An Immersive Learning Approach to Teach Science Experiments

Dr. Abhijit Joshi, Arindam Bose, Rahul Mehta, Kashyap Shah

Abstract-In this paper we propose a solution to problem faced by most school/high school students of the current generation, i.e., the inability to perform science lab experiments at their comfort and their own pace. We are combining Augmented Reality, Real-time Digital Image Processing technologies with the latest advancements in Gaming on smart-phones to create an application that enables school students to perform their school lab sessions virtually on their smart-phones, with a level of interactivity and lifelikeness never seen before.
The exponential growth in smartphone CPU and GPU performance over the last couple of years now allows the deployment of rich, high fidelity; Augmented Reality (AR) based applications on low to mid-range mobile hardware, this in-turn helps to develop a solution that is more cost effective than actually buying/renting laboratory equipment. Students will be able to turn ordinary pieces of paper into interactive lab instruments using AR on their smartphones.

Index terms- Augmented Reality, Android, iOS, Mobile, Embedded GPU, Real-Time Image Processing, e-Learning.

1 Introduction

In the last three years the computing power in mobile devices has far outpaced Moore's Law, mobile System-on-Chips, i.e., the CPU+GPU in a phone, have doubled in performance every 9 months. This has resulted in mobile devices that, now in 2014, are about 6 to 8 times faster than those in 2011 [3]. While this growth may be staggering, software developers have failed to keep up. Standard mobile applications make little use of such massive performance gains; the only place that some progress is observed is in mobile games. Thus we are using the latest in mobile Game Development technology to bring an immersive and interactive e-learning experience to students on mobile devices that are now low cost and still decently powerful.

It is now possible, using Digital Image Processing Techniques, to identify and track, in a 3D space a special marker, which has a unique image printed on it [12]. By comparing the image seen by the mobile-device camera, to the one stored in the database, one can compute the exact position and orientation of the image w.r.t the device in real-time more than 20-times a second [10]. Using this information one can super-impose virtual 3D objects on those markers, thus creating a virtual object in a real world. This enables augmented reality to be used in scenarios where high cost-efficiency is required, such as in education [11]. This technology is used to create science lab equipment such as beakers, lenses, lasers, burettes, test tubes etc. on these markers. This enables the student to interact in an almost lifelike manner with markers, just as he would be able to with real lab equipment. For example, a student can physically pick a marker which has a beaker on it and then tilt it to pour liquid out, just like with a real world beaker.

The rest of this paper is organized to portray the development of the application, the shortfall's found in existing systems and how we seek to overcome them, the tools we used in development of the system.

In Section 2 we discuss the various existing techniques and their shortfalls; here we also discuss the frameworks we use in our solution. We proceed to describe our implementation in section 3 and the design of the application in section 4. Finally in Section 5 we experiment with our app and make observations on how we have improved the existing systems.

2 Literature Review

This section consists of the research conducted by the team. It includes a cursory preface of the existing system and its drawbacks. It also gives an idea about all potential frameworks that could have been used in building this application.

2.1 Review of Frameworks

In this section, discussions took place to decide as to which frameworks would be most useful for the development of the application that can teach students basic science experiments.

2.1.1 Unity 3D:

Unity is a cross-platform game engine with a built-in IDE. It is used to develop video games for web plugins, desktop platforms, consoles and mobile devices. It currently supports development for iOS, Android, Windows, Blackberry 10, OS X, Linux, web browsers, Flash, PlayStation 3, Xbox 360, Windows Phone 8. We selected this framework because it has a highly modular nature which allows us to expand it, along with that it provides a relatively small learning curve and excellent Android support [4].

- *Dr. Abhijit Joshi is the Vice-Principal (academic) and Head of Department of Information technology at Dwarkadas J. Sanghvi College of Engineering. E-mail: abhijit.joshi@djsce.ac.in*
- *Arindam Bose, Rahul Mehta, Kashyap Shah are currently final year undergraduate students for a Bachelor of Engineering Degree in I.T at Dwarkadas J. Sanghvi College of Engineering. E-mails:arindam.1993@gmail.com, rahul10992@gmail.com, kashyap2492@gmail.com*

2.1.2 Augmented Reality for Unity:

There are various packages available for Augmented Reality in Unity 3D, some of them are: Catchoom [5], DFusion, Metaio, Total immersion [6] and Qualcomm's Vuforia [1].

From the above toolkits we chose Qualcomm Vuforia as it is highly modular like Unity and provides a highly robust and extensible C# API to program in. This allows the creation of powerful experiences with little code. Apart from this Vuforia is a free toolkit.

2.2 Existing System Study

In this section the functioning of the current system and the various pitfalls that exist in the current system are put forth.

2.2.1 Current System for Performing Science Experiments

There are three main approaches for students to be able to learn about virtual lab experiments. They are categorized as follows.

A. Computer Based Approach

Currently students at the school and high school level have to either rent or buy equipment in order to perform practical labs outside of their main institute. This is an expensive process and not exactly feasible for most students. So, most students turn to online media to be able to receive guidance. Students are able to go online and watch the experiments being performed or they can use simple tools such as flash games to assist them [2].

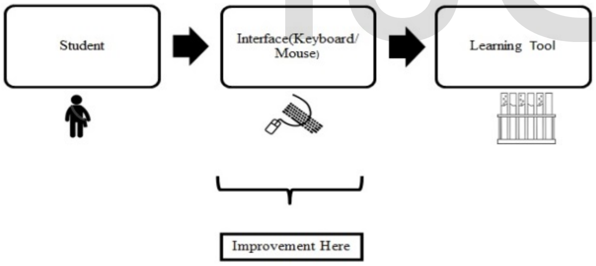


Fig 1: Existing Interaction model for Computer Based System

As seen in fig.1, students generally use a mouse and keyboard combination to able to use current e-learning tools. While mouse and keyboard are useful in the desktop space, it is not intuitive to perform real world tasks using the mouse and keyboard as an interface. As these input devices were designed with a 2D desktop environment in mind which makes it unsuitable for interacting with real world objects. It is this component that will be the area of focus.

B. Online Video Based Approach

As shown in fig.2 students can go and look at various online videos for guidance on experiments. This approach however remains completely one way interaction and the student cannot perform any experiments physically. Also, it is observed that students learn in a better way when they perform the experiments themselves.

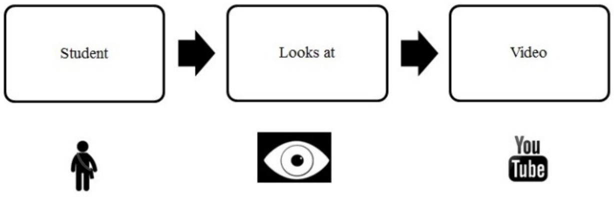


Fig 2: Online Video based approach

C. Traditional approach

This approach mainly consists of classroom learning where experiments are performed using scientific equipment. It also proves to be inefficient since the learning only depends on individual's grasping power.

Also, if student is unable to understand the problem during the teaching hours, he/she cannot practice the same since he does not have access to lab apparatus and also the experiment needs to be performed under the supervision of an instructor which is not viable for home learning.

2.2.2 Observations on the Existing Systems

Looking at all the existing systems in section 2.2.1, the following issues are observed.

A. One Way Interaction

The primary way to be able to revise labs outside of the institute now is to watch online video's on a streaming site. In such a situation the interaction of the student is only one way, i.e., the student receives information from the teacher but there is no response on the student's question. This is an important aspect in the E-learning scenario and is addressed using this application.

B. No Spatial Awareness

Even if the student is using a simple flash tool on the internet to perform the experiments the student gets no spatial awareness of the same. In order to perform a simple chemistry experiment a student needs an idea of what apparatus goes where. And this is not possible in case of simple flash tools/animations.

C. Poor Interaction Medium (Keyboard+ Mouse)

There are few 3D tools available that can be used to perform experiments in the 3D space but even when using such tools the primary interaction medium remains the keyboard and mouse combination. Tough useful in manipulation of text and icons it is not really useful for manipulation of 3D objects in a 3D space.

D. Poor Discoverability

There are various sites and channels to be able to receive learning tools currently and it is highly fragmented. Locating the source of required information becomes difficult and standards might not match at various locations. This application will be a consolidated source for all virtual lab experiments.

E. Stationary and Expensive Platform

Most current learning tools work on desktop PCs or laptops. Such a platform is not mobile and also very expensive. Mobile devices are now fast approaching the performance of stationary platforms and can run similar type of software. A desktop platform is also very expensive whereas there are cheap mobile platforms such as the AAKASH tablet is an alternative.

3 The Proposed Approach

We are developing an android application that uses augmented reality to create virtual lab instruments for school and high-school students. The plan is that an android device acts as a virtual window which the students look into and can interact with the marker directly with their hands. For example, if a student wants to interact with 2 beakers he can place android device between him and the markers and then use his hands to directly pick-up markers and perform the experiment. This will create a much more interactive experience for the student and he need not learn any new gestures which would differ from the process of actually performing the experiment.

The main problem with the current system is the usage model; it involves the use of keyboard and mouse input where the best scenario is to have the student actually perform the experiment by hand. The keyboard and mouse though highly efficient for the manipulation of text and icons; is not really useful for manipulating real world 3d objects. Also, the student does not feel that he is performing that specific task.

In the E-learning scenario, interaction is a very important aspect. So the main objective is to fix the interaction model, i.e., have the student actually be able to manipulate virtual instruments by hand rather than some other form of input so that he/she achieves a true sense of immersion (See Fig 3). Hence, we have applied Augmented Reality on the Android platform to achieve our goal. The student will basically manipulate AR-markers by hand while he looks through a virtual "window" in the android phone/tablet. Through the window he basically sees various lab instruments superimposed on the AR-marker and he can manipulate them to whatever extent the user wants. Now, let us see the proposed architecture of the system.

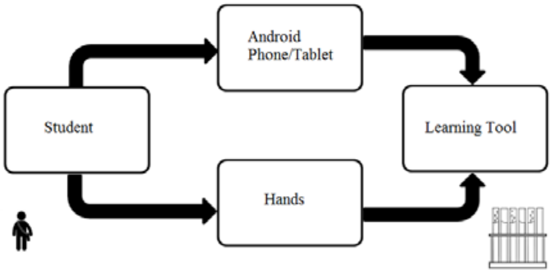


Fig 3: The Interaction Model

3.1 Proposed Architecture

The proposed architecture for the system is shown in Fig 4. It consists of various components organized in a layered fashion as discussed below.

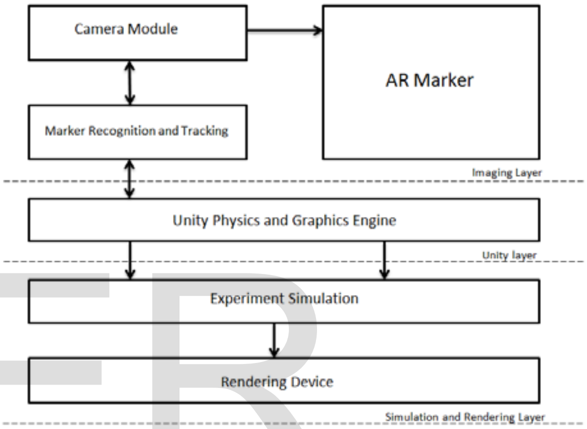

Camera Module
AR Marker
Marker Recognition and Tracking
Imaging Layer
Unity Physics and Graphics Engine
Unity layer
Experiment Simulation
Rendering Device
Simulation and Rendering Layer

Fig 4: Proposed Architecture for Virtual Laboratory

3.1.1 Imaging Layer

This layer consists of various components that deal with the various imaging and marker acquisition and tracking of the system. It consists of various components described as follows.

A. AR-Marker

This is the actual physical component of the system. It consists of physical AR-markers whose positions can be tracked by the other modules. The student uses these markers to interact with the virtual instruments [7].

B. Camera Module

This is the physical camera sensor that is placed behind the android device, it enables the android device to act as a virtual window and the student basically points this camera to towards the markers.

C. Marker Recognition and Tracking

This component of the system deals with taking the video input from the camera and is able to extract the position and orientation of the marker from the video input. This is the key component of the system as this is the one that basically feed the engine with the position and orientation of the virtual instruments so that the student is able to interact with them. As the student manipulates the marker with his hands the tracking and recognition module updates the orientation and position if the virtual objects in real-time. This component is essentially provided almost in its entirety by the Vuforia toolkit.

3.1.2 Unity Layer

This layer consists of the core unity physics and graphics engine. It acts as an intermediary between the imaging layer and the simulation and rendering layer. It handles the draw calls to the GPU to create the virtual instruments and it also handles the positioning of the objects in a virtual 3D space.

3.1.3 Simulation and Rendering Layer

This layer has two components, namely, experiment simulation and rendering device. Experiment simulation is the component of the system that takes the positioning and physics data from the game engine and runs the simulation of the experiment as per the orientation of the virtual instruments. For example, if the student is performing a chemistry experiment this component will basically recognize the various chemicals and provide the necessary changes in color or materials that are required. The second component of this layer is the rendering device that takes information from the simulation component and sends OpenGL draw calls to the GPU on the device.

4 Implementation

Looking at role played by each component of the system, now, let us see the implementation of the system as a realization of the approach outlined in section 3. In order to implement such a system various tools and technologies are required. We describe them along with our rationale for selecting these tools in the subsequent sections.

4.1 Technology Used

In the development of Virtual Laboratory technologies such as Unity Game Engine, Qualcomm's Vuforia Toolkit and various 3D Modeling softwares were used.

4.1.1 Unity Game Engine

Unity is a cross-platform game engine with built in IDE developed by Unity Technologies. It is used to develop videogames for web plug-ins, desktops platforms, consoles, and mobile devices. It grew from an OS X supported game development tool in 2005 to a multi-platform game engine [8].
The reason for selecting Unity instead of any other engine is its excellent mobile support. This engine was built with mobile in mind, thus it offers excellent performance on mobile hardware with a rendering layer built in OpenGL ES 3.0 which is an optimized version of OpenGL targeted for the mobile platform. Along with that Unity is also easily extensible by downloading a wide range of add-ons and plugin's from the asset store. This significantly reduces development time and effort by allowing us to focus on the essentials.

4.1.2 Qualcomm's Vuforia Toolkit

Vuforia is an Augmented Reality Software Development Kit (SDK) for mobile devices that enables the creation of Augmented Reality applications [9]. It uses Computer Vision technology to recognize and track planar images (Image Targets) and simple 3D objects, such as boxes, in real-time. This image registration capability enables developers to position and orient virtual objects, such as 3D models and other media, in relation to real world images when these are viewed through the camera of a mobile device. The virtual object then tracks the position and orientation of the image in real-time so that the viewer's perspective on the object corresponds with their perspective on the Image Target, so that it appears that the virtual object is a part of the real world scene.
The reason for selecting Vuforia over other AR tools for Unity such as Catchoom or UARToolkit is the unique way in which Vuforia deals with tracking images. It allows custom images to be used as targets for tracking and wraps all the tracking functionality into a prefab called ImageTarget. By populating this prefab with our own custom image we can track it and associate a 3D object to it. Along with that, Vuforia allows these ImageTargets to be partially obscured and still maintain tracking of the image. This is a powerful feature that allows implementing Virtual Buttons on the ImageTarget which allows the user to interact intuitively with the lab equipment. Rather than having to press interacts with the equipment using on-screen buttons one can place these buttons on the actual marker and the user can interact physically with the marker by pressing an invisible button on it.

4.1.3 3D Modeling Software

The various 3D modeling tools used in the development of this application are Blender and Autodesk 3D Studio Max. 3DSMax allows creating high quality 3D models of lab equipment speedily. Blender allows easy conversion of these models to .blend files that work better with the Unity Engine in terms of quality and various conventions such as orientation of XYZ axes.

4.2 Implementation Steps

This section goes over the high-level tasks that were performed in the development of this application. This starts with integrating the Vuforia toolkit with Unity, then proceeds to adding and registering ImageTargets and then completes with the programming of the experiment simulation. For better understanding, all the above steps are described with respect to the implementation of an acid-base Titration experiment which is frequently performed at the high school level. This experiment involves the simulation of a burette as well as the simulation of a chemical reaction in which the color of a chemical solution changes to indicate that it has changed from acidic to basic.

4.2.1 Integration of Vuforia with Unity

In this step we import the various Vuforia prefab's, extensions and classes into the Unity IDE.

The recommended development environment for Android Development using Vuforia is Microsoft Windows 7 32/64-bit or Windows XP. Note that Unity 4.1 or later is required. After ensuring the above requirements the steps given below are followed to integrate Vuforia with Unity.

1. Download the archive file from https://developer.vuforia.com/resources/sdk/unity#sdkModal
2. Extract the archive and copy the content (vuforia-unity-android-ios-xx-yy-zz.unitypackage) to a location that is convenient for the development environment.
3. After installing the package, create a new project in Unity.
4. Select vuforia-unity-android-ios-xx-yy-zz.unitypackage from the list of "Import the following packages" in unity.
5. Fig 5 shows the folder structure that should be present in the Project panel after successfully importing the Vuforia extension into Unity.

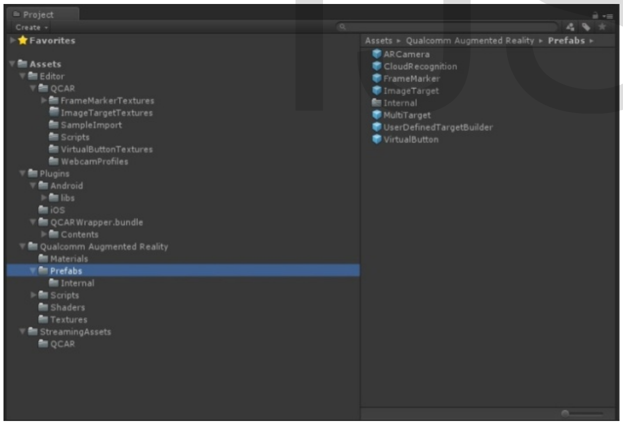


Fig 5: Folder Contents

4.2.2 Registering ImageTargets in Unity

The next step involves creating "ImageTargets". An Image Target is a 2D image whose position and orientation in 3D space can be tracked by a smartphone camera.

So as to turn a simple JPEG or PNG image into a format that Vuforia can understand involves the creation of a package referred to as a "Device Database". Each Device Database contains information that helps the Vuforia toolkit track the image in real time.

Qualcomm provides an easy-to-use web portal called the Target Manger to allow developers to create Device Databases from JPEG images. Steps A to C show the procedure to create the augmented reality image using the Target Manager.

A. Using the Target Manger to create a Device Database

1. Go to https://developer.vuforia.com/target-manager and log in or create an account.
2. Click on the blue "Create a Database" button, in the dialog that pops up give a suitable name to your database. One database can contain multiple Image Targets.
3. Now click on the "Add a Target" button.
4. Fig. 6 shows the Add Target dialog that appears. Fill Target name with a suitable name and click Single Image and click Choose File and Browse and select an image file. Enter the real with of the image in inches in the Target Dimension field.
5. Click the Add button and after a few seconds a created device Database should appear.

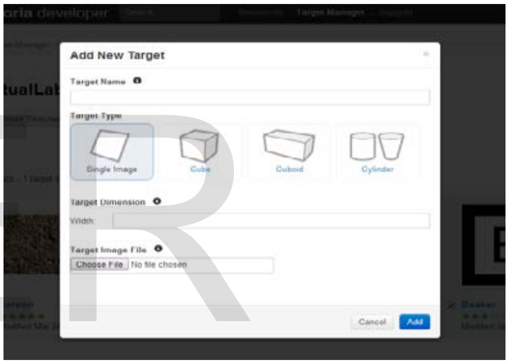


Fig 6: The Add Target Dialog

6. Click on the required Target and click "Download Selected Targets". In the pop-up select the Unity editor option and click create to download a unitypackage of the Database.

B. Importing the Database into Unity

1. Double-click the downloaded package or right-click on Unity Project for "Import Package" and "Custom Package".
2. Select the downloaded package.
3. Click Import to import the target Device Database.

C. Add AR assets and prefabs to scene

1. Now the Vuforia AR Extension for Unity has been imported, it adapts the project to use augmented reality.
2. The prefabs for creating an Augmented Reality enabled project are located in the Qualcomm Augmented Reality/Prefabs folder.

3. The default "Main Camera" in the scene is then deleted, this camera cannot be used for AR applications, instead the ARCamera prefab used by dragging it into the scene. In the inspector view of the ARCamera there is an attached script called "Data Set Load Behaviour". Below it are listed all the dataset's imported in the database.
4. ImageTarget prefabs are required for tracking 2D images. Once added to the scene they can be loaded with data from the Device Database.
5. Select the ImageTarget object in the scene, and look at the Inspector. There should be an Image Target Behaviour attached, with a property named Data Set. This property contains a drop-down list of all available Data Sets for this project. When a Data Set is selected, the Image Target property drop-down is filled with a list of the targets available in that Data Set.
6. Import the required 3D model (in this case it is the 3D model of a burette) in the scene and make it a child of the required ImageTarget. Then position the model in the scene relative to the image target as required.
7. Fig. 7 shows how the scene is set up for a titration experiment. In the Hierarchy tab it can be seen how various elements that need to be tracked with the "BuretteMarker" ImageTarget are made children of that marker.

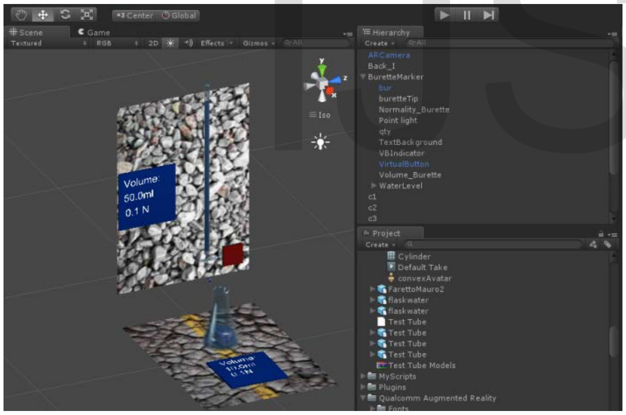


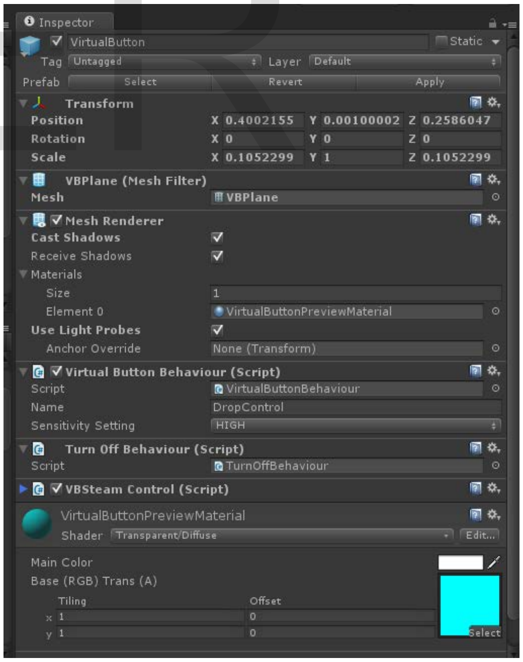
Fig 7: Titration Scene Set Up

8. Repeat steps 4 through 8 to set up another image target with the conical flask.

4.2.3 Programming the Experiment Simulation

Now, the code is written to simulate the working of various lab apparatus. In case of titration there is a need to create an interactive burette. This is achieved by creating a Virtual Button on the BuretteMarker. On pressing this Virtual Button the tap of the burette is opened to release liquid at a steady rate. The following steps are performed to implement a Virtual Button.

1. Drag the VirtualButton prefab present in the Qualcomm Augmented Reality/Prefabs folder into the scene.
2. Make this VirtualButton prefab a child of the ImageTarget that it is to be positioned upon.
3. Move the VirtualButton on the ImageTarget to a preferable location. In this case it is the red square shown in Fig. 9.
4. Create a new C# script and attach it to the VirtualButton. The event handling code must be written in this script file.
5. Fig. 8 shows the Inspector view of the VirtualButton prefab. There are two scripts attached. The VirtualButtonPrefab is a predefined script already attached to the button. The Name parameter in this script helps us to uniquely identify this button. The VBSteamControl is a custom script written to simulate the operation of the burette.
6. The C# class created in the VBSteamControl script must extend the IVirtualButtonEventHandler interface. This interface provides these two methods that must be overridden, onVirtualButtonPressed() and onVirtualButtonReleased(). These methods are called as and when the virtual button is pressed and released respectively by the user.


Inspector

☑ VirtualButton □ Static

Tag Untagged Layer Default

Prefab Select Revert Apply

Transform

Position X 0.4002155 Y 0.00100002 Z 0.2586047

Rotation X 0 Y 0 Z 0

Scale X 0.1052299 Y 1 Z 0.1052299

VBPlane (Mesh Filter)

Mesh VBPlane

☑ Mesh Renderer

Cast Shadows ☑

Receive Shadows ☑

Materials

Size 1

Element 0 VirtualButtonPreviewMaterial

Use Light Probes ☑

Anchor Override None (Transform)

☑ Virtual Button Behaviour (Script)

Script VirtualButtonBehaviour

Name DropControl

Sensitivity Setting HIGH

Turn Off Behaviour (Script)

Script TurnOffBehaviour

☑ VBSteam Control (Script)

VirtualButtonPreviewMaterial

Shader Transparent/Diffuse Edit...

Main Color

Base (RGB) Trans (A)

Tiling Offset

x 1 0

y 1 0

Select

Fig 8: Inspector view of the VirtualButton prefab

4.3 Advantages of proposed system

1. Two-way Interactivity

The interaction of the student with the system is two-way and interactive. The student can interact with ease and can provide valuable feedback to the system. Since, rather than using a keyboard and mouse, user is interacting with the objects in a realistic fashion.

2. Cheap and Easy to procure

As this is just a simple android application the entire student needs to do is download and the run the application to be able to get access to wide range of lab instruments and experiments.

3. Easy Discoverability

This application basically provides a one stop shop for all experiments so the student need not search all over the web to obtain different experiments. This easy discoverability definitely improves the current system.

5 Evaluation And Results

In this section we start with the walkthrough of the system as a user and we end it with an evaluation study conducted to assess the teaching approach and a survey.

5.1 Working with the system

When the application is selected a menu screen appears as shown in Fig. 9 which provides an option as to which of the experiments to choose from. In this case, the options are titration (shown by the burette and conical flask) or the refraction of lenses (shown by the light source and the 2 lenses).

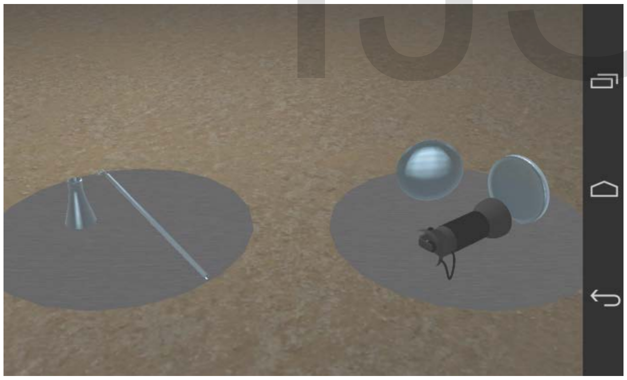
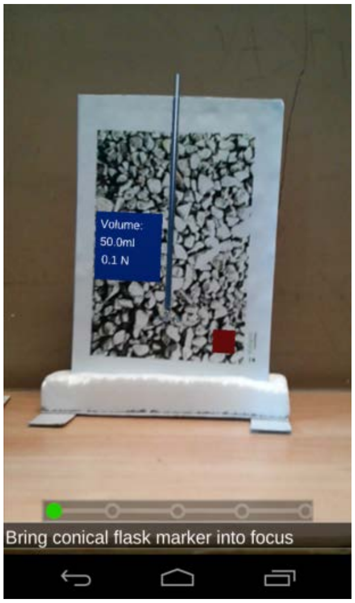


Fig 9: Menu to select experiment

On selecting one, a window opens up showing a progress bar for the experiment and the instructions to proceed to the next step. As shown in Fig. 10, the first instruction is given with a progress bar. This instruction guides the user to bring the burette marker into focus so that the camera can track the marker effectively.



Bring conical flask marker into focus

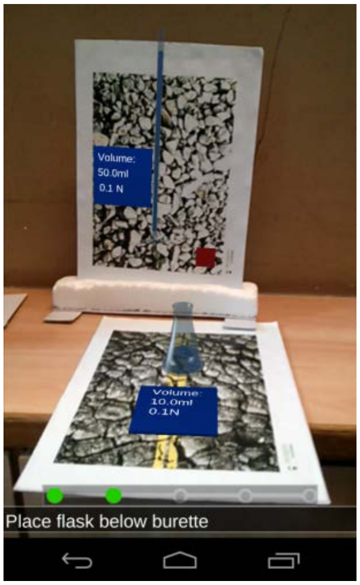
Fig 10: First Step of Experiment

Bring Burette Marker into focus

Figure 11: Step 2 of the Experiment

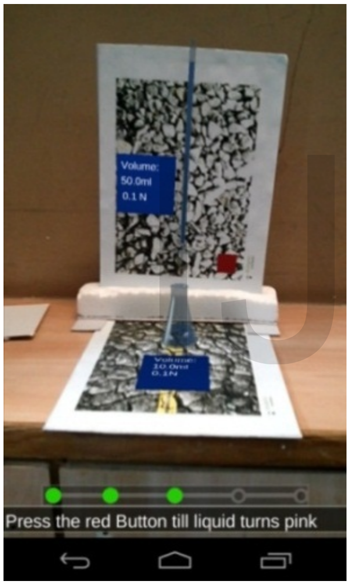
Once the first step is complete, it implies that the marker is successfully tracked and we can see a virtual burette. The next instruction is shown on the screen and the progress bar shows the first circle as completed which is indicated with a green light. This is shown in Fig. 11.

The next step involves changing the instruction again and incrementing the progress bar to show the completion of another task as shown in Fig. 12.



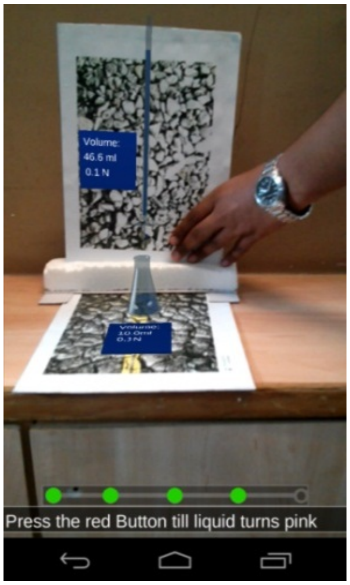
Place flask below burette

Figure 12: Step 3 of the Experiment



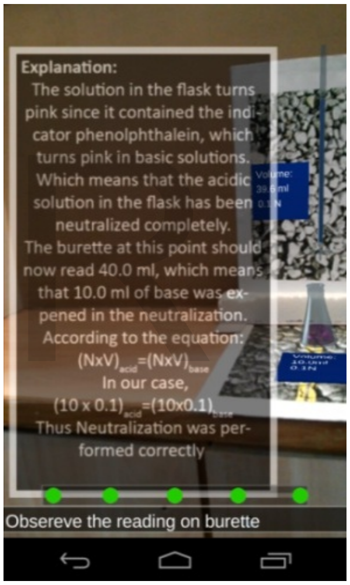
Press the red Button till liquid turns pink

Figure 13: Step 4 of the Experiment



Press the red Button till liquid turns pink

Figure 14: Step 5 of the Experiment



Explanation:
The solution in the flask turns pink since it contained the indicator phenolphthalein, which turns pink in basic solutions. Which means that the acidic solution in the flask has been neutralized completely.
The burette at this point should now read 40.0 ml, which means that 10.0 ml of base was expened in the neutralization.
According to the equation:
$(NxV)_{acid}=(NxV)_{base}$
In our case,
$(10 x 0.1)_{acid}=(10x0.1)_{base}$
Thus Neutralization was performed correctly

Obsereve the reading on burette

Figure 15: Explanation

The next step (Fig. 13) is placing the conical flask below the burette so as to ensure that the base drops directly into the conical flask.

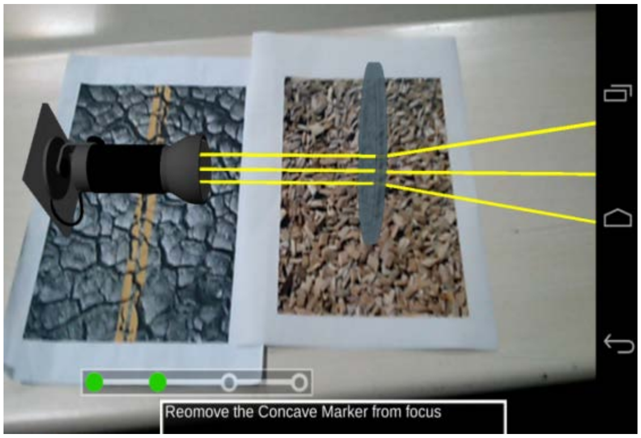
Fig. 14 shows the usage of the virtual button, which when pressed releases the base drop by drop into the conical flask. The virtual button is registered as pressed when the red square in Fig. 15 is obstructed from the camera view.

The final step of the titration experiment will be triggered when the liquid in the conical flask turns pink due to the overall increase in the base and the presence of phenolphthalein. This step involves providing an explanation so as to help the student understand what happened as he performed all the tasks as shown in Fig. 15.

If the user wants to do this experiment again or choose another experiment, they are free to do so by just clicking on the 'back button' and then choosing the experiment he/she would like to do.

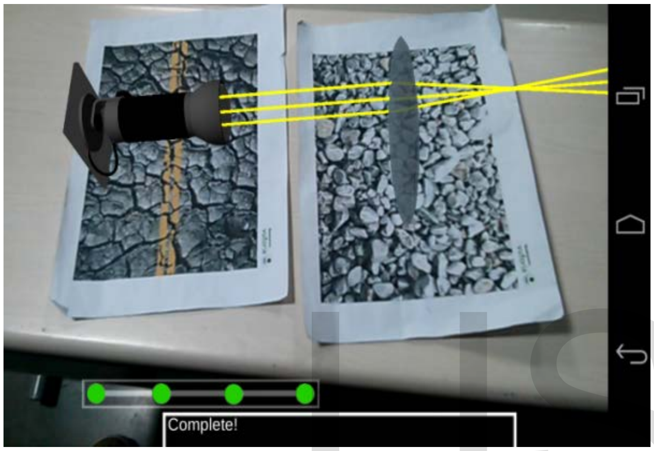
This gives a sample of how we can create a simple chemistry experiment using the power of Unity and Vuforia. The student is able to perform a lab experiment completely virtually, be guided through the procedure and receive an explanation at the same time.

The next sample experiment is a physics experiment to demonstrate the versatility of this approach. This sample experiment demonstrates the refraction of light through concave and convex lenses.



Reomove the Concave Marker from focus

Fig 16: Rays of light diverging through a concave lens



Complete!

Fig 17: Rays of light converging through a convex lens

As seen in Fig. 16 and Fig. 17, each marker replicates the functionality of each apparatus in this simple optics experiment. Similar to the titration experiment the user is guided through the process with a progress bar and instructions and he/she can see the results in real time.

We achieve the functionality in Fig. 16 and Fig. 17 by performing dynamic mesh modification using the Mesh class in Unity. The rays of light from the lamp are actually cylindrical meshes which are loaded into a mesh object as soon collision is detected with a lens. The locations of the vertices in the mesh are then modified so as to make them converge or diverge as required.

5.2 Experimental Evaluation

A study was conducted to understand that which approach of learning was preferred by the students for better understanding of the experiments. School students were asked to use all three learning approaches and provide their opinion about which approach helped them in better understanding of the experiments.

Table 1: Test Cases and Results

No.	Evaluation Parameter	Evaluation method	Evaluation Result
1.	Application Design	Usability testing was performed with the help of few students who are supposed to use the application in future.	Students found the application design and layout attractive and easy to use.
2.	Performance	Application was run under different scenarios such as under low intensity of light, phones with lower quality cameras etc.	Application was running smoothly and was generating correct output under all the circumstances.
3.	Covering of syllabus	All the important school level experiments were carefully studied to develop the application.	Important experiments from the syllabus were used to develop the application.

A survey was conducted in which students were asked to weigh our application on a scale of 100 based on different parameters. Aggregate of the ratings from the users was calculated and were used to plot the chart shown in Fig. 18 and Fig. 19.

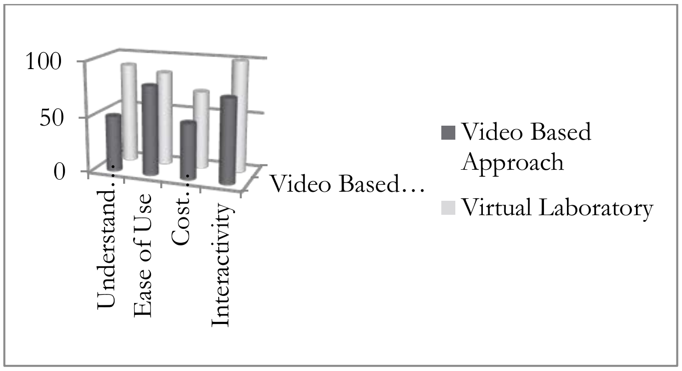


Fig 18: Survey Results

The result from the survey proved that the Virtual Reality (VR) based approach is more effective than the traditional approach.

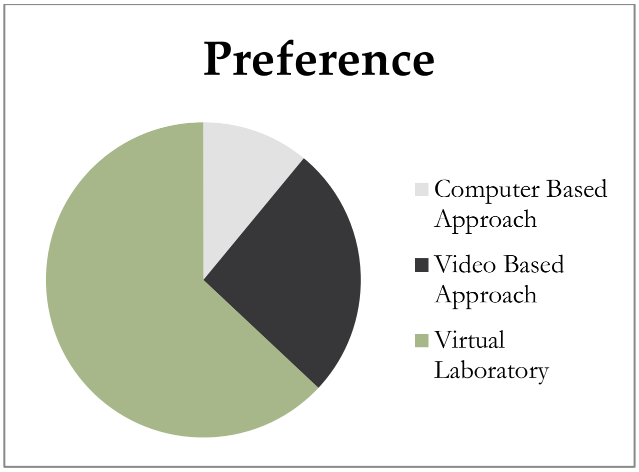


Fig 19: Survey Results

Study proved that the interactive approach, i.e., virtual reality based approach was preferred by 63% of the students whereas Video Based Approach and Computer Based Approach were preferred by only 26% and 11% of students respectively.

6 Conclusion And Future Scope

The application developed is extremely useful to the students since it enables them to perform many science experiments just by using their smartphones. Also the experiments can be performed very easily since the application has user friendly UI and functionality. The application helps students to understand each and every aspect of the experiment effortlessly since the application provides students with the on screen instructions/guide. Since the application provides with so many advantages over traditional way of performing experiments and also enhances the learning ability of the students and hence is a better way of performing experiments.

This application having been developed using modular design, more and more experiments can be added in the future to expand its scope.

Also, it can be combined with the functionality provided by VR technologies such as oculus rift headsets so as to make the application more immersive.

7 References

[1] VuforiaToolkit, https://developer.vuforia.com/resources/sdk/unity, Last accessed on April 1, 2014.

[2] Animations by Project OSCAR at IIT-Bombay, http://oscar.iitb.ac.in/availableanimations.do Last accessed on April 2, 2014.

[3] Performance Growth in Mobile, http://www.anandtech.com/show/5762/nvidia-plots-mobile-soc-gpu-performance-surpassing-xbox-360-by-2014, Last accessed on April 3, 2014.

[4] Unity Features Website, http://unity3d.com/features/whatsnew, Last seen on April4, 2014.

[5] Catchoom Website, http://catchoom.com, Last accessed on April 4, 2014.

[6] Total immersion Website, http://www.t-immersion.com, Last accessed on April 4, 2014.

[7] Vuforia Markers, https://developer.vuforia.com/resources/dev-guide/frame-markers, Last accessed on April 5, 2014.

[8] History of Unity 3D, http://news.dice.com/2013/06/03/how-unity3d-become-a-game-development-beast/, Last accessed on April 8, 2014.

[9] Vuforia Features, https://developer.qualcomm.com/mobile-development/add-advanced-features/augmented-reality-vuforia, Last accessed on April 8, 2014.

[10]M Gervautz, D Schmalstieg, Anywhere Interfaces Using Handheld Augmented Reality, *IEEE Computer*, 2012.

[11] Stephen Voida, Elizabeth D., Mynatt, Blair MacIntyre, and Gregory M. Corso, Integrating Virtual and Physical Context to support Knowledge Workers, *IEEE Pervasive Computing* 2002, 1(3), 73-79.

[12] Ronald Azuma, Yohan Baillot, Reinhold Behringer, Steven Feiner, Simon Julier, Blair MacIntyre, Recent Advances in Augmented Reality, Computer Graphics and Applications, *IEEE* 2001,21(6), 34-47.