International Journal of Computer Trends and Technology
ISSN: 2231–2803 / https://doi.org/10.14445/22312803/IJCTT-V72I8P106

Volume 72 Issue 8, 33-41, August 2024

Original Article

House of Cards: Framework for Memory Experiments in VR

Priyank Singh

Rochester Institute of Technology, US.

Corresponding Author : erpriyanksingh@gmail.com

Received: 07 June 2024 Revised: 13 July 2024 Accepted: 30 July 2024 Published: 13 August 2024

Abstract *- Playing games is something that everyone likes at some point in their lives. It seems that playing a game is wasting time, but that is not true if you play memory games. Mind games are supposed to help a human being improve concentration, enhance cognitive skills, improve brain functionalities, increase short-term memory, train visual memory and much more. The paper discusses developing a memory card game bundled with a game-development engine, Unity3D, to execute the game. The primary goal of the project was to use the game for research purposes. The study is aimed to be on the human brain, specifically how memory works in the brain. Different papers and books were used and studied to find how the human brain works when a subject is presented visually. The same thing was tried to be explored with this particular card game. The results collected from different subjects are discussed and shown in the report later. The game was developed in C#.*

Keywords *- Unity3D, Game development, Human brain, Game engine, Software architecture.*

1. Introduction

Card games are a great learning platform. All the games include some kind of rules which a player has to listen to and follow. This improves the listening skills, sharpens the memory and develops good social skills. Concentrating on cards increases your ability to learn and improves your concentration skills. The sequence of the numbers on the cards helps you memorize them, which is needed for the game and helps improve your memory.

The game of cards has always been interesting and exciting for the users to play. Card games are not only played for money as seen in casinos. There are other card games that can be played for fun. This project is all about creating one such fun game using virtual reality. The main aim is to develop a card game that can be played using Google cardboard, and the player feels his/her existence in the virtual world while playing this game. This Virtual Reality environment will increase the excitement for the players playing the game.

1.1. How does Memory Work

Every one of us sees different things around us daily. Do we remember all of them? There are things or objects that we remember for a decade and on the other hand, there are things that we forget as soon as they are out of our sight. As it has been said, "out of sight, out of mind" [9]. Some of us do remember where we used to live or play in our childhood. We may also remember how our school looked. Consider meeting a person at a party for the first time; we will greet him/her, but when we meet the person for the second time at the same party, we will not greet that person again as we already remember that person. But now again it is not necessary that we remember that person for a long time. I would talk a little about visual Memory and how it works.

What are the visual memories that fade away immediately and that last for decades?

Visual Memory is defined as the preservation of visual information after the optical source of information is no longer available to the visual system Palmer et al.[18].The amount of time the information is preserved in memory is unspecified; it could be from milliseconds to an entire life.

Memory has been categorized into three parts Palmer et al.[18]:

1.1.1. Sensory Information Store (SIS)

Stores visual and auditory information.

1.1.2. Short-Term Memory(STM)

Stores information that is being processed.

1.1.3. Long-Term Memory(LTM)

This stores semantic memory, procedural memory and episodic memory.

1.1.4. ICONIC MEMORY

An experiment was performed to study the iconic memory. In the experiment, the subjects were shown four letters in one row, and a total of four rows were shown. If the rows were shown one after the other, the subjects were able to remember four letters at the end. Then the experiment was changed, and now all the 16 letters were displayed at once, and then they disappeared.

However, the results were the same; subjects were able to remember four letters out of these 16 letters. Now the researchers thought that if a subject can recognize four letters when displayed all 16, then they should have other letters somewhere in their mind which they cannot recall. Differences in their basic characteristics define different memory systems. Those characteristics are Duration, Content, Loss, Capacity and Maintenance. Let us look at the iconic memory based on these different characteristics.

1.1.5. Duration

Averbach and Sperling introduced a delay between the termination of the letter display and the onset of the tone using the target row Palmer et al.[18]. The data tracked in the memory showed that information lasts for more than two seconds in the iconic memory if the prestimulus and poststimulus fields are dark and half a second if they have light. Typically, it is less than half a second as the pre and post-have light.

1.1.6. Content

Von Wright conducted additional experiments to illustrate different uses of cues in memory[18]. In the experiment, he displayed eight items separated in two rows, each having four items. The items were four letters and four digits. Also, four amongst those were red colored, and the other four were black colored. Each time the items were presented to the subject, a tone was played, letting the subject know what to choose. First, spatial cue condition which asked to report the upper row. Secondly, the Color cue condition where subjects were asked to report a single color, red or black. Thirdly, the categorical cue condition asks the subject to report the letters or digits.

When the results were compared, it was known that they were way better when used as spatial and color cue conditions. On the other hand, the results for categorical cue conditions were not that good. From the results, we conclude that location, color, size and shape are the features that are represented in iconic memory Palmer et al.[18].

1.2. Requirements

The primary goal of the project was to create a real-time game that people of all ages could play. The game should be interactive and fun to play. The focus was to design and develop the game on an Android device. As of today many latest devices running Android on them are in the market. Our goal was to take a basic but not too old version and make sure the game was compatible with all of the higher versions.

1.2.1. Hardware

Development of the Virtual Reality game for the Android device requires a Unity 3D game engine. Hence, For the development of the game, Unity 3D was chosen to be the gaming engine. Unlike other game engines, Unity 3D is heavy software and requires some minimum hardware configuration to run on. For the development purpose it was required to have a Graphics card with DX9 (shader model 3.0) or DX11 with feature level 9.3 capabilities [14]. Additionally, Mac computers running a minimum OS X 10.9.4 version and Xcode 7.0 or higher were required [14].

All the requirements mentioned above are important as Unity 3D is computationally expensive software and needs a good platform to execute on.

1.2.2. Android Device

Though Android SDK comes along with the simulator to test the real-time game a physical device was needed. As per our goal, the device needed to be in the middle generation of an Android operating system so that all the higher versions also support the same game with the same software and compatibility. The game was targeted to be tested on version 4.0(Ice Cream Sandwich) and 6.0(Marshmallow). The device chosen to test the game was Nexus 5. Without the physical device, it becomes difficult to predict the performance issues. Nexus 5 was chosen for the reason that it has one of the most stable pedometer sensors installed in it. Gyroscope and Accelerometer are also the sensors that were present in the device and made use of while developing the game.

1.2.3. Software

Android SDK

The Android SDK contains all the tools needed for the development and deployment of the application onto an Android device. The tool is also very helpful in testing, running and debugging the application on the simulator as well as on the physical device. With the Android Studio, Android SDK is pre-installed into the package. Android Studio also contains an Android simulator to test the application on different Android OS. Android Studio 2.0 was used for the development of the game which is available to download for free and is compatible with the version of OS X from [1]. Android Studio is the most powerful IDE that exists as of today. Google develops Android Studio, and so is Android; hence the compatibility is good and reliable.

Unity

Unity is a cross-platform game engine developed by Unity Technologies [15]. The engine is used to develop mobile games, video games and much more stuff related to

graphics. Unity 3D 5.0 was used for the development of the game. Unity has inbuilt support for the Android phone. This makes it easier to deploy the application on the device and test it. Unity compiles all the files needed for the project, including the scripts written in c#, utilities, and all the dependencies, into a single APK and makes it compatible with the device we are deploying on. All the process is taken care of by unity internally, and this makes the development more fun, as we do not have to worry about these small things and can focus more on our main idea and goal of the project.

Unity is a great platform and was very helpful in the development of our application like the free and good customer support provided by Unity.

C#

Unity supports JavaScript and C# as the development language with the Android interface. Our project was developed using C# as the primary and only language of execution. The reason behind choosing the language is the previous experience with Java and c++, which is mostly similar to C#. However, the major challenge was that most of the documentation was available in JavaScript, which made it a little challenging to solve some complex problems we came across. On the other hand, the game that is being developed would perform better with C# and easier as the advantage of pointers can be taken in C#, and it is an object-oriented language [3].

Roadmap

The organization of this paper will be as follows, Chapter 2 will talk about the challenges faced throughout developing and implementing the project. Challenges from small, like software related to big, something related to ideas, are also discussed here. Chapter 3 will present the design for the game used. Why a particular design is being chosen, and how would that design help me in my research on the human brain? Chapter 3 will discuss the approaches taken to implement the design explained in Chapter 3. All the software used and languages used are discussed in the chapter. Chapter 4 discusses the results achieved from different experiments. All the results are compared and explained. Chapter 5 concludes the whole project from start to end. This chapter talks about the current status, Future Work and Lessons Learned.

2. Challenges

Throughout the development of the project, minor and major challenges were faced. All the challenges faced are listed below:

- The game engine chosen to develop the game was unity3D (version: 5.5.1). Unity was updated in December 2016, just when the project was about to begin. The application was targeted to run on an Android device (Nexus 5) using Google Cardboard. Because of the recent update, there were compatibility issues with the Google VR package and Unity 3D. Some of the functions are now deprecated in the newer versions, e.g., optimize function.
- Secondly, Unity, by default, has a path set for Java 7 and the machine I was using had Java 8. It took us a long time to figure out. The only thing that needed to be fixed was to give the path to the Java version that existed in the machine. This issue was causing the build to break whenever we tried using an Android device (Nexus 5).
- Thirdly, we have had no previous experience with C# and started the project with it. Initially, it was a big hurdle as most of the examples and documentation were in JavaScript. But as we went through with time, we got a good command over it.
- Fourthly, we wanted a time frame for which the card should remain visible once clicked by the user. Achieving this was challenging earlier as Unity operated on one thread, and if we try to use Thread.sleep(), then the only thread goes to sleep, and the application hangs for that particular time. It was finally achieved with StartCorou tine(function).

3. Design and Implementation

3.1. Design

3.1.1 Game Summary

House Of Cards is a very popular card matching memory game that existed since our childhood. This game was also known as Concentration, which could be played with any standard deck of cards or any commercial cards that have some similarities to them. There are a total of 52 cards in a deck, 13 of each kind (if we exclude the houses and consider the numbers). The game has some simple rules, as stated below:

All the cards would be placed upside down with the back facing the player.

Flip one card, remember the card and location of the card and keep it back with face down.

Flip another card and repeat the above step unless you find two matches in the cards.

When a match is found (Ace-Ace), remove those two cards from the game.

Repeat the process unless all the cards are finished in the game.

The main goal is to finish the game in the minimum number of moves. One move is counted when one card is flipped. The complexity of the game can be increased if instead of finding only the match in numbers (Ace-Ace), also color should be the same(Ace of red - Ace of red). This rule will add a level of complexity to the game. The idea behind

the *House Of Cards* game is the same, and the rules are the same the only different thing is that it has been developed to be played in a Virtual Reality(VR) environment instead of a static 2D environment. Playing it in VR makes it more interesting and also challenging for the players. The biggest challenge faced by the players is to remember the cards in a 3D environment instead of a 2D. If there is no perspective where the cards are placed the game becomes more challenging to a human brain because the human brain is trained to remember things with respect to something. The study has been explained in the Introduction section 1 of the report.

The design for our game would consist of several stages. In the first level, the player would be present in the four wall room, where there would be different cards scattered all over the room. The user has to pick two consecutive cards to get them both out of the scattered cards. As soon as all the cards are finished, the game will stop.

3.1.2. Concept

The motive of the game was not only for fun but to do some study on the human brain. The game had to be interactive and easy and should keep the person engaged. With the fun part, some data was also being collected to see how the human brain, specifically the hippocampus of the brain, responds. Data collected included the number of clicks the user did in the whole game, the total time taken and whether the user had specs or not.

Before the game, the user was asked to fill out a questionnaire that would help in the study. The survey asked a user about the age, time in hours the user spends on video games per week and games played using VR/AR gadgets per week. These questions were asked to know how familiar a user is with playing games and using VR/AR gadgets.

As the game was built on an Android device, the game made use of mobile features such as the touchscreen of the phone. Apart from the touch feature as the game is implemented for a Virtual Reality platform, the sensors of the device were also taken advantage of. Motion and rotation sensors were made for controlling the VR environment.

The game was designed to be played by players of any age. The game was targeted to be designed in a way that it has fireworks all over in the VR environment. This makes the game more interactive and fun to play. For several levels, the images chosen for the walls and background are the illusion images to distract the user and make the user lose concentration.

3.1.3. Target Platform

The game was targeted to be played on Android devices using a Google Cardboard. Google Cardboard is a virtual reality platform developed by Google Inc. [2]. Google Cardboard uses a smartphone to complete the VR experience for the user. The project also made use of the Google Cardboard SDK for Android to integrate the application with it. The design of the cardboard is very simple and easy to use. The only limitation faced was that it has only one control for clicking. The whole game was developed keeping the restriction in mind as no more than one control could be given to the user. The only control was used to flip a single card over.

3.2. Levels

3.2.1. Level 1

The first level of the game was designed for the user to understand the game and get acquainted with the Virtual Reality environment. Also, the first level made sure that the user was comfortable using the Google Cardboard (a virtual reality environment platform) before proceeding to the next level, which would help in the study of the human brain for which the application was created.

The cards are arranged around the camera, as shown in the figure below. The cards are static, and the view seen through the camera is also shown in the image below. The view from the camera has a pointer(white circle in the center of the image) that points to any card. When the button on the cardboard is clicked, the card is flipped, and the player can see which card it is.

As seen from the image, two screens would be shown in the camera of the Android phone. When the phone is used with the Google Cardboard, the player will only see one card as it gives the feel of a Virtual Environment and makes it very interactive to play.

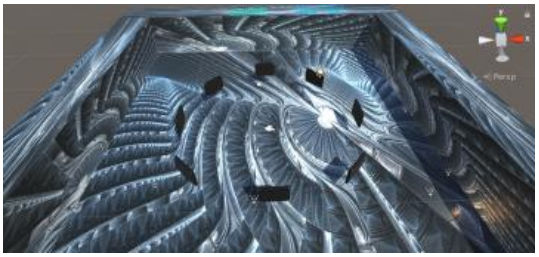

Fig. 3.1: Level 1

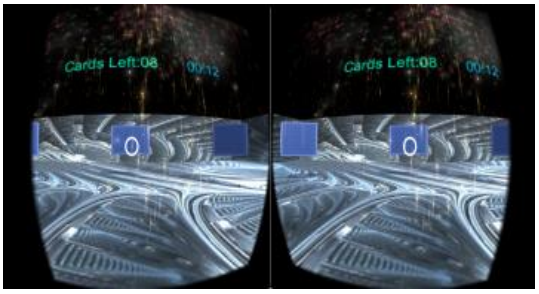

Fig. 3.2 View from camera

When a player points the pointer at the card and clicks it using the controller on the Google cardboard, the card flips and looks as shown in the image below. There are, in total, eight cards (4 pairs), as can be seen at the top left of the screen by the user. The user needs to use the pointer and the control from the Google cardboard, find all the matching pairs, and click them consecutively to disappear them. Once all the cards are finished, the game will be over, and the user will land on a different scene stating" Game over" and show the total clicks done by the user on the cards and the total time taken to complete the level. This data is used for further study as we will see later in the report.

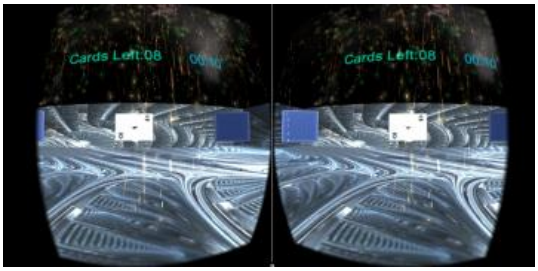

Fig. 3.3. View from camera

3.2.2. Level 2

The second level of the game was designed for the users to get acquainted with the motion and movement of the camera. This level also has a pointer as the first level, but the difference is that here the user will move in the direction wherever the pointer is facing. This makes the game more interesting as the user can walk(not physically) all over the area and find the cards. The idea is the same: to match the pairs, and the cards would disappear. To make the game interesting and interactive fireworks were added in the sky that gives a good user experience. Also, the images chosen for the wall are illusion images that make the area look bigger than it actually is. This Level makes the user comfortable with the motion and knows how to have control over the game.

The learning from the first two levels would be used in all the Levels, and then it would be fair to time the user and learn about the human brain. The figure shown below is the screenshot of the game for level 2. At the center, the white object is the camera and the starting position for the user. All the rectangle objects are the cards that users have to find while walking all over the place and also find the matches to find the pair and finish the game.

The speed of motion is kept slow and feasible so that users get enough time to see the card and also not that slow that the user gets bored walking in one direction. Several experiments were done with the moving speed and finally, one speed was chosen. The floor in this level is kept black because it causes a headache to the players with the illusion photo on it while walking. This level was made square and without any obstacle, as the motive was to make the user familiar with the game and technology.

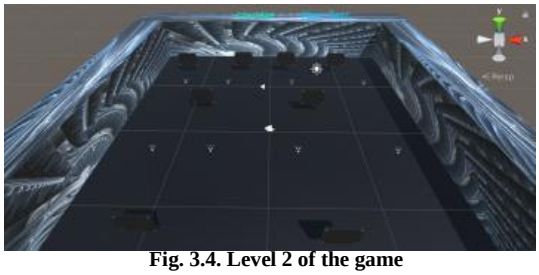

Fig. 3.4. Level 2 of the game

3.2.3. Level 3

The third level of the game was designed for the main study. The level is very similar to the first level of the game, as can be seen in the figure below. The difference between the third and second levels is that now the cards are rotating around the camera in a clockwise direction. With all the cards having a blue background and rotating around the user(camera), it is not easy to track the exact position of the cards and find the pairs. The speed of rotation is also kept feasible so that the user does not feel dizzy if the game also does not become impossible to play.

A user needs to have full concentration on the game. If the user misses the cards and rotation, then they have to start again. A human mind always needs some perspective to remember things [4]. If you lose track of the cards once, you lose the perspective and hence need to start again to develop the perspective again.

Fig. 3.5. Level 3 of the game

3.2.4. Level 4

The fourth level of the game is the same as the third level, with a slight change. The only difference between the fourth and the third level is that in the previous level, the background for all the cards was blue. In level 4 the background for the cards is blue and red alternatively.

The reason behind this design is that, as we talked about earlier, the human brain always needs a perspective to remember things. Here the idea is to keep the alternate colors and give the brain a perspective to remember the cards.

The study later showed that the idea proved to be true as the user took less time to complete level 4 than they took to complete level 3. We will talk about it later in the report.

3.2.5. Level 5

Level five of the game is designed to take longer than 120 seconds to play. The reason behind this was to focus the research on the short-term memory of the brain. All the earlier levels were related more towards the iconic memory.

Level five has a maze, as shown in the figure below. The white object shown in the image is the camera(user) starting position. There are cards placed at different locations in the maze. The challenge here is for the human brain to remember the path in the maze and also the path to reach the previously discovered cards. The cards are placed in a way that if a user is trying to explore new areas, they would be able to find the cards in a pattern.

The previous study has shown that the "CA3" part of the hippocampus in the brain is mainly responsible for deciding between a familiar thing seen before or not [5]. The decision could be right or wrong. More details about the human brain and hippocampus are provided later in the report.

Fig. 3.6: Level 5 of the game

3.3. Art Design and Restrictions

Designing a game from the basics takes a lot of brainstorming. Things with a convenient design for users and with normal colors are a must. The color and images chosen were not too harsh or something that would make the user feel dizzy playing the game. The game, as said before, was targeted to be played by users of all age groups. Hence the game has to be very easy and user-friendly to play. Also, the game was targeted to be played on an Android device with only one controller(available in Google cardboard). At last, a roadmap was designed with the deadline to finish the game in time and what all things need to be done and when, so the project is ready on time. As the application is developed to work on an Android device, it comes with some restrictions as well. The first and most important was the processing speed of a smartphone. The specification for the Android smartphone(Nexus 5) we targeted has: "a System on the chip with Qualcomm Snapdragon 800, CPU 2.26 GHz quad-core Krait 400, GPU Adreno 330, 450 MHz and memory 2 GB of LPDDR3-1600 RAM" [7]. The processors of all the devices are different but still, we know that the processors in phones are much slower than that in iPads and computers. We have restricted our application to a smaller number of utilities.

For example, firecrackers emitting less number of particles per frame were used to ensure the smooth functioning of the game. Secondly, Alpha texture planes were used for the complex structure. This saves much more computational speed and improves the performance of the game. The texture in Unity has a restriction that it has to be in the power of 2. For example, 256x256, 512x512 and so on [16]. If the images are not of the power of 2, it will increase the work for the hardware reducing the performance speed of the game.

Unity supports different types of audio formats as input, such as .mp3, .ogg, .wav, .aif, .mod, .it, .s3m, and .xm [11]. Our project used .mp3 format and used the feature of auto-repeat which saved storage by uploading small audio files with the repeat mode on. The audio file chosen was a horror theme, which goes well with a card game [10]. It was also made sure that the song should not have a sudden increase or scary volumes that would scare the user. The audio was just to provide slow music in the background.

3.4. Implementation

The project was planned to be built within 15 weeks from scratch to deployment. The first 2 weeks were kept for brainstorming and deciding on all the technical dependencies in the game and installing all the software required. The last three weeks were kept for testing, debugging and gathering the results from different subjects.

The first major consideration was to build an application for a touch screen that makes it different from all other platforms with controllers. Unity provides the feature of accepting the touch and providing the data to the system [17]. As the project was using the smartphone with Google Cardboard, which has only one control that taps at the center of the screen whenever clicked, the coordinates remained the same.

However, an event was triggered whenever a screen was clicked. This event triggered function was used to flip the cards and to find the matches on the current card and the previous card.

The user playing the game should not cross the cards from between and also should not cross any of the walls; for that reason, the Box Collider functionality of Unity was used [12]. For this reason, now users cannot cross any of the solid walls makes it fun to play the game as they have to follow the path designed to be followed.

When a user taps the screen using the Google Cardboard, a function is called using the Event trigger functionality of Unity 3D[13]. This feature is used to call a particular function whenever a system event is triggered. Multiple functions could be assigned to a single event system trigger[13].

For Level 2 and Level 5, when the user is moving all over the place to find the cards, it is important to stop the user when they are near the card and point towards it. For this reason "Physics. Raycast" function of Unity was used. This helps to detect an object from a distance and returns true. Using this condition, the movement of the user could be stopped, which makes it feasible for them to select the card and look at it. While moving, it was very hard to point at it and click it as a user would sometimes pass the card before clicking it, making the game non-user-friendly.

For flipping the cards, Material Objects were used, and all the different cards were stored in them separately. When an event trigger calls the function, a "select.cs" script is executed. This script works as the brain of the application. The first card clicked is stored in an object, then when any other card is clicked the two objects are checked if they are the same or not. If they are, nothing happens(nothing should be done when one card is clicked consecutively) and should be flipped back. If the objects are different, then they should be checked if they have the same cards or not. If not, flip the cards back; otherwise, make the cards disappear from the game and decrease the counter for cards left by 2.

Also, after several experiments, the time chosen to display the card to a user after clicking using the Google cardboard was set to 0.5 seconds. If it was less, the game became very difficult to play, and if it was more, it was very easy as the user could click the next card before even the first card was closed.

The results that users see at the top while playing the game have to be implemented using the "canvas" functionality of Unity. Unity has only one canvas in which everything has to be done. All the text and buttons that have to be added could only be added to the ONLY canvas we had. The best plan we could think of is to make the canvas dynamic, and it should move with the camera view(wherever the user looks, the canvas would always be in front of them). On the canvas, we presented the time elapsed and the total no. of cards left in the game. This data kept the user motivated and knew how far they had to go in the game. The counter of cards starts with x and reduces by 2 (x-2) always, as whenever a match is found, two cards are removed from the game. When a user finishes all the cards in the game, a new scene is loaded representing "game is over", the audio is changed, and the total time and the total number of clicks done by the user are shown. That was later used in the study of the brain.

4. Analysis

The hardware chosen for the testing and experimentation was an Android device(Nexus 5). The device has a pretty suitable configuration for the game to run on. The game was developed using the MacBook Pro. The Macbook Pro has the following configuration: 2.7GHz dual-core Intel Core i5, Turbo Boost up to 3.1GHz, with 3MB shared L3 cache, 8GB of 1866MHz LPDDR3 onboard memory and Intel Iris Graphics 6100 [6]. These configurations are much higher than we could get in a smartphone. Every level after implementation has to be tested on the Android phone, as there are performance issues compared to the application running on the Macbook Pro. Several experiments were done to ensure the smoothness of the application and a simple user interface. As all users are not comfortable with a Virtual Reality device, hence it was made sure that whoever plays the game it is easier for them to get along.

The software chosen was unity3D and the language chosen to develop the application was C#. The language chosen to develop the game turned out to be very handy and useful. The language has a lot of inbuilt features. This saved development time and made it easier to implement the game and execute it on an Android smartphone device.

Results obtained from the application were satisfactory and replicated the research as expected by us. The application was easy to use for all the users who have used the Virtual Reality platform before or not. Some children played the game for fun and enjoyed it. Though, we did not get much data from the children. Our leading research was focused on users aged 18-34 as this was the range of age of subjects we tested on.

Results obtained from the projects are shown below.

Figure 4.1: Result of the subjects with age 18-24 marked with RED and 25-24 marked with BLUE.

Subjects playing video games each week with RED indicating 1- 4, BLUE 0 and YELLOW 5-10 hours/week.

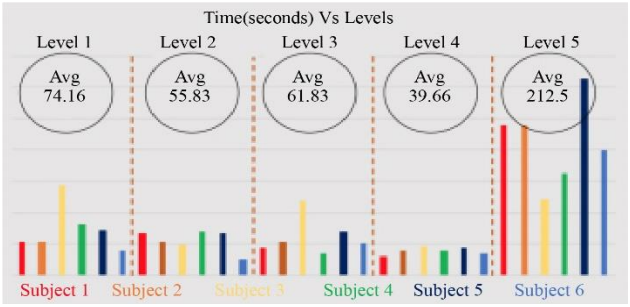

Fig. 4.5. Result of subjects playing video games each week with RED indicating 1-4 and BLUE 0 hours/week

5. Conclusion

The current version of the application has five levels of the game. Where the first two levels are to make the user familiar with the application, before starting the research on the next 3 levels. Level 3 and Level 4 were most useful for the research. Level 5 was a bit long, and some users were not able to complete it or quit. The final result did show that a human brain could remember four items at a time in the iconic memory. Cards used for the study were 8(4 pairs). Later the

cards were increased to 12 and then to 16. The results for the same are shown below. Results show how, with an increase of four cards each time, i.e. 8-12-16, the brain reacts. Specifically, the memory starts taking exponential time to remember it.

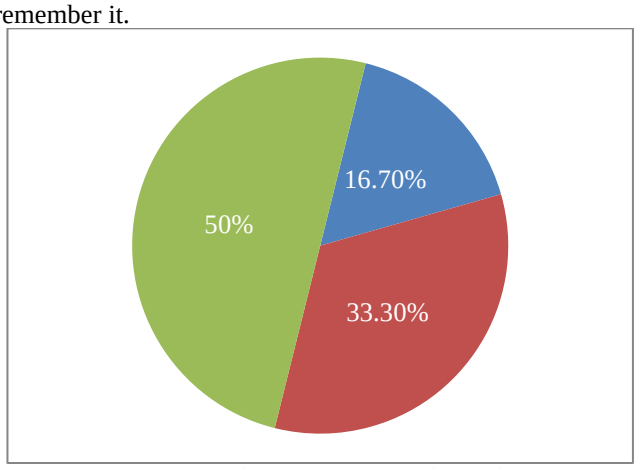

Fig. 5.1. Result generated from the application for the first six subjects

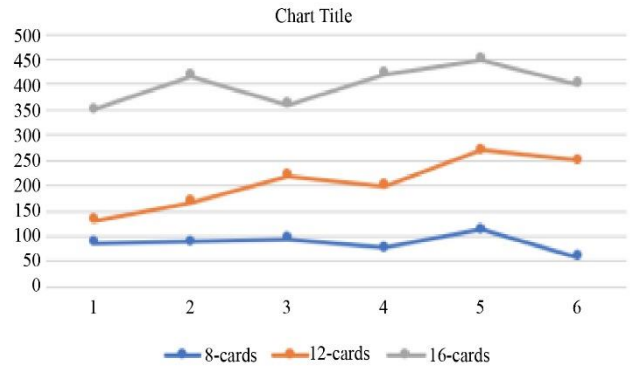

Fig. 5.2. Result of six subjects with different numbers of cards. Where the BLUE line represents 8 cards, the YELLOW line represents 12 cards, and the GRAY line represents 16 cards. All the units are different.

5.1. Future Work

The research can be extended further by adding different levels to the game and recording the responses from the subjects. As of now, the gravity has been disabled, and a subject is on the floor.

In the future, gravity restrictions can be removed so that the subject can even fly, basically doing the movements in the "Z-axis" direction as well. Responses recorded with the movement in the "Z-axis" will show more insight into how the different parts of brain memory work and study can be extended. Secondly, cards are static at a particular position where they are placed, but in future, they can be made to flash. According to the study in the book [18], if an object is flashed within a circular frame or if it is flashed under a bar, then a subject tends to remember the object flashed under the bar. This research was done in a 2D environment. In future, the same can be tried and tested in a virtual environment and a 3D environment. Also, the application could be extended to other VR devices, such as Oculus Rift VR by Microsoft[8], that have more functionality rather than one control as we currently have in Google Cardboard.

5.2. Lessons Learned

Developing a Virtual Reality based game on a mobile device to integrate it with a Google cardboard requires in-depth knowledge of hardware as well as Unity 3D. The reason is that performance is the top concern since you start developing the game. At each level you create or add anything to the application, it becomes a must to check each time for performance. All the software keeps on updating and sometimes even deprecates some functions. Codes should be well documented all the time so that if a function needs to be replaced, the person reading the code knows what the function is doing.

References

[1] Download Android Studio and SDK Tools - Android Studio. [Online]. Available: https://developer.android.com/studio

[2] Google Cardboard, Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/ Google_Cardboard

[3] How Should I Decide if I Should Use c#, javascript (unityscript) or Boo for My Project? - Unity Answers. [Online]. Available: https://discussions.unity.com/t/how-should-i-decide-if-i-should-use-c-javascript-unityscript-or-boo-for-my-project/114

[4] How We Remember, and Why We Forget, Brain Connection, brainHQ, 2023. [Online]. Available: https://www.brainhq.com/brain-connection/how-we-remember-and-why-we-forget/

[5] Richard Gray, Awkward! Why You Think You Recognise Strangers in the Street: Human Brain Behaves Like a Rat Lost in a Maze, Daily Mail Online, 2015. [Online]. Available: https://www.dailymail.co.uk/sciencetech/article-3205000/Awkward-think-recognise-strangers-street-Human-brain-behaves-like-rat-lost-maze.html

[6] MacBook Pro, Technical Specifications 2015, Apple. https://support.apple.com/en-in/111959

[7] Nexus 5, Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Nexus_5

[8] Oculus Rift, Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Oculus_ Rift.

[9] Out of Sight, Out of Mind – Idioms, The Free Dictionary. [Online]. Available: https://idioms.thefreedictionary.com/out+of+sight%2C+out+of+mind

[10] Scary Horror Music - Best Halloween Mega Remix 2015 - Creepy Anthem - New Dark Songs, Youtube. [Online]. Available: https://www.youtube.com/watch?v=cCbd38uibNg

[11] Audio files, Manual, Unity Documentation. [Online]. Available: https://docs.unity3d.com/Manual/AudioFiles.html.

[12] Scripting API: Boxcollider, Unity Documentation. [Online]. Available: https://docs.unity3d.com/ScriptReference/BoxCollider.html
[13] Scripting API: Eventtrigger, Unity Documentation. [Online]. Available: https://docs.unity3d.com/2018.2/Documentation/ScriptReference/EventSystems.EventTrigger.html
[14] System Requirements for Unity 2022.3, Unity Documentation. [Online]. Available: https://docs.unity3d.com/Manual/system-requirements.html#player
[15] Unity (Game Engine), Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/ Unity_(game_engine)
[16] Whats With the Power of 2 Restriction?, Unity Community. [Online]. Available: https://discussions.unity.com/t/whats-with-the-power-of-2-restriction/417795
[17] Elisabeth Beinke et al., "*Memory Dash-a Multiplayer iPhone Game*," PhD Thesis, Worcester Polytechnic Institute, pp. 1-65, 2009. [Google Scholar] [Publisher Link]
[18] Stephen E Palmer, *Vision Science: Photons to Phenomenology*, MIT Press, pp. 1-810, 1999. [Google Scholar] [Publisher Link]