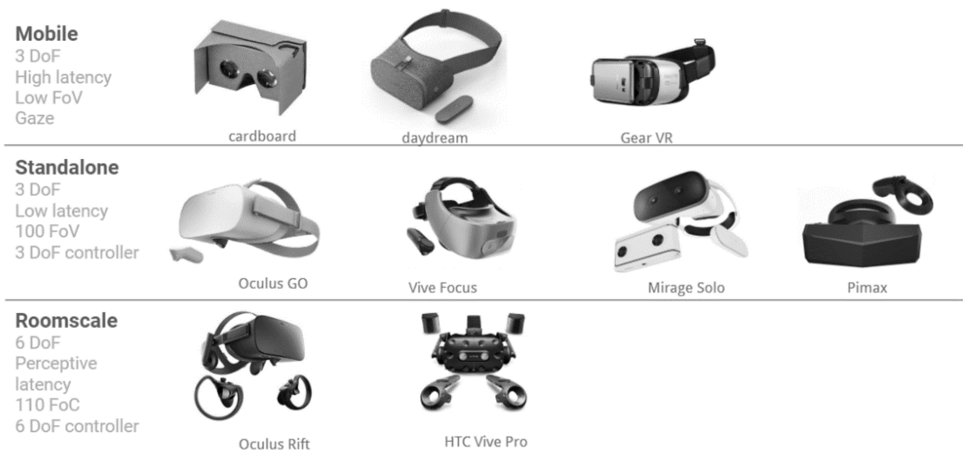
2.2. Virtual reality

There are four major key elements [9] that should be included in VR experience: (1) experience of being in the virtual world, (2) sense of feeling immersion, (3) receiving sensory feedback from the user input, and (4) interactivity. VR convince a human's brain about the existence of the computer-generated world. It combines the display technology with a 3D model in real-time providing an interactive environment [10]. There are several popular VR headsets, such as Google Cardboard, VR box, Oculus Go and HTC Vive. The built-in accelerometer in the mobile devices is often used to detect the head-position for generating movement or interact in the virtual world [11]. VR in the mobile device with headset such as Google Cardboard that is made with inexpensive materials and cable-less has gained its popularity among the consumers due to its affordable and portable characteristic [12]. Some VR SDK that are

IOP Conf. Series: Materials Science and Engineering **979** (2020) 012003 doi:10.1088/1757-899X/979/1/012003

compatible with Unity3D game engine are Google Cardboard VR SDK, OpenVR SDK, SteamVR SDK and Oculus Mobile SDK. Table 2 shows the typical VR gears consist of mobile, standalone, and roomscale. Mobile and standalone have a boundary limits to 3 degree of freedom (DoF). While roomscale is enable 6 DoF. In this research, we use cardboard with high latency and low field of view (FoV) due to the gears is low in cost and its mobility to work with mobile device that comes with the built-in accelerometer that can be used to detect user head-position in the virtual environment [11].

Table 2. Typical VR gears.



2.3. Remote collaboration
The collaboration between the users can be achieved through: (1) face-to-face collaboration or (2) remote collaboration. All the VR users are required to wear a HMD that fully hide the vision of the real world's environment. Hence, the face-to-face collaboration is unachievable where the fusion of AR and VR can only be achieved through remote collaboration [13]. The remote collaboration between AR and VR refers to a mixed platform collaboration that enables multi-users with a different interface, either AR or VR interfaces to interact together when the users are located at different location [14]. The major goals of the collaboration in AR and VR are to change the human's sense allowing users to feel co-present through immersion of 3D environment while collaborating with each other [3]. The remote collaboration system of VR and AR (CoVAR) is discussed in [14] and [15].

The networking system connects the users via the network in a collaborative interface. The most renowned networking systems in the Unity3D are: (1) Photon Unity Networking (PUN) and (2) Unity Network (UNET). However, UNET may induces a higher latency in transmitting networked data, increase the device's workload hence reduce the performance in compares to PUN which might cause registration and interaction error in the remote collaboration [18].

2.4. Software development kit (SDK)
The SDK contains components, tools or functions that can be used to build an entirely new program, system or application. A successful SDK can reduce the developing time and eases in the creation process [19]. Each SDK has its own license type, targeted platforms, programming languages and related features and functionalities [6].

3. Methodology

After the literature review has been completed, the waterfall model is used to actualize this work, the following subsections explained about the methodology.

3.1. Analyze remote collaboration between AR and VR
The first phase focuses on the preliminary analysis of the remote collaboration in AR and VR. The analyses are conducted to have a better understanding of the whole methodology. The comparison

between the related tools has been made to ease in choosing the suitable tool during the prototyping in the later phase. Some existing projects have been studied to gain a better understanding of this work and the experiments. Several suitable options such as AR SDK and tracking technique, VR SDK and the headset, the networking service and the development platform are discussed. Besides, the gameplay for both card matching game and chess game are also explored to ease in scripting the game logic.

3.2. Develop the remote collaborative AR and VR mobile games

Based on the result analyzed in Phase 1, the suitable AR SDK, VR SDK and collaborative technology are determined in this phase. The Vuforia AR SDK is used to developed the AR interface. The image feature detection provided in the Vuforia is used for rendering the game in AR. Camera in mobile device constantly detects the marker and the 3D virtual objects are rendered and superimposed onto the real world accordingly. Figure 1 shows the AR tracking process.

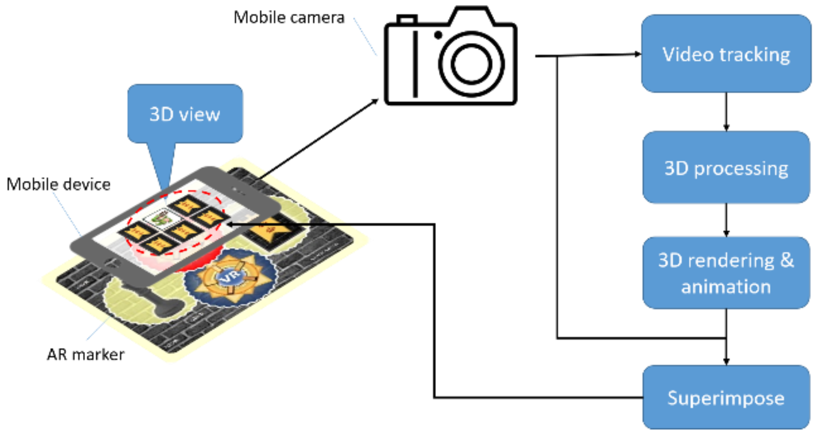


Figure 1. AR tracking process.

Google Cardboard VR SDK is used to developed the VR interface. The mobile device with gyroscope traces the x, y and z rotation and angular velocity. The VR system then match the in-game camera to follow the mobile's gyroscope and ray selection is used to select virtual objects in the virtual environment.

A mobile device is placed in the VR box to experience the 3D view through the mobile device's screen. Figure 2 shows the VR tracking process.

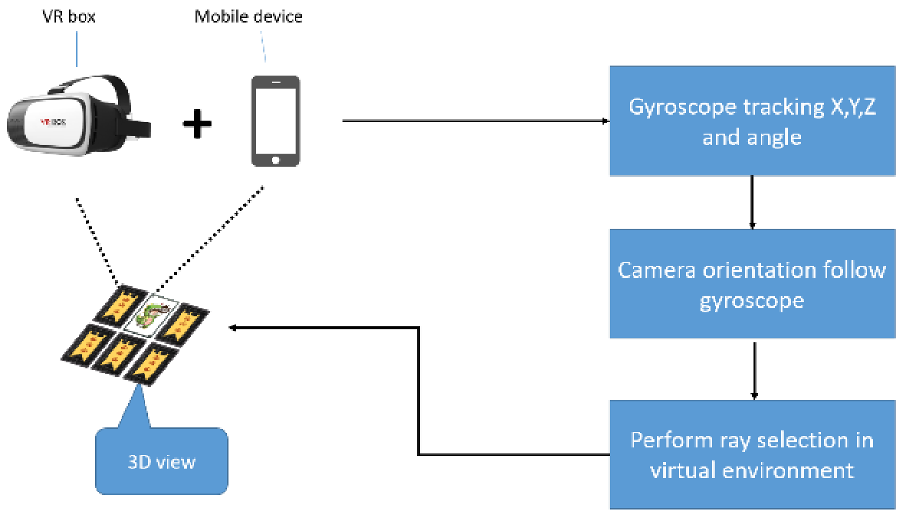


Figure 2. VR tracking process.

PUN is chosen to be the collaborative interface to connect both AR and VR users remotely. The algorithm for the collaboration between AR and VR are scripted such that the users from both AR and VR interface are able to take turns to interact with the same virtual object in real-time.

Figure 3 shows the system architecture. The whole methodology implementation is conducted as illustrated in the system architecture to actualized the remote collaboration between AR and VR.

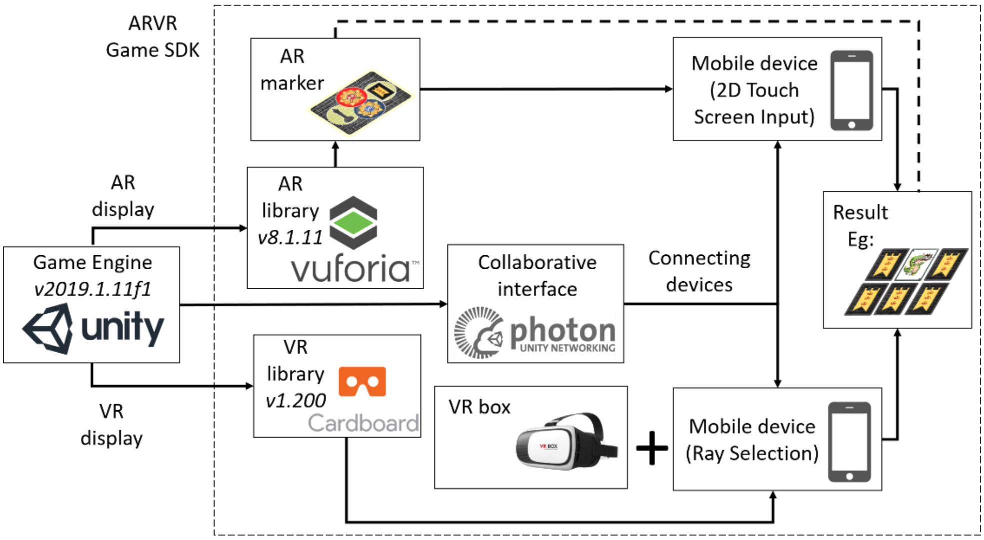


Figure 3. System architecture.

3.3. Construct the Mobile Game SDK with remote collaboration between AR and VR

The mobile games with remote collaboration between AR and VR that have been developed in the previous phase is constructed as SDK to achieve the research aim. Figure 4 shows the flow chart of the SDK.

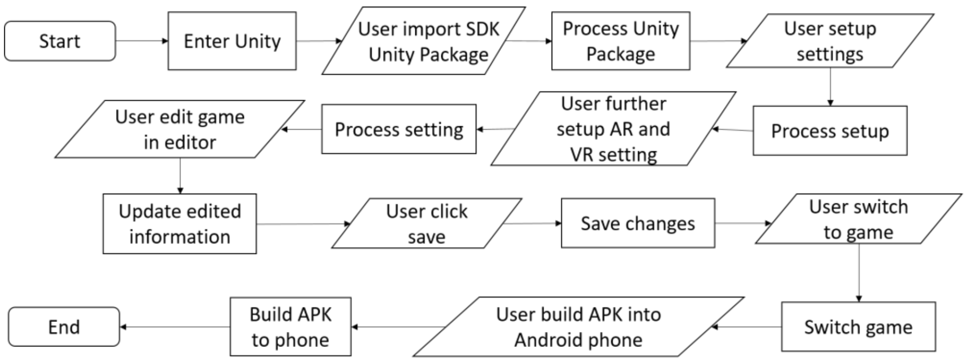


Figure 4. SDK flow chart.

The SDK is constructed such that the users can easily build their own collaborative AR and VR games into mobiles and play immediately without writing a single line of code. Besides, the users are allowed to change the desired texture or colour of the game object with ease

4. Implementation

The implementation of remote collaboration between augmented and virtual reality in mobile games is carried out according to the design as shown in the system architecture.

4.1. Develop card matching game

The card play is designed according to the traditional card game's concept and rules. There are a total of 3 pairs of cards facing down, that make it a total of 6 cards. A CardSpawner class acts as a controller to spawn all the 6 cards to the respective position. A GameController class acts as the card matching game controller. This class controls the game algorithms like determining cards' flipped, determining card matched and handling users' turns.

Figure 5 shows the gameplay flow chart for card matching game. Once the game start, the algorithm is scripted to spawn the card and set the user turn. The user then opens the first card and continues by opening the second card. If both cards opened by the user is the same, user gets the point and both cards are removed from the game. If both cads opened by the user happened to be different, both cards turn back faced down and it switches to the other user's turns. After the cards being removed, if there is no card left on the game, the game ends with the user with most points win. However, if there are cards left on the game, it continues by switching to the other user's turns.

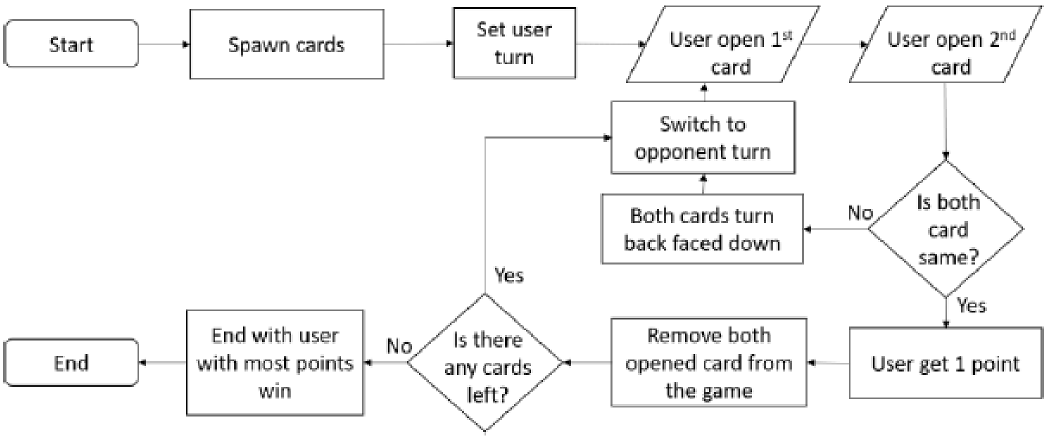


Figure 5. Card Matching game flow chart.

4.2. Fusion between AR and VR

PUN connects the users between AR and VR games. When running the application, a connection is sent to the Photon Server. A room is allocated in Photon Cloud for Master Client, the user who request to create a new room. The user that join the room later is known as the Client and is connected to the Photon Cloud server after joining the room. A real time synchronization of the gameplay's status is implemented to both users across the network by applying the Photon View component. Remote Procedure Calls (RPC) synchronized the function to be executed so that when a user performs an action that called a function, that particular function on the other user's side is called as well. This allows both AR and VR interface to be synchronized across the network via internet connection.

Figure 6 shows the fusion between AR and VR users. Both AR user and VR user are required to connect to the server before joining the room. After the game starts, AR user needs to scan the AR marker with the mobile device, while the VR user is required to wear a VR box with a mobile device placed inside to play the game.

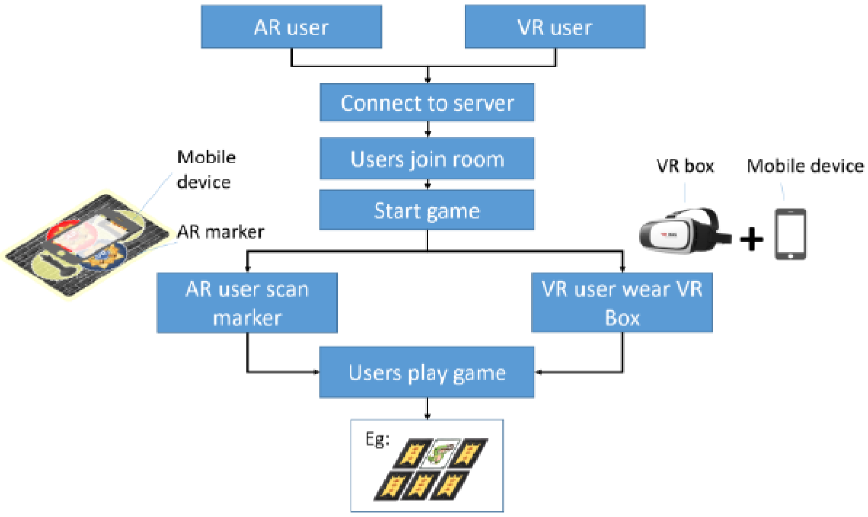


Figure 6. Fusion between AR and VR.

Figure 7 shows the collaboration between two users, the AR user and the VR user. The flow of the hardware and software works to enable the fusion of AR and VR is shown. In the hardware, VR box is attached with the mobile device to run VR application, while networking is required to connect VR user to the AR user that consists of a AR marker and a mobile device. For user interaction, gaze interaction is used for VR user and touch screen input is used for AR user.

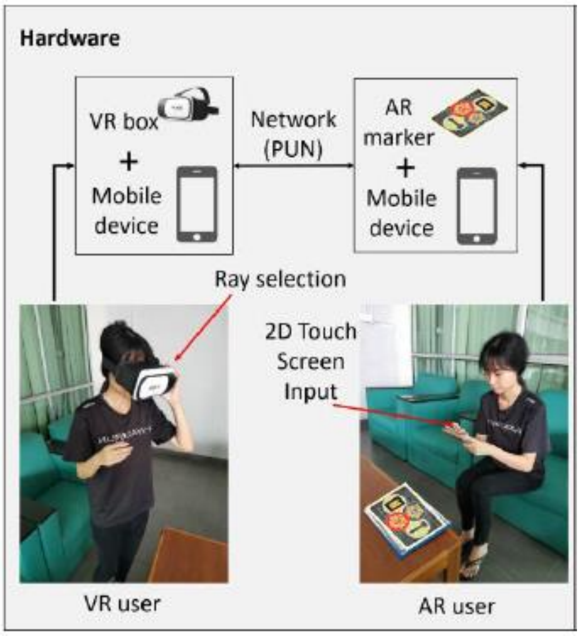


Figure 7. Collaborative between two users in AR and VR fusion.

4.3. User Interaction for AR and VR users

Multi-user interaction is defined as the action of more than one user able to interact concurrently in the environment. However, the basic multi-user interaction approach is incapable of distinguishing individual users, which leads to conflicts in collaborative tasks that regularly happen. Figure 8 illustrates the multi-user interaction workflow in the shared space context of integration between two different interface settings of AR space and VR space.

Multi-user interaction implements the client-server architecture to integrate AR and VR users in a shared space, where either the AR or VR user can be the host server. The shared space is independent of the system. Its position and rotation data are transmitted to one another between the AR or VR users via a network-enabled by internet protocol in real-time. Thus, the system eventually provides real-time 3D space interaction in VR and 2D touch input interaction for handheld AR applications. The integration phase is separated into two categories which are a server and a client. The server is a host computer that handles all connection requests from clients, maintains and manages the virtual content shared by users, and processes data storage and retrieval. The manipulated object properties are later updated through the network and are sent to the server where the shared space metaphor is situated. The virtual object properties consist of information that holds the transformation value of the virtual object and the boolean value that notify whether the object is currently manipulated by another user or not. The boolean value assists in organizing the interaction cue of a multi-user interaction in a collaborative system for the interaction to be smooth and sequential. The feedback received from the server as the new properties of the virtual object information is later updated and synchronizes on the client-side of the respective user spaces, either AR or VR.

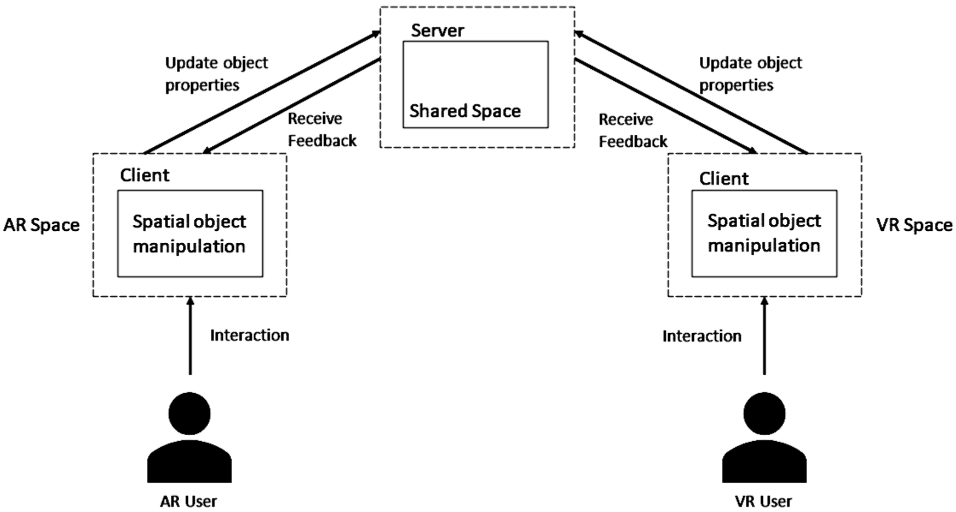


Figure. 8 Multi-user interaction workflow in the shared collaborative space.

Table 3. Interaction techniques for users in AR and VR fusion.

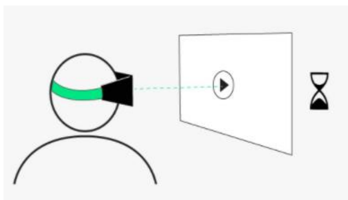
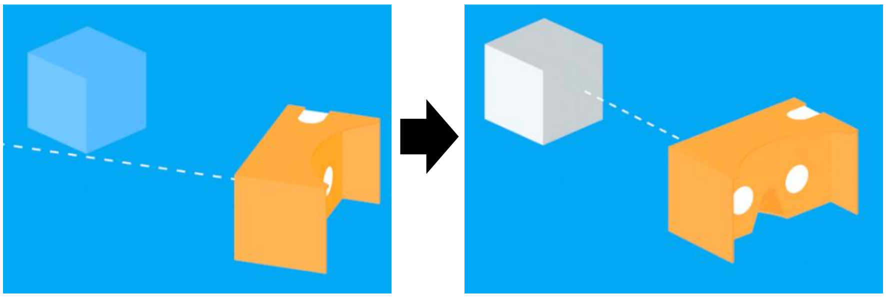
	AR	**VR**
Technique	Touchscreen interaction	Gaze interaction
Description	A touchscreen allows the user to interact by using their finger or stylus on a display device.	The gaze input uses a visual reticle to aim for user selection.
Input		

Table 3 shows the interaction techniques, when the user moves their head, a ray pointing was triggered the selectable object. The ray pointing was moving corresponding to user’s head movement. *PointerEnter* function was called and the gaze was loading to fill up clockwise loading to complete the timer. The fill-up speed was according to the time set in the *ARVRInput* class. *PointerExit* was triggered when the loading was completed within the time.

4.3.1. Gaze Input

This study uses a gaze input, where the study does not have to use a magnetic trigger. Magnetic trigger uses a magnetometer that built-in in the smartphones and also used for a compass. Gaze input is when the user just keeps looking for a certain time (in our tests, 1.2 seconds (+ 0.2 seconds for final decision) worked best) on a button and then it gets selected, as shown in Fig 9.

Figure 9 (a) shows the targeted object is not yet hit by the gaze input, Figure 9 (b) shows the user is able accurately target the objects by moving their gaze. This interaction uses a visual reticle where there is a ray pointing used as a visual aid to track the object. Displaying a reticle, user can do fine targeting tasks and can help them to aim. This method displays the reticle only when the user approaches a target that they can activate as presented in Figure 9 (b). It is obvious hover states when the user moves their head to aim the targeted object.



(a) The gaze does not look at the object (b) The gaze has triggered the object

Figure 9. Gaze input for user interaction in VR.

4.3.2. Touch Input

The mobile technologies nowadays typically allow a user to interact with a screen. Some technologies may work with only the user's finger, and other technologies may allow other tools, like a stylus. The mobile technologies have a capacitive touch screen which a screen that coated with a special material. When user touches a capacitive touch screen, there was a point of contact to indicate where the fingers have touched the screen.

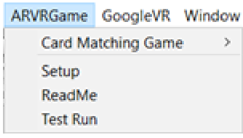
In this study, we use touchscreen interaction since it was common interaction used for user to interact with AR content. Tapping is a single touch or briefly tap on the screen surface with a finger. When compared to an old-fashioned computer, a tap is the same as clicking with a mouse.

5. Results

The result from the experiment is a working SDK that acts as a plugin to Unity. This SDK can be used to develop mobile games where the remote collaboration of AR and VR were involved in the game. The Android mobile games were the card matching game and the chess game. These games support multiplayer where the users of the game were a AR user and a VR user.

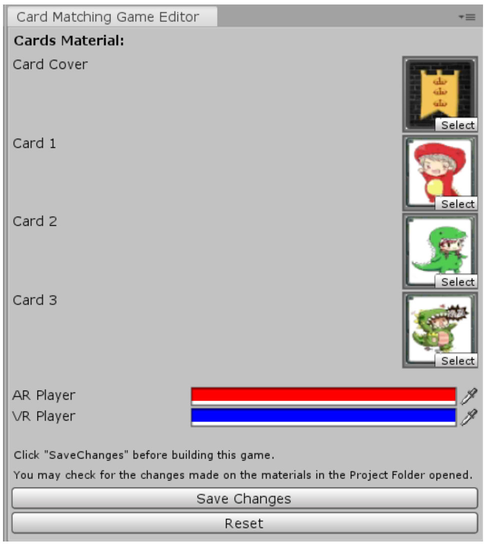
During the gameplay, the User Interface (UI) for both AR and VR users were different. For an AR user, the game objects only appeared when the marker was detected. For card matching game, all the six cards were laid facing down during the game start, while chess game has all chessman arranged accordingly on the chessboard. During the gameplay, AR user interacts with the game using touchscreen.

The SDK developed adds a new menu item named "*ARVRGame*" to the main menu in the Unity3D game engine as shown in Figure 10 (a), and Figure 10 (b) shows the materials that available. Figure 10 (c) shows the matching game editor when the user started to build the card matching game.



(a) SDKs in Unity IDE

(b) Materials for Matching Card



(c) Card matching game editor

Figure 10. Game SDK with remote collaboration AR and VR.

Figure 11 shows the AR matching game, and AR user's in-game decided to quit. The confirmation panel appears to ask the user whether want to choose cancel or quit. Figure 12 (a) shows the VR user's in-game UI for card matching game and Figure 11 (b) shows the VR user's in-game UI in card matching game when the quit button was being pressed.

Figure 11. Card matching game in AR.

(a) Matching card game in VR using Cardboard headset

(b) User chooses to quit; the confirmation panel appears.

Figure 12. Card matching in VR.

Acknowledgement

We would to express our appreciation to Mixed and Virtual Reality Laboratory (mivielab), ViCubelab at Universiti Teknologi Malaysia (UTM) for the facilities and technical supports. Special thanks to UTMHIR Grants (Q.J130000.2451.04G77) for the funding.

IOP Conf. Series: Materials Science and Engineering **979** (2020) 012003 doi:10.1088/1757-899X/979/1/012003

References

[1] M. N. A. Nor'a and A. W. Ismail, "Integrating Virtual Reality and Augmented Reality in a Collaborative User Interface," *Int. J. Innov. Comput.*, vol. 9, no. 2, pp. 59–64, 2019

[2] A. A. Oyelude, "Virtual reality (VR) and augmented reality (AR) in libraries and museums," 2018

[3] T. Piumsomboon, G. A. Lee, and M. Billinghurst, "Snow Dome : A Multi-Scale Interaction in Mixed Reality Remote Collaboration," no. May, 2018

[4] L. Muñoz-Saavedra, L. Miró-Amarante, and M. Domínguez-Morales, "Augmented and virtual reality evolution and future tendency," *Appl. Sci.*, vol. 10, no. 1, 2020

[5] S. Benbelkacem, N. Zenati-Henda, D. Aouam, Y. Izountar, and S. Otmane, "MVC-3DC: Software architecture model for designing collaborative augmented reality and virtual reality systems," *J. King Saud Univ. - Comput. Inf. Sci.*, 2019

[6] A. Hanafi, L. Elaachak, and M. Bouhorma, "A comparative study of augmented reality SDKs to develop an educational application in chemical field," *ACM Int. Conf. Proceeding Ser.*, vol. Part F1481, 2019

[7] R. T. Azuma, "A Survey of Augmented Reality," vol. 6, no. 4, pp. 355–385, 1997

[8] Ulfah, S., Ramdania, D. R., Fatoni, U., Mukhtar, K., Tajiri, H., & Sarbini, A. (2020, June). Augmented reality using Natural Feature Tracking (NFT) method for learning media of makharijul huruf. In *IOP Conference Series: Materials Science and Engineering* (Vol. 874, No. 1, p. 012019). IOP Publishing.

[9] W. R. Sherman and A. B. Craig, "Introduction to Virtual Reality," *Underst. Virtual Real.*, pp. 5–37, 2003

[10] P. Vichitvejpaisal and T. Chotined, "Virtual Reality Photo Hunt Game Application for Amblyopia Patient," *ICSEC 2017 - 21st Int. Comput. Sci. Eng. Conf. 2017, Proceeding*, vol. 6, pp. 57–60, 2018

[11] W. Hürst and M. Helder, "Mobile 3D graphics and virtual reality interaction," p. 1, 2011

[12] S. Yoo and C. Parker, "Controller-less Interaction Methods for Google Cardboard," no. August, pp. 127–127, 2015

[13] S. Ranade, M. Zhang, M. Al-sada, J. Urbani, and T. Nakajima, "Clash Tanks : An Investigation of Virtual and Augmented Reality Gaming Experience," pp. 0–5, 2017

[14] T. Piumsomboon, A. Dey, B. Ens, G. Lee, and M. Billinghurst, "CoVAR: Mixed-Platform Remote Collaborative Augmented and Virtual Realities System with Shared Collaboration Cues," in *Adjunct Proceedings of the 2017 IEEE International Symposium on Mixed and Augmented Reality, ISMAR-Adjunct 2017*, 2017, pp. 218–219

[15] T. Piumsomboon *et al.*, "Mini - Me : An Adaptive Avatar for Mixed Reality Remote Collaboration," *Proc. 2018 CHI Conf. Hum. Factors Comput. Syst. - CHI '18*, pp. 1–13, 2018

[16] A. A. Kubena, "Pexeso ('Concentration game') as an arbiter of bounded-rationality models," *28th Int. Conf. Math. Methods Econ. 2010, Pts I Ii*, pp. 377–380, 2010

[17] M. Diaz, M. Alencastre-Miranda, L. Munoz-Gomez, and I. Rudomin, "Multi-user Networked Interactive Augmented Reality Card Game," pp. 13–16, 2006

[18] C. S. Yusof, T. S. Low, A. W. Ismail, and M. S. Sunar, "Collaborative augmented reality for chess game in handheld devices," *2019 IEEE Conf. Graph. Media, GAME 2019*, pp. 32–37, 2019

[19] A. Kolsanov, A. Nazaryan, A. Ivaschenko, and A. Kuzmin, "Intelligent SDK for 3D surgery simulation," *Proc. 2016 6th Int. Conf. - Cloud Syst. Big Data Eng. Conflu. 2016*, pp. 384–387, 2016

[20] D. Mardanbegi and T. Pfeiffer, "EyeMRTk: A toolkit for developing eye gaze interactive applications in virtual and augmented reality," *Eye Track. Res. Appl. Symp.*, 2019.

[21] R. Van Herk, J. Verhaegh, and W. Fontijn, "Espranto SDK: An adaptive programming environment for tangible applications," *Conf. Hum. Factors Comput. Syst. - Proc.*, pp. 849–858, 2009

[22] D. Amin, & S. Govilkar, (2015). Comparative study of augmented reality SDKs. International *Journal on Computational Science & Applications*, 5(1), 11-26.