基于Flutter的校园交友平台的开发

作者姓名：肖骏 专业班级：软件工程4班 指导老师：邓飞

摘 要

近几年，随着物联网时代到来、移动互联网高歌猛进，日新月异，在很多业务场景中，传统的纯原生开发已经不能满足日益增长的业务需求。主要表现在：动态化内容需求增大；当需求发生变化时，纯原生应用需要通过版本升级来更新内容，但应用上架、审核是需要周期的，这对高速变化的互联网时代来说是很难接受的，所以，对应用动态化(不发版也可以更新应用内容)的需求就变的迫在眉睫。业务需求变化快，开发成本变大；由于原生开发一般都要维护Android、iOS两个开发团队，版本迭代时，无论人力成本，还是测试成本都会变大。原生应用程序是指某一个移动平台（比如iOS或安卓）所特有的应用，使用相应平台支持的开发工具和语言，并直接调用系统提供的SDK API。比如Android原生应用就是指使用Java或Kotlin语言直接调用Android SDK开发的应用程序；而iOS原生应用就是指通过Objective-C或Swift语言直接调用iOS SDK开发的应用程序。原生开发有以下主要优势：可访问平台全部功能（GPS、摄像头）；速度快、性能高、可以实现复杂动画及绘制，整体用户体验好；主要缺点：平台特定，开发成本高；不同平台必须维护不同代码，人力成本随之变大；内容固定，动态化弱，大多数情况下，有新功能更新时只能发版。目前常用的跨平台技术有：（1）H5+原生（Cordova、Ionic、微信小程序）（2）JavaScript开发+原生渲染 （React Native、Weex、快应用）3）自绘UI+原生(QT for mobile、Flutter) 。（杜文，2020）

关键词: Flutter,Android,SpringBoot,跨平台

**Development of campus dating platform based on Flutter**

**Abstract**: In recent years, with the advent of the Internet of Things and the rapid advancement of the mobile Internet, with each passing day, in many business scenarios, traditional pure native development has been unable to meet the growing business needs. Mainly reflected in: the demand for dynamic content increases; when the demand changes, pure native applications need to update the content through version upgrades, but the application shelves and audits require a cycle, which is difficult for the rapidly changing Internet era Accepted, therefore, the need for application dynamics (the application content can be updated without publishing a version) becomes imminent. Business requirements change rapidly, and development costs become larger. Since native development generally maintains two development teams, Android and iOS, when the version is iterated, regardless of labor costs and testing costs, it will become larger. A native application refers to an application unique to a mobile platform (such as iOS or Android), uses the development tools and languages ​​supported by the corresponding platform, and directly calls the SDK API provided by the system. For example, Android native application refers to an application developed by directly calling Android SDK using Java or Kotlin language; iOS native application refers to an application developed by directly calling iOS SDK through Objective-C or Swift language. Native development has the following main advantages: access to all functions of the platform (GPS, camera); fast speed, high performance, complex animation and drawing, and good overall user experience; main disadvantages: platform specific, high development cost; different platforms must be maintained With different codes, the labor cost will increase accordingly; the content is fixed and the dynamics are weak. In most cases, the version can only be issued when there are new feature updates. Currently commonly used cross-platform technologies are: (1) H5 + native (Cordova, Ionic, WeChat applet) (2) JavaScript development + native rendering (React Native, Weex, fast application) 3) self-painting UI + native (QT for mobile, Flutter)

**Key words**: Flutter,Android,SpringBoot,Cross-platform

目 录

[第1章 前 言 1](#_Toc483610767)

[1.1 软件开发背景及目的 1](#_Toc483610768)

[1.2 研究现状 1](#_Toc483610769)

[1.3 本文的主要工作 2](#_Toc483610770)

[1.3.1 需求整理与软件设计 2](#_Toc483610771)

[1.3.2 系统开发与开发成果 3](#_Toc483610772)

[第2章 开发工具简述 4](#_Toc483610773)

[2.1 开发工具与iOS系统的介绍 4](#_Toc483610774)

[2.1.1 iOS系统及核心组件 4](#_Toc483610775)

[2.1.2 iOS系统流行的两种设计模式及开发工具介绍 4](#_Toc483610776)

[2.2 开发环境的搭建 6](#_Toc483610777)

[2.3 关键技术的研究 8](#_Toc483610778)

[2.3.1 EasyAR-SDK的使用与研究 9](#_Toc483610779)

[2.3.2 Unity3D和iOS的混合开发及3DMAX介绍 9](#_Toc483610780)

[第3章 需求分析和概要设计 11](#_Toc483610782)

[3.1 软件需求分析与开发任务 11](#_Toc483610783)

[3.3 总体设计 13](#_Toc483610784)

[3.3.1 系统流程 13](#_Toc483610785)

[3.3.2 EasyAR说明 14](#_Toc483610786)

[第4章 系统实现与测试 16](#_Toc483610787)

[4.1 用户交互UI等相关的设计 16](#_Toc483610788)

[4.2 核心AR功能的实现 16](#_Toc483610789)

[4.2.1 识别图片的设计 16](#_Toc483610790)

[4.2.2 3D模型的获取与展示 18](#_Toc483610791)

[4.2.3 3D模型实景渲染 22](#_Toc483610792)

[结 论 24](#_Toc483610793)

[致 谢 25](#_Toc483610794)

[参考文献 26](#_Toc483610795)

# 第1章 前 言

## 1.1 软件开发背景及目的

近年来，随着移动互联网的迅速发展，手机日渐深刻的在改变着人们的生活方式。越来越多的app渗入我们的日常生活，包括网约打车，点外卖，线上购买商品，远程医疗等等。而这一切都需要有专门的公司负责相关app的开发，但是作为主流的手机操作系统，Android和IOS,他们都有自己的开发平台和开发语言，所以在app开发过程中，许多公司使用的解决方案是分别招聘Android和IOS的开发人员，分别负责相关的开发。这无疑给开发过程中增添了许多不必要的麻烦，包括兼容性问题，适配问题。为了使开发者更加专注app的开发，而不是跨平台的学习，业界提出了跨平台的方案，包括Facebook 于2015年四月开源的跨平台移动应用开发框架ReactNative，阿里巴巴公司与2016年6月开源的一种用于构建移动跨平台的 UI 框架Weex, （陈阿票，2016）此外包括Google 开源的 UI 工具包Flutter，帮助开发者通过一套代码库高效构建多平台精美应用，支持移动、Web、桌面和嵌入式平台。（百度，2020）可以说，未来的主流开发应该是不分平台的，这也会大大减少各个公司的开发成本。

本题主要使用了Flutter，这种技术最早于2015年提出，Flutter的第一个版本被称为“Sky”，运行在Android操作系统上。它是在2015年Dart开发者峰会 上亮相的，其目的是能够以每秒120帧的速度持续渲染。Flutter包括许多特性，例如，快速开发，绚丽UI,响应式，访问原生功能。Flutter 组件采用现代响应式框架构建，这是从React中获得的灵感，中心思想是用组件(widget)构建你的UI。 组件描述了在给定其当前配置和状态时他们显示的样子。当组件状态改变，组件会重构它的描述(description)，Flutter 会对比之前的描述， 以确定底层渲染树从当前状态转换到下一个状态所需要的最小更改。（百度，2018）

## 1.2 发展现状

校园交友平台选择了Flutter， Flutter是当今热门的跨平台开发框架，Flutter 构建应用时，可以让开发者对 UI 实现像素级的控制。意味着原本的设计意图可以被完美地执行出来，从而将品牌个性忠实地传达给用户。Flutter 的 UI 渲染性能很好。在生产环境下，Flutter 将代码编译成机器码执行，并充分利用 GPU 的图形加速能力，因此使用 Flutter 开发的移动应用即使在低配手机上也能实现每秒 60 帧的 UI 渲染速度。Flutter 同时做到 “美观” 和 “快速” 的原因，在于其架构本身。Flutter 引擎使用 C++ 编写，包括高效的 Skia 2D 渲染引擎，Dart 运行时和文本渲染库。这个引擎使得 Flutter 框架可以自由、灵活、高效地绘制 UI 组件。而应用开发者则可以用 Flutter 框架来轻松实现各种设计语言和动画效果。Flutter 是开放的，它是一个完全开源的项目。全球的开发者都可以免费使用和拓展 Flutter 的源代码，并为 Flutter 的生态和文档作贡献。 我们已经看到许多中国开发者活跃在社区中，并为 Flutter 做出了坚实的贡献。 Flutter 是开放的，它是一个完全开源的项目。全球的开发者都可以免费使用和拓展 Flutter 的源代码，并为 Flutter 的生态和文档作贡献。 我们已经看到许多中国开发者活跃在社区中，并为 Flutter 做出了坚实的贡献。(董韬,2019)

## 1.3 本文的主要内容

### 本次毕业设计的基本要求是利用Flutter设计出一款跨平台的校园交友平台app。实现利用Flutter进行快速的开发以及构建富有表现力和灵活的UI，达到和iOS和Android上原生应用一样的性能。实现后端数据基Springboot+Jpa+MySQL的数据存储，实现用户登录，异常处理，密码加密等功能。

### 1.3.1 需求整理与软件设计

通过对“一次性买到合适家具难”这个问题进行分析，结合用户的期望和实际生活情况以及本APP的宗旨，整理需求如下：

（1）用户可以通过本APP对大型家具模型进行各个角度的观察；

（2）用户可以通过本APP将大型家具模型放在实景中来观察、搭配；

（3）在本APP中用户可以操作大型家具模型的角度和大小；

（4）在本APP中用户可以对模型的角度和大小进行锁定；

（5）在本APP中用户可以将模型和实景的搭配结果进行截图保存。

易摆AR家具预摆放APP-iOS客户端的总体设计风格的就是把最核心功能以最直观的方式展示到应用用户面前，所以应有的操作流程如下：

安装完成后，打开APP，会直接进入摄像头权限获取部分，当获取成功后，会直接来到摄像界面，所有的UI也都是存在于摄像界面，用户可以最直接的使用本APP。利用手机摄像头取景识别图，就能进行3D模型渲染，操作简单易懂，能让用户最快学会应用的使用方式。

### 1.3.2 系统开发与开发成果

本AR-APP开发主要分为以如下阶段：

（1）家具模型的设计。按照实际1:1大小进行3DMAX家具模型设计制作；

（2）AR SDK的选择和研究。对AR SDK的文档和使用规范说明进行研究，并对其合理使用和拓展，进行APP的核心设计开发；

（3）APP的设计和实现。在选定了AR SDK后，就该着手APP部分的设计和编码了，首先通过Axure进行APP的原型设计，包括UI、交互等设计；

（4）把制作好的家具模型放进Unity 3D开发工具里，对模型进行成像和操作等方面的设计，包括模型的缩放，模型的旋转操作，模型脱离识别图片后持续存在等，并通过Unity 3D导出为iOS应用；

（5）使用Xcode打开导出的项目文件，实现之前原型图设计的UI和UI交互，优化APP界面，使操作更加人性化和简便，突出APP的合理性设计，此部分完全使用Objective-C实现；

（6）在开发的过程中不断对已开发部分进行功能测试和真机调试，对收集到的Crash和Bug进行及时处理，提高应用稳定性。

易摆AR家具预摆放APP-iOS客户端的功能有：

（1）摄像头取景，识别预览图片；

（2）通过识别出来的相应的预览图片进行相应的3D模型的渲染和展示；

（3）调整好了模型的角度，可以通过模型识别锁定按钮进行模型角度锁定；

（4）当和实景配合完成可以通过按下home键和锁屏键进行截图保存。

易摆AR家具预摆放APP-iOS客户端能够稳定运行，能够实现预期设计的功能，对可能出现的Crash和BUG进行了一定的解决，设计合理，逻辑清晰，本应用交互合理，实用价值大，是一款非常有价值的应用。

第2章 开发工具简述

## 2.1 开发工具与iOS系统的介绍

### 2.1.1 iOS系统及核心组件

iOS应用的开发是基于Objective-C语言以及Swift语言进行的，iOS是美国APPLE公司推出的非开源智能移动设备操作系统，主要应用于APPLE公司的智能移动设备如：iPhone、iPad、APPLE Watch等。iOS系统的设计理念主要有三点：简洁扁平化、操作流畅化、交互人性化。首先，iOS系统最受人喜爱的一点就是其界面极其简单大气，iOS系统的工程师擅长将很繁琐的东西简洁化，这也使得该系统能在Android横行的年代脱颖而出并反超。再次，iOS系统操作极其流畅，通过对系统的高度优化以及对应用交互的严格规定，iOS系统的用户体验相当好，很少出现卡顿、延迟、交互机械化等问题。最后，iOS系统是一款非开源的系统，虽然非开源对其可扩展性造成了重创，但是非开源也很好的规范了iOS系统的开发模式，大大提高了各种APP的规范化和稳定性。这也侧面提升了iOS系统最优秀的易交互特点。

iOS系统的核心组件可以分为以下三个部分：

1. NavigationController，iOS APP的导航控制器，是一个应用的核心部分，控制了APP的交互流程，从一个界面如何转到另一个界面，这就是它的职责。一个合理的APP设计第一步就是设计好NavigationController的逻辑结构；
2. ViewController，iOS APP的视图控制器，是iOS系统MVC模式的最直接体现，一个应用的界面核心，界面的UI设计、数据处理、交互设计都是在这上面完成的；
3. 沙盒存储空间，iOS APP的数据存储空间，是iOS系统沙盒特色的体现，通过分配给各种不同的APP 不同的沙盒空间，可以将不同的APP完全分隔开来，不会出现数据的混乱等情况[[[1]](#endnote-2)]。

### 2.1.2 iOS系统APP的开发

iOS系统目前流行的设计模式主要有MVC模式和MVVM模式。MVC模式即模型-视图-控制器模式，是iOS系统中最常见了的设计模式，也是系统最基本的设计要求。该模式中，要求将数据模型和视图完全分隔开来，将数据的处理以及相应的UI变化放在控制器中实现，提高程序的可维护性，UIViewController就是这个设计模式的最直观体现。但MVC模式还是存在一定的缺点，缺点就是它的控制器部分会很臃肿，为了解决这个问题，MVVM这种设计模式就应运而生，并在iOS系统开发中流行起来。

MVVM即，Model-View-View Model模式，实际数据交互模式为Model-View Model-ViewController-View，其中ViewModel模块进行数据的处理和操作，ViewController进行界面UI的变化等操作，相当于把MVC中的C拆分为两个部分，一定程度减少了某一个模块的臃肿化。本题选用MVC设计模式，因为本题UI方面的内容不多，ViewController更多的是对数据和对象的操作，所以不会造成ViewController的臃肿。

iOS系统的开发工具Xcode，Xcode是Apple的专用开发工具套件，支持项目管理、编辑代码、构建可执行程序、代码级调试、代码的版本管理、性能调优等。Interface Builder的控件库中可以通过拖拽创建诸如窗口，视图，菜单，按钮等，保存后的nib文件包括了控件对象，控件配置，以及控件布局等信息，程序加载nib文件时根据这些信息创建对象。同时使用Xcode和Interface Builder工具，简化代码，便于维护，加快开发，使用Xcode来设计程序的逻辑，使用Interface Builder 来设计程序的界面。iPhone模拟器是可以在电脑上部署和测试你的应用程序的目标平台iPhone模拟器提供了类iPhone的环境，通过菜单选项，可以模拟屏幕锁，屏幕转向，内存警告等功能。因为电脑没有提供一些必要硬件，所以iPhone模拟器也有功能限制[[[2]](#endnote-3)]。

优秀的游戏开发引擎Unity3D，Unity3D是由Unity Technologies开发的一个让玩家轻松创建诸如三维视频游戏、建筑可视化、实时三维动画等类型互动内容的多平台的综合型游戏开发工具，是一个全面整合的专业游戏引擎。Unity类似于Director, Blender game engine, Virtools 或 Torque Game Builder等利用交互的图型化开发环境为首要方式的软件其编辑器运行在Windows 和Mac OS X下，可发布游戏至Windows、Mac、Wii、iPhone和Android平台，也可以利用Unity web player插件发布网页游戏、手机游戏，支持Mac 和Windows的网页浏览[[[3]](#endnote-4)]。

## 开发环境的搭建

本题开发过程中使用了Mac OS系统，IDE使用了Unity3D和Xcode，技术使用了Unity3D开发和iOS开发，详细的开发环境搭建如下：

（1）通过下载Unity3D开发环境，通过下载下来的程序进行Unity3D的安装和配置。图2-1为Unity3D安装界面。

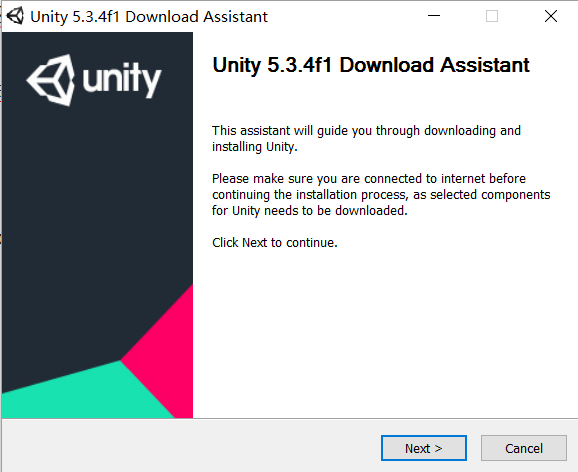


图2-1 Unity3D安装界面示意图

如图2-2所示，Unity 默认勾选的，是 Unity 游戏引擎本身；建议不要去掉，其余组件如Android Build Support 和iOS Build Support可以选择性支持。本次设计中需要安装iOS Build Support；

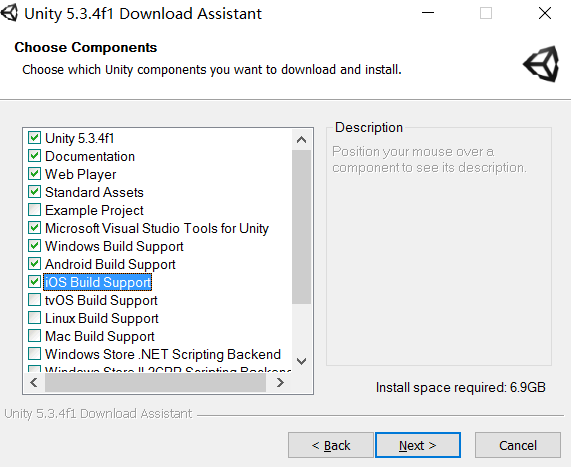


图2-2 Unity3D安装界面-安装组件选择

（2）Unity3D安装完成后，需要安装一款Mac OS上的C#编译器Xamarin，其对应网站https://developer.xamarin.com/。Xamarin作为一个跨平台开发框架，Xamarin.Mobile有很多优点。在这一框架内，开发iOS、Android、Windows Phone和Mac App应用可以不用转到Eclipse 或者额外购买Mac并使用Xcode，而继续在Visual Studio之中使用C#与.NET Framework进行[[[4]](#endnote-5)]。如图2-3所示为Xamarin欢迎界面；

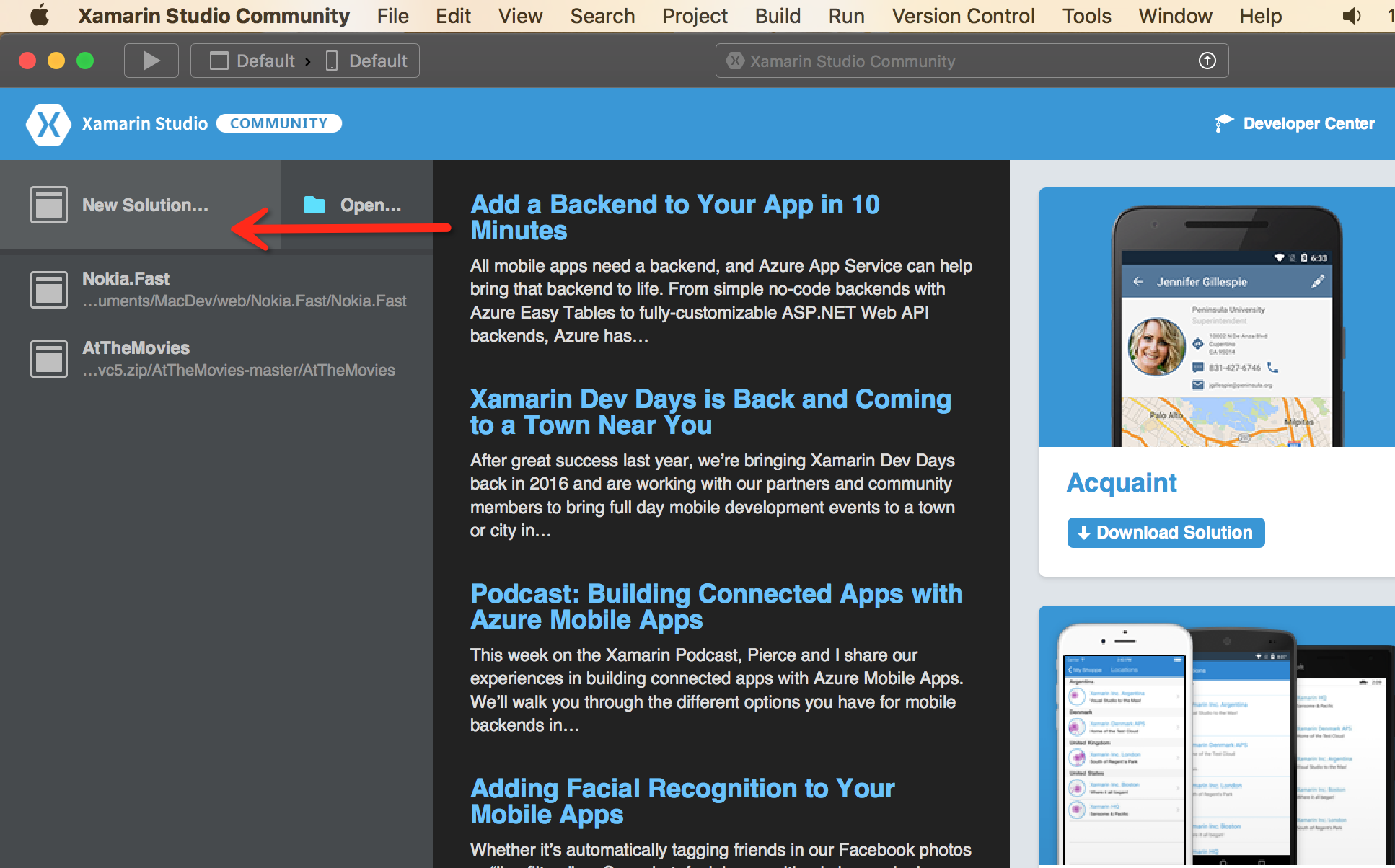


图2-3 Xamarin界面

（3）Xcode安装以及配置，从Mac OS上的APP Store搜索开发工具Xcode，进行一键下载安装流程，安装完成后可以通过Xcode欢迎页面进行工程新建等操作。进入主界面后可以同住Xcode-Preferences进行编译器相关环境的设置以及开发者账号的添加。图2-4位Xcode的欢迎页面，在该页面上可以创建新的工程。



图2-4 Xcode欢迎页面

如图2-5所示，在该界面上可以设置Xcode的很多偏好属性，如字号、缩进风格、字体等内容，还可以设置自己的开发者账号，方便之后的真机调试部分。

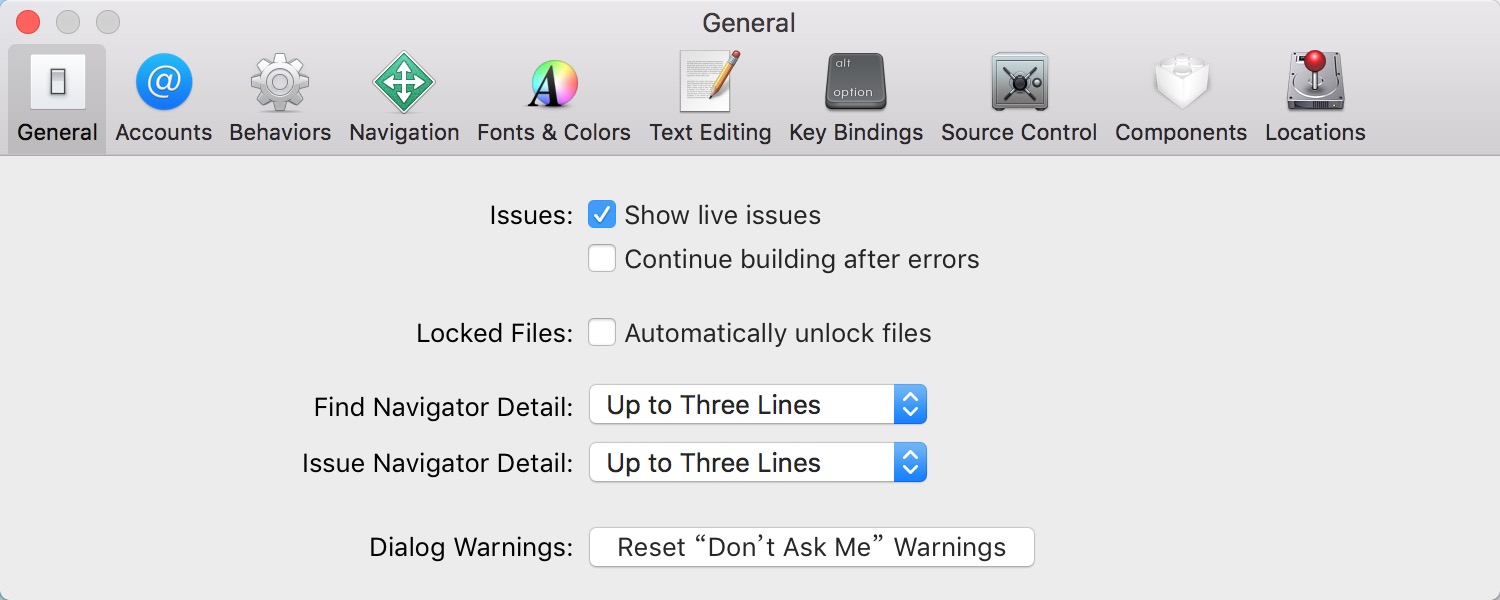


图2-5 Xcode设置页面

如图2-6所示，为Xcode主界面，在该页面上能设置许多项目部署时需要的属性，例如本题的项目部署：Display Name为EasyPut、Budnle Identifier为com.gokgod.easyput、Team为Ang Li、Deployment Target为6.0等；

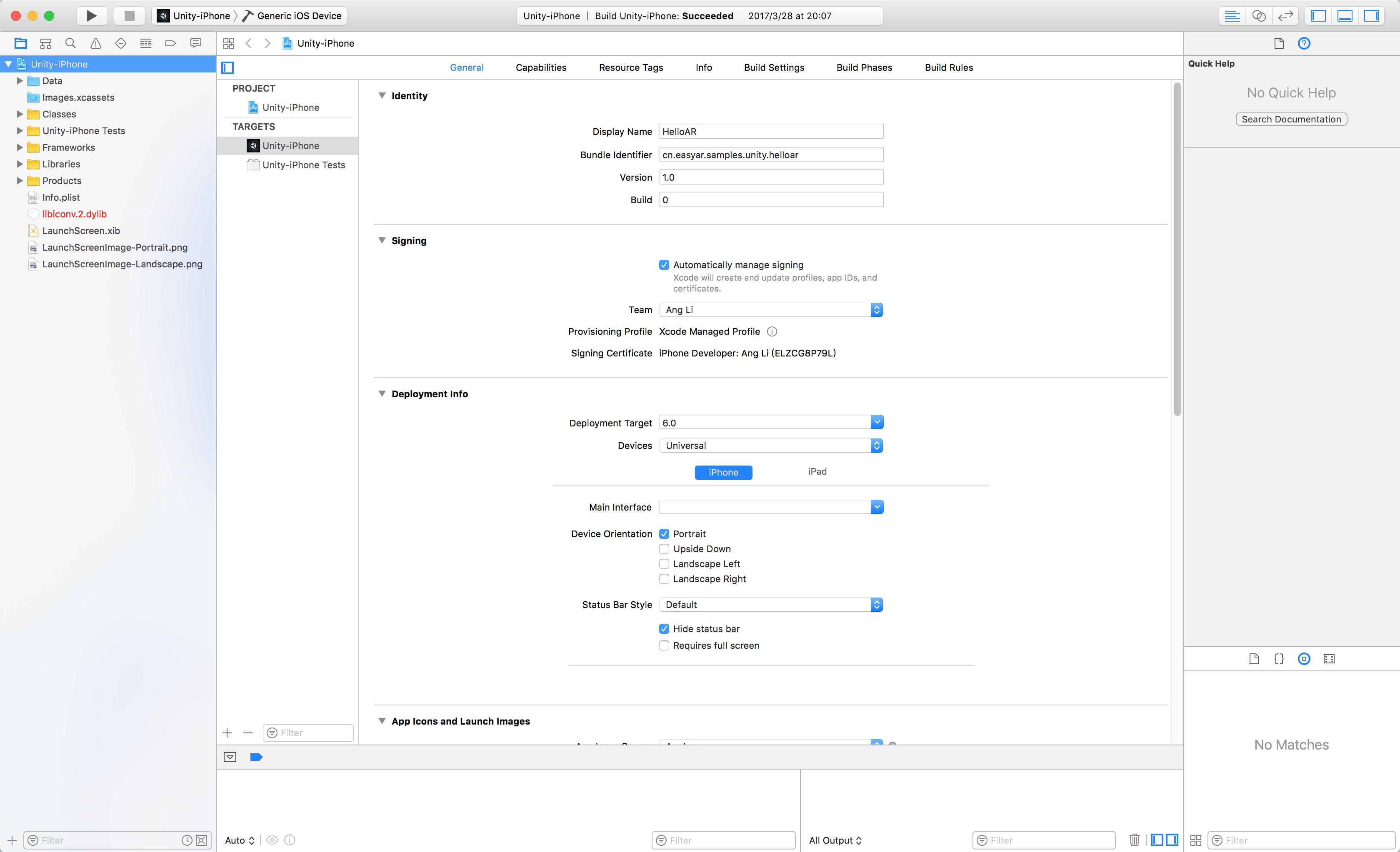


图2-6 Xcode主界面

（4）在Unity3D的欢迎页面上，选择创建3D类型工程，并命名为EasyPut，到此相关开发环境的配置和搭建全部结束。

## 2.3 关键技术的研究

本节主要对设计中涉及到了的EasyAR、Unity3D开发、iOS开发、3DMAX模型制作等技术进行研究和介绍。

### 2.3.1 EasyAR-SDK的使用与研究

EasyAR是好用免费的全平台AR引擎。EasyAR支持使用平面目标的AR，支持1000个以上本地目标的流畅加载和识别，支持基于硬解码的视频（包括透明视频和流媒体）的播放，支持二维码识别，支持多目标同时跟踪。

EasyAR支持PC和移动设备等多个平台，EasyAR不会显示水印，也没有识别次数限制。在拿到EasyAR package或EasyAR样例之后，需要一个key才能使用。

如图2-7所示，使用EasyAR之前需要使用邮箱在www.easyar.cn注册，为了初始化 EasyAR SDK，需要先在网页系统中生成key。创建应用，输入应用名称和Bundle ID（移动应用必填），点击确定后生成 Key。

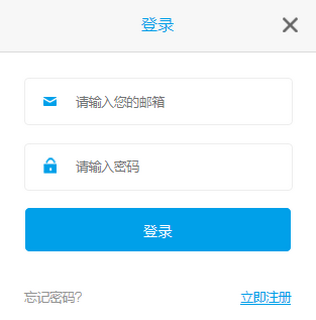


图2-7 EasyAR登录界面-创建应用Key界面

### 2.3.2 Unity3D和iOS的混合开发及3DMAX介绍

（1）本次采用Unity3D/Xcode混合开发。Unity3D是一款多平台的游戏开发引擎，其在模型渲染、图像处理等功能上的性能和效率都要高于iOS系统原生的游戏开发引擎，因为本应用中大量使用到图片的识别、模型的渲染、图像的调整等功能，所以必须要使用一款专业的游戏开发引擎来满足这些需求，所以就有了Unity3D/Xcode混合开发模式。其主要内容为Unity3D负责3D模型的渲染与处理，Xcode提供iOS系统支持和交互设计，二者有机结合、强强联手，一定能使本应用的最终结果如虎添翼。如图2-8所示，为易摆开发模式示意图，Unity3D主要负责EasyAR-SDK的接入与图形、模型等对象的处理；iOS主要负责提供合适的运行环境以及实景效果截图保存等功能的实现。

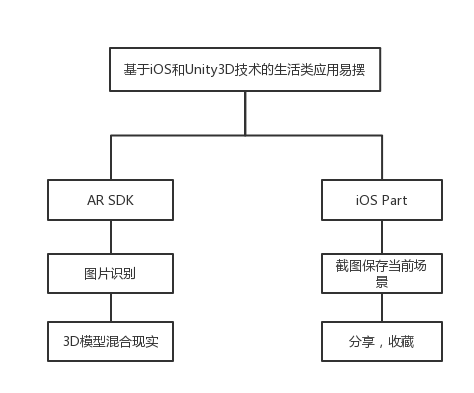


图2-8 Unity3D-Xcode混合开发

（2）3D Studio Max，常简称为3ds Max或MAX，是Discreet公司开发的（后被Autodesk公司合并）基于PC系统的三维动画渲染和制作软件[[[5]](#endnote-6)]。本次设计中主要使用其对家具模型进行1:1真实设计，包括模型颜色、大小、风格等都做到最接近于真实物件，并将模型文件格式保存为Unity3D支持的格式，即\*.3ds。

第3章 需求分析和概要设计

这章主要是对易摆AR家具预摆放APP-iOS客户端软件的需求分析和总体的程序设计进行介绍，同时对EasyAR SDK的各种接口、功能的使用方法进行分析，以此来做出合理的设计。

## 3.1 软件需求分析与开发任务

在现在越来越智能化的生活方式中，有些很麻烦的事情依然保持着自己原本的“模样”，但传统一般都很难满足现在人们对快速、便捷的追求，因此必须改变有些落后的“传统”，例如传统的家具选购方式。随着时代的发展，新技术的出现，人们希望自己身边的一切都能用一种便捷的方式来呈现，比如：“AR早教”、“AR博物馆”、“VR游戏体验馆”这类新产业，正是当代的真正需求。当然面对家具购买问题，很多人也提出了自己的看法，“家具一次很难买到合适的”、“家具预览只有预览图，根本无法看到真实效果”、“家具很大、很重，要是一次买到不合适的，就算能退货也是个很艰辛的过程”等都是人们的呼声，因此目前社会急需要这样一款手机应用来解决这个问题。

针对这个问题，本APP应该含有两大功能：一是家具预览图片的识别，二是家具3D模型的渲染与实景展示。

家具预览图片的识别，可以将家具图册上的图片识别进手机系统中，从而方便之后对相应的家具模型进行渲染。3D模型在APP识别预览图片之后，会按照1:1渲染在摄像头当前摄取的实景中，就像“真的”家具摆在实景中一样，用户可以得到一个立体、完整的家具摆放预览，满足用户在购买家具前希望准确衡量家具大小、样式、颜色的需求。

两大功能模块之间的联系如图3-1所示，用户可以通过进入APP，打开摄像头；然后将需要识别的家具预览图放到自己想摆放家具的位置，并用摄像头对准识别图片；最后在APP识别完预览图之后，会从3D模型库中提取相应的3D家具模型，并在预览图的位置渲染出3D家具模型，得到最终的实景展示效果。

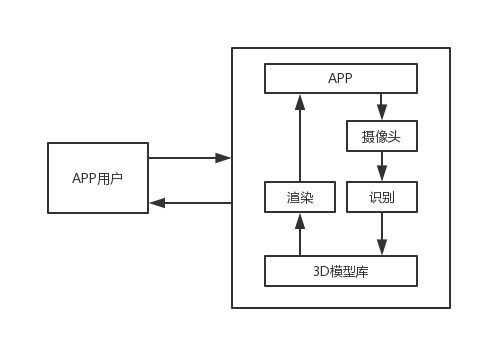


图3-1 App两大功能模块之间的联系

作为一个生活类AR-APP，一切的根源都是要解决存在于生活中的问题，因此各种交互、操作的设计应该简单亲民，APP的主要设计也应该是直接切入主题，避免不必要的设计。

根据上面的需求分析，对本APP的开发任务进行详细的规划：

（1）家具模型的设计。通过3DMAX来制作具体的家具3D模型，按照实际家具的模型，按照实际1:1大小进行3DMAX模型设计制作。在制作的时候要注意导出后的模型应该符合Unity3D的模型规范，以免在后期APP开发过程中出现返工现象，或者出现模型的错误导致APP成品效果不佳。模型应该和实际家具完全一样，包括颜色、形状、大小，从而可以更加准确的获得真实的预摆放效果；

（2）AR SDK的选择和研究。因为近几年AR技术获得了突飞猛进的发展，越来越多的图形学技术公司开始着手AR技术的研究和探索，因此市面上的AR SDK质量也越来越高了。通过在网上收集各种AR SDK的表现和用户反馈信息，最后Easy AR（www.easyar.cn）的AR SDK。对AR SDK的文档和使用规范说明进行研究，并对其合理使用和拓展，就能开始核心的APP设计开发了；

（3）APP的设计和实现。在选定了AR SDK后，就该着手APP部分的设计和编码了，首先通过Axure进行APP的原型设计，包括UI、交互等设计。之后把制作好的家具模型放进Unity 3D开发工具里，对模型进行成像和操作等方面的设计，包括模型的缩放，模型的旋转操作，模型脱离识别图片后持续存在等。在Unity 3D开发完全后，通过Unity 3D导出为iOS应用，之后便进入iOS开发部分。使用Xcode打开导出的项目文件，实现之前原型图设计的UI和UI交互，优化APP界面，使操作更加人性化和简便，突出APP的合理性设计，此部分完全使用Objective-C实现。

## 3.3 总体设计

本节主要介绍了项目的总体设计，包括系统运行流程、EasyAR-SDK的接入。

### 3.3.1 系统流程

易摆AR家具预摆放APP-iOS客户端目前包括如下功能：图片识别，模型渲染，实景摄取，实景预览，模型锁定，预览截图，其中应用的主要流程如图3-2所示。

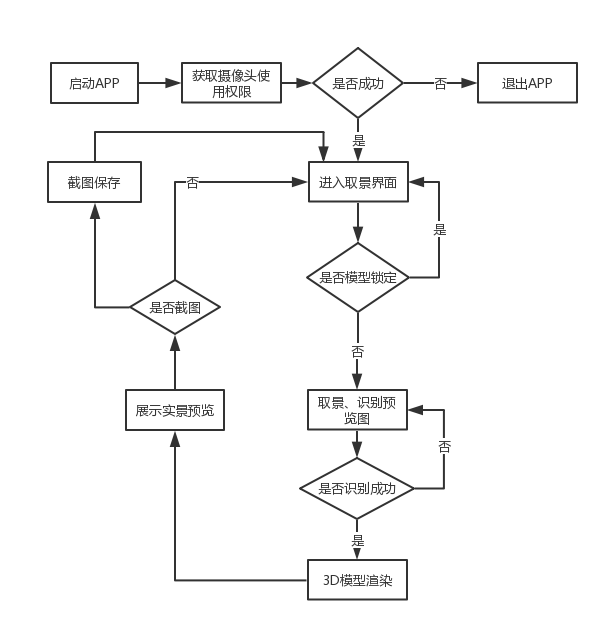


图3-2软件功能结构图

如图3-2所示，APP一启动就直接进入核心功能，在获取系统摄像头使用权限后，就进入取景界面，取景界面有用于交互的UI，接着就是进行预览图的识别和模型的渲染，如果检查到用户打开了识别锁定开关，APP将停止识别预览图，并将最后渲染出来的3D模型进行锁定，以便用户更好观察和实景的搭配效果。

### 3.3.2 EasyAR说明

EasyAR的组件在结构上是互相独立的。因此在EasyAR可以工作之前需要将这些组件结合起来。一个完整的EasyAR场景，需要有CameraDevice、ImageTracker、Augmenter三大组件。其中ImageTracker需要attach上CameraDevice，并使Augmenter周期性地获取当前的Frame；newFrame和attach操作的运行使整个AR系统开始工作，数据也在这个结构中流动[[[6]](#endnote-7)]。

从CameraDevice识别产生的Image会以每帧进入ImageTracker并推动tracker运行，这些Image将跟随Frame进入Augmenter。

Target从图像和json文件中构造出来，可以被加载进ImageTracker。当它被tracker检测或跟踪到，将会跟随Frame进入Augmenter。最终移动到AugmentedTarget。EasyAR的数据流图如图3-3，图3-3出自于《EasyAR文档：EasyAR数据与内存》[8]。

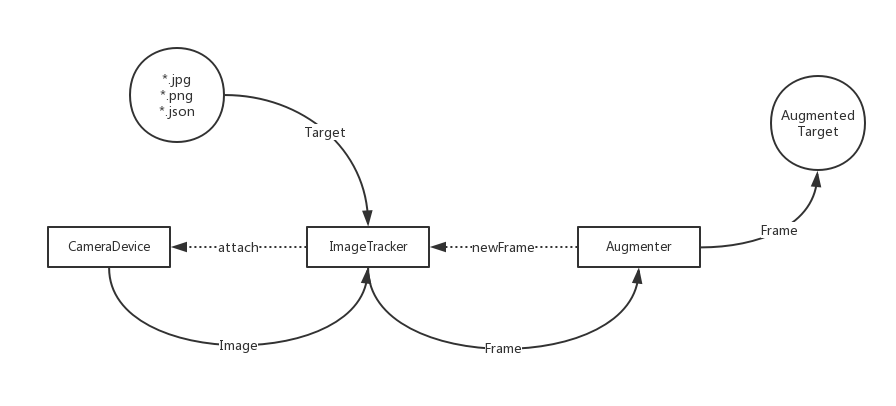


图3-3 EasyAR数据流图

EasyAR的C++对象与C++的std::shared\_ptr行为类似，它是一个对真实对象很简单的封装。对于一个继承自RefBase的类C，可以像std::shared\_ptr<C\_internal>一样使用。C 所引用的内存是引用计数的，并且以类似于std::shared\_ptr的方式管理。同时，C 的线程安全性与std::shared\_ptr相同。EasyAR的对象通常是延迟初始化的，也就是说在特定方法被调用之前，对象是无效的（operator bool()返回false）。如图3-4所示，为EasyAR的内存模型图，出自于《EasyAR文档：EasyAR数据与内存》[8]。

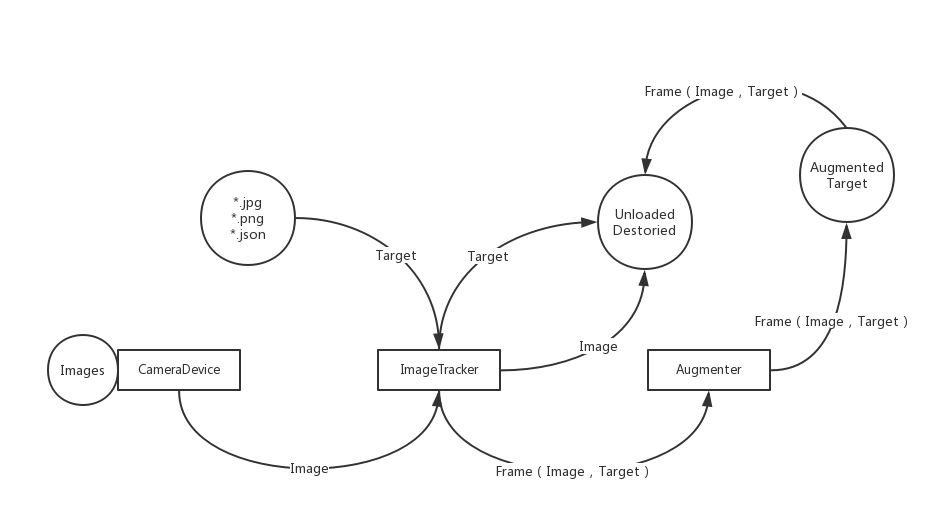


图3-4 EasyAR内存模型图

如果从camera获取的Image被newFrame所捕获，它将会在Frame被释放之前一直驻留在内存中。

ImageTracker会保持加载到自身的target的引用。这个引用可以通过从tracker卸载或销毁tracker来移除。Frame和AugmentedTarget也会保持被跟踪到的target的引用。加载之后的Target在这些引用被移除之前将不会被释放[[[7]](#endnote-8)]。

# 第4章 系统实现与测试

## 4.1 用户交互UI等相关的设计

易摆AR家具预摆放APP-iOS客户端是面向普通用户的手机应用，所以一定要有实用大气的功能和简介明了的UI界面，以此来便于用户使用以及吸引用户下载使用。iOS程序界面的设计包括图标切图和Logo的设计，程序界面的元素设计和Autolayout布局等。本客户端主要采用了Autolayout布局，这是iOS5之后推出的一种全新的布局模式，通过设置各种UI元素之间的约束进行布局，因此APP能够对各种分辨率的屏幕进行适配。为了保证APP的易用性，易摆没有很复杂的界面，主要是对识别图片与模型渲染等操作交互的设计。在确定APP需求和APP风格之后，将APP主界面设计为核心功能界面——摄像取景界面。



图4-1 易摆LOGO

## 4.2 核心AR功能的实现

### 4.2.1 识别图片的设计

易摆选择了三种类型的家具进行设计和测试，分别是灯、沙发、桌子，三种极具代表性的生活中的家具。为了正常的演示，设计了三张高区别度的识别图片，如图4-2所示。



图4-2 家具预览图

将这些预览图放进Unity3D中Assets文件夹中，作为识别图形库；下面为图片识别部分的关键代码设计：

这部分代码是图片识别的实现代码，使用了ImageTrackerBaseBehaviour，它主要有几大方法，这些方法从开始图片识别的StartTrack方法（代码：12行）到在识别图片位置建立3D模型的TargetLoad方法（代码：22行）再到对建立模型进行销毁的TargetUnload方法（代码：29）都有涉及，它是后面ImageTrackerBehaviour的父类实现，对子类的扩展进行了一定的标准化，是非常重要的一个类，图片识别功能的核心。

1. //建立ImageTrackerBaseBehaviour类型，用于继承实现ImageTracker
2. //识别图片后的操作
3. protected ImageTrackerBaseBehaviour ();
4. protected override void DetachAndStop ();
5. public void LoadImageTargetBehaviour (ImageTargetBaseBehaviour imageTargetBaseBehaviour);
6. protected virtual void OnDestroy ();
7. protected override void OnDeviceStart (DeviceAbstractBehaviour device, bool status);
8. protected virtual void Start ();
9. public bool StartTrack ();
10. public bool StopTrack ();
11. public void UnloadImageTargetBehaviour (ImageTargetBaseBehaviour imageTargetBaseBehaviour);
12. protected virtual void Update ();
13. public event Action<ImageTrackerBaseBehaviour, ImageTargetBaseBehaviour, Target, bool> TargetLoad {
14. [MethodImpl (MethodImplOptions.Synchronized)]
15. add;
16. [MethodImpl (MethodImplOptions.Synchronized)]
17. remove;
18. }
19. public event Action<ImageTrackerBaseBehaviour, ImageTargetBaseBehaviour, Target, bool> TargetUnload {
20. [MethodImpl (MethodImplOptions.Synchronized)]
21. add;
22. [MethodImpl (MethodImplOptions.Synchronized)]
23. remove;
24. }
25. }
26. }

测试：彩色打印三张识别图片，运行DEMO对这三张彩色图片分别进行识别测试，若均能正常识别则证明该部分设计无误，可以进行下个部分的设计。

### 4.2.2 3D模型的获取与展示

这是易摆的核心功能，首先通过3DMAX对家具模型进行建模修改，做到按照实际1:1大小进行3DMAX模型设计制作。在制作的时候要注意导出后的模型应该符合Unity3D的模型规范，导出文件格式应该为\*.3ds。对3D家具模型的设计一定要遵循尽量贴近现实，包括颜色、大小、风格和现实物品统一，如图4-3所示。

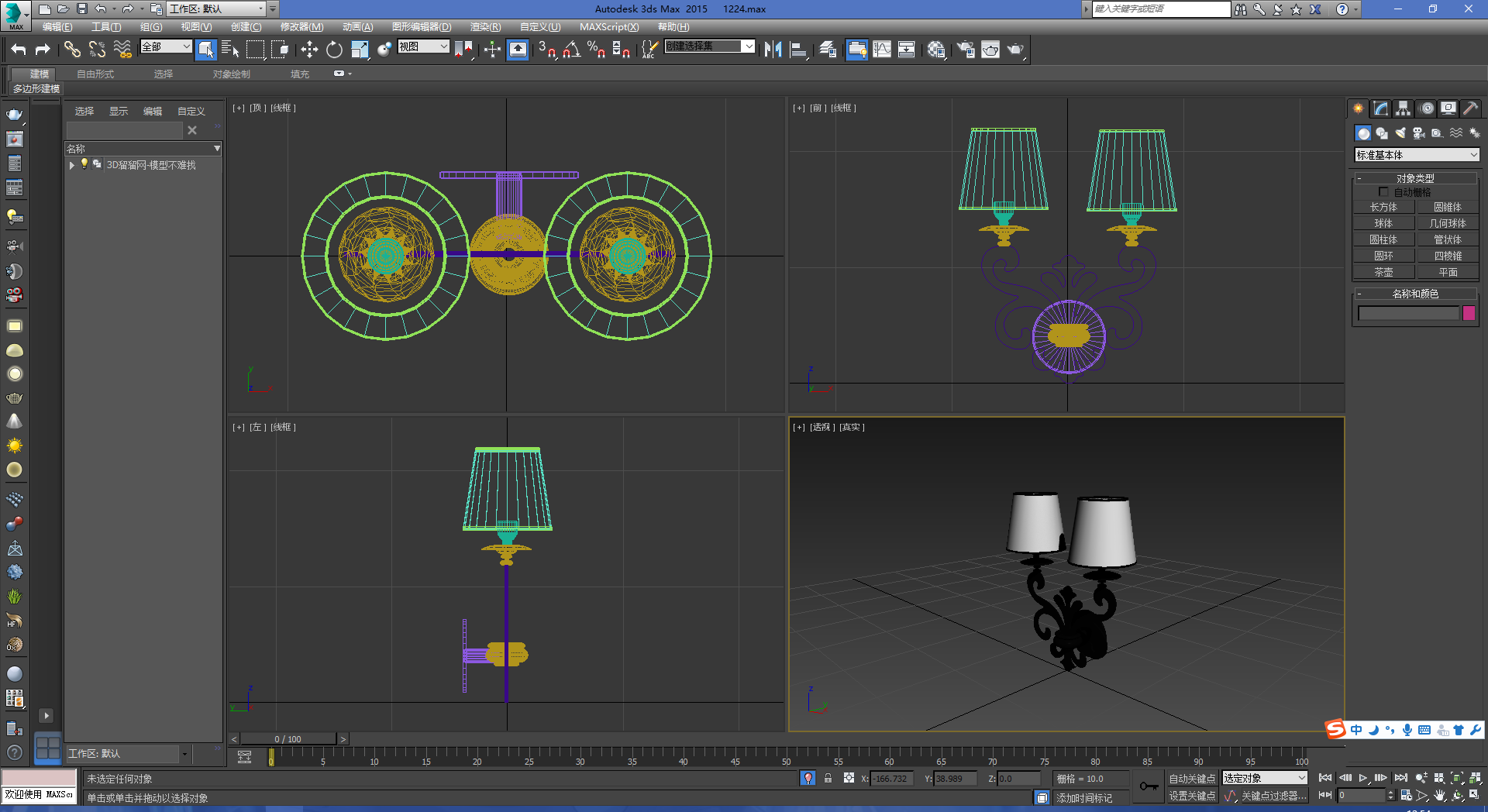


图4-3 3DMAX家具模型制作过程

将三种测试模型制作完成之后，就需要将模型文件导入Unity3D进行编辑、处理。主要是对识别图片与3D模型识别大小差的设计以及3D模型在系统识别图片成功后3D模型的展示功能的实现，如图4-4所示。



图4-4 Unity3D家具模型制作过程

在对3D模型导入设置完毕后，就该对EasyAR-SDK的模型渲染部分进行设置和代码的编写。

如图4-5 [[[8]](#endnote-9)]所示，EasyAR通过两个步骤来管理target：

第一步是加载配置来设置target。关于target的详细说明和加载/卸载可以参考Target和ImageTarget（C++）或是Target和ImageTarget（Unity）[[[9]](#endnote-10)]。

第二步是将target加载进入tracker中用以跟踪。EasyAR通过异步方式加载和卸载target到tracker，不会阻塞调用线程，因而可以动态增量加载[11]。

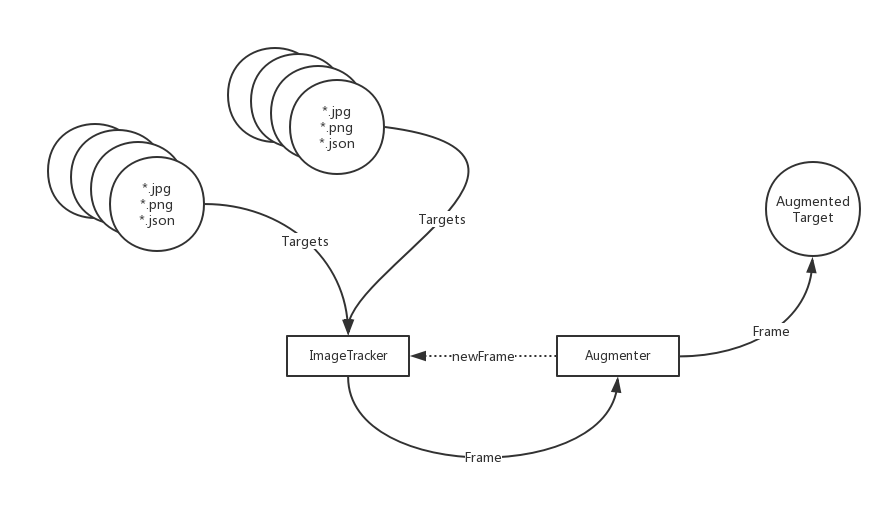


图4-5 EasyAR-Target管理图

加载进tracker的target会在Augmenter::newFrame调用中被传入Frame，并最终成为AugmentedTarget的一部分[[[10]](#endnote-11)]。

ARIsEasyBehaviour是AR服务的核心代码，其主要是用于提示目前APP正在进行哪一项任务，控制整个APP的执行顺序，例如：当识别到预览图片时，控制台Log出识别图片的信息，并开始在3D模型库中搜索相应的模型。

该部分核心代码如下：

//ARIsEasyBehaviour.cs

1. //来识别到图片时，打印获取目标的信息
2. void OnTargetFound(AugmenterBaseBehaviour augmenterBehaviour, ImageTargetBaseBehaviour targetBehaviour, Target target)
3. {
4. Debug.Log("<Global Handler> Found: " + target.Id);
5. }
6. //丢失识别图片时，打印丢失目标的信息
7. void OnTargetLost(AugmenterBaseBehaviour augmenterBehaviour, ImageTargetBaseBehaviour targetBehaviour, Target target)
8. {
9. Debug.Log("<Global Handler> Lost: " + target.Id);
10. }
11. //载入3D模型时，打印模型相关信息
12. void OnTargetLoad(ImageTrackerBaseBehaviour trackerBehaviour, ImageTargetBaseBehaviour targetBehaviour, Target target, bool status)
13. {
14. Debug.Log("<Global Handler> Load target (" + status + "): " + target.Id + " (" + target.Name + ") " + " -> " + trackerBehaviour);
15. }
16. //卸载3D模型时，打印模型相关信息
17. void OnTargetUnload(ImageTrackerBaseBehaviour trackerBehaviour, ImageTargetBaseBehaviour targetBehaviour, Target target, bool status)
18. {
19. Debug.Log("<Global Handler> Unload target (" + status + "): " + target.Id + " (" + target.Name + ") " + " -> " + trackerBehaviour);
20. }

代码：1行用来识别到图片时，打印获取目标的信息；

代码：7行用来丢失识别图片时，打印丢失目标的信息；

代码：13行用来在载入3D模型时，打印模型相关信息；

代码：19行用来卸载3D模型时，打印模型相关信息；

下面代码为识别对象EasyImageTargetBehaviour的实现代码，该对象为ImageTarget的行为对象，是ImageTarget对象与模型对象的连接。通过这段代码可以实现从识别到渲染模型的功能，实现模型的获取与展示功能[[[11]](#endnote-12)]。

//EasyImageTargetBehaviour.cs

1. //开始识别方法
2. protected override void Start()
3. {
4. base.Start();
5. HideObjects(transform);
6. }
7. //隐藏图片对象上存在的模型对象
8. void HideObjects(Transform trans)
9. {
10. for (int i = 0; i < trans.childCount; ++i)
11. HideObjects(trans.GetChild(i));
12. if (transform != trans)
13. gameObject.SetActive(false);
14. }
15. //显示图片对象上存在的模型对象
16. void ShowObjects(Transform trans)
17. {
18. for (int i = 0; i < trans.childCount; ++i)
19. ShowObjects(trans.GetChild(i));
20. if (transform != trans)
21. gameObject.SetActive(true);
22. }
23. //当该对象被识别时
24. void OnTargetFound(ImageTargetBaseBehaviour behaviour)
25. {
26. if (EasyImageTargetBehaviour.lockFlag == 0) {
27. ShowObjects (transform);
28. Debug.Log ("Found: " + Target.Id);
29. }
30. else{
31. Debug.Log ("lock!!!");
32. }
33. }
34. //当该对象丢失识别时
35. void OnTargetLost(ImageTargetBaseBehaviour behaviour)
36. {
37. if (EasyImageTargetBehaviour.lockFlag == 0) {
38. HideObjects (transform);
39. Debug.Log ("Lost: " + Target.Id);
40. }
41. else{
42. Debug.Log ("lock!!!");
43. }
44. }
45. //当该对象载入其对应的模型对象时
46. void OnTargetLoad(ImageTargetBaseBehaviour behaviour, ImageTrackerBaseBehaviour tracker, bool status)
47. {
48. Debug.Log("Load target (" + status + "): " + Target.Id + " (" + Target.Name + ") " + " -> " + tracker);
49. }
50. //当该对象卸载其对应的模型对象时
51. void OnTargetUnload(ImageTargetBaseBehaviour behaviour, ImageTrackerBaseBehaviour tracker, bool status)
52. {
53. Debug.Log("Unload target (" + status + "): " + Target.Id + " (" + Target.Name + ") " + " -> " + tracker);
54. }
55. }
56. }

第一歩，当ImageTarget被系统加载，对其相应的模型对象进行渲染，并将该ImageTarget添加至tracker对象中；

第二步，ImageTarget初始化完成后，进入识别事件，在该事件开始前先使用HideObjects方法对已经渲染过的模型进行隐藏，避免重复显示问题吗，如代码：1行Start()方法；

第三歩，当ImageTarget被识别后，执行ShowObjects方法，对属于它的模型对象进行显示操作，如代码：16行ShowObjects(Transform trans)方法；

第四歩，当ImageTarget未被识别后，执行HideObjects方法，对之前显示过的模型对象进行隐藏操作，避免重复显示问题，如代码：7行HideObjects(Transform trans)方法；

第五歩，当ImageTarget被系统销毁，对其之前渲染过的模型对象也都进行销毁，避免内存泄漏问题，并将其移出tracker对象。

测试：利用上一阶段的测试用例，对打印的相应的识别图片进行识别，将3D模型渲染接口连接在识别接口上，测试是否在识别成功后出现相应的3D模型，若出现且稳定，说明此部分设计完整可行，可以进行下一阶段的开发任务[[[12]](#endnote-13)]。

4.2.3 3D模型实景渲染

3D模型的获取与展示完成之后，下面的任务就是将已经渲染好的3D模型放入实景中让用户真正的预览到家具和实景的搭配效果，运行效果如图4-6。



图4-6 易摆实际使用效果图

锁定按钮，用于锁定当前模型的状态并停止Tracker对图片的不断识别，如上图右上角UI所示。其核心代码如下：

//LockButtonTouch.cs

1. using UnityEngine;
2. using System.Collections;
3. using EasyAR;
4. using UnityEngine.UI;
5. //LockButtonTouch类申明，该类用于锁定按钮的点击事件
6. public class LockButtonTouch : MonoBehaviour {
7. public Sprite m\_lock;
8. public Sprite m\_unlock;
9. private GameObject tracker;
10. private ImageTrackerBehaviour tracker\_behaviour;
11. // Use this for initialization
12. void Start () {
13. EasyImageTargetBehaviour.lockFlag = 0;
14. tracker = GameObject.Find ("ImageTracker");
15. tracker\_behaviour = tracker.GetComponent<ImageTrackerBehaviour>();
16. }
18. // Update is called once per frame
19. void Update () {
21. }
22. //LockButton被点击时更改Lock状态，从而调整模型的角度锁定等情况
23. public void Click ()
24. {
25. if (EasyImageTargetBehaviour.lockFlag == 0) {
26. EasyImageTargetBehaviour.lockFlag = 1;
27. this.GetComponent<Button> ().image.sprite = m\_lock;
28. tracker\_behaviour.StopTrack ();
29. }
30. else{
31. EasyImageTargetBehaviour.lockFlag = 0;
32. this.GetComponent<Button> ().image.sprite = m\_unlock;
33. tracker\_behaviour.StartTrack ();
34. }
35. Debug.Log ("Button Clicked");
36. }
37. }

该段代码为锁定模型按钮的实现代码，首先在初始化方法中EasyImageTar­getBehav­iour.lockFlag进行初始化，该变量用于判断当前系统状态。然后当点击锁定按钮时，对EasyImageTargetBehaviour.lockFlag进行修改，根据修改后的值对系统中tracker对象进行控制，从而锁定当前模型对象状态。让模型变为静态模型，方便用户细致观察与截图保存。

该部分功能测试：使用测试用识别图进行识别，并进行3D模型的渲染和展示，按下锁定按钮，测试模型状态是否锁定，若锁定则该部分功能正常可行。

# 结 论

易摆AR家具预摆放APP-iOS客户端就人们难以一次性购买正确合适家具这个问题进行了深入的思考，详细地分析了用户群体的需求，并结合EasyAR的学习和研究，使用AR技术实现了家具3D模型实时渲染与多角度实景观察这两大功能，为这个常见的生活难题提供了一种不错的解决方法[[[13]](#endnote-14)]。

易摆AR家具预摆放APP-iOS客户端是一款利用AR技术进行家具预摆放预览的生活类AR-APP。它的优势很明显，它能帮助用户节省非常多的曾经因为未买到合适家具后的生活开销。它能让用户通过最直观的方式，多角度的去预览一件家具在实际景物中的真实效果，这是非常实用的一种体验，也是最好能解决购买家具问题的可行方法。虽然刚刚确定这个选题时，曾经感到了不小的压力和畏惧，但是通过自己的努力研究；在各大门户网站诸如：简书、GitHub、StackOverflow等学习和收集资料，一步一步地实现一个本来觉得难以实现的功能，让自己在设计过程中收获很大[[[14]](#endnote-15)]。

易摆AR家具预摆放APP-iOS客户端也有遗憾之处，由于时间原因以及自己对3DMAX软件的不熟悉，最终定稿的易摆APP只能识别出三个测试家具模型，这是相当遗憾的一点，不能将易摆这个AR-APP的优势完全发挥出来。如果能够将整本家具预览图册的模型全部制作出来，这款APP的功能将更加强大。AR技术目前还在上升发展阶段，如果未来的AR技术性能更好、体验更加，那易摆APP也会随之变得越来越好。希望在今后的日子里有机会能将它做得更加完整，由一个毕业设计项目做成一款真正能发布的商业级项目，发挥它全部的价值。

# 致 谢

我是从大一开始学习iOS开发，大四开始学习Unity3D开发的，但是本次开发更多的是使用了Unity3D开发的技术，这使我在开发过程中遇到了不少的难点和坎坷。我非常感谢我的指导老师邓飞老师对我在选题以及毕设过程中的各种帮助，如果没有他合理又及时的帮助，我相信这款有重大实际意义的AR-APP我一定设计不出来。在今后的生活和工作中，这次设计中积累的经验和解决问题的思路一定会使我在之后的竞争道路上更加具有竞争力。

本次毕业设计让我真正的体会到了开发的乐趣，从新技术的略知一二到对新技术有所认识，从一个问题的无法解决到重新定义一种生活方式。这个从无到有的过程是辛苦的，但结果是幸福的、开心的。我以能为人们解决生活各种奇奇怪怪的小问题而感到自己这几年大学生涯没有白费，自己慢慢在社会中实现自己的价值。易摆AR家具预摆放APP-iOS客户端只是我的第一步，从学院迈入真实社会的第一步，今后的日子里我希望我能开发出更多诸如此类的，对我们的生活有真正实际价值的软件，这样的方式是我的理想，也是我最终实现自己全部价值的方式。非常感激能有这次难得毕业设计机会。

还要感谢我身边的很多同学，在Unity3D这方面给我提供的帮助以及建议。感谢EasyAR社区成员，对我对SDK提出的疑问的及时解答，使我没有耽误自己的开发计划。最后我要感谢我的室友，在我毕设期间不断的与我共同思考需求的合理实现以及改进功能实现。当然，最重要的是感谢APPLE公司，把这么好的系统呈现在了用户面前，也深深地把我吸引在了iOS开发这条道路上，我希望今后我能在这条路上走的更出色，成为这个行业真正的精英。

# 参考文献

1. [] iOS development documentation. 2017. [↑](#endnote-ref-2)
2. [] 莫志超. 基于IOS智能手机的移动学习终端的设计与开发,2012. [↑](#endnote-ref-3)
3. [] Unity3D development documentation. 2017. [↑](#endnote-ref-4)
4. [] 施坡. 基于Unity3D引擎的《保护村庄》小游戏的设计与开发,2014. [↑](#endnote-ref-5)
5. [] Unity3d快捷键大全、Unity3d如何学习, 《互联网文档资源（http://wenku.baidu.c）》,2016. [↑](#endnote-ref-6)
6. [] EasyAR文档-EasyAR数据与内存,2017. [↑](#endnote-ref-7)
7. [] 朱建基. 基于安卓的实物互动系统,2016. [↑](#endnote-ref-8)
8. [] EasyAR文档-EasyAR Target 配置,2017. [↑](#endnote-ref-9)
9. [] 刘立奇. 游戏宣传手册增强现实APP的设计与开发,2016. [↑](#endnote-ref-10)
10. [] 宋春雨. 基于移动终端的增强现实装配技术[D].广东工业大学,2014. [↑](#endnote-ref-11)
11. [] 王慧星. 基于iOS的增强现实技术的研究与应用[D].西安电子科技大学,2014. [↑](#endnote-ref-12)
12. [] 戴茜. 基于Unity3D的虚拟故园系统开发与实现[D].北京林业大学,2014. [↑](#endnote-ref-13)
13. [] 吴景. 基于Unity3D的虚拟实验系统的设计[D].广东工业大学,2015. [↑](#endnote-ref-14)
14. [] 王征. 比较IOS探讨安卓系统的优势[J]. 才智,2013,(20):243.

    [1]百度.Flutter[EB/OL].https://baike.baidu.com/item/Flutter/22498985?fr=aladdin,2018-4-17.

    [2]陈阿票.微信小程序、Facebook-RN、阿里-Weex 三者分析比较[EB/OL].https://www.jianshu.com/p/fb28b8e14bc5,2016-6-11.

    [3]董韬.Flutter 最新进展与未来展望[EB/OL].https://www.sohu.com/a/321869129\_670669,2019-6-20. [↑](#endnote-ref-15)