

Simple AR Basketball For Android Devices

Nur Hadisukmana*, Dias Teddo Wicaksono

Information Technology Study Program, President University

*Corresponding author: nurhadisukmana@gmail.com

Abstract — These days many people doing a lot of activities. Entertainment is one solution to reduce stress from those activities. Game is one of entertainment type, most of people playing games every day. People in these days using their smartphone to do a lot of things, smartphone becomes most important thing for people handles their life, people bring it every day and everywhere. Now, many game developer develop game on smartphone, most of them develop the 2D game or 3D game. This research will develop an casual game application on android device, combine 3D environment and Augmented Reality (AR) environment.

I. INTRODUCTION

Nowadays many people have a lot of busyness. Some people reduce their stress by listen music or watching movies. Games could became one of many entertainment that help people to reduce their stress. In recent years Game have many types, such as education, puzzle, adventure, action, casual, etc. Casual game is most easiest game to play, casual game require no long-term commitment or special skills to play, a game that can be played in short session. This research will develop an casual game application on android device, combine 3D environment and Augmented Reality (AR) environment. The development of this application aims to help people reduce their stress every day and everywhere. Meanwhile, this application also aims to people feel new user experience by playing casual game in Augmented Reality (AR) environment. This research focus on android mobile apps that is played by one player only, provide 3 levels. The 3 levels have different difficulty for each. The features of this application, such as main menu, turn off/on sound, and pause game.

II. METHOD

To enhance the development of this research, the proposed model that is used to design the application is Waterfall model. There are five phases of six phases in this model to examine and improve the application.

2.1 Requirement Phase

This phase's purpose are to define and collecting every related data to the application. Finding related source about game engine, script language, software development tools including for Augmented Reality (AR), and object editor.

2.2 Analysis Phase

The following phase is analyzing the gathered data from previous phase in order to properly generate what the

application must do and characteristic the application must have.

2.3 Design Phase

This phase largely covers technical design requirements, such as programming language. Also, logical design and physical design must be created here in order to achieve this.

2.4 Coding Phase

In this phase is the actual source code is finally written. Implementing all models and service integrations that were specified in the prior phases. This phase will be discussed more detail in next chapter.

2.5 Testing Phase

The following phase covers system testing and report issues within the application that need to be resolved. The details for this phase will be discussed on chapter 5.

III. IMPLEMENTATION

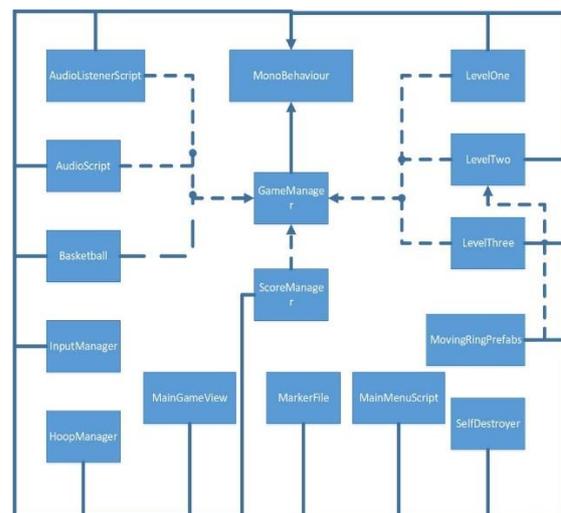


Figure 3.1 Class Diagram of Simple AR Basket Ball

After gathered, analyzed, and designed the data and sources. The data and sources need to implement into source code. The figure 4.1 show classes in this application.

3.1 Throw the Basketball

How the basketball can drifting towards the ring? This funtion handled by InputManager class. The class give the basketball power to drifting towards the ring using *void GetForce()* method, and the rotation of the basketball when it's drifting towards the ring using *rg.AddRelativeTorque*.

3.2 Ring Distance

This function handled by HoopManager classes. The class give minimum distance between the smartphone and the marker using *void Update()*, so the player have ideal distance to play the application.

3.3 Level Manager

There is no level selection on this application, the level automatically move to the next level after the player finish the current level. GameManager class, handles this level manager function using *IEnumerator StartLevel (int value)* method.

3.4 Score and Ring Collision

To validate target score and how the application counted the basketball that put into the basket? All handled by ScoreManager, this class define the target score to continue to the next level and the collision between the ring and basketball that will be counted as a score using *void OnTriggerEnter(Collider col)* method.

3.5 Count Down Time

LevelOne class handles the time for level 1 and also two other levels using *void Update()* method. The difficulty for level 1 is only count down time, player must achieved target score before the time's up to continue to the next level.

3.6 Movement Ring

Movement ring is a function to move ring from side to side, it's the difficulty for level 2 and handled by MoveRingPrefabs class.

3.7 Wind

The following function is to give wind on level 3 as the difficulty, it's randomized the power and the direction of the wind. It's handled by LevelThree class.

3.8 Turn On/Off Sounds

To handles turn on/off sounds of this application, the *AudioListenerScript* class using *void OnChangedToggle()* method, to change the volume value from default 1 to 0 or vice versa.

3.8 Exit the Application

MainMenu class handles this function using *void QuitGame()* method. Inside this method there is a unity method to terminated the application.



Figure 4.1 User Interface

Those figures are the results of the system testing, by doing system testing, the application can achieve the expected results. In Figure 4.1, it's testing for Main Menu scene, the test are by taps all of the buttons on Main Menu, such as Sound button, How to Play button, Exit button and Play button. Next Figure, which is Level 1 Scene, the test are, swipe up the ball, to see whether the basketball drifting towards the ring or not, also, if the basketball hit the board or ring, is it bounce back or not, and taps on Pause button including Pause panel, after taps on Pause button. Figure 4.1 is the Level 2, the test are not quite different as Level 1, the different is just about movement ring, and test is to check after the application enter the level 2, is the ring move side to side or not. As well as Figure 4.1, the Level 3 test is to check whether the wind working or not.

In Figure 4.1 also shown the condition after the player complete the current level, the test is to check after taps the Proceed button, is the scene automatically move to next level show the ring on marker for the next level. Furthermore in Figure 4.1 is Game Over Scene, after the player failed to complete current level, Game Over Scene will be displayed, the test is taps on Restart button will play the application again or not.

Last figure in Figure 4.1 displayed Win Scene. After the player win the game, Win Scene will be displayed, on this scene there is a button, Back to Menu button, the test is after taps on the button is it going to Main Menu or not.

IV. EXPERIMENTAL RESULT



V. CONCLUSIONS

Simple AR Basketball application has managed to achieve the objectives of the research, such as developing a 3D casual game by using Unity 3D and C# as the programming language which successfully run on mobile

devices with minimal Android OS version 6.0 or higher installed and also implement successfully an interesting gameplay on the touch screen devices which could be play anywhere. Even though the application is purposed to teenager. But also everyone could play this application. Runnable on Android devices makes this application is easy to install in entire platform that contain Android OS inside, because most of portable devices contain Android OS inside.

References

- [1] Blackman, Sue (2014). Unity for Absolute Beginner. Apress
- [2] Casual Games. From URL: <https://www.engadget.com/2009/04/29/what-defines-a-casual-game/>. Accessed on March 7th, 2018
- [3] Comparison Table Between Android. From URL: <http://socialcompare.com/en/comparison/android-versions-comparison>. Accessed on March 10th, 2018
- [4] Definiton of Android. From URL: [http://en.wikipedia.org/wiki/Android_\(operating_system\)](http://en.wikipedia.org/wiki/Android_(operating_system)). Accessed on March 10th, 2018
- [5] Definition of Augmented Reality. From URL : https://en.wikipedia.org/wiki/Augmented_reality. Accessed on February 27th, 2018
- [6] Jerry Gao; H.-S. J. Tsao; Ye Wu (2003). Testing and Quality Assurance for Component-based Software. Artech House
- [7] Definition of Video Game. From URL: <http://www.contempaesthetics.org/newvolume/pages/article.php?articleID=492>. Accessed on February 24th, 2018
- [8] History of Video Games. From URL: http://en.wikipedia.org/wiki/History_of_video_games. Accessed on March 11th, 2018
- [9] MonoBehaviour Scripting. From URL: <http://www.dannygoodayle.com/2013/06/11/unity-monobehaviours-c/>. Accessed on March 11th, 2018
- [10] Norton, Terry (2013). Learning C# by Developing Games with Unity 3D. PACKT Publishing.
- [11] Paper Toss Image. www.techhew.com/mobile-game-paper-toss/. Accessed on March 13th, 2018
- [12] Use Case Definiton. From URL: <http://www.uml-diagrams.org/use-case-diagrams.html>. Accessed on March 23rd, 2018
- [13] Waterfall Method. From URL: <https://economictimes.indiatimes.com/definition/waterfall-model/>. Accessed on April 16th, 2018