National Conference on Advancements in Computer Science and Engineering
In association with International Journal of Scientific Research in Science, Engineering and Technology
Print ISSN: 2395-1990 | Online ISSN : 2394-4099 (www.ijsrset.com)

Corona Killer A Free Augmented Reality Game Using Unity

Savion Mario Sequeira, Dinesh G

Department of Computer Science and Engineering, New Horizon College of Engineering, Outer Ring Rd, near Marathalli, Kaverappa Layout, Kadubeesanahalli, Bengaluru, Karnataka, India

ABSTRACT

Corona Killer is an Augmented Reality game developed for Android devices using Unity 3D Engine and Google AR Core. This paper will focus on the process and the technologies used in order to achieve the game.

Keywords — Augmented Reality, Unity, Android, Game Development, XR Development

I. INTRODUCTION

There are two major components introduced in more detail in this paper, Augmented Reality Development and Game Development.

"Augmented reality combines the physical world with a computer-generated or simulated environment. It is accomplished by superimposing computer-generated images over real-world images. There are four forms of augmented reality: marker-based, marker-less, projection-based, and superimposition-based. It has a wide range of real-world applications. AR is used in a variety of areas, including medicine, education, manufacturing, and robotics, to name a few. "[1]

The way I would define Augmented Reality is bringing the digital word into the physical space with the means of a device, the device being a smartphone in this scenario. It is augmenting the digital world to the real world.

Augmented Reality is a subset of XR (Extended Reality), this includes Virtual Reality (VR), Augmented Reality (AR) and Mixed Reality (MR). Mixed Reality combines both Virtual Reality and Augmented Reality.

"Augmented Reality has come a long way and has recently been popularized in mainstream entertainment through games like Pokémon GO and Face filters.

Game Development is a very complicated process, in reference to the book "Agile Game Development with Scrum" by Clinton Keith, he mentions about how the pioneer days of game development has now disappeared. It is no longer a sole programmer who used to specialise in every aspect of game development from programming and even rendering their own art. They have all been replaced by an army of specialists. "[2]

Game Development is a very vast process, it goes beyond the scope of just being a programmer or an artist, I classify game development as a multidisciplinary field. From your basic programming and scripting, to animation, 3d modelling, game design, level design, sound design and so much more. It has become a very complicated process. It can take an experienced game developer a year or more to develop a full-fledged game from scratch for production.

This should not scare away any solo developers trying to get started or create a game. The beauty of game

development is how huge the community is, there are millions of free and paid assets which anyone can use under certain conditions for their game. As a result if I as a developer is good at game scripting I need not worry about learning to make good 3D models anymore as I can just download them. This is why game development is so accessible in general to anyone who is willing to give it a go.

Unity 3D game Engine, is one if not the most commonly used game engine by solo and even some company game developers. Unity has a huge resource online from their own sources and even from third party sources. It is a very well documented game engine which makes it easier for anyone new trying to get started.

The reason I used Unity is because it supports both Game Development as well as Augmented Reality Development. It is suitable for any cross-platform development without having to change the codebase for each platform. Speaking of codebase Unity can be scripted in C# or JavaScript. There are other ways to do scripting for people who are weak in coding, using UI drag and drop to create scripts.

II. REQUIREMENTS

The requirements for developing a game in AR is surprisingly almost nothing. Let's start with AR. When it comes to developing in XR, VR can get very expensive to develop fir, starting with getting the equipment required to handle VR which can cost a lot of money, and moreover the equipment required to handle VR is also expensive. As a result for anyone dipping their toes in XR development and does not currently have the resources to build for VR, they can always develop for AR.

The main requirement for AR development is a smartphone. Smartphones nowadays are built to handle augmentation. Any Android device running Android 7 or above can support AR. This is because firstly Google Play AR services is only compatible with Android version 7 or above. Moreover, the new smartphones come with depth sensors and this is very important since any 3D object to be Augmented requires all three axis coordinates to be spawned into. This being the x, y and z. A normal camera takes a picture or records a video in 2D, but depth sensors allow the camera to now differentiate between an object in the foreground and an object in the background. From just the AR perspective these are the only requirements to allow a device to use augmentation.

When it comes to the developing phase things can get a bit expensive depending on the scale of your project. Firstly, you need a laptop with around 16 GB of RAM to use Unity without any hassle, Unity is a heavy application since it is very extensive. The next question would be do you require a GPU? Well from a game that does not depend on too many heavy graphics you will not require a GPU as your PC does not need to render any heavy graphics, you can use a few 3D models and effects and get away with a good CPU. Saying this the moment you start adding a lot of graphics into your game be in in AR or normal games you will start to face lagging issues; hence GPUs are a must in those scenarios.

To end the requirements chapter, I would say compared to some of the other project you can start on, AR development is very easy to get it done and get started with for even a student with almost no budget.

III. SETTING UP YOUR WORK ENVIRONMENT

"The Unity Hub is a stand-alone programme that makes finding, downloading, and managing Unity projects and installations much easier.

Unity will prompt you to instal the Hub if you start the editor without it installed. You can generate an empty project from the Hub installation prompt if you have a licence from a previous version of Unity. "[3]

Unity is very easy to setup, as mentioned earlier it is very well documented engine. As a result almost any

issue you might face will have been resolved. To get started all you need to do is download a version of Unity from the Unity website. Unlike other platforms you can download multiple versions of Unity each being stored as a separate application. This is because some versions have LTS (Long Term Support) while the newer ones are still in Beta. Moreover, some versions are compatible with some features like AR while some might not be, this makes developing very flexible as you can choose which version of Unity suits your needs and which version of Unity you are comfortable developing in.

When downloading you can add several modules that you will require for your project , in this case you can add the Android Module which includes JDK and NDK tools that is required for creating an APK to run on your smartphone.

Once everything is installed you can now create a new project, a 3D project with the version of Unity you require. The last step you need to do to allow development in AR, is install Unity AR foundation and Google AR core. Let's look at each individually.

"Within Unity, the AR Foundation allows you to work with augmented reality platforms across several platforms. This kit provides an interface for Unity developers to use, but it does not provide any AR functionality. To use AR Foundation on a target system, you'll also need separate packages for the Unity-supported target platforms:

AR Core XR Plugin on Android and AR Kit XR plugin on iOS" [4]

AR Core is the official plugin created for allowing AR applications to run on Android devices. It is not exclusive to Unity; it can also be used in Android Studio.

To install these packages, you can download both AR Foundation and AR Core from Unity's package manager. Once installed you can begin developing an AR application.

IV. BRIEF SUMMARY OF THE GAME

Corona Killer is a 3D AR shooter, essentially the game requires you to move your phone around finding viruses and destroy them by vaccinating them. When the player downloads the game , they are first shows a home screen where you can see how to play the game, about menu, and even an achievements menu. In a later chapter we will talk about the importance of minor details that can make a game exciting or boring to play.

The main button is the Play Button which the player will click to start the game, once the player clicks that the game begins. After a small delay the viruses start spawning randomly in various places, using a cross hair on the screen the player will need to point at that virus and click the vaccinate button. The more points the player scores the harder the game gets, there is a health bar to add stakes to the game, once the health reduces to 0 the player loses. To add advantage to the player, random heart points are also spawned in between the player on shooting them gains back some health. Once the player dies, the high score is checked. The goal of the game is to beat the previous high score. In addition to that, the player can also gain some achievements to make it more interesting. This is a brief summary of the game, we will now look at the game development process.

V. APPLYING SCRUM TO THE GAME DEVELOPMENT PROCESS

"A game is a type of software designed to entertain people. However, in real-world game development, merely following the software development life cycle (SDLC) is insufficient, as developers face numerous obstacles during the life cycle. To resolve the problem, game development employs a method known as the game development life cycle (GDLC) to guide game development. None of the current GDLCs, on the other hand, specifically discuss how to produce a high-quality game. This paper introduces a new game development life cycle model as well as recommendations for delivering a high-quality game.

At each point, a number of quality parameters are specifically considered." [5]

With reference to the above paragraph, I realised there was no particular way to go about a game development process, so you are required to create a workflow which works for you, especially when you are doing it solo. You can then look back at the process and refine it for your next project until you have a very functional workflow.

For this project I followed a SCRUM approach and broke down my project into separate components tacking each component one at a time. Below was my entire thought process for this project.

- Figure out the main game mechanic of being able spawn objects in AR at random intervals. To do this we can us an IEnumerator.
- Next would be to figure out if our phone is pointing at the spawned AR object in free space, this can be solved using RAYCASTING. We will look at Raycasting with more detail in Paragraph VI, as it is the crux of this project.
- Once we have figured this out, we need now a point which can be used to point at the objects, we can create a cross hair on the center of the screen now. This cross can be referenced to act as the point for applying Raycasting to detect the object.
- Now that we are able to detect the objects, we can now start performing an action when the object is detected, to do this we need a trigger, that trigger can be a button at the side of the screen.
- Once the button is clicked and the objected was detected we can now destroy the object and perform a random effect.
- Now that the main part of the game is done, we need to start adding game mechanics, first is a score system.
- The score system will give players a reason to play the game, a sense of achievement. Each time an object is destroyed we can increase the score by one.
- Now that the player can achieve something we need to start adding a risk factor, in this case a health bar, this health bar gives the player a reason to play well. If the virus on screen starts spreading, we then need to start reducing the health overall. Until the player eventually dies.
- Now that we have implemented the health system, we need to allow the players to gain some health back, we can start generating hearts for players to collect.
- With this we are done with the game mechanics. We can now start working on UI components. We first need to make the main screen which the user will access.
- The main screen will have multiple components as well, starting with the Play button.
- Then we can add an Options button to control the game volume.
- We can now add a help menu to make the player understand the concept of the game.
- Lastly it is important to give goals for the player to achieve so we will add an achievements menu.
- We can also add an about menu to give some information about the developer of the game.
- To end the game, we need a game over menu, with option to retry and quit the game. We add this after the player loses all health.
- The last feature to show if the player improved from last time, we can add a high score, which only gets updated once the player beats his previous score.
- With this we have developed the entire game. We can refine it by adding some sound effects and music to enhance the overall experience.

VI. RAYCASTING

Raycasting is the method of firing an invisible ray from a point in a specific direction to see if any colliders are in its way.

This is a simple definition of raycasting but this technique is what allowed me to develop my entire game.

"AR technology, which shows virtual objects overlapping in the real world, is gaining popularity, and a variety of AR applications are being created. These apps were first created as mobile AR for mobile devices including smartphones and tablet computers. Mobile AR systems used the smartphone's camera to display the real world while augmenting simulated objects on the camera screen. Since then, smartphones have become more widely available, making AR applications accessible to all. With the development of computer vision technology, the floor and wall can be distinguished by a mobile device with Apple ARKit or Google AR Toolkit [6] and the furniture can be placed in the desired position without markers. The most significant disadvantage of AR based on mobile devices is that the user must continue to keep the device in order to superimpose the virtual entity, which is the most significant limitation of AR based on mobile devices." [7]

The reason I cited the above article is because to show the current flaw behind the system, raycasting will only work if the player moves the phone to the location to detect an existing object in front of it.

The description of raycasting in unity is "Casts a ray, from point origin, in direction direction, of length maxDistance, against all colliders in the Scene." [8]

Basically, when you point your phone at an object, the application shoots a beam to the object and if the beam collides with the object a hit is detected. We can configure for only certain object to be detected and then perform any action of those objects when detected. It shoots a beam from the centre of the phone.

In my game I have named the virus as "Cube" and any clone created of the virus would be called "Cube(Clone)" so if the player points the crosshair at the virus and click the vaccinate button, the program first checks if the object detected is named "Cube(Clone)" and if so then I programmed the object to be destroyed while displaying a special effect and incrementing the score counter.

This is the main technique behind which the game works, luckily in Unity we do not have to worry about coding the entire technique as Raycasting is available as a function to which we can input parameters to it.

While saying this , it has been pointed out in several papers about the flaws of Raycasting and new techniques are being developed to improve this based on Raycasting itself. For my project it worked well and I did not face any issue using this technique.

VII. CONCLUSION

The future of gaming is very fruitful and bright. The market is ever growing and there seems to be no stop to its growth. As estimated to hit staggering numbers of over 250 billion dollars in the coming years, game development is extremely sustainable for any engineer willing to become a part of the gaming industry.

Technology is ever growing, withing a year the technology can become obsolete since new and better ways are being developed. The amount of game developers are also rising with time, due to the ease of accessibility of the game engines being developed such as Unity and for sure in the coming years they will become more powerful and more easier and efficient to use by anyone willing to learn and become a game developer.

The current project being worked upon can be improved upon vastly in the coming years with better 3D rendering upgrades being developed and more powerful processors and graphical cards being made, it will be very easy to quickly render projects and test them. It will enable developers to try to make more complex games and execute more interesting ideas due to this, as they would not have to spend years working on a single project.

Augmented Reality has leaps and bounds to go further as it is still in its early stages. Augmented

Reality which is a subset of Extended Reality is collectively being developed by vast number of companies. In this field particularly Facebook is investing heavily to make more complex interpretations of what Augmented Reality and Virtual Reality currently can do. The technology will become even more accessible to the developers and more consumer friendly in the coming years. The technology needs a lot of improvement in how functional the portability of the sensors can be.

Currently the smartphones are being developed with sensors and hardware components equipped to handle augmented reality applications. While they work pretty well, the detection at times can still be very clumsy with irregular surfaces and it can cause glitches in the application.

This project has ways to go and can be improved in several ways, with more time available and with better hardware equipment the time taken to make a more complex game can be greatly reduced.

VIII. ACKNOWLEDGMENT

I would like to thank my college and the department for providing me the opportunity to work on this project. I have learnt a lot from this work and will apply them to future projects. I would also like to thank my parents for supporting in my endeavours. Special thanks to my mentor Mr. Dinesh for guiding me throughout this project with any queries I had. I am also grateful to the multiple people out there who have done various researches for me to be able to learn from their work and apply it to my project. It was very insightful.

IX. REFERENCES

[1]. Riya Aggarwal; Abhishek Singhal, “Augmented Reality and its effect on our life”, 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2019

[2]. Clinton Keith, “Agile Game Development with Scrum”, Chapter 1

[3]. The Official Unity Webpage for getting started with Unity. https://docs.unity3d.com/Manual/GettingStarted.html

[4]. The Official Unity Webpage for getting started with XR Foundation https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@4.1/manual/index.html

[5]. R. Ramadan and Y. Widyani, "Game development life cycle guidelines," 2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS), 2013, pp. 95-100

[6]. Bellarbi, A.; Zenati, N.; Otmane, S.; Belghit, H.; Benbelkacem, S.; Messaci, A.; Hamidia,” A 3D interaction technique for selection and manipulation distant objects in augmented reality.”, n Proceedings of the 2017 5th International Conference on Electrical Engineering-Boumerdes (ICEE-B), Boumerdes, Algeria, 29–31 October 2017; pp. 1–5.

[7]. Ro, H.; Byun, J.-H.; Park, Y.J.; Lee, N.K.; Han, T.-D., “AR Pointer: Advanced Ray-Casting Interface Using Laser Pointer Metaphor for Object Manipulation in 3D Augmented Reality Environment.”, Appl. Sci. 2019, 9, 3078.

[8]. The Official Unity Webpage for getting started with Raycasting https://docs.unity3d.com/ScriptReference/Physics.Raycast.html