

# Implementation of Augmented Reality in Multiplayer Games

Andi Nur RACHMAN<sup>1</sup>, Muhammad Adi Khairul ANSHARY<sup>2\*</sup>, and Nazmi  
RIDWAN<sup>3</sup>

<sup>1</sup>*andy.rachman@unsil.ac.id Dept. Informatics Siliwangi University Tasikmalaya Indonesia*

<sup>2</sup>*adikhairul@unsil.ac.id Dept. Informatics Siliwangi University Tasikmalaya Indonesia*

<sup>3</sup>*nazmi.ridwan14@unsil.ac.id Dept. Informatics Siliwangi University Tasikmalaya Indonesia*

*\*Corresponding author: adikhairul@unsil.ac.id*

## ABSTRACT

This research develops augmented reality games by applying multiplayer features. Some previous research on augmented reality games has suggestions for features similar to multiplayer. And the weaknesses of some previous studies also exist in augmented reality which still requires technology-based tracking markers. The markers used must match the Markers available in the database game. For this reason, an augmented reality game needs to be made by adding multiplayer features and also applying marker less technology. This study succeeded in making a Multiplayer Action Game by utilizing Augmented Reality technology without using the luther-sutopo method. Based on the tests that have been carried out, alpha test results that contain functional tests are obtained and the results of beta testing with a questionnaire obtained 90,6% value which is declared "Good" then this game is feasible to use and support.

**Keywords:** *augmented reality, Action Game, marker less, multiplayer*

## INTRODUCTION

One type of multiplayer games is action games. Action Games Are a type of game with the main features in the form of many actions where players must have fast reaction skills to avoid enemies or avoid obstacles [1]. Players usually control characters in the form of protagonists. Action games based on shootings, for example, give players a lot of weapons that can be improved, while other games based on a fantasy world will provide swords and magical powers. When the game starts, he must still pay attention to the health and life of his character. Characters can usually receive several attacks, but if they receive too many attacks the character will lose his life. If the character's life is lost, the game ends.

As we know that multiplayer games lately are becoming a trend, for example we can see several types of MMOG (Massively multiplayer online games) such as PUBG, mobile legend, DOTA and others as its. Even the MMOG is currently included in national and international competitions familiar with eSport. Multiplayer games that are developing at this time, we know together that the arena of the game was formed artificially. Do not rule out the possibility, in the future to add a sense of reality in playing games, then the arena of play is made in a real environment.

To realize this, we need to implement Augmented Reality as a multimedia technology to multiplayer games. But in the application of this technology, there are several problems that must be solved, including (1) What kind of architecture program can meet the needs of Augmented Reality multiplayer games? And (2) What form of data communication must be prepared to build Augmented Reality Multiplayer games?

To strengthen the discussion, several related theories are presented in this study. Among them are Client-server program architecture and Augmented Reality technology. Client server network is a computer network architecture where the client requests data and a server that provides data responses to the request. The application of this network can support game development with the Real-time Multiplayer feature. Multiplayer is a type of game that can be played by more than one player in the same game arena and at the same time. In multiplayer games it allows each player to interact with other players, work together on the same team become a rival, to be able to provide a form of social communication that is hardly found in games with a single-player orientation [2].

Augmented Reality (AR) is an environment that inserts 3D virtual objects into real environments [3]. There are several types of Augmented Reality, one of which is Marker less Based Tracking. In implementation because markers are not prepared beforehand, while algorithms running in AR applications must recognize patterns, colors or other features that can be displayed on the camera frame. After the pattern is read, virtual objects can appear on the smartphone screen [4]. Although using a virtual environment created by computer graphics, the main arena of Augmented Reality is the real environment. Therefore, the projected object can be seen from various angles in the real environment. Augmented reality technology has been used in various fields. Among them are health education [5][6], industrial sector[7] and of course in the field of games [8][9].

## METHODOLOGY

The research method used in this study uses the following research:

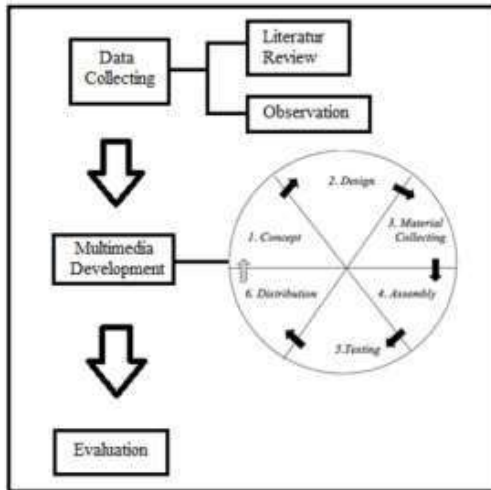


Figure 1 Research Methodology

**Data Collection**

Data collection aims to collect data and materials that support and are used in the research and manufacture of this game. In the process of collecting data using two stages. The first step is to collect related literature data.

Collect literature using literature studies and observations. Then the next data collection is the collection of material in the form of multimedia elements that will be used in making this game

**System Development**

The method used is Luther Sutopo's version, where there are six stages to making a multimedia product, namely: concept, design, collecting, assembly, testing and distribution materials. Sutopo adopted Luther's methodology with the scheme must begin with the concept and end with the distribution phase, while the collecting material can be done in parallel with the assembly stage [10].

- a. Concept, is the determination of the initial goal, the user's target, the type of product, the limitation of the function and the final target. In this concept stage the determination of goals and final targets influences the nuances of multimedia as a reflection of organizational identity.

Table 1. Concept of the Application

Title	One on One AR
Theme	Action
Audience	General
Duration	Unlimited
Animation	3D Character (*.FBX)
Image	Picture, Button, Icon (*.JPG dan *.PNG)
User	2 User
Interactivity	Button, Virtual Joystick
Description Game	This game features battles between 2 players. Each player uses each character, where each character has a life and weapon. If the weapon is about the player's character then the life will be reduced, and if the life runs out then the game is finished.

The rules of the game are arranged at this stage. The rule of the game is that each player must be able to approach and bring the weapons held close to other players. The player who is approached by an opposing weapon, his health status will decrease. The player whose health status runs out is declared to be lost, and the player whose health status remains is declared the winner.

- b. Design, is the stage of making specifications about product architecture, interface, design and supporting products. Specifications are made as detailed and detailed as possible because they will affect the next stage.

This stage uses storyboards to describe events and storylines from the construction of products that are built. As well as the navigation structure as a navigation or direction of information, making it easy for users to use.

Table 2. Storyboard

Scene	Display
1	Login
2	Main Menu
3	Lobby
4	Game

- c. Material Collecting, where the material or material needed for this game is collected such as text, images, audio and animation.

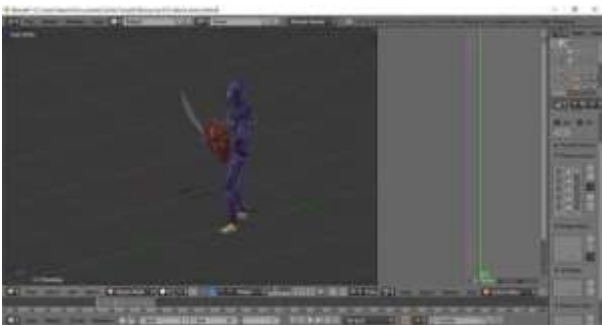


Figure 2 3D Animation Character

Figure 2 available on the internet to be used as the main characters of each player and also as enemies in the game. Collecting several images as an Interactivity button between the player and the system, each button has different functions such as a button to exit, a button to enter the game and an action button. The following picture for the button that has been collected.



Figure 3 Interactivity Button

The making of program code is made using the C # programming language, the process of writing program code is done on visual studio in this application also added code from the library of vuforia packages and asset programs from unity asset store.

- d. Assembly.

At this stage, the first thing to do in making this Multiplayer Action Game is to create a game scene. In its implementation, making this game uses Unity using C # language. The stages that are built are:

1. Determine the player's identity

The player's identity is obtained from the Google Play id. Each player must log in using the Google Play id so that it can be properly identified.

2. Determine which players to host the game and make the game lobby.  
The player who is first identified will be set as the host and automatically creates a game lobby.
3. Other players can find the lobby that has been created.  
The next player must be able to find and join the lobby that has been created.
4. Send and receive player position, rotation and health data.

When the game is in progress, the application must be able to send and receive data related to the player's position, rotation and health status.

- e. Testing, this stage is done after the manufacturing stage by running the game then seeing if there are errors or deficiencies in this game.

- f. Distribution, in this last stage, all stages have been completed in a structured manner according to the development method studied so the game is ready to be published through the Google Play Store.

**Evaluation**

The stages are carried out after all the process of making the application has been successfully carried out including the drawing of conclusions and the idea of development.

**RESULTS AND DISCUSSION**

In developing AR-based multiplayer game applications, several features are needed so that the application can run as expected. The most important thing is how the application can send and receive some data in real time. The data sent is the player's position, player rotation, player animation and player health status.

Player position and rotation data is sent when changes occur, sent from host to client or from client to host. Position data sent is the last coordinate point of the player. Similar to player rotation data, data is sent in the form of the degree of rotation and direction of rotation of the

player. For player animations, the data sent is in the form of changes in the player's movements. For the player's health status the data sent is a size from the blood bar

image. The worse the player's health status, the smaller the size of the image from the blood bar.

```

String data ="Position:" + player.position.x + ":"
+ player.position.y + ":" + player.position.z; byte[]
bytedata = System.Text.ASCIIEncoding.Default.GetBytes(data);
PlayGamesPlatform.Instance.RealTime.SendMessageToAll(reliability, bytedata);

string datarot = "Rotation:" + player.eulerAngles.x
+ ":" + player.eulerAngles.y + ":" +
player.eulerAngles.z;
byte[] bytedatarot =
System.Text.ASCIIEncoding.Default.GetBytes(datarot)
;
PlayGamesPlatform.Instance.RealTime.SendMessageToAll(reliability, bytedatarot);
}
}
if (!player.gameObject.activeSelf)
{ menang.SetActive(false); kalah.SetActive(true);
string lost = "Lost";
byte[] bytelost =
System.Text.ASCIIEncoding.Default.GetBytes(lost);
PlayGamesPlatform.Instance.RealTime.SendMessageToAll(true, bytelost);
}

```

Figure 4 Implementation of Send Position and Rotation player data

```

string data = "Position:" + player.position.x + ":"
+ player.position.y + ":" + player.position.z; byte bytedata
System.Text.ASCIIEncoding.Default.GetBytes(data);

PlayGamesPlatform.Instance.RealTime.SendMessageToAll(reliability, bytedata);

string datarot = "Rotation:" + player.eulerAngles.x
+ ":" + player.eulerAngles.y + ":" +
player.eulerAngles.z;
byte[] bytedatarot
System.Text.ASCIIEncoding.Default.GetBytes(datarot)
;
PlayGamesPlatform.Instance.RealTime.SendMessageToAll(reliability, bytedatarot);
}
}
if (!player.gameObject.activeSelf)
{
menang.SetActive(false); kalah.SetActive(true);
string lost = "Lost";
byte[] bytelost
System.Text.ASCIIEncoding.Default.GetBytes(lost);
PlayGamesPlatform.Instance.RealTime.SendMessageToAll(true, bytelost);
}

```

**Figure 5** Implementation of Receive Position and Rotation player data

In this way, this AR based multiplayer game application can run. However, there are some things that need to be improved further as follows. Data exchanges that occur have a fairly frequent intensity. For the stability of this game, a very stable internet connection is required. The data sent has a size that is not large, but because of its intensity quite often, if accumulated the data size can be large. In the current conditions, the game is limited to a maximum of 2 players. But in the future, it can be developed into more players. Can be imagined if in the future the player has added more than 2, then the intensity of data exchange will be even greater.

This application architecture even though the client server looks like a point to point connection. This is because the players are still limited to a maximum of 2 players. In addition, this application also does not fix a server that functions as a host. In this application, the player who first made the lobby, then the player will be set as a host.

## CONCLUSION

This research was building the multiplayer game with augmented reality and marker less technology. This application can run by send and receive data player's position, player rotation, player animation and player health status in real time. Position data sent is the last coordinate point of the player. Similar to player rotation data, data is sent in the form of the degree of rotation and direction of rotation of the player. For player animations, the data sent is in the form of changes in the player's movements. For the player's health status, the data sent is a size from the blood bar image. Data exchanges that occur have a fairly frequent intensity. For the stability of

this game, a very stable internet connection is required. The data sent has a size that is not large, but because of its intensity quite often, if accumulated the data size can be large.

## REFERENCES

- [1] George, Louis. "Designing the North Sulawesi Traditional Games Augmented Reality Game Application Dodorobe". Universitas Sam Ratulangi, 2017
- [2] Shobana,. Augmented Reality Multiplayer Gaming. Loyola ICAM College of Engineering and Technology, 2017.
- [3] Azuma, R., A survey of augmented reality. Presence: Teleoperators and Virtual Environments, 6(4), p.355-385, 2013
- [4] Lee, Youngo, Jongmyong choi. "Tideland Animal AR:Superimposing 3D Animal Models to user Defined Targets for Augmented Reality Game", 2014.
- [5] Zhu E, Hadadgar A, Masiello I, Zary N. (2014) Augmented reality in healthcareeducation: an integrative review. PeerJ2:e469<https://doi.org/10.7717/peerj.469>
- [6] Calle-Bustos A-M, JuanM-C, Garcí a-Garcí a I, AbadF (2017)An augmentedreality game to support

therapeutic education for children with diabetes. PLoS ONE 12(9):e0184645.

[7] Nirit Gavish, Teresa Gutierrez, Sabine Webel, Jorge Rodriguez and Franco Tecchia. Design Guidelines for the Development of Virtual Reality and Augmented Reality Training Systems for Maintenance and Assembly Tasks BIO Web of Conferences, 1 (2011) 00029. DOI: <https://doi.org/10.1051/bioconf/20110100029>

[8] Liberati, N. Phenomenology, Pokémon Go, and Other Augmented Reality Games, A Study of a Life Among Digital Objects. Hum Stud (2018) 41: 211. <https://doi.org/10.1007/s10746-017-9450-8>

[9] Annetta L, Burton EP, Frazier W, Cheng R, Chmiel M. Augmented reality games: using technology on a budget. Science Scope. 2012;36(3):54-60. <https://search.proquest.com/docview/1114793801?accountid=38628>.

[10] Sutopo, Ariesto Hadi. Interactive Multimedia with Flash. Graha Ilmu. Yogyakarta, 2003.