##### AUGMENTED REALITY SETUP WITH MOBILE PHONES FOR REALTIME

##### SUMMER INTERNSHIP PROJECT REPORT

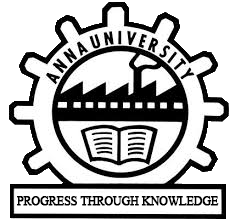
###### ***Submitted by***

**S SASHANK (2016103060)**

**APARAJIT K R (2016103505)**

**S HARINAARAAYAN (2016103029)**

**COLLEGE OF ENGINEERING GUINDY**



**ANNA UNIVERSITY: CHENNAI 600 025**

**MAY-JUNE 2018**

**ANNA UNIVERSITY: CHENNAI 600 025**

**BONAFIDE CERTIFICATE**

Certified that this project report **“Augmented reality setup with mobile phones in real time”** is the bonafide work of “**Sashank S(2016103060),**

**Aparajit K R(2016103505), S Harinaaraayan(2016103029)”** who carried out the project work under my supervision.

**COORDINATOR MENTOR**

**Dr. T Raghuveera**  **Dr.Arockia Xavier Annie R**

Assistant Prof Sr., Assistant Prof,

CS Dept, CS Dept,

CEG CEG

Anna University Anna University

**Dr. D Manjula**

**HEAD OF THE DEPARTMENT**

Department of Computer Science and Engineering

College of Engineering, Guindy

Anna University

**ACKNOWLEDGEMENT:**

We would like to thank our respectable Dean **Dr.T V Geetha** for providing us with this internship at our college where we could get a chance to do this project.

We thank our Head of Department **Dr. Manjula D** who provided us with the lab facilities and all that was necessary for us to complete the project.

We would also like to thank **Dr.Arockia Xavier Annie R** who mentored and guided us throughout our work. She provided us with the necessary direction to move forward with our work.

We thank our coordinator **Dr. T Raghuveera** whose efforts ensured proper communication which was instrumental in us completing our project on time.

We would also like to thank **Dr S Sudha** who provided us with the devices and instrumentation needed to test out our project.

We thank the research scholars **Mrs. Shiney Jeyaraj** and **Mr. Naveen Raju** who provided us with timely guidance and motivation throughout our intern.

**Thanking You,**

**Sashank S**

**Aparajit K R**

**S Harinaaraayan**

**TABLE OF CONTENTS**

**CHAPTER NO. TITLE PAGE NO.**

**LIST OF FIGURES**

**LIST OF TABLES**

**1. PROBLEM STATEMENT 7**

**2. INTRODUCTION 7**

**2.1 OBJECT DETECTION 7**

**2.2 OBJECT RECOGNITION 7**

**2.3 REVERSE IMAGE SEARCH 8**

**2.4 TEXT TO SPEECH 8**

**3. PROBLEM DESCRIPTION 8**

**4. WORKFLOW 9**

**5. IMPLEMENTATION 9**

**5.1 IMPLEMENTATION 9**

**5.1.1 TAKE PICTURE 9**

**5.1.2 SEND IMAGE TO CLOUD 10**

**5.1.3 SEND IMAGE URL TO GOOGLE 10**

**REVERSE IMAGE SEARCH**

**5.1.4 OXFORD DICTIONARY SEARCH 10**

**5.1.5 TEXT TO SPEECH CONVERSION 11**

**5.2 TOOLS 12**

**6. ANALYSIS 13**

**7. CONCLUSION 14**

**8. REFERENCES 15**

**8.1 ONLINE REFERENCES 15**

**9. RESULTS AND SCREENSHOTS 16**

**10. APPENDIX 18**

**LIST OF FIGURES**

**FIGURE NO. TITLE PAGE NO.**

**4.1 GENERAL OUTLINE OF WORK FLOW 9**

**5.1 START CAMERA 9**

**5.2 SEND IMAGE TO CLOUD 10**

**5.3 DISPLAY OF WORD AND MEANING ON 11**

**SCREEN**

**9 SCREENSHOTS 16**

**LIST OF TABLES**

**TABLE No. TITLE PAGE NO**

**5.1 TABULATION OF TOOLS 12**

**6.1 TYPES OF ANALYSIS 13**

**1 PROBLEM STATEMENT:**

To detect and recognize objects in an image. To do this we use reverse image search and vision API to determine the object and use text to speech to describe it.

**2.INTRODUCTION:**

**2.1 OBJECT DETECTION:**

**Object detection** is a computer technology related to [computer vision](https://en.wikipedia.org/wiki/Computer_vision) and [image processing](https://en.wikipedia.org/wiki/Image_processing) that deals with detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos. Well-researched domains of object detection include [face detection](https://en.wikipedia.org/wiki/Face_detection) and [pedestrian detection](https://en.wikipedia.org/wiki/Pedestrian_detection). Object detection has applications in many areas of computer vision, including [image retrieval](https://en.wikipedia.org/wiki/Image_retrieval) and [video surveillance](https://en.wikipedia.org/wiki/Video_surveillance).

**2.2OBJECT RECOGNITION:**

**Object recognition** – technology in the field of [computer vision](https://en.wikipedia.org/wiki/Computer_vision) for finding and identifying objects in an image or video sequence. Humans recognize

a multitude of objects in images with little effort, despite the fact that the image of the objects may vary somewhat in different view points, in many different sizes and scales or even when they are translated or rotated.

Objects can even be recognized when they are partially obstructed from view. This task is still a challenge for computer vision systems. Many approaches to the task have been implemented over multiple decades.

**2.3REVERSE IMAGE SEARCH:**

Google Reverse Search, available at **reverse.photos**, lets you search by images instead of keywords. Upload a picture from your desktop, tablet or mobile phone, and Google will show all the other web pages on the Internet that have similar images.

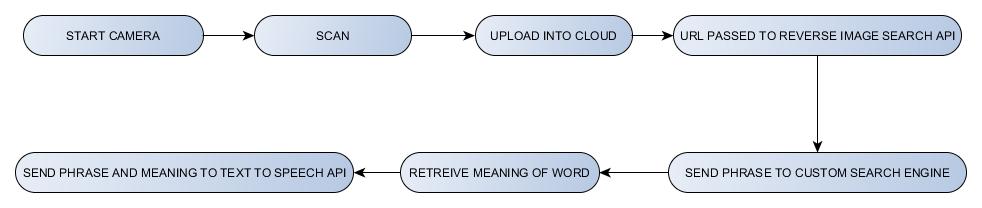
The [images are hosted](https://ctrlq.org/images/) anonymously on the Internet and cannot be discovered by other users.

**2.4TEXT TO SPEECH:**

**Text-to-Speech** (**TTS**) refers to the ability of computers to read text aloud. A **TTS Engine** converts written text to a phonemic representation, then converts the phonemic representation to waveforms that can be output as sound. TTS engines with different languages, dialects and specialized vocabularies are available through third-party publishers.

**3.PROBLEM DESCRIPTION:**

We capture the object as an image and upload it to a cloud service and then we pass the image url to google reverse search API.The API will return a key-phrase which we can then pass to the google custom search engine (or) Oxford dictionary API to retrieve the meaning.

**4.WORKFLOW:**

**4.1 GENERAL OUTLINE OF WORKFLOW**

**5 IMPLEMENTATION**

**5.1.1 TAKE PICTURE:**

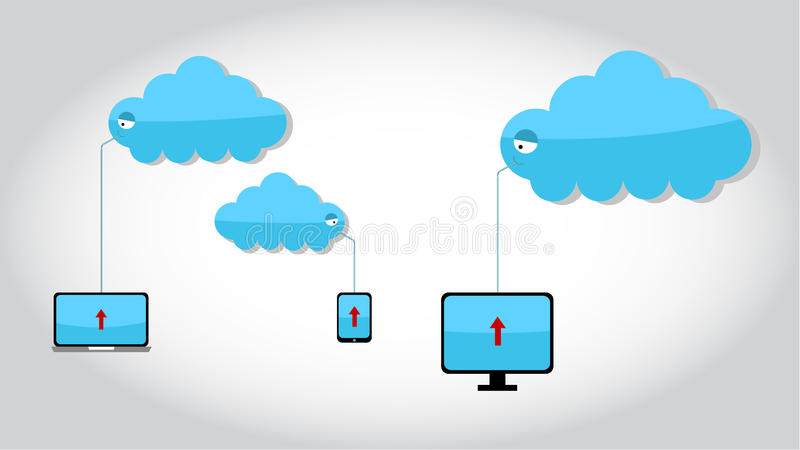
We first start the camera. We capture the frame and store it in a byte array.



**5.1 START CAMERA**

**5.1.2 UPLOAD IMAGE INTO CLOUD:**

Uploading image into cloudinary cloud service



**5.2** SENDING IMAGE TO CLOUD

**5.1.3.SENDING CLOUD URL TO GOOGLE REVERSE IMAGE SEARCH API:**

The URL of the image from the cloud service is sent to the reverse image search

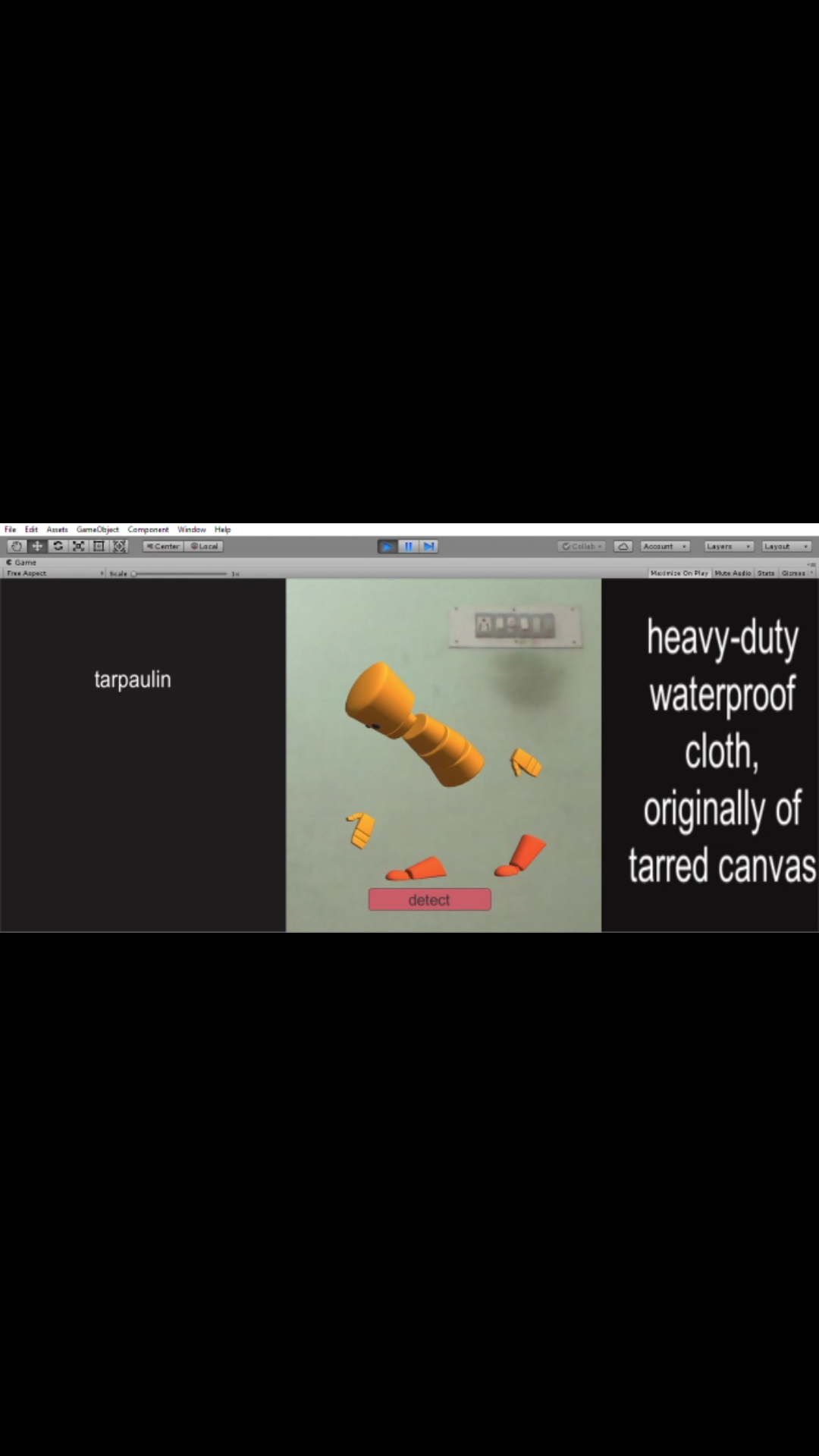
Engine and the word is acquired.

**5.1.4.OXFORD DICTIONARY SEARCH:**

Send the word or phrase to oxford dictionary API to find the description (or) meaning of the object :

**5.1.5. TEXT TO SPEECH CONVERSION:**

SEND THE WORD AND ITS MEANING TO TEXT TO SPEECH API:



5.3 DISPLAY OF WORD AND MEANING ON THE SCREEN

**5.2Tools:**

Since we use three APIs we will describe about them.

|  |  |  |
| --- | --- | --- |
| S.No | TITLE | DESCRIPTION |
| 1 | **Google reverse image search API** | This app lets you search by images using ( Google Reverse Search engine ) instead of keywords. |
| 2 | **Oxford Dictionary API** | The Oxford Dictionaries API allows easy access to our world-renowned dictionary content. We can pass the words to it and we can extract the meaning of those words. |
| 3 | **Google Text to speech API:** | We can say that this helps in speech synthesis by the name of the API.We recognise the word or definition using the other APIs and passing them to this API we can spell it out |
| 4 | **Cloudinary** | Cloudinary is the media management platform for web and mobile developers.  An end-to-end solution for all your image and video needs. |
| 5 | **Vuforia** | **Vuforia** is an Augmented Reality [Software Development Kit](https://en.wikipedia.org/wiki/Software_Development_Kit) (SDK) for mobile devices that enables the creation of [Augmented Reality](https://en.wikipedia.org/wiki/Augmented_Reality) applications. |
| 6 | **Unity** | Unity offers everything you need to build beautiful and engaging content, boost your productivity, and connect with your audience. |

5.1. TABULATION OF TOOLS

**6.ANALYSIS:**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TITLE** | **DESCRIPTION** |
| **1** | *REQUIREMENTS* | * The user’s device needs to have a standard camera * The environment should have proper lighting conditions * User’s device should have internet services enabled * The minimum API level required is API 24 which is the Android version Nougat(7.0) |
| **2** | *USE CASES* | * We can use this app to help kids improve their vocabulary and help them things about all the things they see at their hands’ disposal * It helps visually-challenged people to know what is before using the speech which comes out from the device * Has a high educational value and this app can be implemented as a futuristic learning way in schools and children will enjoy their learning |
| **3** | *FURTHER IMPROVEMENTS* | * This app can be integrated for security purposes in homes at the lowest cost possible * As of now, the app kind of struggles to differentiate between two similarly-structured objects placed close to each other which will be done in the future |

**6.1 TYPES OF ANALYSIS**

**7.Conclusion:**

A project well started is half solved. The better the clarity around what the team is attempting to fix, the more efficient they'll be in solving the problem, the solution will better 'fix' the issues, and the team can get back to executing the business versus fixing it.

We aimed at making a useful software that makes learning enjoyable.

This is not to say that the app is perfect and needs no improvement , we still feel the need to improve the project on its object recognition, say to recognize many objects at once.

We would like to improve the working of the app under low light and artificial lighting conditions

**8.REFERENCES:**

**8.1** **ONLINE REFERENCES:**

**Unity documentation:**

https://docs.unity3d.com/Manual/index.html

**Vuforia documentation:**

https://library.vuforia.com/api

**Wikipedia :**

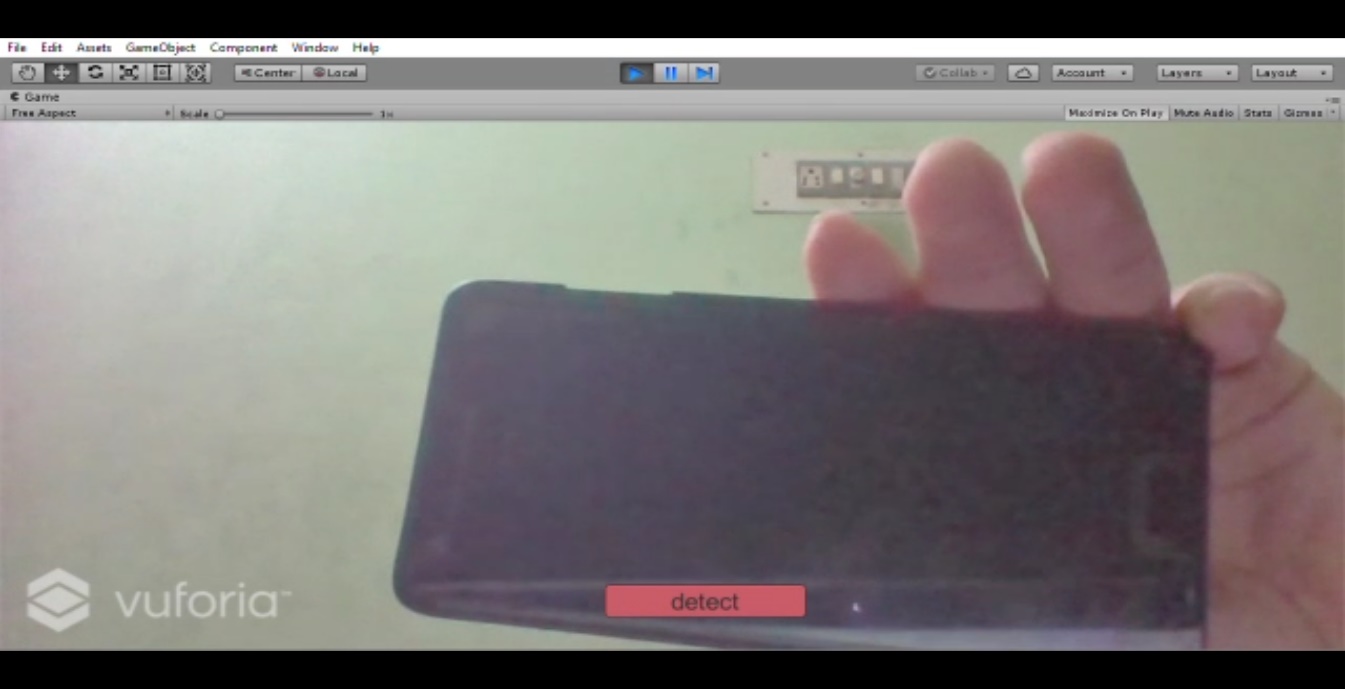
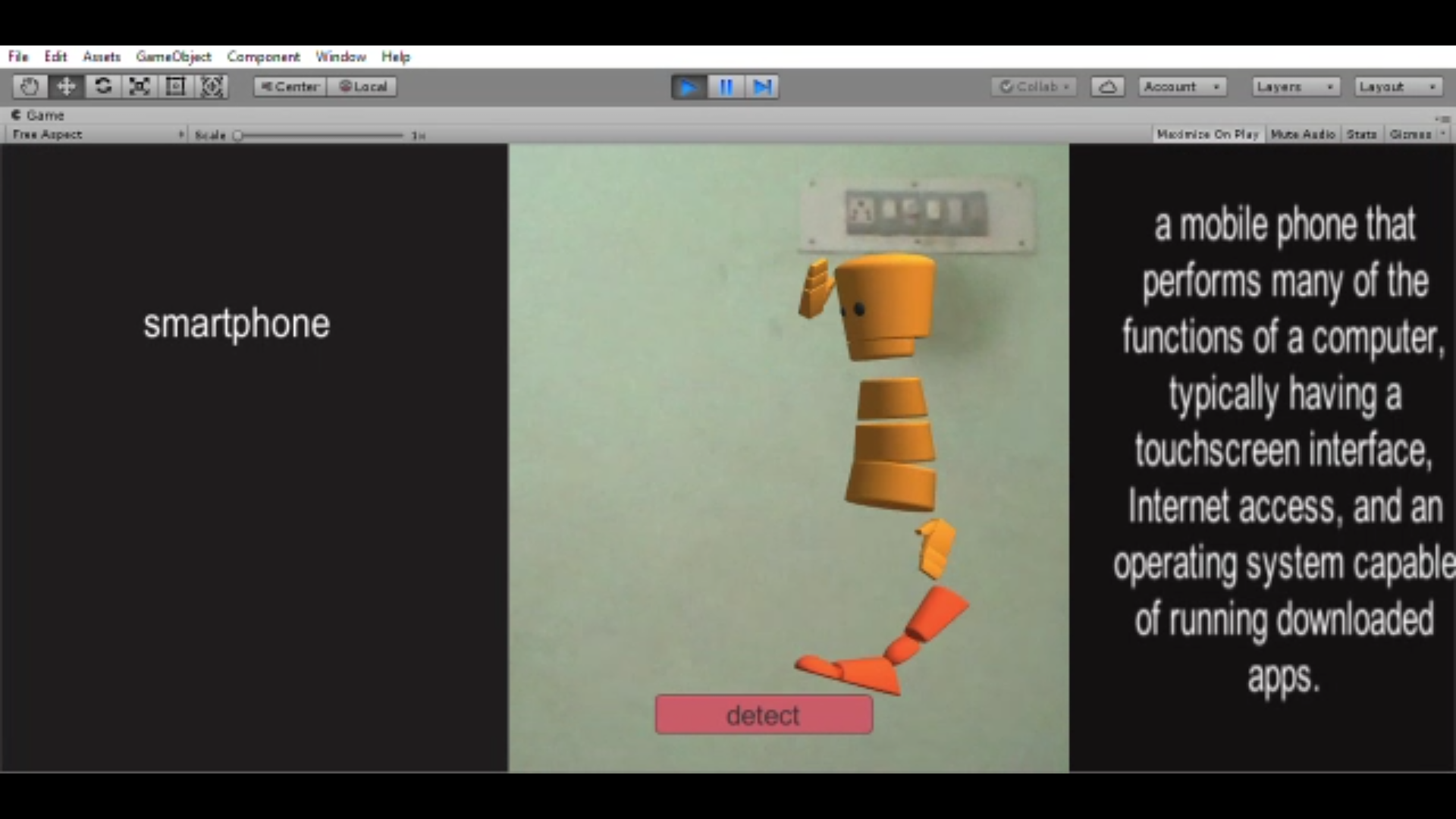
https://en.wikipedia.org/wiki/Augmented\_reality

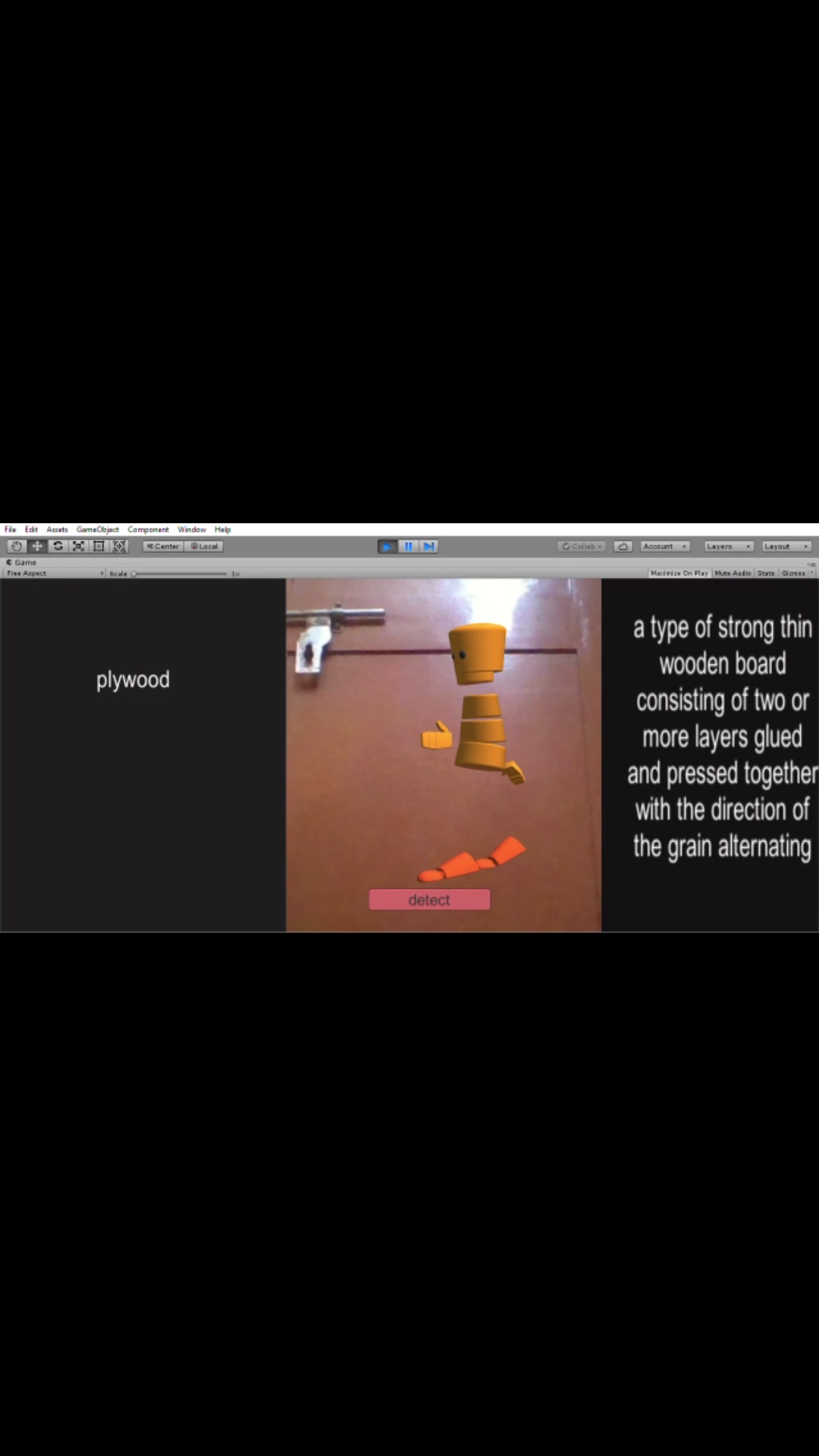
**Google reverse image search:**

http://in.pcmag.com/smartphones/96330/feature/how-to-do-a-reverse-image-search-from-your-phone

**Object recognition:**

https://virtualrealitypop.com/object-recognition-in-augmented-reality-8f7f17127a7a

**9.RESULTS AND SCREENSHOTS:**



**10.APPENDIX:**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using System.IO;

using System.Text.RegularExpressions;

using System;

using UnityEngine.UI;

public class Photo : MonoBehaviour {

public string definition { get; set; }

public AudioSource audio1;

byte[] imageByteArray;

private const string CLOUD\_NAME = "dylrioik3";

private const string UPLOAD\_PRESET\_NAME = "qylb43yg";

private const string CLOUDINARY\_API\_KEY = "284551392584861";

private const string CLOUDINARY\_SIGNATURE = "K5IE-CzFZS5HCZw\_DHz5G6MVQVo";

private string imageURl;

private GameObject status;

private GameObject scanningObject;

private GameObject cat;

public GameObject buttonObject;

public GameObject t1;

public GameObject t2;

private GameObject b1;

private GameObject b2;

private const string BASE\_URL = "http://www.google.com/searchbyimage?hl=ru&image\_url=";

private string wordsToSearch;

private const string GOOGLE\_API\_KEY = "AIzaSyD7SHc\_jTdUmsYteNArB2f7ME9LXzoTM-g";

private const string GOOGLE\_CUSTOM\_ENGINE\_ID = "002966606582515264909:32sslnvo-4g";

private const string GOOGLE\_SEARCH\_URL = "https://www.googleapis.com/customsearch/v1?cx=" +

GOOGLE\_CUSTOM\_ENGINE\_ID+"&key="+GOOGLE\_API\_KEY+"&cref&q=";

private const string OXFORD\_API\_KEY = "63c0e519bf3923479a4f666f1d27e947";

private const string OXFORD\_APP\_ID = "d866d9df";

private const string OXFORD\_SEACRH\_URL = "https://od-api.oxforddictionaries.com/api/v1/entries/en/{0}/definitions";

// Use this for initialization

private GameObject lb;

private GameObject rb;

private GameObject ub;

private GameObject bb;

private GameObject statusbg;

private GameObject c2;

void Start () {

cat = GameObject.FindGameObjectWithTag ("catjump");

cat.SetActive(false);

c2 = GameObject.Find("cat\_Walk");

c2.SetActive(false);

buttonObject = GameObject.Find ("scan");

t1 = GameObject.Find("word");

t1.SetActive(false);

t2 = GameObject.Find("meaning");

t2.SetActive(false);

lb = GameObject.Find("lborder");

lb.SetActive(false);

rb = GameObject.Find("rborder");

rb.SetActive(false);

ub = GameObject.Find("uborder");

ub.SetActive(false);

bb = GameObject.Find("bborder");

bb.SetActive(false);

b1 = GameObject.Find("bg1");

b1.SetActive(false);

b2 = GameObject.Find("bg2");

b2.SetActive(false);

status = GameObject.Find("status");

statusbg = GameObject.Find("statusbg");

statusbg.SetActive(false);

}

public void StartCamera(){

StartCoroutine ("TakePhoto");

}

public IEnumerator TakePhoto()

{

lb.SetActive(true);

rb.SetActive(true);

ub.SetActive(true);

bb.SetActive(true);

statusbg.SetActive(false);

string filePath;

//on mobile platforms persistentDataPath is already prepended to file name when using CaptureScreenshot()

if (Application.isMobilePlatform) {

filePath = Application.persistentDataPath + "/image.png";

ScreenCapture.CaptureScreenshot ("/image.png");

//must delay here so picture has time to save unfortunatly

yield return new WaitForSeconds(1.5f);

//Encode to a PNG

imageByteArray = File.ReadAllBytes(filePath);

}

else {

filePath = Application.dataPath + "/StreamingAssets/" + "image.png";

ScreenCapture.CaptureScreenshot (filePath);

//must delay here so picture has time to save unfortunatly

yield return new WaitForSeconds(1.5f);

//Encode to a PNG

imageByteArray = File.ReadAllBytes(filePath);

}

lb.SetActive(false);

rb.SetActive(false);

ub.SetActive(false);

bb.SetActive(false);

print ("photo done!!");

statusbg.SetActive(true);

status.GetComponent<Text>().text = "photo done!!";

StartCoroutine("UploadImage");

buttonObject.SetActive (false);

}

public IEnumerator UploadImage(){

c2.SetActive(true);

status.GetComponent<Text>().text = "uploading image...";

print ("uploading image...");

string url = "https://api.cloudinary.com/v1\_1/" + CLOUD\_NAME + "/auto/upload/";

WWWForm myForm = new WWWForm ();

myForm.AddBinaryData ("file",imageByteArray);

myForm.AddField ("upload\_preset", UPLOAD\_PRESET\_NAME);

WWW www = new WWW(url,myForm);

yield return www;

print (www.text);

status.GetComponent<Text>().text = "done uploading!!";

print ("done uploading!");

c2.SetActive(false);

yield return new WaitForSeconds(1.5f);

//parse resulting string to get image url

imageURl = www.text.Split('"', '"')[43];

print ("IMAGE URL: " + imageURl);

StartCoroutine ("reverseImageSearch");

}

public IEnumerator reverseImageSearch(){

status.GetComponent<Text>().text = "reverse image search...";

//create the full search url by adding all 3 together

c2.SetActive(true);

string fullSearchURL = BASE\_URL + WWW.EscapeURL(imageURl);

print (fullSearchURL);

WWWForm form = new WWWForm ();

var headers = form.headers;

headers ["User-Agent"] = "Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/56.0.2924.87 Safari/537.36";

WWW www = new WWW (fullSearchURL, null, headers);

//create a new www object and pass in this search url

yield return www;

string response = www.text;

print(response);

Match m = Regex.Match (response, "style=\"font-style:italic\">(.\*?(?=<))");

wordsToSearch = m.Groups [1].Value;

print (wordsToSearch);

status.GetComponent<Text>().text = "word identified!!";

c2.SetActive(false);

yield return new WaitForSeconds(1.5f);

StartCoroutine ("GoogleSearchAPI");

}

public IEnumerator GoogleSearchAPI(){

c2.SetActive(true);

status.GetComponent<Text>().text = "Searching for meaning...";

string words = wordsToSearch.Replace (" ", "\_").ToLower();

string url = String.Format(OXFORD\_SEACRH\_URL, words);

var headers = new Dictionary<String, String>();

headers ["app\_id"] = OXFORD\_APP\_ID;

headers ["app\_key"] = OXFORD\_API\_KEY;

headers ["Accept"] = "application/json";

WWW www = new WWW (url, null, headers);

yield return(www);

string result = www.text;

Match m = Regex.Match(result, "definitions\":.\*?(?=\")\"(.\*?(?=\"))", RegexOptions.Singleline);

definition = m.Groups[1].Value;

print(definition);

status.GetComponent<Text>().text = "Definition found!!";

c2.SetActive(false);

yield return new WaitForSeconds(1.5f);

cat.SetActive(true);

b1.SetActive(true);

t1.SetActive(true);

t1.GetComponent<Text>().text = wordsToSearch;

StartCoroutine(play(wordsToSearch));

int len = wordsToSearch.Length;

yield return new WaitForSeconds(2.0f);

t2.SetActive(true);

b2.SetActive(true);

t2.GetComponent<Text>().text = definition;

len += definition.Length;

StartCoroutine(play(definition));

yield return new WaitForSeconds(0.1f\*len);

buttonObject.SetActive (true);

t1.SetActive(false);

t2.SetActive(false);

b1.SetActive(false);

b2.SetActive(false);

cat.SetActive(false);

}

public IEnumerator play(string str)

{

string url = "http://translate.google.com/translate\_tts?ie=UTF-8&tl=en&client=tw-ob&q=";

string[] strArr = null;

strArr = str.Split(' ');

string txt = null;

foreach (string item in strArr)

{

txt += item;

txt+='+';

}

url+=txt;

WWW www = new WWW(url);

yield return www;

yield return new WaitForSeconds(1.5f);

print(www.isDone);

AudioClip au = www.GetAudioClip(false,true,AudioType.MPEG);

audio1.clip = au;

audio1.Play();

}