项目设计报告

姓 名： 罗强

班 级： 物联网61班

学 号： 2161200091

2020年5月4日

题目： 知识森林AR可视化系统安卓版本及功能改进

# 研究背景

增强现实（AR）技术的兴起为诸多领域带来新的活力，在教育领域也得到了广泛应用。增强现实技术，被定义为一种将现实世界与虚拟对象相结合的技术，它提供了真实和虚拟对象之间的交互。本项目设计的工作是将实验室研究成果——知识森林，进行基于增强现实技术可视化，实现一款安卓版的应用并改进和完善一些相关功能，为学习者提供更好的学习工具和资源。

随着知识爆炸式增长，知识碎片化问题日益凸显，为学习者带来认知过载、认知片面或偏差、认知迷航等学习问题。应对上述问题，实验室多位老师共同主导研发了知识森林[1]，它是一种知识碎片化问题的解决方案。为了让研究成果能够为学习者直接使用，需要进行可视化开发。可视化方案有多种，例如：(1) 基于网页Web技术的可视化；(2) 基于虚拟现实技术（狭义的）的可视化；(3)基于增强现实技术的可视化等。相对于网页端的可视化，虚拟现实和增强现实技术会给学习者带来新鲜感、立体感、真实感等体验，为此更能吸引学习者注意力、提高学习兴趣。从便捷性和广泛性考虑，虚拟现实技术的应用往往需要专用设备支持，而增强现实技术可以不需要专用设备而得到使用，仅仅是使用手机或者带摄像头的电脑等。因此，使用增强现实技术实现知识森林可视化具有众多优。有研究表明，在教育中使用AR技术工具为增加个人之间的互动和娱乐性学习提供了新的机会，也使学习过程增加注意力和学习动机，更加活跃、有效和有意义[2]。

# 相关研究

AR（增强现实）技术已经遍布在生活中了，例如拍照时添加的实时动画效果，亦或网上购物常用的“试衣服”等等。欧美等发达国家已经相当广泛地将AR应用于教育，其中包括高等教育和K12教育，并且取得不少积极的效果[3]。由于先前信息技术的不发达等原因，导致我国的虚拟现实技术起步晚于欧美各国，但是近年我国开始重视虚拟现实技术（广义的，包含VR/AR）[4]，在教育领域也是迅速的发展起来。首先是在高等教得到越来越广泛应用，K12教育在逐步跟进[5]。本项目设计的系统应用ARKF是同时面向高等教育和K12教育的，包含的课程数量还在逐步增加。

本项目是由Vuforia提供的AR SDK结合Google提供的AR Core作为AR引擎，以Unity 3D作为开发环境，C#为编程语言。Unity 3D是做游戏开发中最流行的软件之一，Vuforia AR也是最早一批做AR研发的，无论是技术成熟度，还是开发的产品兼容性，它们都是最优选择。为了适配更多安卓机（手机、平板等），还需要使用由Google开发AR Core作为本地AR引擎。

# 研究内容

鉴于在本项目之前已有一个适用于iOS的初版应用，应用研发不必从零开始。本项目主要工作：（1）切换AR引擎，在安卓系统实现AR场景；（2）添加安卓应用配置文件；（3）代码重构，功能模块化；（4）修复bug和功能改进，并发布知识森林AR可视化系统安卓版本应用——ARKF （1.5版本）。

知识森林AR可视化系统的整体框架，如下图：

手机屏幕截图

描述已自动生成

图1.系统整体框架图[[1]](#footnote-1)

## 切换AR引擎

系统中采用的AR引擎架构如下图所示：

手机屏幕截图

描述已自动生成

图2.系统中关于AR部分的框架图

先前iOS版本的应用是采用Vuforia+AR Kit作为引擎，但是AR Kit是嵌入iOS系统的，它仅支持苹果的设备。

对于Android系统的设备，由Vuforia引擎直接支持的设备是很有限的，主要是三星华为的高端手机和平板[6]，为了让更多用户使用该应用，因此本项目推荐采用Vuforia搭配AR Core作为AR引擎。AR Core是由Google开发的AR引擎，理论上采用Android 8及以上版本的设备都是可以拥有AR能力。

要使用AR Core需要有三个步骤：

1. 开发时，将AR Core开发包导入项目，本项目采用1.7.0版本的AR Core，需要将core-1.7.0.aar包（文件）导入到“Plugins/Android”目录下；
2. 生成应用APK之前，需要在配置文件AndroidManifest.xml中声明使用AR Core，如图3.；



图3.AR Core配置截图

1. 用户端需要安装“Google Play Services for AR”应用，又称“AR Core APK”。因为原生Android系统并不包含AR引擎，因此需要额外安装。

考虑到部分用户可能并不想下载AR Core APK以支持AR Core引擎，因此步骤（2）将配置中的“value”设置为optional，表示对于AR Core是可选的，设备没有安装AR Core APK时，将会由Vuforia引擎直接作为AR引擎，当然这也只限定于Vuforia支持的设备。

## 添加安卓应用配置文件

安卓应用配置文件AndroidManifest.xml是每一个安卓应用所必须的，它的主要作用是声明相关权限或者配置资源位置。Unity 3D这款开发软件也会自动地为开发者配置这个文件。但是这里还是需要单独提出来，是因为一些重要的配置是Unity没有自动添加进去的，例如上文中提到的AR Core的配置。

对于配置文件主要做了如下修改：

1. 修改支持的安卓系统最低版本，即设置Android SDK API level 为24，表示支持安卓7.0以及上系统。现在安卓手机和平板系统绝大多数都在该版本以上。
2. 添加相机的权限，相机是AR场景必需的一部分。
3. 授予网络权限，因为在使用应用过程中需要从知识森林数据库获取数据。
4. 设置屏幕旋转，分为竖屏、左旋横屏、右旋横屏；本应用都是允许的。
5. 添加AR Core配置，设为可选。

图4.是配置文件的部分截图：

电脑屏幕的照片上有文字

描述已自动生成

图 4. 安卓配置文件AndroidManifest.xml部分截图

## 代码重构

代码重构，这项工作主要目的是为了让整个工程代码更加规范。将代码中的功能模块化，可以很大程度上减少代码重复率、提高代码利用率，使程序更易阅读和逻辑更加清晰，也方便于后续开发，甚至是多人合作开发。

该应用是基于Unity 3D开发的，使用的编程语言为C#，它是一款面向对象的编程语言。在代码重构工作中可以充分利用其代码封装思想和面向对象设计思想，为项目开发提高效率和便捷性。

下面以有关网络请求的功能为例，简单叙述工作的内容以及意义：

需要网络请求的功能有：调用百度OCR服务、判断数据库是否存在课程该功能、获取课程数据功能等。先前的做法是为上述功能分别完整的编程：设定请求的URL和参数、发起网络请求、使用“try-catch”捕获异常及异常处理、接收网络响应结果、解析数据。其实，可以发现虽然上述功能不同，但是都做了一些同样的事情——网络请求。因此，可以做进一步改进，将这一部分通用的代码抽取出来，封装为了一个网络请求类（class），在这个类中实现如下功能：接收将要请求的URL和参数、配置参、发起网络请求、使用“try-catch”捕获异常及异常处理、接收网络响应结果、返回结果。之后，在各个功能模块中，只用调用网络请求类，给它传入相关参数，然后接收这个类返回的结果。可以看到，使用了类封装之后，各个功能模块不需要关心网请求的事情，自身代码量有可观的减少量，并且程序的逻辑和思路更加清晰。

由于该项工作涉及代码量较大，将在附录代码中展现。

## 修复bug和功能改进并发布应用

Bug修复主要有如下4个：

1. Bug描述：第二次进入树的界面时，在未点击小球的情况下，树不能旋转和缩放。解决过程：在先前版本中，从逻辑上分析，这个bug是不应该存在的；为此造成该bug的原因尚未具体明确；但是，自从把项目中主要代码进行了重构之后，该bug“自动”得到了修复。因此，该bug可能还是逻辑上的失误造成的，也验证了前文所提到的规范的代码的重要性。
2. Bug描述：触控太灵敏，很容易点一下操作两下。解决方案：修改触控模块中关于触控的持续时间。这是由于计算机执行速度极快，达到微秒甚至纳秒级别，而人的手指操作的反应时间最多在0.01s的级别。也就是说，在用户点击屏幕的“一瞬间”，而设备已经识别到几十次点击事件，因此会做出让用户感觉是点击了多次的操作，这里需要调整触控的持续时间。
3. Bug描述：旋转放大缩小时直接进入下一个界面。解决方案：改进触控算法，当识别到多指触摸屏幕时，屏蔽点击事件，只能执行放缩功能。
4. Bug描述：当请求第一个API出现错误时，没有处理这种情况。一直停留在正在识别，请稍等界面。解决方案：重新设计相关功能，让应用在请求网络API出现错误后，能够自动返回先前状态。

项目设计期间主要是针对OCR文字识别模块功能改进。存在问题：文字识别比较慢。解决方案：对将要进行OCR文字识别的图片进行预处理，现阶段项目采用的是固定阈值的图片二值化处理方法，为项目引入OpenCV for C#插件，调用该插件提供的图片处理函数[7]，先对图进行灰度化处理，然后进行二值化处理。实验中，图片二值化处理使用的是全局阈值，经过多次设定不同的阈值处理一批样本图片，通过对比处理后的图片，选定了处理效果较好的阈值：180 。进行测试发现二值化处理会提高OCR文字识别速度。本实验结果将在4.1小节展示。

应用发布，首先需要生成APK安卓应用安装包，之后可以分发给用户。曾经试图上架到华为、三星、小米等几大安卓设备厂商的应用商店，但是需要软件著作权证书，或者是企业开发者账户，目前还不具备相关条件，后续会尽量上架。目前可以通过ARKF的网站免费获取。

# 结果及分析

本节主要展现内容有：OCR文字识别提速的实验结果及分析、生成的ARKF应用使用时的部分截图。

## OCR文字识别提速的实验结果及分析

下图是以《计算机组成原理》这本书的封面图片进行二值化处理的结果：

手机屏幕的截图

描述已自动生成

图 5.不同阈值二值化处理结果

可以看出不同阈值处理的图片效果有些许差异，选定一个好的阈值，对图片进行二值化处理，将会明显区分出背景与文字信息。

之后对一组样本进行了OCR文字识别的测速，结果如下图，分别是原图、灰度处理的图片、二值化处理的图片：

手机屏幕的截图

描述已自动生成

手机屏幕的截图

描述已自动生成

图 6.不同处理方式的图片进行OCR测速结果

将测试结果列入下表：

表 1.不同图片处理方法OCR识别速度

|  |  |
| --- | --- |
| 图片处理方法 | 每张图片平均用时（s/张） |
| 原始图片  灰度处理  阈值为180的二值化处理 | 2.40955  1.83955  1.67814 |

从实验结果可以看出，图片预处理对OCR文字识别速度是有效果的。预处理的方法也很关键，经过不同方法处理的图片OCR识别速度有差异，采用阈值为180的二值化处理方法明显优于灰度处理。后续还会继续对这部分内容进行研究，采用动态的阈值来二值化处理不同的图片，希望更近一步改善该功能模块。

## 应用的实际使用

图7.中，左图是AR场景中知识簇布局，右图是点击某个知识簇进入其内部的布局，展示该知识簇下所有主题的布局，沙盘左上角蓝色标签内显示课程名：

图片包含 桌子, 游戏机, 显示器

描述已自动生成 图片包含 游戏机, 电脑

描述已自动生成

图 7.ARKF应用的部分截图

以上实例是运行在陕西省天地网技术重点实验室提供的一款三星平板（三星Tab S6）上的截图。

本项目中，知识森林AR可视化系统安卓版本基本实现，并且可以流畅地运行在安卓设备上，现实了主要功能：拍照识别课程、AR场景展示课程下的知识簇、AR场景展示知识簇内的主题树、展现具体一颗主题分面树、展示知识碎片、关键词的实时跟踪的AR场景，达到了预期目标。

# 后续研究

系统应用中还存在一些有待改进的功能，例如OCR识别模块提速、知识簇内布局优化等；为了充分体现知识森林的作用，还可以在应用中添加导学路径场景，进一步解决学习者“迷航”问题。总之，以后将会以“提升用户体验、提高学习效率、充分发挥知识森林优势”为宗旨，为系统增添或优化一些功能。这将作为之后毕业设计的主要内容。

# 参考文献

1. Zheng Q, Liu J, Zeng H, Guo Z, Wu B, Wei B. Knowledge forest: a novel model to organize knowledge fragments[R]. Science China Information Science,2019.
2. 莫灿.沉浸式增强现实与高等教育的融合探索[A]. 教育教学论坛，2020.
3. 赵文静, 曹忠. 国外增强现实教育应用的研究进展 \*——基于104篇国际教育技术期刊论文的质性元分析[J]. 中国教育信息化·基础教育 / The Chinese Journal of Ict in Education. 2019;(5):30.
4. 徐子译, 崔虹云, Xu Ziyi, Cui Hongyun. 国内外增强现实技术及其在教学领域的现状分析 / Augmented Reality Technology at Home and Abroad and Its Current Situation in the Field of Teaching[J]. 广东化工 / Guangdong Chemical Industry. 2019.
5. 张栌月, 程明智, 李豪等. 增强现实的K12教育应用综述 / A summary of the application of augmented reality in K12 education[J]. 北京印刷学院学报 / Journal of Beijing Institute of Graphic Communication. 2019.
6. Vuforia支持的Android设备. <https://library.vuforia.com/content/vuforia-library/en/platform-support/vuforia-engine-recommended-devices.html>
7. Adrian Kaehler , Gary Bradski. Learning OpenCV 3 COMPUTER VISION IN C++ WITH THE OPENCV LIBRARY[M]. 2017.

# 附录

## 关于封装网络请求相关的部分代码

|  |
| --- |
| /\*  \* 百度OCR接口部分  \* \*/  public static string BaiduOCR(object imageObject)  {//文字识别，使用百度。  baiduOCRapi.Timeout = 60000; // 修改OCR超时时间  byte[] image = (byte[])imageObject; // 拆箱  var options = new Dictionary<string, object>{  {"detect\_direction", "true"}  }; // 增加识别方向参数,识别单字结果  var ocrResultJson = baiduOCRapi.General(image, options); // 带参数调用通用文字识别（含位置信息版）  string ocrResultString = ocrResultJson.ToString(); // ocrResultJson为OCR结果json格式  Debug.Log(ocrResultString);  return ocrResultString;  }  public static string GetKeywords(string ocrResult = "", object imageObject = null)  {//将OCR结果 通过实验室API1 与 实验室数据库匹配，判断是否有该 关键字（词）。  if (ocrResult == "")  {  if (imageObject == null)  {  return "";  }  ocrResult = BaiduOCR(imageObject);  }  return JudgeDomainAPI1(ocrResult);  }  public static string JudgeDomainAPI1(string ocrResult)  {  // 开始请求API1获得追踪位置坐标与关键词及其位置坐标  string api1Url = root + "judgeDomainAndGetLocation/?OCR\_result=" + ocrResult;  string OCR\_RESULT = HttpHelperGetJson(api1Url);  return OCR\_RESULT;  }  //从实验室API3 获取课程全部数据。  public static string GetAllByTopicNameAndDomainId(string itemName, string domainid)  {  string api3Url = root + "get\_AllInByTopicNameAndDomainId/?topic=" + itemName + "&domainid=" + domainid;  // Debug.Log("GetAllByTopicNameAndDomainId(string itemName, string domainid):"+itemName+domainid);  // var api3Url="http://yotta.xjtushilei.com:8000/crystal/get\_AllInByTopicNameAndDomainId/?topic=天气&domainid=488";  string api3\_result = HttpHelperGetJson(api3Url);  return api3\_result;  }  //API2 得到这门课程所有主题划分好的簇，以及簇间的顺序,同时会返回domainid.  public static string GetClusterDivided(string clusterName)  {  string api2Url = root + "clusterDivided/?domain=" + clusterName;  string api2\_result = HttpHelperGetJson(api2Url);  return api2\_result;  }  static string HttpHelperGetJson(string url)  {  string result;  HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);  request.Method = "GET";  request.ContentType = "application/json";  // 捕获请求失败的异常  try  {  HttpWebResponse response = (HttpWebResponse)request.GetResponse();  StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);  result = reader.ReadToEnd();  Debug.Log("Http请求响应的结果：" + result);  }  catch (WebException e)  {  // WebResponse webRsp = (HttpWebResponse)e.Response;  // StreamReader reader = new StreamReader(webRsp.GetResponseStream(), Encoding.Default);  Debug.Log("网络请求出现了问题:");  Debug.Log(e.ToString());  result = "";  }  request.Abort();  return result;  } |

## 关于OCR文字识别的代码

|  |
| --- |
| tempFrame = VuforiaRenderer.Instance.VideoBackgroundTexture; // 取到一帧  frameOCR = new Texture2D(tempFrame.width, tempFrame.height, TextureFormat.RGB24, false);  Utils.textureToTexture2D(tempFrame, frameOCR); // texture 转换成 texture2D  Mat imageMat = new Mat(frameOCR.height, frameOCR.width, CvType.CV\_8UC3);  Utils.fastTexture2DToMat(frameOCR, imageMat);  Mat grayMat = new Mat(frameOCR.height, frameOCR.width, CvType.CV\_8UC3);  Mat binMat = new Mat(frameOCR.height, frameOCR.width, CvType.CV\_8UC3);  Imgproc.cvtColor(imageMat, grayMat, Imgproc.COLOR\_BGR2GRAY);  Utils.matToTexture2D(grayMat, frameOCR);  var frameOCR2 = ProcessImages.VerticalFlipPicture(frameOCR); // 翻转图片。需要这一步操作，否则是镜像的。  byte[] frameOCRbyte = frameOCR2.EncodeToJPG(OCR\_IMAGE\_QUALITY); // 转换成PNG格式的图片  //SaveImage.Save(frameOCRbyte, "image3.jpeg");  object imageOCR = frameOCRbyte; // 装箱  Thread thread = new Thread(new ParameterizedThreadStart(GetDataFromWeb.GetKeywordsT)); // 新建一个线程  thread.Start(imageOCR); // 开启线程，并传入参数image |

1. 该图片由武云封师兄提供。 [↑](#footnote-ref-1)