

**A  
Project Report  
On  
"Educational App"**

**Prepared by**  
Mit Patel (15IT093)

**Under the guidance of**  
Prof. Pritesh Prajapati

A Report Submitted to  
Charotar University of Science and Technology  
for Partial Fulfillment of the Requirements for the  
Degree of Bachelor of Technology  
in Information Technology  
( 8<sup>th</sup> Semester Software Project Major-IT407 )

**Submitted at**



**SMT. KUNDANBEN DINSHA PATEL DEPARTMENT OF  
INFORMATION TECHNOLOGY  
Chandubhai S. Patel Institute of Technology  
At: Changa, Dist: Anand – 388421  
April 2019**

## **CANDIDATE’S DECLARATION**

We hereby declare that the project entitled “**Educational App**” is our own work conducted under the guidance of **Ms. Payal Joshi and Prof. Pritesh Prajapati**.

We further declare that to the best of our knowledge, the project for B. Tech does not contain any part of the work, which has been submitted for the award of any degree either in this University or in other University without proper citation.

**Mit Patel (15IT093)**

**Prof. Pritesh Prajapati**

**Assistant Professor,**

**Smt. Kundanben Dinsha Patel Department of Information Technology,**

**Faculty of Technology & Engineering,**

**Changa – 388421.**



# HariSkyline

April 19<sup>th</sup>, 2019

## Internship Completion Certificate

This is to certify that **Mr. Nikunj P. Rathod** and **Mr. Mit A. Patel** from **Charotar University of Science & Technology**, has successfully completed their internship project named **Educational App with AR/VR in Unity 3D** at Hari Skyline from December 17<sup>th</sup>, 2018 to March 29<sup>th</sup>, 2019.

During their internship, they were exposed to the various activities in Unity 3D.

We found them extremely inquisitive and hard working. They were very much interested to learn the functions of our core division and also willing to put their best efforts and get in to the depth of the subject to understand it better.

Their association with us was very fruitful and we wish them all the best in their future endeavours.

For Hariskyline



Authorized Signatory  
Payal H. Joshi  
(Project Manager)



**CHARUSAT**  
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

## CERTIFICATE

This is to certify that the report entitled “**Educational App**” is a bonafied work carried out by **Mr. Mit Patel (15IT093)** under the guidance and supervision of **Ms. Payal Joshi & Prof. Pritesh Prajapati** for the subject **Software Project Major (IT407)** of 8<sup>th</sup> Semester of Bachelor of Technology in **Information Technology** at Faculty of Technology & Engineering – CHARUSAT, Gujarat.

To the best of my knowledge and belief, this work embodies the work of candidate themselves, has duly been completed, and fulfills the requirement of the ordinance relating to the B.Tech. Degree of the University and is up to the standard in respect of content, presentation and language for being referred to the examiner.

Under supervision of,

Prof. Pritesh Prajapati

Assistant Professor

Smt. Kundanben Dinsha Patel Department of  
Information Technology  
CSPIT, Changa, Gujarat.

Ms. Payal Joshi

Project Manager

Hari Skyline,  
B-704,705 Business Empire Hub,  
Sola, Ahmedabad, Gujarat.

Prof. Parth Shah

Head & Associate Professor

Smt. Kundanben Dinsha Patel Department of Information  
CSPIT, Changa, Gujarat.

---

---

**Chandubhai S Patel Institute of Technology**

At: Changa, Ta. Petlad, Dist. Anand, PIN: 388 421. Gujarat

## ABSTRACT

Unity is the primary tool that was used to create this project. It was chosen primarily because of the fact that it is free to use and has a vast amount of support available from the community including an asset store from which some free assets can be added to a project in order to enhance it as well as saving time and money.

In **Educational App** the app's focus has a strong connection to the KIDS which are bad at mathematics. A game is much more than just its software. It has to content to become enjoyable. User will enhance learning basic by starting from basics and to higher level like VEDIC mathematics. The project was carried out to develop a 3D game that could be used for fun and as an efficient way of teaching. The player could play on or off the Background Music and the author and start or end this game on the GUI scene. There is also AR and VR additional functionality. The app is available over the internet through [nikunj Rathod.me](http://nikunj Rathod.me) site and will be available in Play Store.

## Acknowledgement

The Objective of the Final Semester industrial training is to provide exposure to the industrial technicalities and work culture to a student. Hence, we are extremely thankful to our institution to provide us with this opportunity. It has been a great learning experience on our technical as well as personal aspect.

We are extremely thankful to our internal guide **Prof. Pritesh Prajapati** (Assistant Professor of IT Department CSPIT, CHARUSAT) whose deep knowledge of the subject and a helping attitude towards the students has always provided us with the strength to achieve the objective. We are very much thankful to sir who's never dying attitude to work for the interest of the Student has always been in itself a great inspiration.

We also extend our sincere thanks to our external guide **Ms. Payal Joshi** (Project Manager, HariSkyline, Ahmedabad) whose zeal and technical know-how kept up the pace of the project till the end. He was my true guardian during this period with just the right mix of discipline and immense care. We heartily thanking **Ms. Payal Joshi** (Project Manager, HariSkyline, Ahmedabad)) for guiding us during the project.

We are also thankful to **Dr. Parth Shah** (Head of Department, Information Technology, CSPIT) for always encouraging me to do even better and always providing support to accomplish any t, ask.

We are also thankful to the entire **HariSkyline Team** for giving us a friendly work culture.

We are also thankful to the entire **IT Department** of **CSPIT** for their immense support and help during the entire project work.

Mit Patel (15IT093)

# TABLE OF CONTENTS

<b>Abstract.....</b>	<b>iii</b>
<b>Acknowledgement.....</b>	<b>iv</b>
<b>List of Figures.....</b>	<b>vii</b>
<b>List of Tables.....</b>	<b>viii</b>
<b>1.0 Introduction.....</b>	<b>1</b>
1.1 Project Overview.....	1
1.2 Problem Definition.....	3
1.3 Purpose.....	3
1.4 Objective.....	4
1.5 Scope.....	4
1.6 Tools and Technology Used.....	4
<b>2.0 Project Management.....</b>	<b>6</b>
2.1 Project Planning.....	6
2.2 Project Work Scheduling.....	10
2.3 Feasibility Study.....	11
<b>3.0 System Requirements Study.....</b>	<b>13</b>
3.1 User Characteristics.....	13
3.2 Hardware and Software Requirements.....	15
3.3 ASSUMPTIONS AND DEPENDENCIES.....	16
<b>4.0 System Analysis.....</b>	<b>17</b>
4.1 Study of Existing Solution.....	17
4.2 Limitations of Existing Solution.....	17
4.3 Requirements of Proposed System.....	17
4.4 System WorkFlow.....	19
4.5 Sequence Diagrams.....	21
4.6 State Chart Diagrams.....	21
<b>5.0 System Design.....</b>	<b>22</b>
5.1 Screen Layout.....	22
5.2 Method Pseudo Code.....	38
<b>6.0 Testing.....</b>	<b>40</b>
6.1 Testing Plan.....	40
6.2 References Suspension criteria & resumption requirements.....	41

6.3	Testing Strategy.....	42
6.4	Test Suites Design.....	43
<b>7.0</b>	<b>Future Enhancements.....</b>	<b>44</b>
7.1	Limitations.....	44
7.2	Future Enhancements.....	44
<b>8.0</b>	<b>Conclusion. ....</b>	<b>45</b>
8.1	Self-Analysis of Project Viabilities.....	45
8.2	Problems Encountered and their Solutions.....	45
8.3	Summary of Project Work.....	46
	References.....	47



## List of Figures

Figure Index	Figure Name	Page Number
Figure 2.1	Iteration cycle of an Agile project .....	7
Figure 2.2	Evolution of a Software during Agile Process.....	7
Figure 2.3	Sample Jira Dashboard.....	9
Figure 2.4	Gantt chart.....	10
Figure 2.5	Project Scheduling Chart .....	11
Figure 3.1	Use-Case Diagram Main Menu.....	13
Figure 3.2	Use-Case Diagram Level.....	14
Figure 3.3	Use-Case Diagram Scene.....	15
Figure 4.1	Activity Diagram.....	20
Figure 4.2	Sequence Diagram.....	21
Figure 4.2	State Chart Diagram.....	21
Figure 5.1	Scene View.....	22
Figure 5.2	Game View.....	23
Figure 5.3	Project Menu.....	24
Figure 5.4	Hierarchy Window.....	25
Figure 5.5	Inspector Window.....	26
Figure 5.6	Asset Store.....	26
Figure 5.7	Unity Recorder.....	27
Figure 5.8	Animation.....	28
Figure 5.9	Animator.....	29
Figure 5.10	Timeline.....	30
Figure 5.11	Particle System.....	31
Figure 5.12	Vuforia.....	33
Figure 5.13	Vuforia (Unity).....	33
Figure 5.14	Google VR.....	35
Figure 5.15	Windows App.....	36
Figure 5.16	Windows App.....	36
Figure 5.17	Android App.....	37
Figure 5.18	Android App.....	37
Figure 5.19	Web App.....	38
Figure 5.20	Web App.....	38

## List of Tables

Table Index	Table Name	Page Number
Table 5.1	Supported Audio format in Unity.....	32
Table 6.2	Testing Strategy.....	42
Table 6.3	Test Cases.....	43

## **1.0 Introduction**

### **1.1 Project Overview**

In the fast growing field of software engineering and development and even more rapidly growing sector of game development the future is hard to predict. We are working with this game and as part of our degree we choose this type of work for doing better with development cycle, development period, graphics, scripting, adopting new technology, animation. In general software project is a project focusing on the creation of software. Consequently, Success can be measured by taking a look at the resulting software. In a game project, the product is a game. But and here comes the point: A game is much more than just its software. It has to provide content to become enjoyable. Just like a web server without content the server is useless, and the quality cannot be measured. This has an important effect on the game project as a whole. The software part of the project is not the only one, and it must be considered in connection to all other parts: The environment of the game, models, game plays, the artwork, and so on.

### **Unity**

Unity is a cross platform game engine with its own built in IDE which allows for development of either 2D or 3D games. Unity is primarily used to create games for a variety of platforms such as web, desktop, consoles and most recently, mobile platforms.

Unity is the primary tool that was used to create this project. It was chosen primarily because of the fact that it is free to use and has a vast amount of support available from the community including an asset store from which some free assets can be added to a project in order to enhance it as well as saving time and money.

Unity features a main scene view which displays the visual elements of the project such as the terrains, models, etc. Any object that is used in Unity, whether it's a terrain, a shape, a model, is referred to as a GameObject. These game objects can be modified using the Inspector view. With the Inspector view you can see the transform, rotation and scale information of the selected GameObject. You can also see the scripts that the object is using and the components attached to it.

Unity offers support for assets that are created in a wide variety of file formats, such as fbx. Assets created in other programs can simply be added to a game project in Unity by

choosing them from the saved location on your computer. Unity makes adding imported art assets to the game scene as simple as drag and drop in the desired location.

Unity's scripting is based primarily on MonoDevelop which is an open source implementation within the .Net framework. When working with Unity, you are given the option of using JavaScript or C#. Unity makes use of MonoDevelop, though developers are free to use whatever scripting tool they desire.

Unity supports project deployment on multiple platforms. When a project is finished, developers can choose which platform they would like their project to run on, e.g. PlayStation 4, Xbox One, Window, Linux, Android, IOS and many others. We are currently using Unity version 2018.2.2 in the development of our game.

## **Unity Asset Store**

The asset store is a resource that is located within the Unity editor and is also available online. The store currently consists of thousands of packages which are divided into sections such as environments, characters, textures/materials etc. that are available to developers to use within their projects. They can be free or paid, depending on the level of work involved in the asset. Without the asset store, all the objects in the game would have to be created manually in a program such as blender and would become extremely time consuming. The asset store was used for all the models and some other websites.

## **MonoDevelop**

MonoDevelop is an IDE which is installed with Unity 3D. It allows you to create and edit C# and JavaScript files used in the game. It differs from the standard IDE because it has in-built functions and auto-completion with coding specifically relating to Unity. MonoDevelop is able to interact with the GameObjects used in Unity. It can access their co-ordinates, scripts attached to them, their tags, etc. These are very useful in creating powerful scripts. It also allows variables to be edited from Unity so changes do not need to be made in the scripts every time. This made it really easy to test different things in my game as I could easily change variables such as distance or try different images or sound clips. Errors that appear within the IDE are also shown in the Unity Debugging Window which recompiles regularly. Double clicking the error will bring you to the line of code in the script that is causing the problem.

## **C#**

C# is a modern programming language which allows developers to create applications which will then run on the .NET framework. I had been introduced to C# in one of my modules in college which is a reason Unity was chosen. As I had been introduced to C# in college I was familiar with it and it made it that bit easier when trying to fix problems in scripts.

## **JavaScript**

JavaScript is a multi-platform, object-oriented scripting language. JavaScript can be connected to the objects of its environment to allow programmatic control over them.

### **1.2 Project Definition**

Unity is a multiplatform game engine that supports 2D and 3D graphics. Games were commonly scripted in C# and UnityScript, similar to JavaScript, which makes them much easier to understand. Earlier people only played games on consoles or computers until now people are more willing to play games on mobile phones. So there are lot of educational apps in market for educational purpose. The app's focus has a strong connection to the KIDS which are bad at mathematics. Also one of the most important subject is mathematics itself. We are going to enhance learning basic by Starting from basics and to higher level like VEDIC mathematics.

### **1.3 Purpose**

In recent years, game industries have entered a stage of rapid development. Also exporting to android version is easy also app will be playable in AR/VR environment on PC as well as mobile supported devices. Majorly Kids will be using this app. More different types of games have appeared on a variety of new platforms, because relatively low cost and huge profits have motivated more people to get involved in this area. . In general software project is a project focusing on the creation of software. Consequently, Success can be measured by taking a look at the resulting software. In a game project, the product is a game. But and here comes the point: A game is much more than just its software. It has to provide content to become enjoyable.

## 1.4 Objective

The app's focus has a strong connection to the KIDS which are bad at mathematics. Many mathematics type game are already available in Play Store. But they are limited to some portion only like addition type basic operations. We are going to enhance learning basic by Starting from basics and to higher level like VEDIC mathematics. Android Studio is globally available for making android apps. But we are using Unity 3D as it is simpler with less code and also is more powerful to make interactive environments which kids like and so on.

## 1.5 Scope

Following are the features in the scope.

- Unity3d installation.
- Used various technologies like C Sharp, Illustrator, Photoshop, 3DSMax, V-ray, After effects, Substance painter and Designer, Cine-Machine etc.
- Support multiple Platform like Windows, Web and Android.
- KIDS can use this app for learning
- Basic 1-100 in words and A-Z with objects
- 1-10 math tables.
- Basics of mathematics operation and Vedic mathematics.
- AR and VR additional functionality.

## 1.6 Tools and Technology Used:

Unity is the primary tool that was used to create this project. It was chosen primarily because of the fact that it is free to use and has a vast amount of support available from the community including an asset store from which some free assets can be added to a project in order to enhance it as well as saving time and money.

### Tools:

- Unity 3D 2018.2.2
- Illustrator
- Photoshop
- 3DSMax

- V-ray
- After effects
- Substance painter and Designer
- Cine-Machine
- Visual Studio Code

**Technology:**

- HTML and CSS
- C Sharp

## 2.0 Project Management

### 2.1 Project Planning

#### 2.1.1 Project Development Approach (Process Model Used)

Agile software development methodology is a process for developing software which differs significantly from other methodologies. In English, Agile means ‘ability to move quickly and easily’ and responding swiftly to change – this is a key aspect of Agile software development as well.

With **Agile development** methodology –

- In the Agile methodology, each project is broken up into several ‘Iterations’.
- All Iterations should be of the same time duration (between 2 to 8 weeks).
- At the end of each iteration, a working product should be delivered.
- In simple terms, in the Agile approach the project will be broken up into 10 releases (assuming each iteration is set to last 4 weeks).
- Rather than spending 1.5 months on requirements gathering, in Agile software development, the team will decide the basic core features that are required in the product and decide which of these features can be developed in the first iteration.
- Any remaining features that cannot be delivered in the first iteration will be taken up in the next iteration or subsequent iterations, based on priority.
- At the end of the first iterations, the team will deliver a working software with the features that were finalized for that iteration.
- There will be 10 iterations and at the end of each iteration, the customer is delivered a working software that is incrementally enhanced and updated with the features that were shortlisted for that iteration.



The iteration cycle of an Agile project is shown in the image below.



**Figure 2.1 Iteration cycle of an Agile project**

This approach allows the customer to interact and work with functioning software at the end of each iteration and provide feedback on it. This approach allows teams to take up changes more easily and make course corrections if needed. In the Agile approach, software is developed and released incrementally in the iterations. An example of how software may evolve through iterations is shown in the image below.



**Figure 2.2 Evolution of a Software During Agile Process**

Agile methodology gives more importance to collaboration within the team, collaboration with the customer, responding to change and delivering working software. In the traditional

approach, each job function does its job and hands over to the next job function. The previous job functions have to sign off before it is handed over the next job function authenticating that the job is full and complete in all aspects. For example, Requirement gathering is completed and handed over to the design phase and it is subsequently handed over to the development and later to testing and rework. Each job function is a phase by itself.

In an Agile way of working, each feature is completed in terms of design, development, code, testing and rework, before the feature is called done. There are no separate phases and all the work is done in single phase only.

### **Advantages of Agile Methodology:**

- In Agile methodology the delivery of software is unremitting.
- The customers are satisfied because after every Sprint working feature of the software is delivered to them.
- Customers can have a look of the working feature which fulfilled their expectations.
- If the customers have any feedback or any change in the feature then it can be accommodated in the current release of the product.
- In Agile methodology, the daily interactions are required between the business people and the developers.
- In this methodology, attention is paid to the good design of the product.
- Changes in the requirements are accepted even in the later stages of the development.

### **Why We Use Agile Methodology?**

Our company finds Agile methodology as the most suitable for this project since it requires frequent updates, meetings, updating in requirements, regular adaptation to changing circumstances.

Also, We give preference to Customer First Always, hence the method serves our purpose to provide quick and reliable solutions to our customers.

#### **2.1.2 Tools Used**

I learnt the usage of various project management tools during my course of training and project work at HariSkyline including

- Jira
- Bamboo
- Bitbucket

## JIRA

Basically, JIRA is Atlassian's popular project- and issue-tracking platform used by development and technical support teams to get more work done and faster. It's available on-demand through a monthly subscription (software-as-a-service), or it can be deployed on your own servers for an upfront license.

It combines:

- issue tracking - a software application that allows to record and follow the progress of every problem or "issue" that user identifies until the problem is resolved.
- agile project management - an iterative approach to planning and guiding project processes.

### Sample Jira Dashboard:-

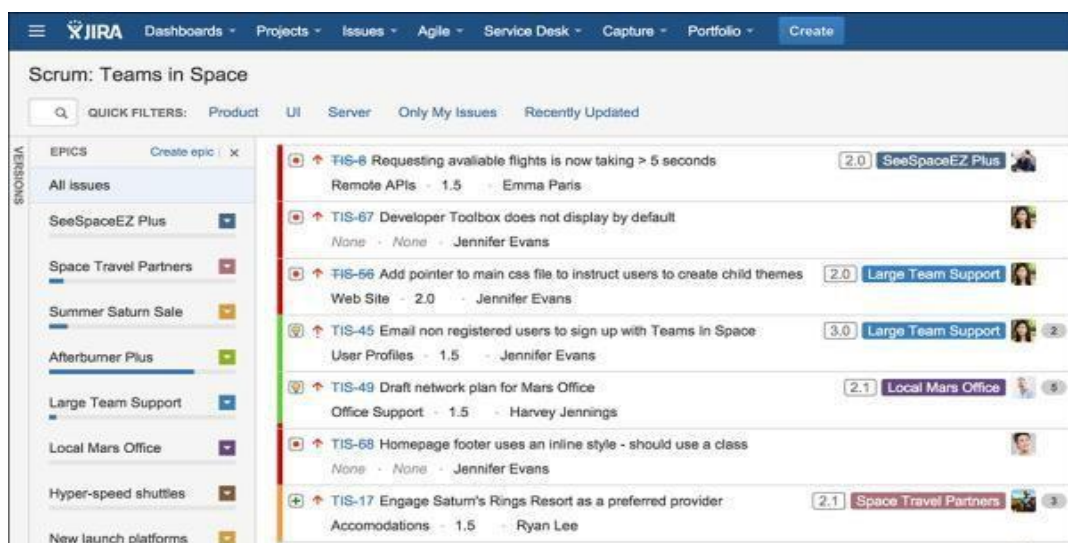


Figure 2.3 Sample Jira Dashboard

## Bamboo

Bamboo is a continuous integration (CI) server that can be used to automate the release management for a software application, creating a continuous delivery pipeline.

### What does this mean?

CI is a software development methodology in which a build, unit tests and integration tests

are performed, or triggered, whenever code is committed to the repository, to ensure that new changes integrate well into the existing code base. Integration builds provide early 'fail fast' feedback on the quality of new changes.

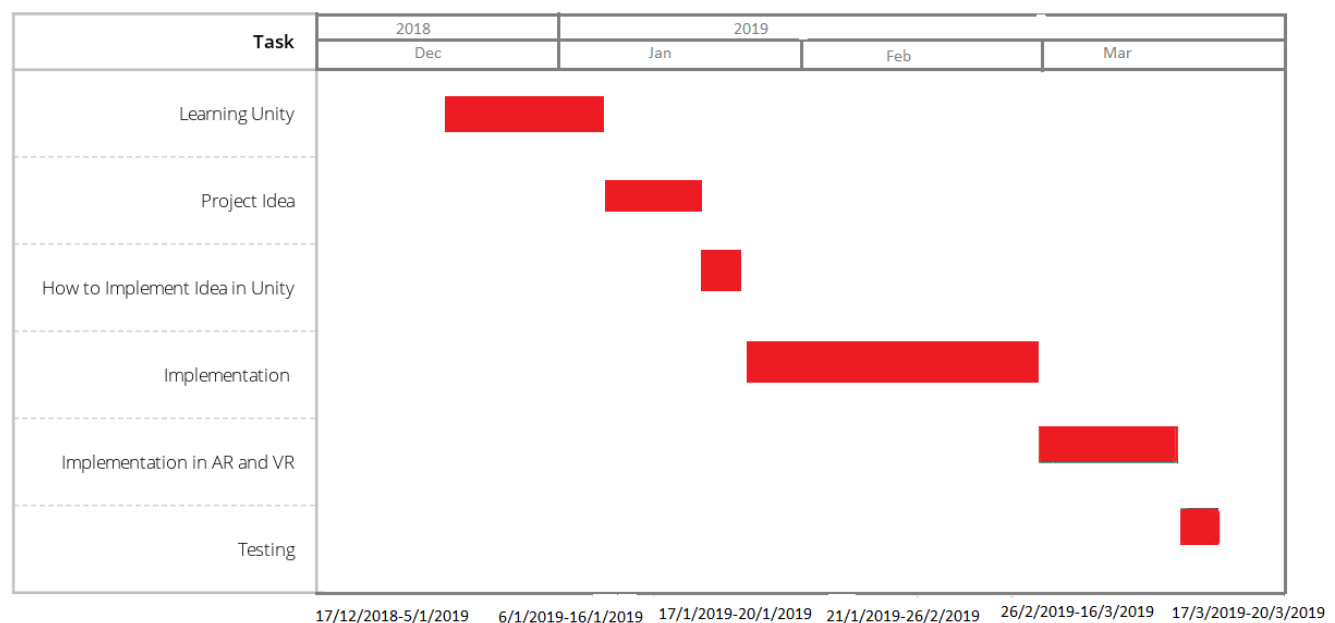
Release management describes the steps that are typically performed to release a software application, including building and functional testing, tagging releases, assigning versions, and deploying and activating the new version in production.

## Bitbucket

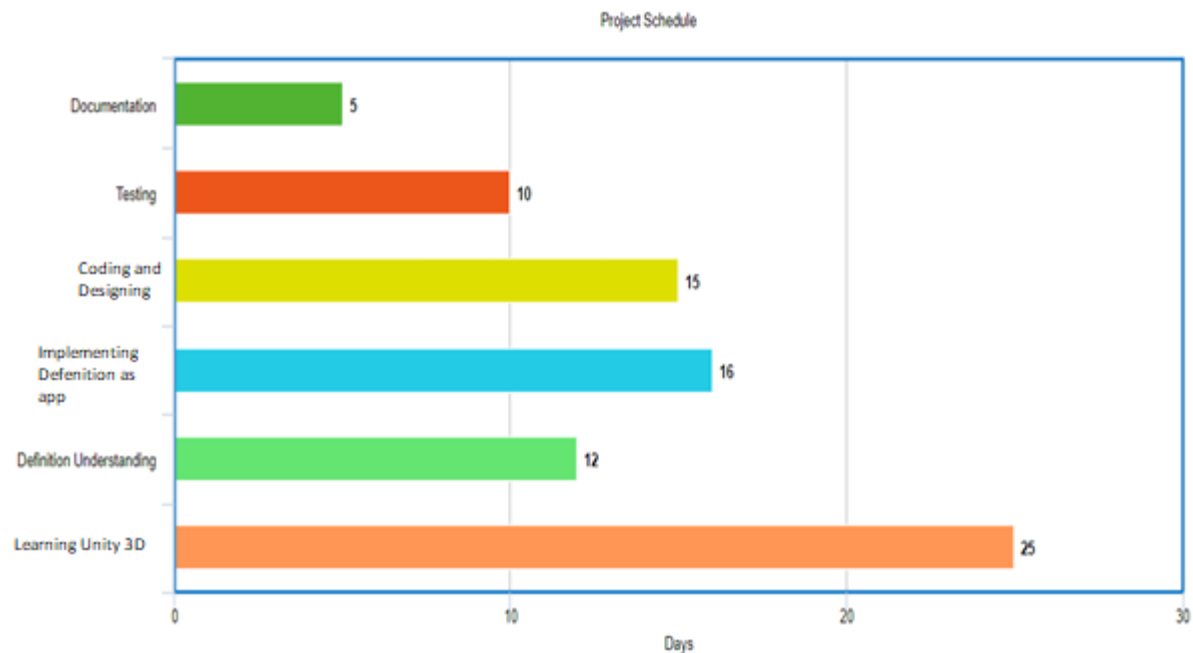
Bitbucket is a web-based version control repository hosting service owned by Atlassian, for source code and development projects that use either Mercurial or Git revision control systems. Bitbucket integrates with other Atlassian software like Jira, HipChat, Confluence and Bamboo.

It is similar to GitHub, which primarily uses Git. Bitbucket has traditionally marketed its services to professional developers with private proprietary software code, especially since being acquired by Atlassian in 2010.

## 2.2 Project Work Scheduling



**Figure 2.4 Gantt Chart**



**Fig 2.5 Project Scheduling Chart**

## 2.3 Feasibility Study

Once the scope has been identified, it is reasonable to ask whether we can build software that meets this scope. Is this project feasible?

The feasibility of software can be tested in four dimensions:

### 2.3.1 Technical Feasibility

What we have planned to implement is technically feasible. Do we have a sufficient amount of knowledge or technology to make it a reality? And the answer is fairly easy because we have found out that we can use Unity to make anything imaginary to reality. So we have all the tools already built in unity. We have tried to maintain the coordination between development, design, and testing. Now we can conclude that the system is technically feasible.

### 2.3.2 Time Schedule Feasibility

We checked whether our system can be ready in time without any error. We have planned all its phase keeping the aspect in our mind, that if we find any bug or error after testing phase then we can move our deadline to 2-4 days, as we set our deadline before the actual

submission date to the client. It requires a minimum of 3 months for the implementation of the complete project with all the features implemented. This also includes the testing and debugging phase.

### **2.3.3 Operational Feasibility**

How the project will work and who will use it, all such concerns arise in this phase. We have to study what the existing system's problem is, and is it worth solving or not.

Most of the current apps are limited to some narrow idea. They are limited to some portion only like addition type basic operations. All the ideas with additional functionality is added.

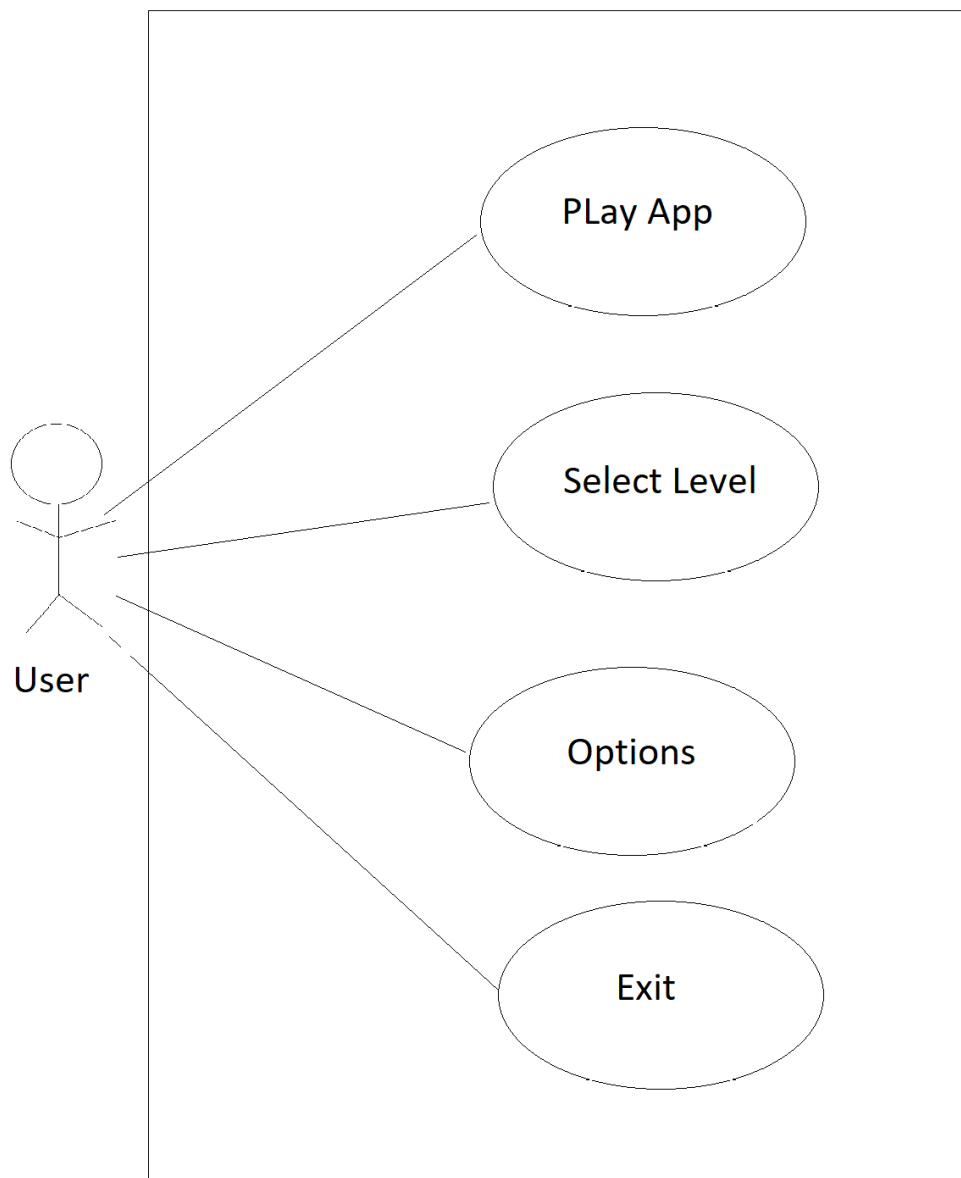
### **2.3.4 Implementation Feasibility**

Unity features a main scene view which displays the visual elements of the project such as the terrains, models, etc. Any object that is used in Unity, whether it's a terrain, a shape, a model, is referred to as a GameObject. Unity supports project deployment on multiple platforms. When a project is finished, developers can choose which platform they would like their project to run on, e.g. PlayStation 4, Xbox One, Window, Linux, Android, IOS and many others.

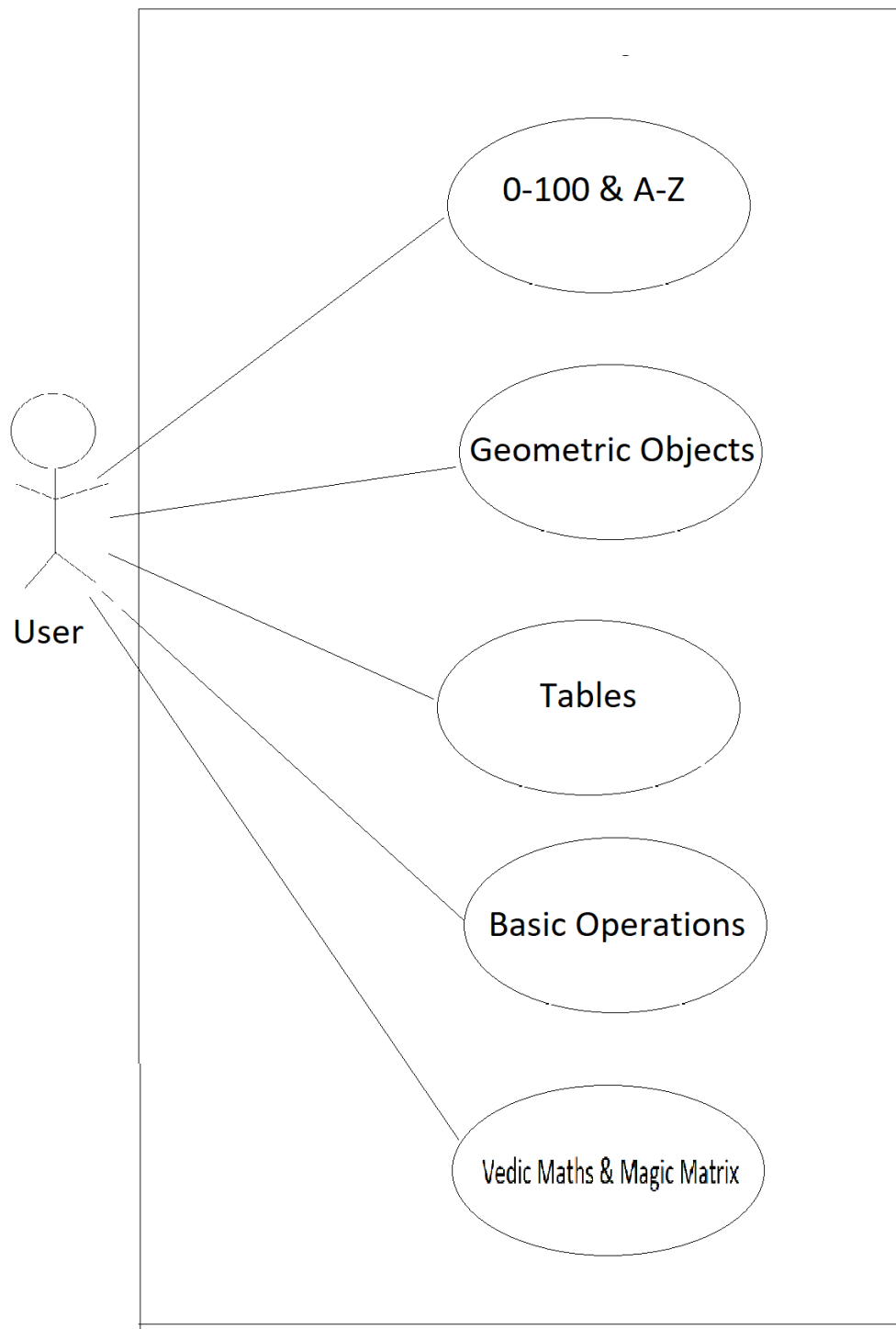
### 3.0 System Requirements Study

#### 3.1 User Characteristics

This application is intended to be mostly used by the KIDS to study mathematics.

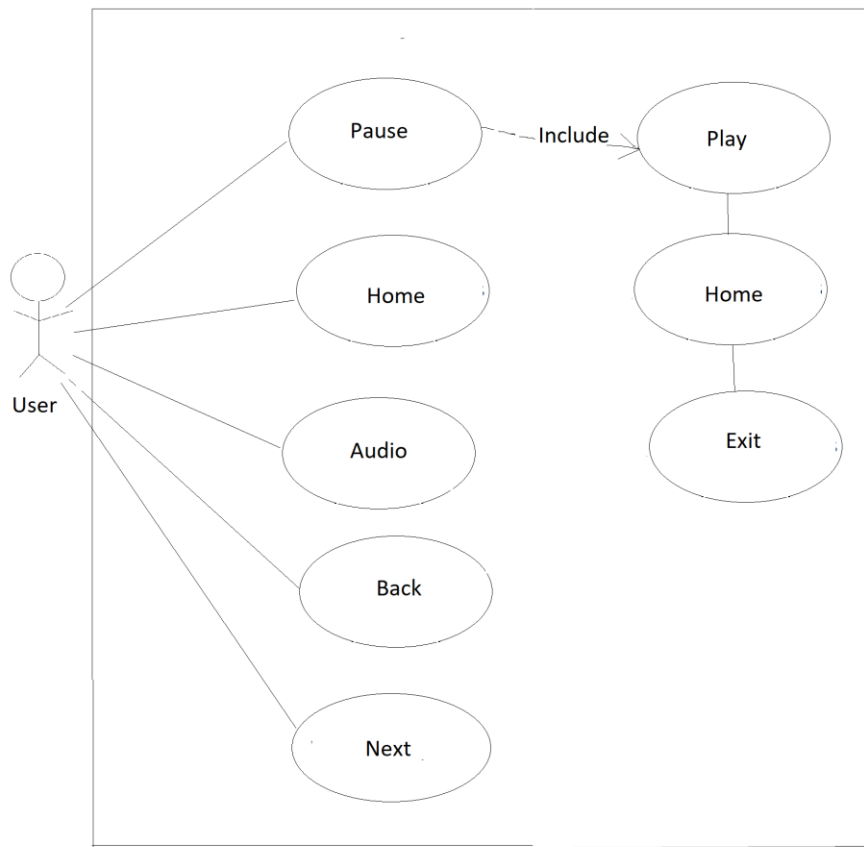


**Fig 3.1 Main Menu Use-Case Diagram**



**Fig 3.2 Levels Use-Case Diagram**





**Fig 3.3 Scene Use-Case Diagram**

## **3.2 Hardware and Software Requirements**

### **3.2.1 Hardware (Recommended)**

- OS: Windows 7 SP1+, 8, 10, 64-bit versions only; macOS 10.11+
- CPU: SSE2 instruction set support.
- GPU: Graphics card with DX10 (shader model 4.0) capabilities
- VR Box & gyroscope sensor in mobile phones for VR app

### **3.2.2 Software**

- Software: Unity 3d, C Sharp, illustrator, photoshop, 3DSMax, After effects, Substance

painter and Designer, CineMachine etc.

- Android 6.0+ (Mobile App)
- Browsers: Chrome, Firefox, Safari (Web App)

### **3.3 ASSUMPTIONS AND DEPENDENCIES**

Following are the assumptions taken while documenting the report:

- The coding is error free
- The instance should have enough storage to run the app.
- User has knowledge of the English language
- All the software and hardware requirements are met

Following are the dependencies:

- The application is dependent only on working on android, windows and web based.

## **4.0 System Analysis**

### **4.1 Study of Existing Solution**

In the Current Apps, there are many apps available in PlayStore.

### **4.2 Limitations of Existing Solution**

The problem in the existing System is that it is not very reliable, there are also a lot of educational apps in market for educational purpose. Our app's focus has a strong connection to the KIDS which are bad at mathematics. Also one of the most important subject is mathematics itself. We are going to enhance learning basic by Starting from basics and to higher level like VEDIC mathematics.

### **4.3 Requirements of Proposed System**

#### **4.3.1 Functional Requirements**

Functional requirements define the internal workings of the software i.e. the calculations, technical details, data manipulation and processing and other specific functionality that show how the use cases are to be satisfied.

The functional requirements of the application are mentioned as follows:

- There should be main menu in the app.
- Thereby though navigating main menu user can choose the correct option required for their convenience.
- There are several scenes for A-Z & 1-10 characters. Afterwards basic mathematics, shapes, operations and symbols for understanding.
- There will be mathematics game like quiz for all operations like addition, subtraction, multiplication, division and their combination.
- Then there will be Basic Vedic mathematics operation for computing operations hastily.

- There will be version of app in VR/AR.

#### **4.3.2 Non-Functional Requirements**

The Non-Functional Requirements are as follows:

- **Usability**

The UI for the app should be easy to use and effective. Adding more functionality into the app may overwhelm the user which should be avoided. Mostly KIDS below 12 years will use the app.

- **Accuracy**

As we develop the application, we must make the app that is very accurate in its functions. We need to make sure that all the scenes should be executed successfully.

- **Reliability**

All possible errors should get handled during the development phase. In case of any error, user should be acknowledged of the error with a meaningful message.

- **Performance**

All scenes of app should be optimized and should take as minimum time as possible to execute.

- **Availability**

For all users free of charge on Android 6.0 and up. Also for windows and web support.

- **Roles of different users to be implemented**

Educational app have normal user. Every user with apk can run the app.

- **Browser compatibility of the application**

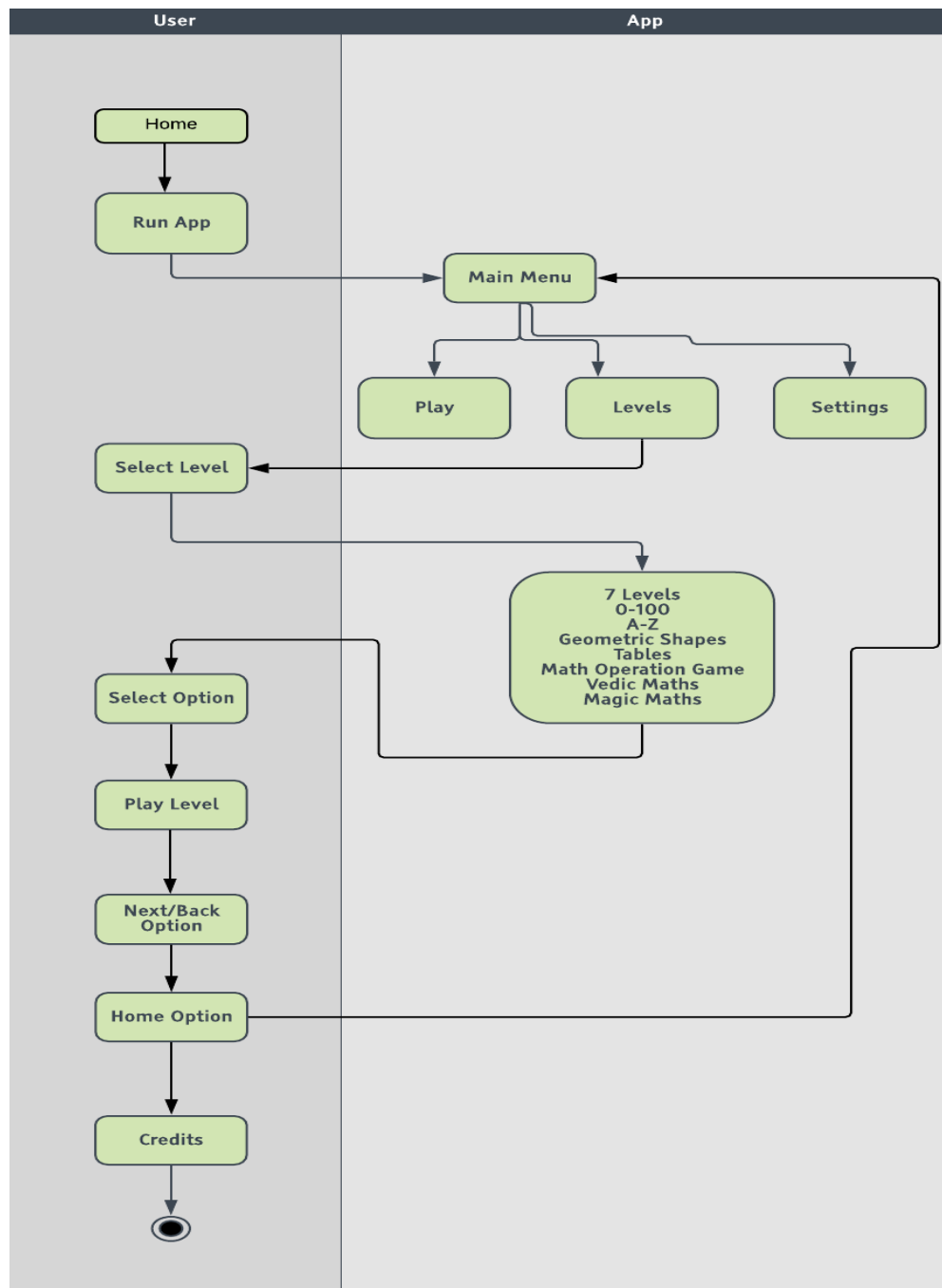
Educational app is supported by Chrome, Firefox, Safari, Chromium, Internet Explorer and Microsoft Edge browser.

- **List of test suits**

Educational app should be well tested with proper documentation.

#### **4.4 System Workflow (Activity Diagram)**

To use Educational App, it's just like a normal Android app. Firstly there is main menu with several options. Selecting Play will directly open the first level. There is also option of selecting levels as per users understanding of mathematics. There is settings option where user can change graphics and Audio of the App. Lastly there is close button to quit the app. In any level there is back and next button to move to respective level. There is also home button to jump to main menu.

**Fig 4.1 Activity Diagram**

### 4.5 Sequence Diagram:

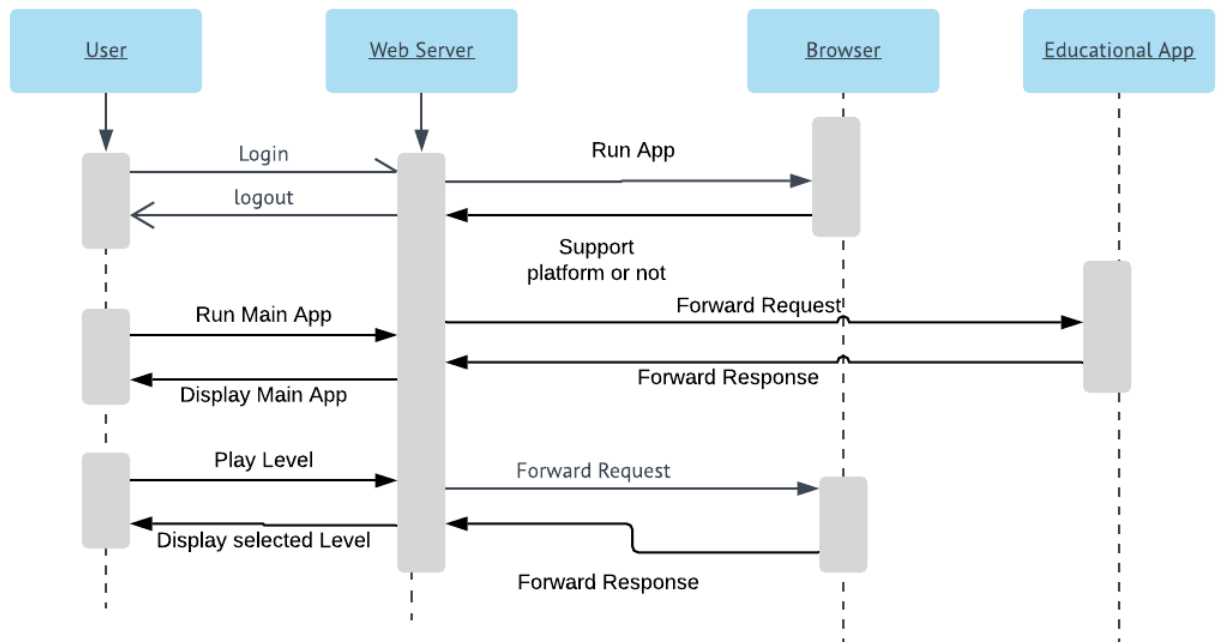


Fig 4.2 Web App Sequence Diagram

### 4.6 State Chart Diagram:

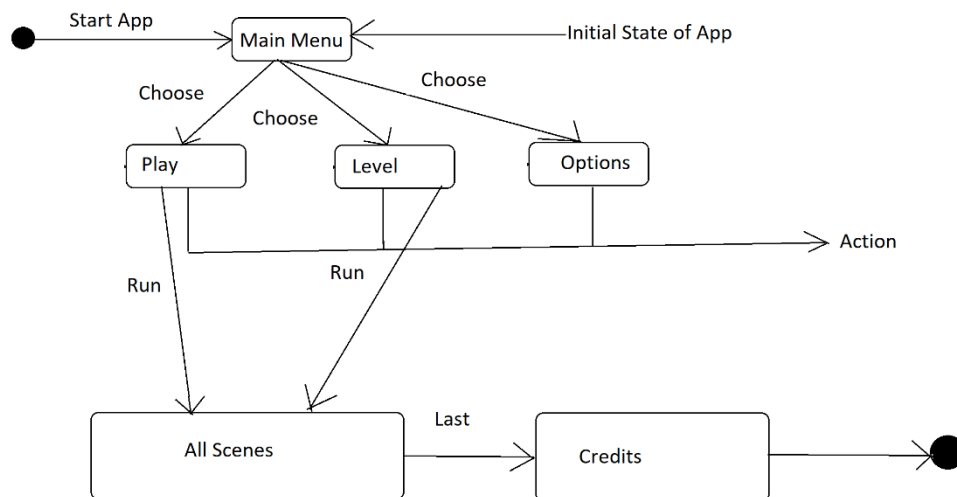


Fig 4.3 State Chart Diagram

## 5.0 System Design

### 5.1 Screen Layout

#### 5.1.1 Scene View

The Scene view is your interactive view into the world you are creating. You will use the Scene view to select and position scenery, characters, cameras, lights, and all other types of Game Object. Being able to Select, manipulate and modify objects in the Scene view are some of the first skills you must learn to begin working in Unity.

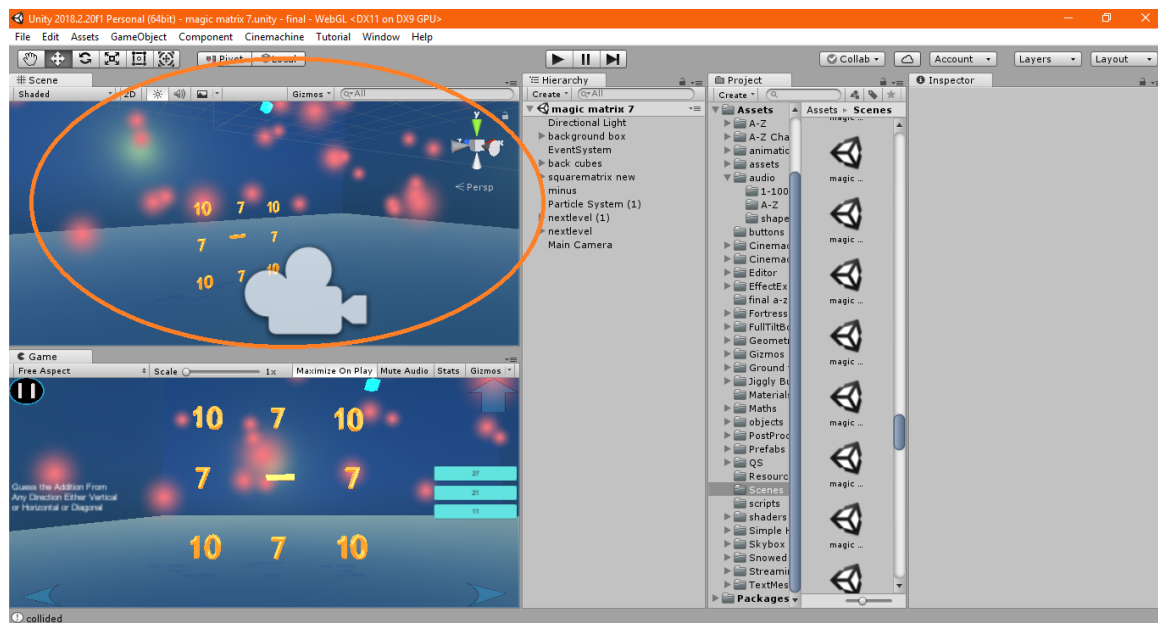
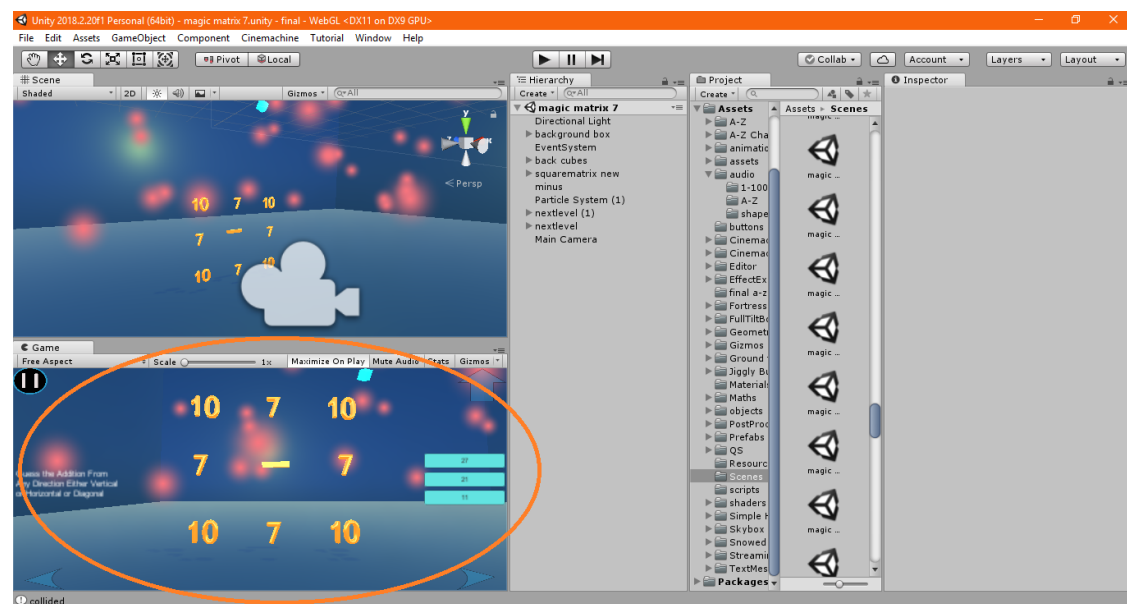


Fig 5.1 Scene View



### 5.1.2 Game View

The Game View is rendered from the Camera(s) in your game. It is representative of your final, published game. You will need to use one or more Cameras to control what the player actually sees when they are playing your game. For more information about Cameras, please view the Camera Component page.

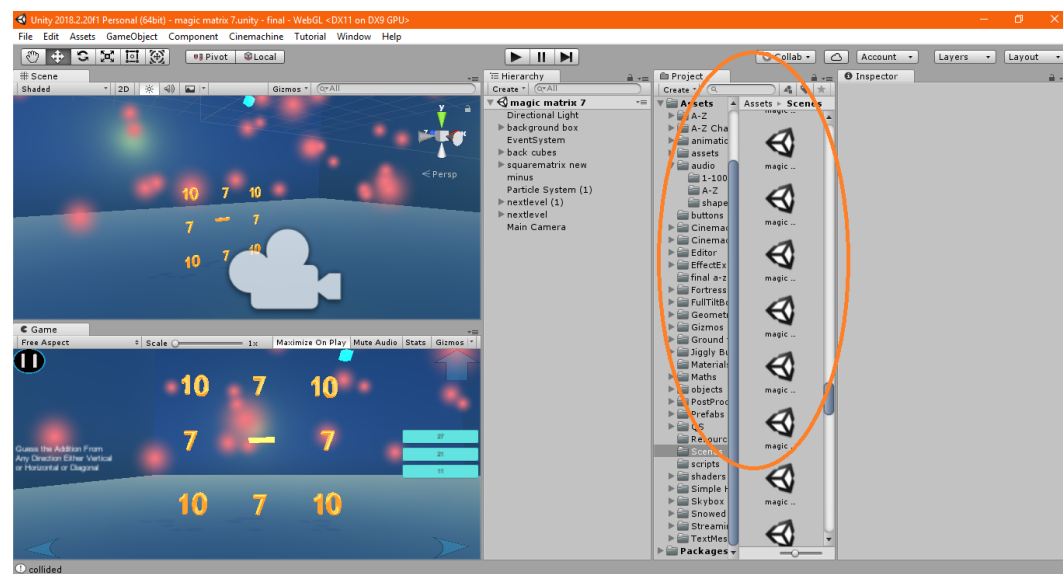


**Fig 5.2 Game View**

### 5.1.3 Project Menu

The left panel of the browser shows the folder structure of the project as a hierarchical list. When a folder is selected from the list by clicking, its contents will be shown in the panel to the right. You can click the small triangle to expand or collapse the folder, displaying any nested folders it contains. Hold down Alt while you click to expand or collapse any nested folders recursively. The individual assets are shown in the right hand panel as icons that indicate their type (script, material, sub-folder, etc.). The icons can be resized using the slider at the bottom of the panel; they will be replaced by a hierarchical list view if the slider is moved to the extreme left. The space to the left of the slider shows the currently selected item, including a full path to the item if a search is being performed. Above the

project structure list is a Favorites section where you can keep frequently-used items for easy access. You can drag items from the project structure list to the Favorites and also save search queries there (see Searching below). Just above the panel is a “breadcrumb trail” that shows the path to the folder currently being viewed. The separate elements of the trail can be clicked for easy navigation around the folder hierarchy. When searching, this bar changes to show the area being searched (the root Assets folder, the selected folder or the Asset Store) along with a count of free and paid assets available in the store, separated by a slash. There is an option in the General section of Unity’s Preferences window to disable the display of Asset Store hit counts if they are not required.

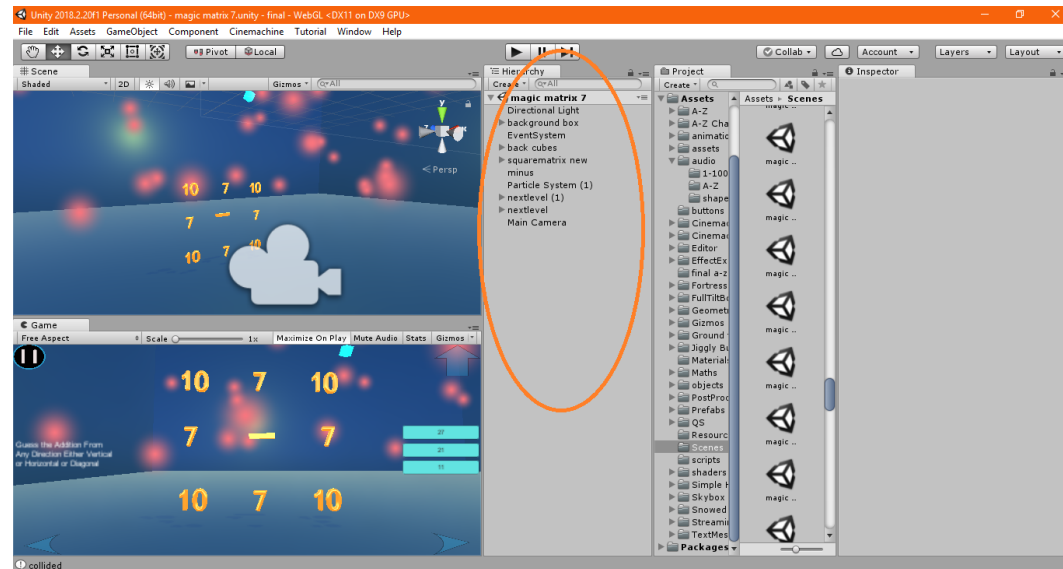


**Fig 5.3 Project Menu**

### 5.1.4 Hierarchy Window

The Hierarchy window contains a list of every GameObject (referred to in this guide as an “object”) in the current Scene. Some of these are direct instances of Asset files (like 3D models), and others are instances of Prefabs, which are custom objects that make up most of your game. As objects are added and removed in the Scene, they will appear and

disappear from the Hierarchy as well. By default, objects are listed in the Hierarchy window in the order they are made. You can re-order the objects by dragging them up or down, or by making them “child” or “parent” objects (see below).



**Fig 5.4 Hierarchy Window**

### 5.1.5 Inspector Window

Projects in the Unity Editor are made up of multiple GameObjects that contain scripts, sounds, Meshes, and other graphical elements such as Lights. The Inspector window (sometimes referred to as “the Inspector”) displays detailed information about the currently selected GameObject, including all attached components and their properties, and allows you to modify the functionality of GameObjects in your Scene. Use the Inspector to view and edit the properties and settings of almost everything in the Unity Editor, including physical game items such as GameObjects, Assets, and Materials, as well as in-Editor settings and preferences. When you select a GameObject in either the Hierarchy or Scene view, the Inspector shows the properties of all components and Materials of that GameObject. Use the Inspector to edit the settings of these components and Materials.

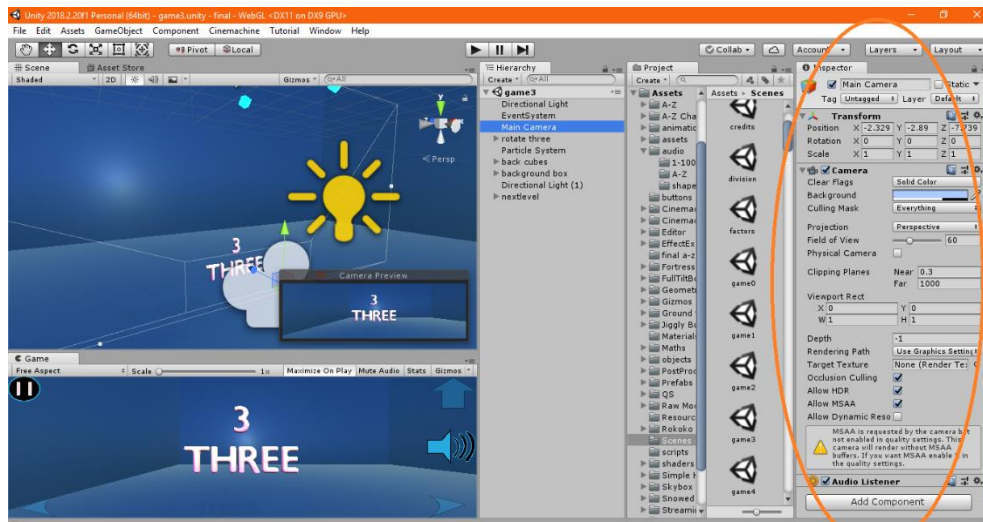


Fig 5.5 Inspector Window

### 5.1.6 Asset Store

The Unity Asset Store is home to a growing library of free and commercial Assets created both by Unity Technologies and also members of the community. A wide variety of Assets is available, covering everything from Textures, Models and animations to whole Project examples, tutorials and Editor extensions. You can access the Assets from a simple interface built into the Unity Editor which allows you to download and import Assets directly into your Project. Unity users can become publishers on Asset Store, and sell the content they have created. To find out more, see Asset Store information on Asset Store Publishing.

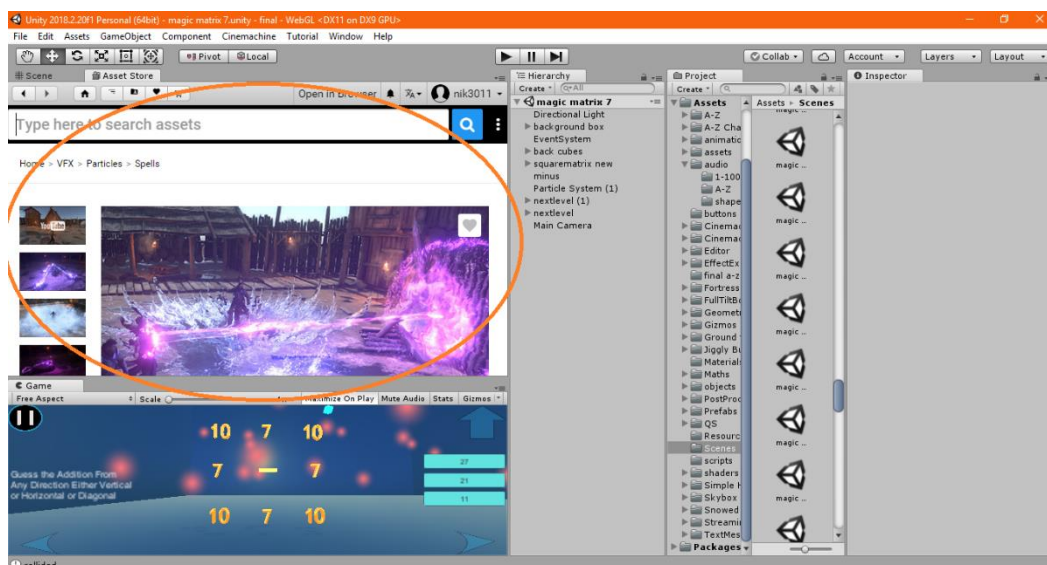
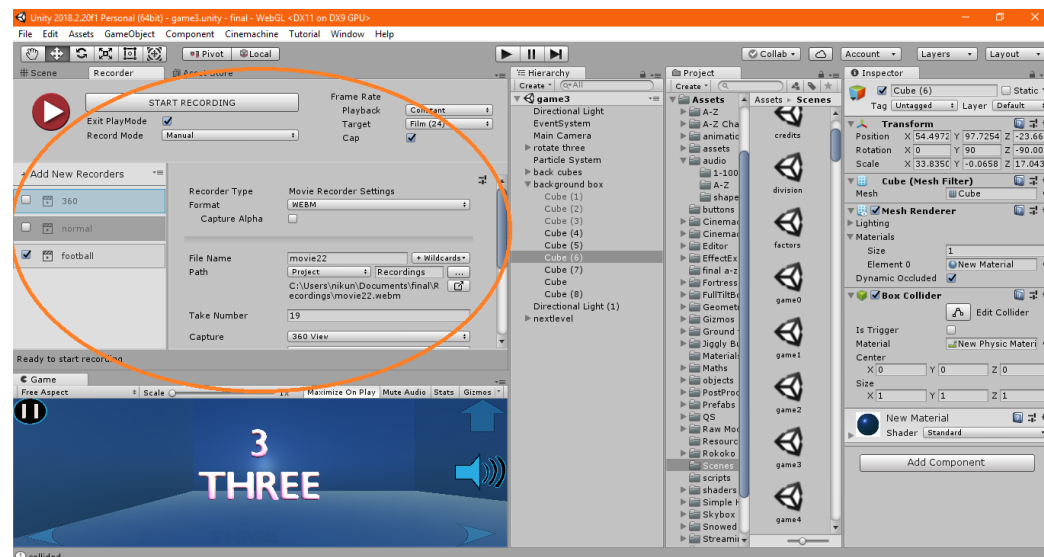


Fig 5.6 Asset Store

### 5.1.7 Unity Recorder

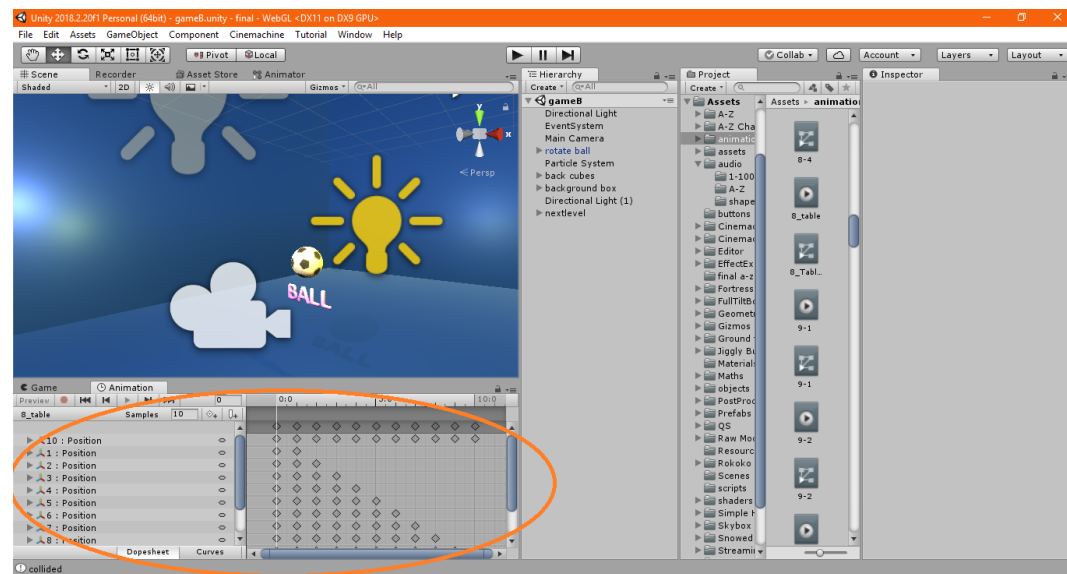
Use the Unity Recorder to capture and save in-game data during Play Mode. For example, use the Unity Recorder to capture and save gameplay as an MP4 file. A recorder specifies details for creating a recording, including a data source, resolution, and output format. The Unity Recorder package includes the following recorders: Animation Recorder: generates an animation clip in ANIM format. Movie Recorder: generates a video in MP4 or WebM format. Image Recorder: generates a sequence of image files in JPEG, PNG, or OpenEXR format. GIF Recorder: generates an animated GIF file. The Unity Recorder also supports Timeline. Use Recorder tracks and clips to trigger recording sessions from Timeline instances.



**Fig 5.7 Unity Recorder**

### 5.1.8 Animation

Unity's Animation features include retargetable animations, full control of animation weights at runtime, event calling from within the animation playback, sophisticated state machine hierarchies and transitions, blend shapes for facial animations, and much more.

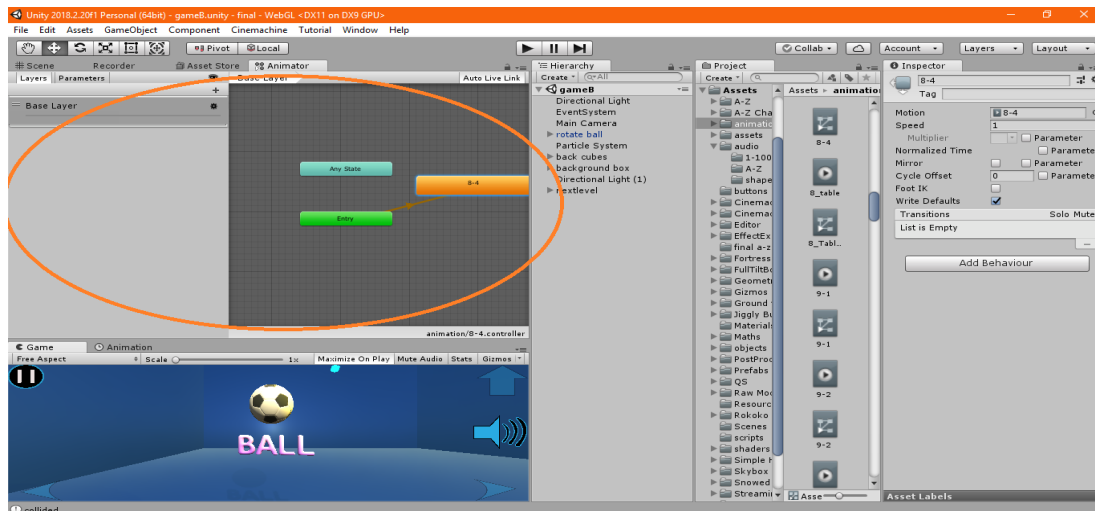


**Fig 5.8 Animation**

### 5.1.9 Animator

Unity has a rich and sophisticated animation system (sometimes referred to as ‘Mecanim’). It provides:

- Easy workflow and setup of animations for all elements of Unity including objects, characters, and properties.
- Support for imported animation clips and animation created within Unity
- Humanoid animation
- retargeting
- The ability to apply animations from one character model onto another.
- Simplified workflow for aligning animation clips.
- Convenient preview of animation clips, transitions and interactions between them. This allows animators to work more independently of programmers, prototype and preview their animations before gameplay code is hooked in.
- Management of complex interactions between animations with a visual programming tool.
- Animating different body parts with different logic.
- Layering and masking features



**Fig 5.9 Animator**

### 5.1.10 Timeline

Use Unity's Timeline to create cinematic content, game-play sequences, audio sequences, and complex particle effects.

- Each cut-scene, cinematic, or game-play sequence that you create with Unity's Timeline consists of a Timeline Asset and a Timeline instance
- The Timeline Editor window creates and modifies Timeline Assets and Timeline instances simultaneously.
- The Timeline Overview section includes details on the relationship between the Timeline Editor Window, Timeline Assets, and Timeline instances.
- The Timeline Workflows section shows how to create Timeline Assets and Timeline instances, how to record basic animation, and how to create cinematics.



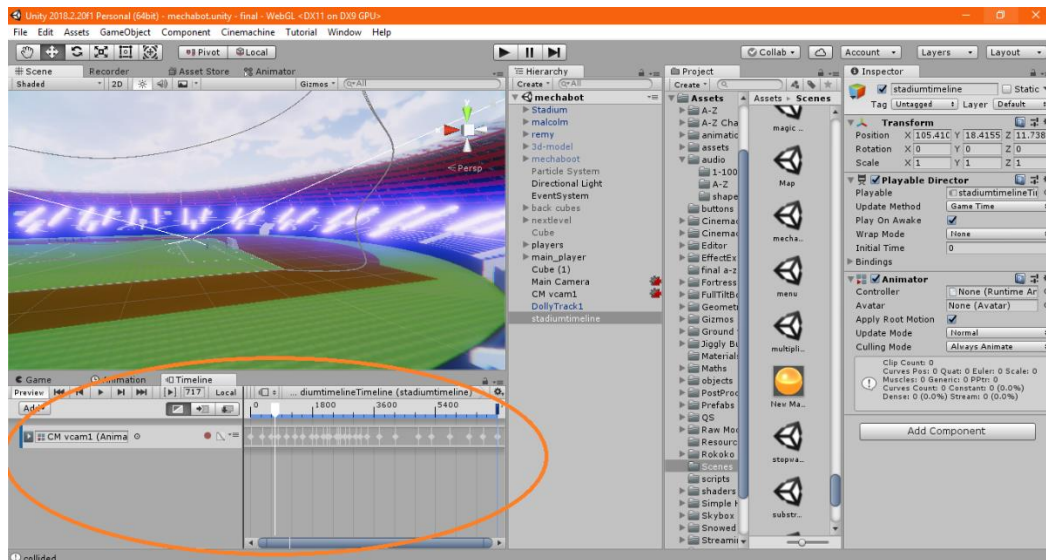


Fig 5.10 Timeline

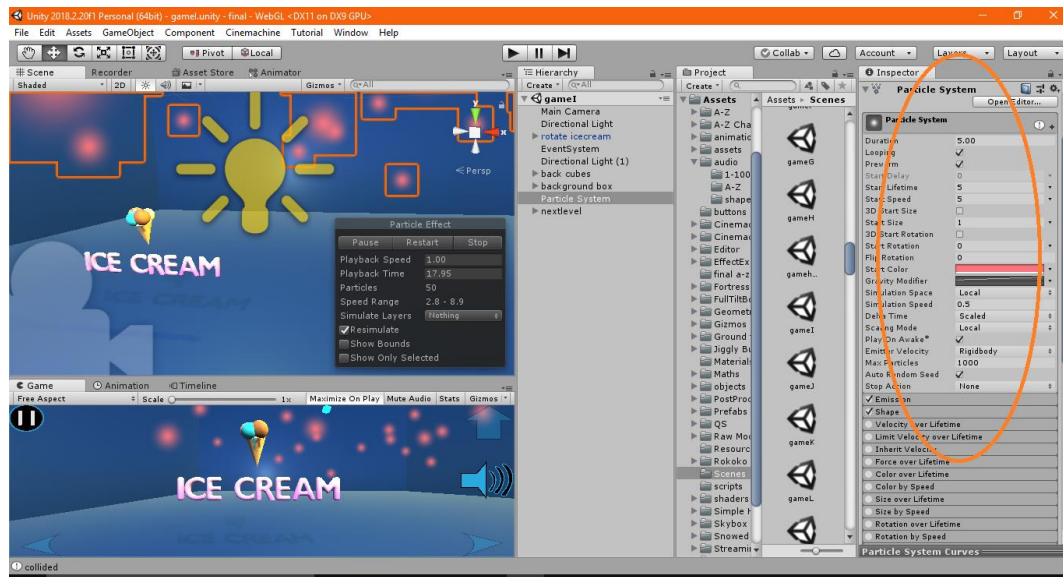
### 5.1.11 Particle System

In a 3D game, most characters, props and scenery elements are represented as meshes, while a 2D game uses sprites

for these purposes. Meshes and sprites are the ideal way to depict “solid” objects with a well-defined shape. There are other entities in games, however, that are fluid and intangible in nature and consequently difficult to portray using meshes or sprites. For effects like moving liquids, smoke, clouds, flames and magic spells, a different approach to graphics known as particle systems

can be used to capture the inherent fluidity and energy. This section explains Unity’s particle systems and what they can be used for.





**Fig 5.11 Particle System**

### 5.1.12 Scripts

The behavior of GameObjects is controlled by the Components that are attached to them. Although Unity's built-in Components can be very versatile, you will soon find you need to go beyond what they can provide to implement your own gameplay features. Unity allows you to create your own Components using scripts. These allow you to trigger game events, modify Component properties over time and respond to user input in any way you like.

Unity supports the C# programming language natively. C# (pronounced C-sharp) is an industry-standard language similar to Java or C++.

In addition to this, many other .NET languages can be used with Unity if they can compile a compatible DLL - see here for further details.

### 5.1.13 Audio

A game would be incomplete without some kind of audio, be it background music or sound effects. Unity's audio system is flexible and powerful. It can import most standard audio file formats and has sophisticated features for playing sounds in 3D space, optionally with

effects like echo and filtering applied. Unity can also record audio from any available microphone on a user's machine for use during gameplay or for storage and transmission.

<b><i>Format</i></b>	<b><i>Extensions</i></b>
MPEG layer 3	.mp3
Ogg Vorbis	.ogg
Microsoft Wave	.wav
Audio Interchange File Format	.aiff / .aif
Ultimate Soundtracker module	.mod
Impulse Tracker module	.it
Scream Tracker module	.s3m
FastTracker 2 module	.xm

**Table 5.1 Supported Audio format in Unity**

#### **5.1.14 Vuforia (For AR)**

Vuforia is a cross-platform Augmented Reality (AR) and Mixed Reality (MR) application development platform, with robust tracking and performance on a variety of hardware (including mobile devices and mixed reality Head Mounted Displays (HMD) such as the Microsoft HoloLens). Unity's integration of Vuforia allows you to create vision apps and games for Android and iOS

using a drag-and-drop authoring workflow. A Vuforia AR+VR samples package is available on the Unity Asset Store, with several useful examples demonstrating the most important features of the platform.

Vuforia supports many third-party devices (such as AR/MR glasses), and VR devices with back-facing cameras

(such as the Gear VR). See the Vuforia page on Devices for a full list of supported devices. See the Vuforia API reference for more information about classes, properties and functions used in the SDK.

You can use any device with a camera to test AR/MR games and applications built in Unity with Vuforia.

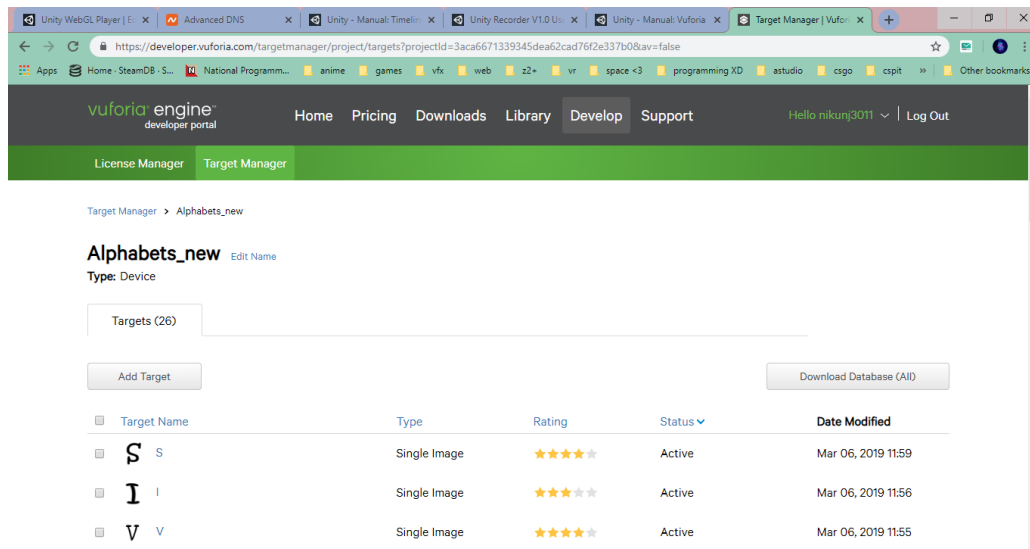


Fig 5.12 Vuforia

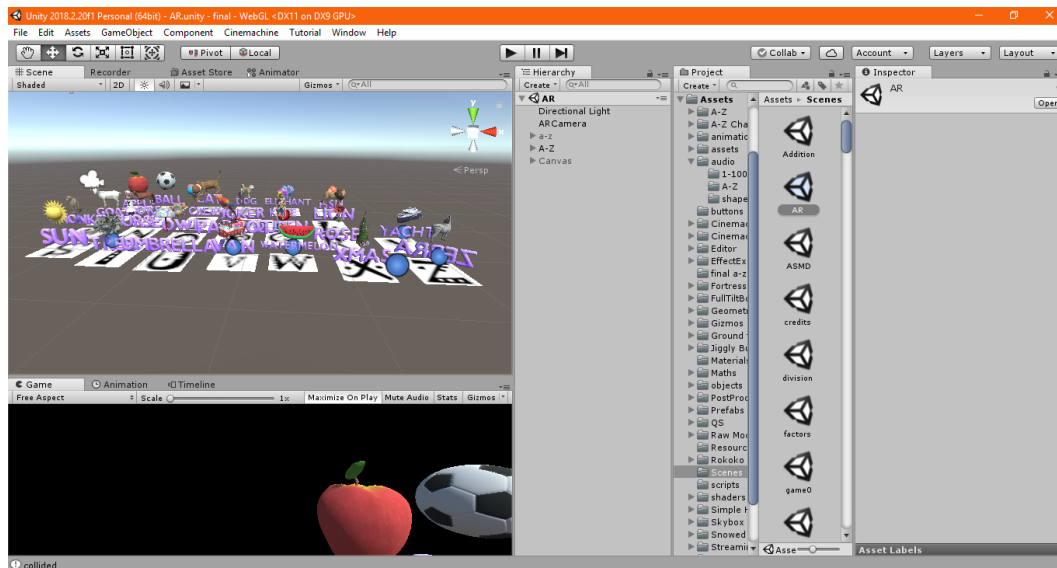


Fig 5.13 Vuforia(Unity)

### 5.1.15 Google VR

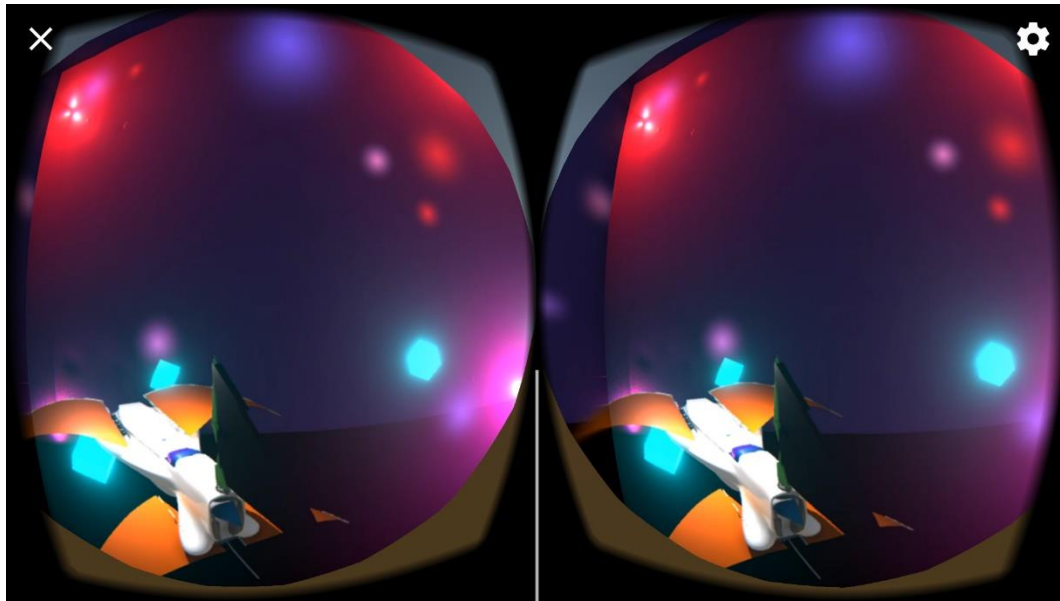
Google VR is the technology behind both Google Daydream and Google Cardboard VR platforms. Google VR encompasses support for smartphones, head mounted viewers and controllers, standalone head mounted displays and applications. Google VR provides an SDK for Unity that allows you to build VR content for this rich ecosystem of devices.

Google VR technologies includes Daydream, a platform for high quality mobile VR and Cardboard, built for bite-size VR experiences and 360 video. Daydream consists of Daydream-ready phones, headsets, controllers and experiences and supports high-end Android mobile devices through Daydream. See the full list of Daydream devices on the Google VR) website. Cardboard

works with almost any smartphone on both iOS and Android.

Google Cardboard is the most accessible mobile VR solution, allowing anyone with an iOS or Android smartphone to experience VR apps. Cardboard does have minimum and recommended requirements for its usage, but these are much less hardware-bound than Daydream, focusing mainly on the operating system and basic control systems (such as the gyroscope). Users can view VR apps with an official Google Cardboard viewing device, or a third-party VR viewing device which supports Google VR content (See Hardware and Software Requirements section). Cardboard also supports a range of third-party handheld controllers, for apps which require more complex interactions.

Daydream allows more feature-rich experiences than Cardboard, but only supports Daydream ready devices such as Daydream-ready phones, built for VR with high-resolution displays, smooth graphics and high fidelity sensors for precise head tracking. Daydream phone apps are accessible with the Daydream View, Google's headset and motion controller for experiencing high quality VR, using a Daydream-ready phone. Daydream will also support a Daydream Standalone HMD built by original design manufacturers. Daydream Standalone is an untethered head mounted display running on the Android OS including richer controller input and motion tracking.



**Fig 5.14 Google VR App**

### **5.1.16 Windows App**

When you build a Unity project to the Windows standalone platform Unity produces the following files (where 'ProjectName' is the name of your project):

ProjectName.exe - The project executable. This contains the program entry point which calls into the Unity engine when launched.

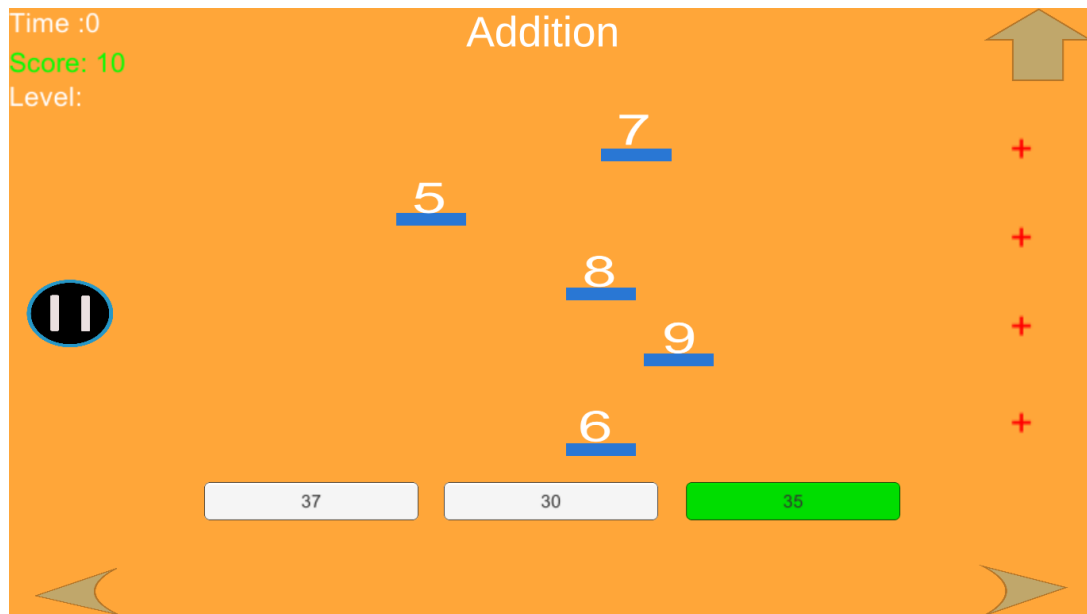
UnityPlayer.dll - this DLL contains all the native Unity engine code. It is also signed using the Unity Technologies certificate, allowing easy verification that the engine was not tampered with.

\*.pdb files - Symbol files for debugging. Unity copies these to the build directory if you enable Copy PDB files in the Build Settings window.

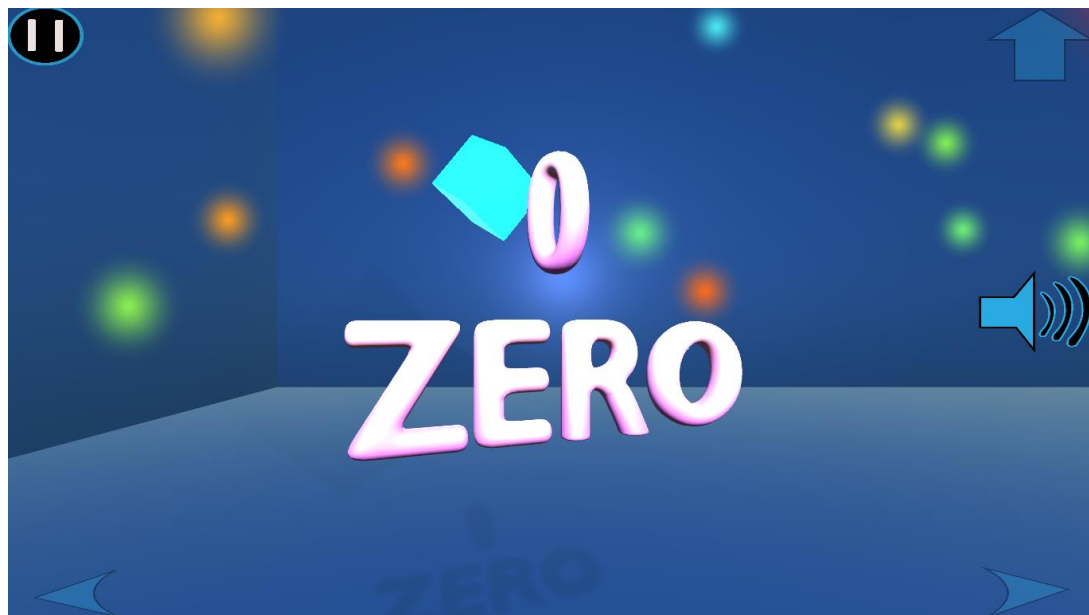
WinPixEventRuntime.dll - This DLL enables Windows PIX profiler support. Unity only creates this file if you check the Development Build

checkbox in the Build Settings window.

ProjectName\_Data folder - This folder contains all the data needed to run your project.



**Fig 5.15 Windows App**

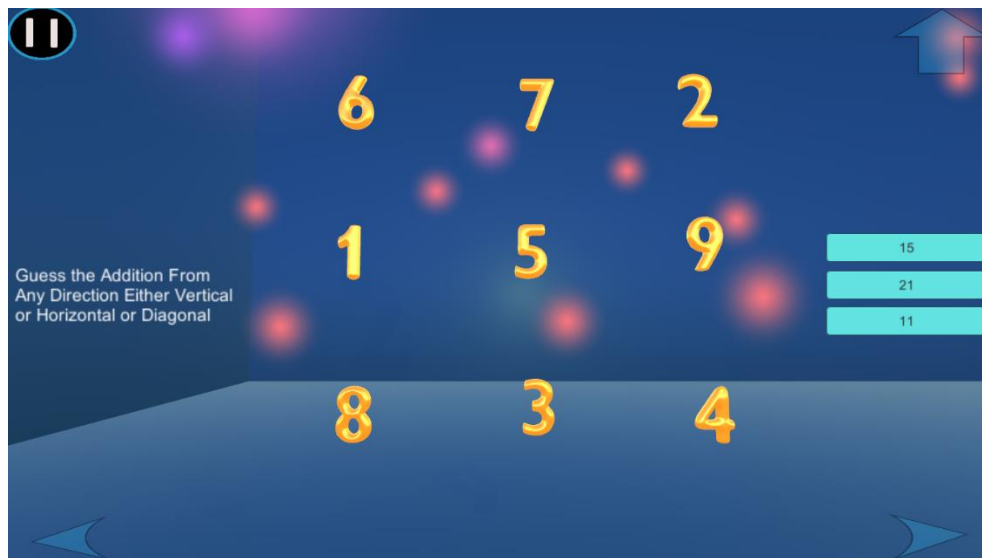


**Fig 5.16 Windows App**

### 5.1.17 Android App

Unity verifies your development environment when building for Android and prompts you to upgrade or download missing components.

Unity provides scripting APIs that allow you to access various input data and settings from Android devices.



**Fig 5.17 Android App**

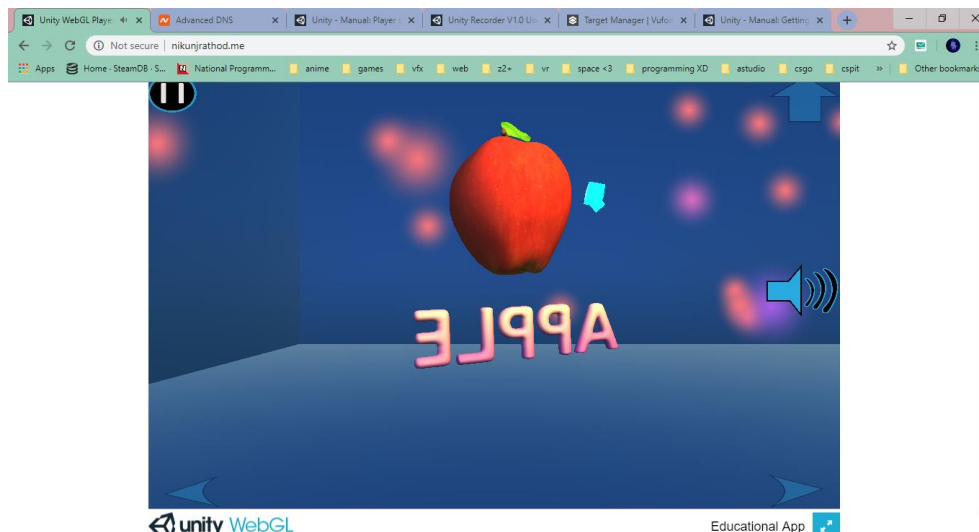


**Fig 5.18 Android App**

### **5.1.18 Web Based App**

The WebGL build option allows Unity to publish content as JavaScript programs which use HTML5 technologies and the WebGL rendering API to run Unity content in a web browser. To build and test your content for WebGL, choose the WebGL build target in the

Build Player window, and click Build & Run.



**Fig 5.19 Web App**



**Fig 5.20 Web App**

## 5.2 Method Pseudo Code

Create Script in C# :

- Input: C# script

Output: create your own Components using scripts. These allow you to trigger game events, modify Component properties over time and respond to user input in any way you like.

Code: Main Menu

If button = play



Load 1<sup>st</sup> scene

If button = level

Open level UI with 7 levels

If button = options

Open options UI with volume change and graphics option

If button = quit

Quit game

Code: Particular Scene

If button = Pause

Open Pause UI with 3 options

And Pause game

If button = Home

Open Main menu

If button = Sound

Play sound of particular object

If button = Back

Open previous scene

If button = next

Open next scene

Code: Mathematics Operations

Scene running on start

3 options for correct answer

Answer should be answered within time limit

If answer is correct

Glow Green button with increase in score by 30

If answer is wrong

Glow Red button with decrease in score by 15

Every time score get increase greater than 100

Increase level by +1

## **6.0 TESTING**

### **6.1 Testing Plan**

#### **Introduction:**

Test plans outline the process of testing the functionality of the software. This test plan is created to comprise project details and an outline of the Testing process and Test Strategy which will be followed during testing. A test plan details each step taken to achieve a certain result and states the objective of each action. The plan also highlights the projected resources, risks, and personnel involved in the testing.

#### **Item to be tested:**

1. Windows app
2. Android app
3. Web app
4. AR app
5. VR app

#### **Feature to be tested:**

1. Runs on Multiple Platform
2. Main Menu
3. Different levels
4. In-game menu/UI
5. Background music/ objects voice
6. All scenes
7. AR working with alphabets symbols in android.
8. VR working in android.

#### **Feature not to be tested:**

1. UI of AR/VR
2. VR app

## 6.2 References Suspension criteria & resumption requirements:

Suspension criteria specify the criteria to be used to suspend all or a portion of the testing Activities while resumption criteria specify when testing can resume after it has been suspended.

- Unavailability of external dependent systems during execution.
- When a defect is introduced that cannot allow any further testing.
- Critical path deadline is missed so that the client will not accept delivery even if all testing is completed.
- A specific holiday shuts down both development and testing.

System Integration Testing in the Integration environment may be resumed under the following circumstances:

- When the external dependent systems become available again.
- When a fix is successfully implemented and the Testing Team is notified to continue testing.
- The contract is renegotiated with the client to extend delivery.
- The holiday period ends.

Suspension criteria assume that testing cannot go forward and that going backward is also not possible. A failed build would not suffice as you could generally continue to use the previous build. Most major or critical defects would also not constituted suspension criteria as other areas of the system could continue to be tested.

### Exit Criteria:

Exit specifies the criteria that denote the successful completion of a test phase. The exit criteria are the targeted results of the test and are necessary before proceeding to the next phase of development. Example: 95% of all critical & major test cases must be executed and are passed. RTM has no Blocker, Critical or Major Issue linked.

**Test strategy:**

Testing technique	Purpose
Functional Testing	To check that the Functionality of the APP is matching the customer Requirement.
Integration testing	To check that there is no merge conflict and all the modules are working as expected after Integration
GUI Testing	To check that UI of the APP is user-friendly and Customer can easily navigate throughout the UI.
API Testing	To Check that the data we are using through API is Correct.
Performance Testing	To check that the APP works with the minimum Required Configurations.
Compatibility Testing	To check that the APP works on all supported OS and Browsers.

**Table 6.1 Test Strategy****6.3 TESTING STRATEGY**

The development process repeats this testing sub process a number of times.

- Unit Testing
- Integration Testing
- System Testing
- Acceptance Testing

Unit Testing tests a unit of code after coding of that unit is completed. Integration Testing tests whether the previous programs that make up a system, interface with each other as

desired. System testing ensures that the system meets its stated design specifications. Acceptance testing is testing by users to ascertain whether the system developed is a correct implementation of the software requirements specification.

Testing is carried out in such a hierarchical manner so that each component is correct and the assembly/combination of a component is correct. Merely testing a whole system at end would most likely throw up errors in component and that would be very costly to trace and fix. We have performed both Unit Testing and System Testing to detect and fix errors.

## 6.4 TEST SUITES DESIGN

### 6.4.1 Test Cases (test ID, test condition, expected output , actual output , remark )

Test Cases	Expected Result	Actual Result	Component	Priority	Status
Verify the Device compatibility	App should run on major platform	Successfully working on android, windows and web	Functional	Critical	<b>PASS</b>
Verify working android devices	App should run with all devices greater than android version 6.0.	Working	Functional	Critical	<b>PASS</b>
Verify working over the internet	App should work under a domain name	Worked under domain nikunj Rathod.me	Functional	Major	<b>PASS</b>
Verify main menu UI of app.	App should have background image with proper UI	Some UI problems with working all main menu options	Functional	Major	<b>PASS</b>
Verify in game UI of app.	App should have back and next button with pause and home button	Has all the functionality	Functional	Major	<b>PASS</b>
Verify different levels	App should have mentioned all working levels	App has mentioned all working levels	Functional	Critical	<b>PASS</b>
Verify working all levels in game	App should have all the levels working	All the levels are working	Functional	Major	<b>PASS</b>
Verify the AR and VR game functionality	App should have working different versions of AR and VR	App has working different versions of AR and VR	Functional	Major	<b>PASS</b>

**Table 6.4 Test Cases**

## **7.0 Future Enhancements**

### **7.1 Limitations**

- Android Version should be greater than 6.0
- UI are not that great
- UI problems in AR/VR app
- Audio for all the scenes are not available.
- VR app requires lot of memory
- AR app supports only few character images.

### **7.2 Future Enhancements**

As the “Educational App” is a premium app, features are decided on the basis of client requirements. So there is a constant flow of requirements and enhancements by users from all over the world.

We can propose enhancement like:

- Add support of the app in other Language
- Add functionality of audio in other Language.
- More User friendly UI.
- Multiple sites can be added under app.
- AR app working on website.

## 8.0 Conclusion

### 8.1 Self-Analysis of Project Viabilities

The Functionalities of “Educational App” have been developed following all customer requirements and has been delivered after thorough code review and testing. I Believe that all the functionalities will serve the purpose developed for and will be helpful for the users to get better in Mathematics and learn some new techniques to make mathematics more enjoyable to learn. Also using AR/VR app the user will find amusement and enjoyment with learning.

### 8.2 Problems Encountered and their Solutions

During the course of training at HariSkyline for the project “Educational App” various problems were faced while implementation.

- Changing User Requirements

User requirements changed many times leading to rework of the feature.

- Understanding Unity Scripting

Unity Scripting in C Sharp was absolutely new concept. So learning it took way too amount of time

- It was a challenge as my knowledge of the tools and technologies used in the project was the beginner level.

- Designing Models in various software was troublesome and sometimes erroneous increasing the overall making time.

- UI changes were the most changed item in the project.

- VR was not working on the first time. We solved by making video of the actual project and then transferring to Unity.

- AR was not working great for some character images. So we used complex image data for the same.

### 8.3 Summary of Project Work

After the successful implementation of “Educational App” during the entire course of training at HariSkyline, We got the understanding of various tools and technologies including,

- Unity
- Illustrator
- C Sharp
- Photoshop,
- 3DSMax
- V-ray
- After effects
- Substance painter and Designer

During implementation, we also came to know about industry standard DevOps tools like JIRA, Confluence, Bamboo and Bitbucket provided by Atlassian by using them in the project.

During training and development, we got familiar with actual in and out of the Agile model, and how to deal with constantly and rapidly changing requirements since so many stakeholders were involved in the project.

We are now also familiar with industry standards for coding.



**References:**

- <https://unity3d.com/learn/tutorials/s/scripting>
- <https://www.coursera.org/specializations/game-development>
- <https://youtube.com>
- <https://www.pluralsight.com/paths/unity-game-development-core-skills>
- <https://answers.unity.com/questions/ask.html>
- <https://www.pluralsight.com>