

Research Using The Google Vr Library To Build Virtual Reality Games On Mobile Platforms

Do Quoc Bao

Dong Nai Technology University, Dong Nai 76000, Vietnam



Abstract – The project "Research using the Google VR library to build virtual reality games on mobile platforms" was conducted by studying the application of the Google VR open source library, combined with the use of Mixamo Fuse graphics software to analyze, design project structure and program, design games according to the built structure. The application of the Google VR library to build virtual reality games on mobile contributes to further popularizing this technology to many people who do not have high conditions; enhance interactivity through user actions to simulate in-game, something that only high-end VR products have, but currently lacking in mobile VR products. The project has aimed to bring VR to everyone at a low cost, using it on your own mobile device is already possible to experience.

Keywords – Open source; Mixamo Fuse; Virtual reality games; Simulations; Mobile platforms

I. INTRODUCTION

Today, when the world is transforming into the era of industrial revolution 4.0, a series of new technology trends are gradually spreading far and wide, bringing outstanding achievements in many fields in life such as science, transportation, culture-tourism, environment, health, education, entertainment, etc.

Particularly in Vietnam, the field of VR is still quite new because it has just been introduced. On the market today, there are still quite a few VR systems and there are few published research works. VR currently has a lot of potential for development in Vietnam due to its newness and different characteristics compared to other technologies, users' access to information through VR technology will become a trend in the future not far. Therefore, research on VR is essential.

Indeed, besides the benefits VR brings to the social fields, the Game is one of the main platforms that help VR develop. However, currently, to be able to experience Game VR technology, the cost of both hardware and software is very high.

Besides the trends of the industrial revolution 4.0 such as AI, Big Data, and Machine Learning, it is impossible not to mention virtual reality technology. However, this is not a new invention, but as early as 1962 American cinematographer Morton Heilig invented the Sensorama simulation device, but like many other technologies, VR was only really developed widely applied in recent years thanks to the development of informatics (software) and computers (hardware).

Therefore, with the project "Research using the Google VR library to build virtual reality games on mobile platforms", we hope to aim at bringing VR to everyone at a low cost, and usable. You can also experience it on your own mobile phone.

A. Research objectives

- Research the Google VR open-source library. Then research and use the Engine to make Unity Games, so that develop VR games on mobile platforms.
- Build a complete game application, highly interactive with users, role-playing incarnations of characters in the game.

B. Research subjects

- Google VR SDK open-source library for Unity 1.50.

- Engine for making Unity games using C# programming language [1].
- Photo recognition with Vuforia library [2].
- Character creation software and Mixamo Fuse animation.

C. Research scopes

- Handling objects and characters in the game on a virtual reality environment.
- Track and recognize player activities for simulated interactions into virtual reality environments.

D. Research methods

- Research application of Google VR open source library.
 - + Read the material available on the home page.
 - + How to integrate Google VR into Unity [3, 4].
 - + Use Google VR to track the accelerometer and gyro sensors in an Android environment.
 - + Building movement features in games that can interact with users.
- Using Mixamo Fuse graphics software.
 - + Design of characters.
 - + Build Animation for characters.
- Analyze and design the structure of the project.
- Programming and designing games according to the built structure.

II. WORKING WITH UNITY 3D GAME ENGINE AND GOOGLE VR LIBRARY

A. Setting up the Android SDK and VR environment in Unity

- * Install: Access the link <https://unity3d.com/get-unity/download/archive>. Download and install Unity like any normal software.
- * Integrate the Android SDK into Unity:
 - To integrate the Android SDK into Unity [5], it is necessary to download and install these 3 libraries:
 - + Install Android SDK: <https://developer.android.com/studio/index.html>
 - + Install JAVA JDK: <https://oracle.com/technetwork/java/javase/downloads/index.html>
 - + Install NDK: <https://developer.android.com/ndk/downloads/index.html>
 - After the installation is complete, start Unity as Figure 1:

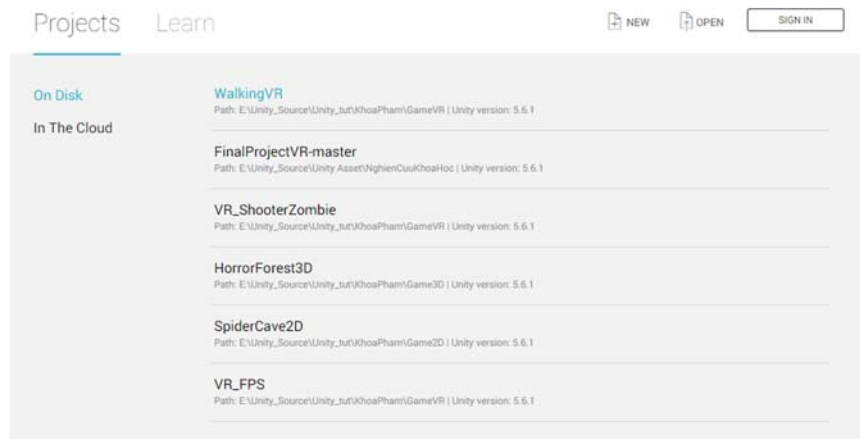


Fig. 1. Interface startup of Unity

- Click New to create a new project
- After entering the name of the project, select 3D mode, and click Create Project, the main interface of the project appears as shown in Figure 2.



Fig. 2. Main interface of a new project

- To be able to compile the game to the Android platform, we need to configure the environment variables in Unity to point to the libraries we just installed above.
- Click "Edit", select "Preferences", and in the dialog box select the "External Tools" tab, in the Android section point the absolute path to the 3 previously installed libraries.

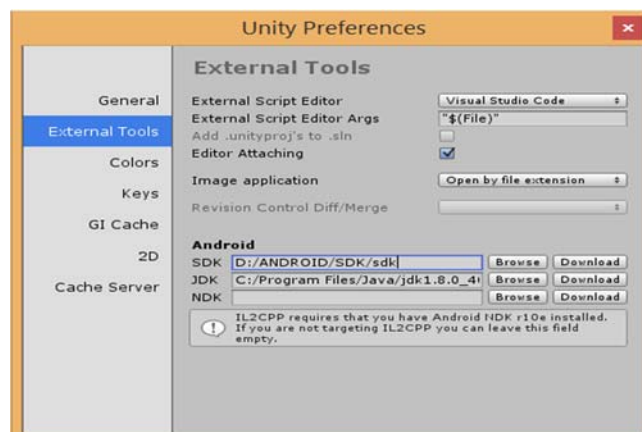


Fig. 3. Environment variable configuration dialog

* Setup VR mode environment in Unity:

- From the main interface of the project, select File -> Build Setting.

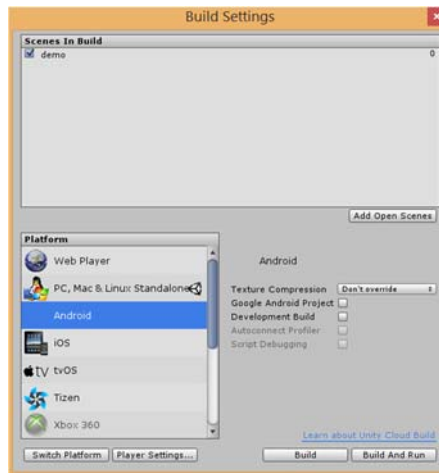


Fig. 4. Build Settings dialog box

- As shown in Figure 4, the project is currently building on a PC environment. To enable VR mode we need to change to Android environment.

- Click on Android environment select Switch Platform.

- After switching environments, click Player Setting.

- Click the “Other Setting” tab, select “Virtual Reality Supported”, and select “Cardboard”.

Note: you can click the “+” button to integrate more VR environments.

B. Integrating Google VR into Unity

- Access the link: <https://github.com/googlevr/gvr-unity-sdk/releases> to download and install the library. There are quite a few versions to choose from, but in this project, we use Google VR version 1.50, because this is one of the most stable versions, and is being recommended by many programmers until now.

- Computer configuration suitable for application development as TABLE I.

TABLE I. CLASSIFICATION TABLE OF CONFIGURATIONS

	Minimum	Recommend
RAM	4GB	8GB
CPU	Intel Core i3	Intel Core i5
OS	Window 7	Window 10
Direct X	9.0c	9.0c
Graphics Card		AMD/NVIDIA 2.0G

C. Theoretical experiment

After learning all the theory as well as setting up the full environment, we will go to practice the Google VR library application combined with Unity through making some simple games about how to move in the game type of virtual reality, from which many genres can be developed based on these sample projects.

* VR AutoWalk: Players will be automatically moved to view the scenes in the game according to the movements of the first part [6]. This is a type of non-immersive VR system that is very suitable for developing in the direction of visiting landscapes, museums, etc.

- * VR LookWalk: Players will also move automatically in the game, but unlike AutoWalk, with LookWalk, in addition to using the head, it will combine with the eyes to move in the field of vision, exiting the view will stop.
- * VR BluetoothController: Help players use additional input devices to support such as controllers, gamepads for applications via Bluetooth connection.

III. GAME ZOMBIE HOUSE VR

A. Game description

- The game is simulated and built in the style of the First Person Shooter (FPS) game series, which is a first-person shooter game [7, 8]. This can be said to be the easiest, accessible and most popular game genre in the gaming industry today. As the name implies, players will really immerse themselves in the main character.
- About the characters in the game, there are 2 types of characters:
 - + Hero (player): the character will move in the game and have the task of finding secret messages in the map, in addition to confronting zombies, running away or shooting zombies.
 - + Zombie: was born to guard the secret letters, has an AI system that helps zombies find and detect players to chase and destroy players.

B. Preparing and building resources

- * Environment: Free to download on Unity's Assets Store, home to Zombies. This is the environment where players will interact such as moving, shooting during the game (Figure 5).

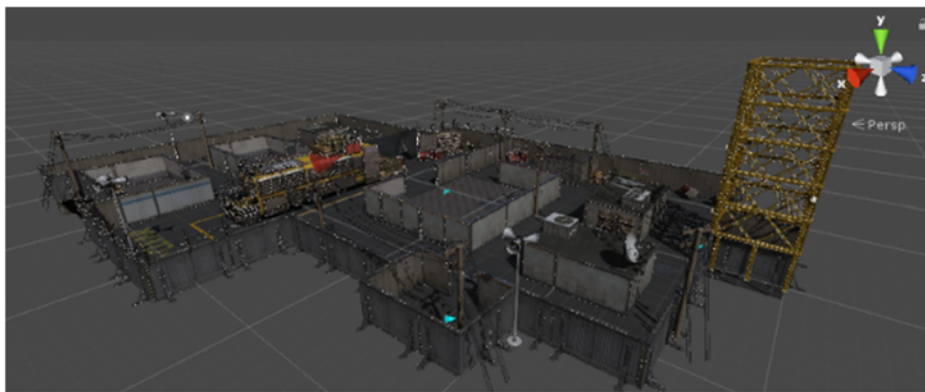


Fig. 5. General environment in the game

- * Zombie: Built from Mixamo Fuse software. Mixamo Fuse software can be downloaded from the link <http://mixamo-fuse.software.informer.com> (Figure 6).

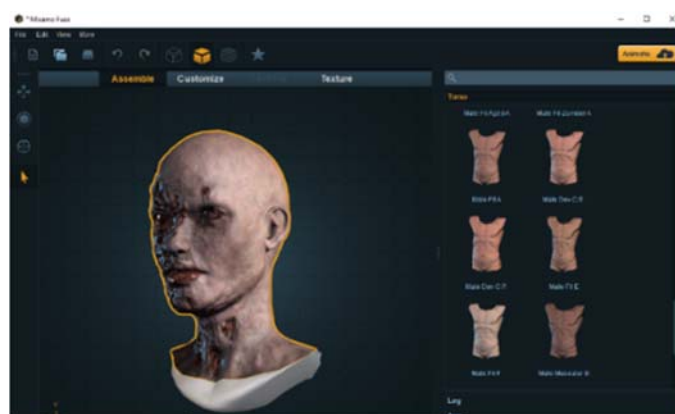


Fig. 6. Mixamo Fuse software interface

* Animation:

- After completing the character design on Mixamo Fuse software, the next job is to create animations for the characters to prepare to import into the game. If you don't have the ability to design animations, Mixamo also provides you with the ability to design animations based on the animations they already have [9]. If you want to modify those animations, simply bend the joints according to the programmer's wishes.

- Animation site and how to use it: <https://www.mixamo.com/#/>

* Vuforia for image processing:

- In this ZombieHouse VR game project, in addition to using the Google VR library for main development, it also uses another type of library combined for image processing for the game, which is the Vuforia library. This is a library commonly used to develop applications of the AR (Augmented Reality) genre. Direct reference link <https://www.vuforia.com/>

- Access the link <https://developer.vuforia.com/downloads/sdk> to download the Vuforia version for Unity. Note that Vuforia currently has a 6.5 version that can be completely downloaded and experienced, but within the scope of the project is using version 6.2 to be most compatible with Google VR version 1.50. After downloading, import Vuforia in the same way as importing Google VR.

C. Practice

* Step 1: Prepare

- Create a Unity project.
- Import Google VR library 1.50.
- Import Assets, 3D objects into Unity.

* Step 2: Build the project structure

Includes 2 main screen scenes:

- Menu Scene:



Fig. 7. MenuScene interface

- Game Scene:

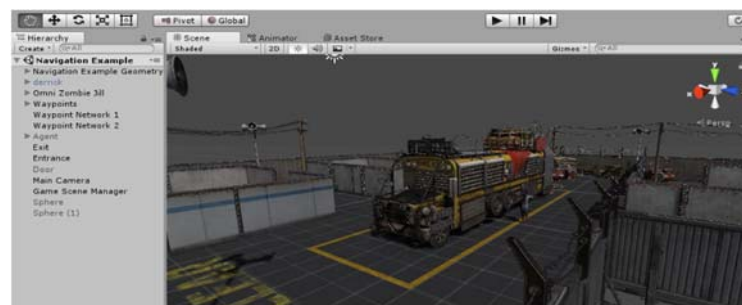


Fig. 8. Structure of components and interfaces of Game Scene

* Step 3: Package the application and build the APK file for Android devices.

- Go to File -> Build Setting to add 2 screens in the game.

- Select “Player Settings”, in the “Other Settings” tab, in the “Identification” section, select Minimum API Level 19 and Target API Level 24.

- Save and click Build.



Fig. 9. Define API level for Android



IV. CONDUCT TESTING

A. System requirements and testing

- Virtual reality applications can be deployed on both Android and iOS platforms. However, within the scope of the project, the application will only deploy on the Android platform.

- TABLE II below is the configuration used for application testing.

TABLE II. TEST CONFIGURATION OF SOME TYPICAL MODELS

Devices	Device info	Results
	<ul style="list-style-type: none"> - Device: Sam Sung Galaxy S6 - Screen resolution: 5.5 inch - CPU: Exynos 7420 3GZ - RAM: 3G - Camera: 16MP - Operating system: Android Nougat 7.0 	The application runs well and stably, the sound, the game screen is switched back and forth quite smoothly.
	<ul style="list-style-type: none"> - Device: HTC One ME dual sim - Screen resolution: 5.2 inch - CPU: MediaTek MT6795 2.2GZ - RAM: 3 GB - Camera: 20MP - Operating system: Android 5.0 	The application runs quite well, when used a lot during the scene transition there is a delay but it is not significant.

* Testing is conducted through the following stages:

- Unit testing: testing separate functions, classes and libraries, recording and analyzing results, quickly identifying causes and fixing errors. This step is performed throughout the system development process.
- Integration testing: Finish unit testing, continue to test complete modules, module assemblies and the communication between modules in the system.
- System testing: conduct a full system test, testing all the functionality of the application. Deploy the app on mobile devices, notice the difference.

B. Reviews and comparisons

The application can run well on Android devices with mid-range configuration and above. Due to time constraints, the product evaluation is done only by collecting the opinions of those who have seen and participated in testing for the project. Specifically:

- About the program interface: the program interface is evaluated quite well in 3D, but it should be noted about the optimization for the application to become lighter and smoother. The game screen is still small, need to add more features to be more attractive.
- About game processing speed: processing speed depends on device configuration. However, for augmented reality and virtual reality applications, the mobile device's GPU must be constantly active to render 3D shapes, the sensors must continuously detect changes from the user and the app's camera must also work continuously. Therefore, the device needs to have a high configuration to be able to run more smoothly.
- About game content: the application meets the initial research needs of setting directions in today's virtual reality game genres. More interaction, helping players transform into the game in a better way.
- About game features: the game can recognize activities from players and then simulate and run 3D models in a virtual environment. The game has an integrated Zombie model with an AI system that makes the game interesting and attractive.

C. Results

- When the game starts, the scene menu runs first.
- After pressing Play Game, the application switches the screen to Game Scene, where the player will search for the secret letters, when they find enough, they will pass the round (Figure 10).
- If the monster attacks until the health runs out, the game will be over.



Fig. 10. Game Scene screen, Zombie detected and is chasing the player

V. CONCLUSIONS

After researching and learning about virtual reality technology, the project has certain results:

- Complete VR game system that simulates actions from users to show in the game.

- The system brings a new experience to users when they want to access virtual reality game technology, and especially for segments of users who do not have too many conditions to experience expensive systems. Although it cannot be compared with specialized VR systems, it has partly met the basic requirements of a required VR system.
- The system also integrates image processing and recognition, making the user experience more diverse, simulating the controllers of specialized VR systems.

REFERENCES

- [1] K. Shafique, B. A. Khawaja, F. Sabir, S. Qazi, and M. Mustaqim, "Internet of things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT scenarios," *Ieee Access*, vol. 8, pp. 23022-23040, 2020.
- [2] M. Radovan and B. Golub, "Trends in IoT security," in *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2017, pp. 1302-1308: IEEE.
- [3] G. Keramidas, N. Voros, and M. Hübner, *Components and services for IoT platforms*. Springer, 2016.
- [4] K. Hamann and R. Smith, "Facial recognition technology," *Criminal Justice*, vol. 34, no. 1, pp. 9-13, 2019.
- [5] L. Li, X. Mu, S. Li, and H. Peng, "A review of face recognition technology," *IEEE access*, vol. 8, pp. 139110-139120, 2020.
- [6] C. Bell, "Introducing the Windows 10 IoT Core," in *Windows 10 for the Internet of Things*: Springer, 2016, pp. 21-70.
- [7] A. N. Ansari, M. Sedky, N. Sharma, and A. Tyagi, "An Internet of things approach for motion detection using Raspberry Pi," in *Proceedings of 2015 International Conference on Intelligent Computing and Internet of Things*, 2015, pp. 131-134: IEEE.
- [8] S. R. Khanal, J. Barroso, N. Lopes, J. Sampaio, and V. Filipe, "Performance analysis of microsoft's and google's emotion recognition api using pose-invariant faces," in *Proceedings of the 8th international conference on software development and technologies for enhancing accessibility and fighting info-exclusion*, 2018, pp. 172-178.
- [9] A. Kumar, A. Kaur, and M. Kumar, "Face detection techniques: a review," *Artificial Intelligence Review*, vol. 52, no. 2, pp. 927-948, 2019.
- [10] S. Ifrah, "Get Started with Microsoft Azure," in *Getting Started with Containers in Azure*: Springer, 2020, pp. 1-26.
- [11] M. I. Mahali, "Smart door locks based on internet of things concept with mobile backend as a service," *Elinvo (Electronics, Informatics, and Vocational Education)*, vol. 1, no. 3, pp. 171-181, 2016.
- [12] N. Krishnamoorthy, R. Kalaimagal, S. G. Shankar, and N. Abdhul, "IoT based smart door locks," *International Journal on Future Revolution in Computer Science Communication Engineering*, vol. 4, no. 3, pp. 151-154, 2018.