

校企合作双元开发新形态信息化融媒体教材  
数字媒体技术专业应用型人才培养实用教材

# 虚拟现实 与增强现实 开发实战

Reality  
Virtual

主编 宫海晓 郭慧 唐梅

西南交通大学出版社  
· 成 都 ·



# 前言

党的二十大报告指出，教育、科技、人才是全面建设社会主义现代化国家的基础性、战略性支撑。必须坚持科技是第一生产力、人才是第一资源、创新是第一动力，深入实施科教兴国战略、人才强国战略、创新驱动发展战略，开辟发展新领域新赛道，不断塑造发展新动能新优势。

随着计算机技术、电子信息技术、仿真技术的快速发展，虚拟现实（VR）、增强现实（AR）和混合现实（MR）的探索研究与应用进入更加广阔的领域，产业生态和业务形态越发多样，在医学模拟、军事航天训练、工业仿真、应急推演以及电子游戏等行业有广泛应用，带来了显著的社会效益，开启了人、计算机和环境之间自然且直观的3D交互。

Unity 3D是业界当前先进的VR/AR/MR 内容制作工具，由于其跨平台能力强、开发流程简便快捷受到开发者们的喜爱。从2005年诞生至今，Unity版本不断更新，功能越来越强大，已成为虚拟现实、增强现实、游戏开发等开发者的首选工具，其开发技术已经逐渐成为相关专业的学生以及从事混合现实开发研究的技术人员必须掌握的技术之一，也成为虚拟现实技术应用专业优选的教学内容。


## 本书具体内容

本书通过AR奇石、AR中草药、AR陶瓷、AR夜市、AR飞机5个AR案例和VR古镇、VR飞机、VR游乐园、VR太空、VR空间站5个VR案例，向读者全面阐述了AR/VR项目开发的全流程，并配备了详细的课程思政内容，非常适合具备Unity编程基础的读者学习。读者可以在短时间内学习书中介绍的所有方法并将其应用到自己的AR/VR项目中。

## 本书主要特点

（1）注重工程项目能力。

本书介绍了大量应用Unity 3D引擎开发的VR和AR实例，这些案例也是本书编写者参加完成的实际应用项目，读者可以直接借鉴使用。



(2) 立体化配套资源丰富。

本书配备了大量的操作截屏及案例视频讲解，针对性强。通过典型的实例分析，帮助读者较快地掌握AR/VR的基本知识、方法、技术应用。本书配套源素材、工程文件、PPT课件、视频教程，视频教程可扫描相关内容二维码观看，其余资源可扫描各章配套资源二维码下载。

(3) 强调理论结合实践。

本书以熟悉和掌握Unity 3D软件操作、VR/AR项目开发基本技能为目标，注重理论与实践相结合，把项目应用开发的技术过程贯穿在教学始终。突出实践的重要性，强调知识的扩展性，支持学习方法的多样性。

## 本书读者对象

本书主要面向广大从事虚拟现实应用开发设计的人员、从事虚拟现实应用设计教育的专职教师和计算机专业的学生等。

在本书的编写过程中，编者参阅和引用了大量专家和学者的书籍、文献以及网络资源，在此向所有资源的作者表示衷心的感谢。另外，蒙颖姗、林德永、岑知蔓、吴雅惠、黄艳芳、李燕青、卢杰、李培平、韦雨奇、莫欢婷、梁开亮、秦正旺等同学在本书的编写过程中也给予了大力协助和支持，在此向他们致以诚挚的谢意。

感谢成都西南交通大学出版社的大力支持，他们认真细致的工作保证了本书的出版质量。由于作者水平有限，书中难免存在疏漏和不足之处，恳请广大读者批评和指正。

编 者

2023年3月



## 本书数字资源

| 序号 | 名称              | 类型   | 页码  |
|----|-----------------|------|-----|
| 1  | 第1章配套资源         | 压缩文件 | 002 |
| 2  | 创建识别卡数据库        | 视频   | 003 |
| 3  | 创建摄像机           | 视频   | 004 |
| 4  | UI界面与交互         | 视频   | 006 |
| 5  | 动画              | 视频   | 013 |
| 6  | 特效              | 视频   | 017 |
| 7  | 音频播放            | 视频   | 018 |
| 8  | 生成与销毁           | 视频   | 020 |
| 9  | 旋转与缩放           | 视频   | 027 |
| 10 | 案例发布            | 视频   | 031 |
| 11 | 第2章配套资源         | 压缩文件 | 033 |
| 12 | 场景切换、退出         | 视频   | 034 |
| 13 | 摄像头及闪光灯切换       | 视频   | 036 |
| 14 | 截屏              | 视频   | 037 |
| 15 | 生成模型UI          | 视频   | 040 |
| 16 | 视频播放            | 视频   | 042 |
| 17 | 旋转与缩放           | 视频   | 048 |
| 18 | 特效              | 视频   | 053 |
| 19 | 案例发布            | 视频   | 056 |
| 20 | 第3章配套资源         | 压缩文件 | 057 |
| 21 | 模型与简介生成         | 视频   | 058 |
| 22 | 生成特效            | 视频   | 063 |
| 23 | 销毁              | 视频   | 063 |
| 24 | 旋转与缩放           | 视频   | 066 |
| 25 | 截屏              | 视频   | 070 |
| 26 | 对焦、摄像头及闪光灯切换    | 视频   | 073 |
| 27 | 场景切换和退出软件       | 视频   | 076 |
| 28 | 案例发布            | 视频   | 078 |
| 29 | 第4章配套资源         | 压缩文件 | 080 |
| 30 | UI界面交互          | 视频   | 080 |
| 31 | 翻页效果            | 视频   | 085 |
| 32 | 播放视频与虚拟按钮控制视频播放 | 视频   | 089 |
| 33 | 纸片风场景生成         | 视频   | 092 |

续 表

| 序号 | 名称              | 类型   | 页码  |
|----|-----------------|------|-----|
| 34 | 生成烟火气特效         | 视频   | 093 |
| 35 | 漫游场景            | 视频   | 095 |
| 36 | 模型旋转与缩放         | 视频   | 101 |
| 37 | 模型长按消失          | 视频   | 104 |
| 38 | 模型的切换与交互        | 视频   | 105 |
| 39 | 案例发布            | 视频   | 111 |
| 40 | 第5章配套资源         | 压缩文件 | 112 |
| 41 | 单指旋转和双指缩放       | 视频   | 113 |
| 42 | 虚拟按钮            | 视频   | 115 |
| 43 | 模型出场动画和拆分动画     | 视频   | 119 |
| 44 | 飞机模型尾气特效和出场特效   | 视频   | 127 |
| 45 | AR手册            | 视频   | 127 |
| 46 | UI交互            | 视频   | 129 |
| 47 | AR手册第三、四页“虚拟立牌” | 视频   | 131 |
| 48 | 案例发布            | 视频   | 135 |
| 49 | 第6章配套资源         | 压缩文件 | 138 |
| 50 | 环境配置            | 视频   | 139 |
| 51 | 场景搭建            | 视频   | 141 |
| 52 | UI交互            | 视频   | 150 |
| 53 | 场景跳转            | 视频   | 153 |
| 54 | 移动              | 视频   | 155 |
| 55 | 背景音效            | 视频   | 157 |
| 56 | 案例发布            | 视频   | 158 |
| 57 | 第7章配套资源         | 压缩文件 | 160 |
| 58 | 场景搭建            | 视频   | 160 |
| 59 | 使用VRTK实现交互      | 视频   | 168 |
| 60 | 游乐项目的切换效果       | 视频   | 174 |
| 61 | 实现三个按钮的点击事件     | 视频   | 175 |
| 62 | 场景加载界面与场景加载     | 视频   | 177 |
| 63 | 场景返回            | 视频   | 178 |
| 64 | 案例发布            | 视频   | 179 |
| 65 | 第8章配套资源         | 压缩文件 | 181 |
| 66 | 场景搭建            | 视频   | 182 |
| 67 | UI制作            | 视频   | 184 |

续 表

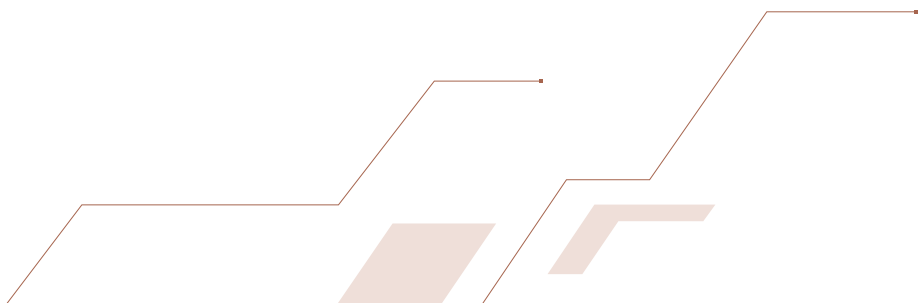
| 序号 | 名称             | 类型   | 页码  |
|----|----------------|------|-----|
| 68 | 发射射线           | 视频   | 186 |
| 69 | 跳转场景           | 视频   | 189 |
| 70 | 射线点击指定按钮弹出相应UI | 视频   | 191 |
| 71 | 飞机飞行           | 视频   | 196 |
| 72 | 案例发布           | 视频   | 199 |
| 73 | 第9章配套资源        | 压缩文件 | 202 |
| 74 | 场景搭建           | 视频   | 203 |
| 75 | UI制作           | 视频   | 207 |
| 76 | 行星自转动画和公转动画    | 视频   | 209 |
| 77 | 太阳爆炸特效         | 视频   | 210 |
| 78 | 实现音频播放和视频播放    | 视频   | 213 |
| 79 | 场景的跳转、返回与退出    | 视频   | 215 |
| 80 | 火箭发射场景返回UI     | 视频   | 218 |
| 81 | 环形选择跳转场景       | 视频   | 218 |
| 82 | 场景异步加载进度的显示    | 视频   | 222 |
| 83 | UI和视频的显示与隐藏    | 视频   | 224 |
| 84 | VRTK射线漫游和瞬移    | 视频   | 227 |
| 85 | 案例发布           | 视频   | 230 |
| 86 | 第10章配套资源       | 压缩文件 | 232 |
| 87 | 场景搭建           | 视频   | 233 |
| 88 | 发光效果           | 视频   | 235 |
| 89 | UI弹出           | 视频   | 239 |
| 90 | 月球车休眠          | 视频   | 240 |
| 91 | 月球车分解          | 视频   | 244 |
| 92 | 音频功能           | 视频   | 247 |
| 93 | 场景漫游           | 视频   | 249 |
| 94 | 案例发布           | 视频   | 251 |



# 目录

## 第一部分 AR 实践案例

|     |            |     |
|-----|------------|-----|
| 第1章 | AR奇石       |     |
| 1.1 | 案例简介 ..... | 002 |
| 1.2 | 案例实现 ..... | 002 |
| 第2章 | AR中草药      |     |
| 2.1 | 案例简介 ..... | 033 |
| 2.2 | 案例实现 ..... | 033 |
| 第3章 | AR陶瓷       |     |
| 3.1 | 案例简介 ..... | 057 |
| 3.2 | 案例实现 ..... | 057 |
| 第4章 | AR西江夜市     |     |
| 4.1 | 案例简介 ..... | 080 |
| 4.2 | 案例实现 ..... | 080 |
| 第5章 | AR飞机       |     |
| 5.1 | 案例简介 ..... | 112 |
| 5.2 | 案例实现 ..... | 112 |



## 第二部分 VR 实践案例

|                     |     |
|---------------------|-----|
| 第6章 VR古镇            |     |
| 6.1 案例简介 .....      | 138 |
| 6.2 案例实现 .....      | 138 |
| 第7章 VR游乐园           |     |
| 7.1 案例简介 .....      | 160 |
| 7.2 案例实现 .....      | 160 |
| 第8章 VR飞机飞行          |     |
| 8.1 案例简介 .....      | 181 |
| 8.2 案例实现 .....      | 181 |
| 第9章 VR太空            |     |
| 9.1 案例简介 .....      | 202 |
| 9.2 案例实现 .....      | 202 |
| 第10章 VR空间站          |     |
| 10.1 案例简介 .....     | 232 |
| 10.2 案例实现 .....     | 232 |
| <b>■ 参考文献</b> ..... | 253 |

第一部分

# AR 实践案例



# 第1章 AR奇石

## 1.1 案例简介

本案例利用Unity增强现实技术实现以“奇石”为主题的图像识别。通过学习制作本案例，使开发者了解我国的奇石文化，增加对传统文化的认知，丰富民族文化的思想和内涵。

用户可以通过手机摄像机识别奇石图像，获取奇石模型，也可以通过手指触屏来实现奇石的缩放、旋转等功能。

本案例开发用到的所有素材，均可从本章配套资源下载，如图1-1所示。



第1章配套资源

|                                 |                 |                    |           |
|---------------------------------|-----------------|--------------------|-----------|
| book.unitypackage               | 2022/7/27 15:36 | Unity package file | 13,286 KB |
| Falling.unitypackage            | 2022/7/25 16:29 | Unity package file | 17 KB     |
| Images.unitypackage             | 2022/11/5 20:51 | Unity package file | 858 KB    |
| stones.unitypackage             | 2022/7/27 16:05 | Unity package file | 96,666 KB |
| The Fading Stories-不再年轻的村庄.m... | 2022/7/23 20:15 | MP3 格式声音           | 2,034 KB  |

图1-1

## 1.2 案例实现

### 1.2.1 素材准备

#### 1. 模型素材

本案例制作需要用到奇石模型，点击资源包“stones.unitypackage”导入即可添加到项目中。

#### 2. 音频素材

本案例制作需要用到的背景音乐素材，点击音频“The Fading Stories-不再年轻的村庄.mp3”导入即可添加到项目中。

#### 3. 动画特效素材

本案例制作需要用到书翻页动画与飘叶特效素材，点击资源包“book.unitypackage”和“Falling.unitypackage”导入即可添加到项目中。



#### 4. 图像素材

本案例制作需要用到的奇石图像素材可从本章配套资源下载，点击资源包“Images.unitypackage”导入即可添加到项目中。

#### 1.2.2 环境配置

##### 1. 创建识别卡数据库

进入vuforia官网，点击“Develop”，创建“TargetManager”用于处理要识别的图片。点击“Add Database”增加数据库，然后输入数据库名称，最后点击“Create”，如图1-2所示。



创建识别卡数据库

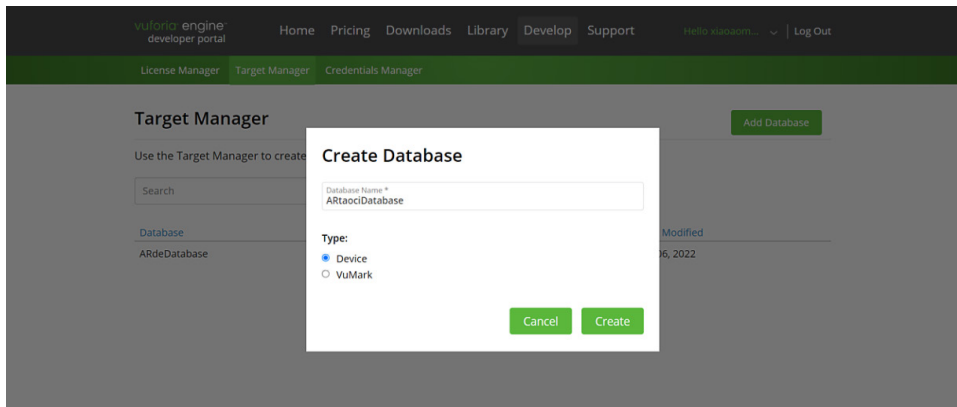


图1-2

添加扫描所需的图片。点击“Add Target”增加图片，Type选择“Single Image”，然后点击“Browse”按钮浏览所需图片，图片宽度“Width”设置为“1”，完成设置，点击“Add”即可完成识别卡的上传。重复以上操作直到完成剩余所有图片的上传，如图1-3所示。

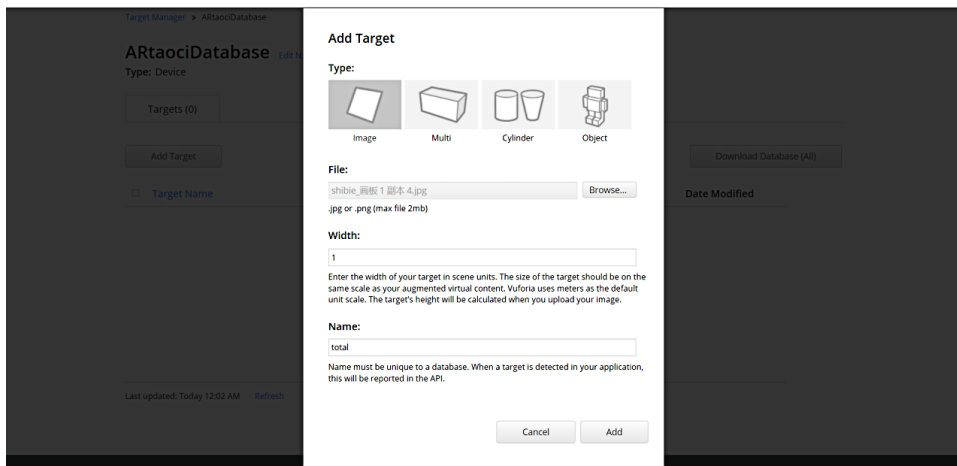


图1-3

下载数据库。选择我们导入的所有图片，点击“Download Database”，选择“Unity Editor”，然后点击“Download”即可完成数据库的下载，如图1-4所示。

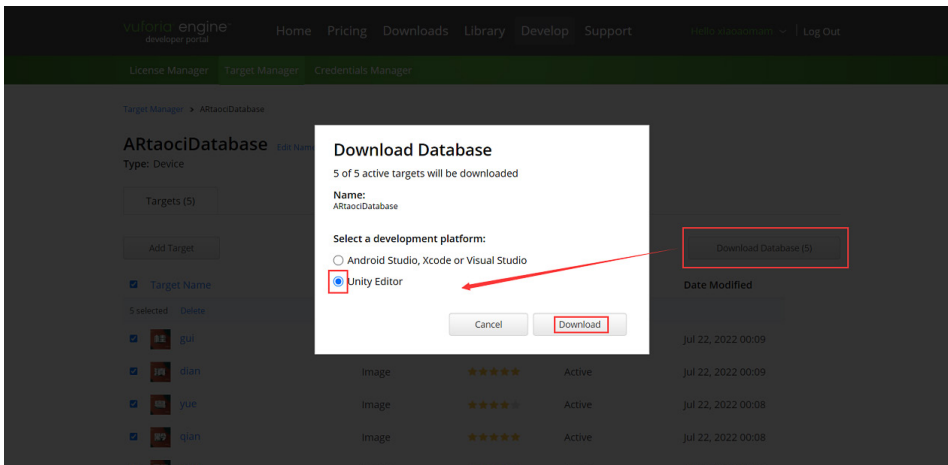


图1-4

将下载好的数据库导入Unity工程文件中，如图1-5所示。

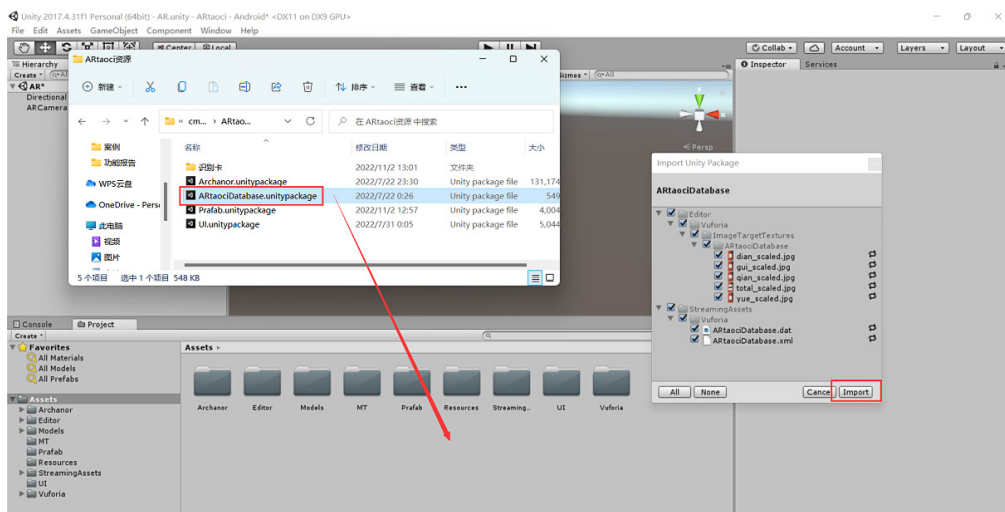


图1-5

## 2. 创建摄像机

首先需要新建一个“AR Camera”，用于调用AR设备的摄像机，负责真实世界的显示。在“Hierarchy”面板中，把原来的“Main Camera”删除，在“GameObject” — “Vuforia”中选择“AR Camera”，创建新的摄像机，如图1-6所示。



创建摄像机

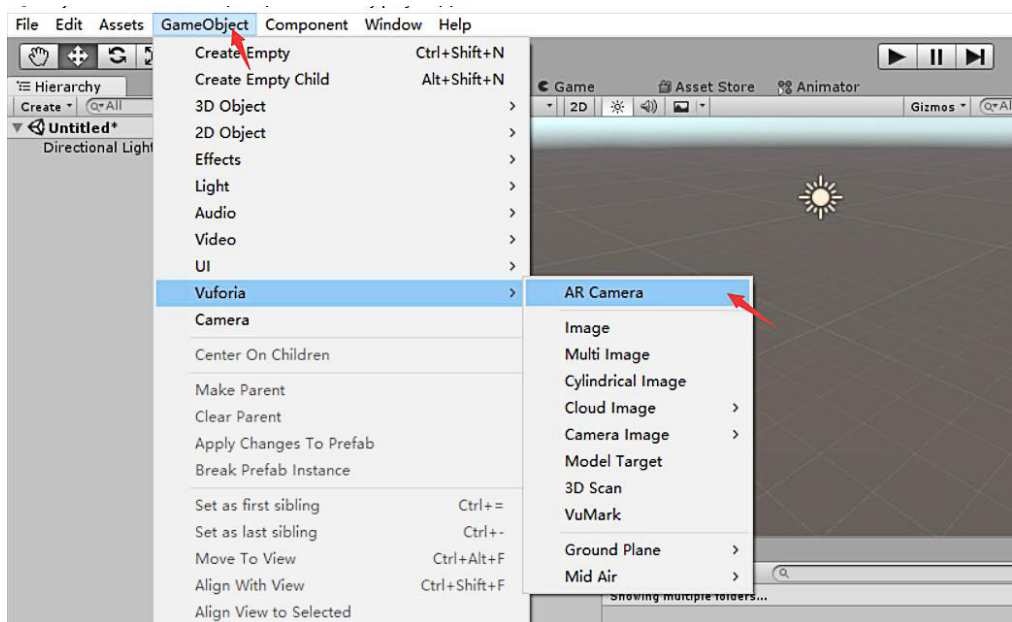


图1-6

点击Unity主目录中的“File”——“Build Settings”，本案例的运行平台是安卓端，需在“Platform”面板中选择“Android”选项，再点击“Switch Platform”转换，如图1-7所示。

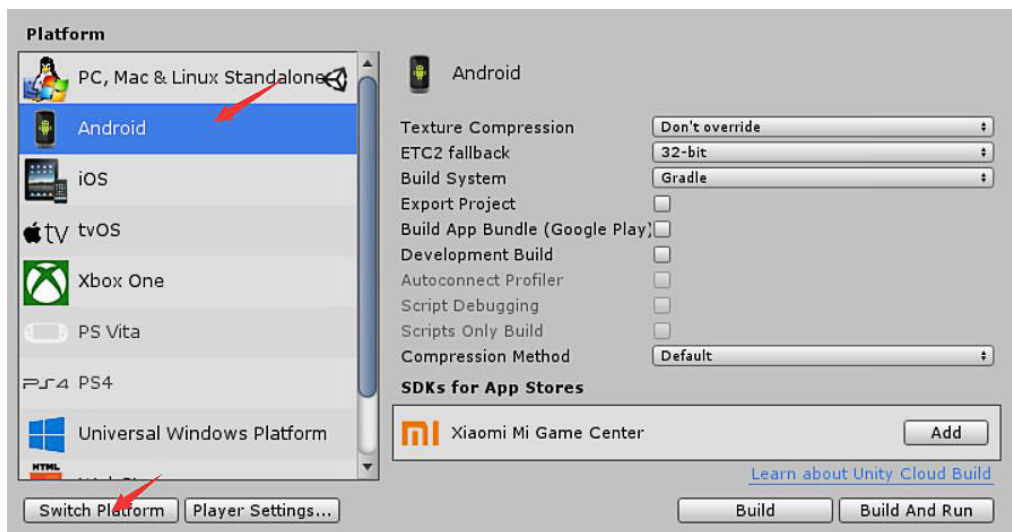


图1-7

接着在“Platform”面板中，点击“Player Settings”，然后在出现的“PlayerSettings”面板中，选择“XR Settings”配置，勾选“Vuforia Augmented Reality Supported”选项，打开增强现实支持，如图1-8所示。

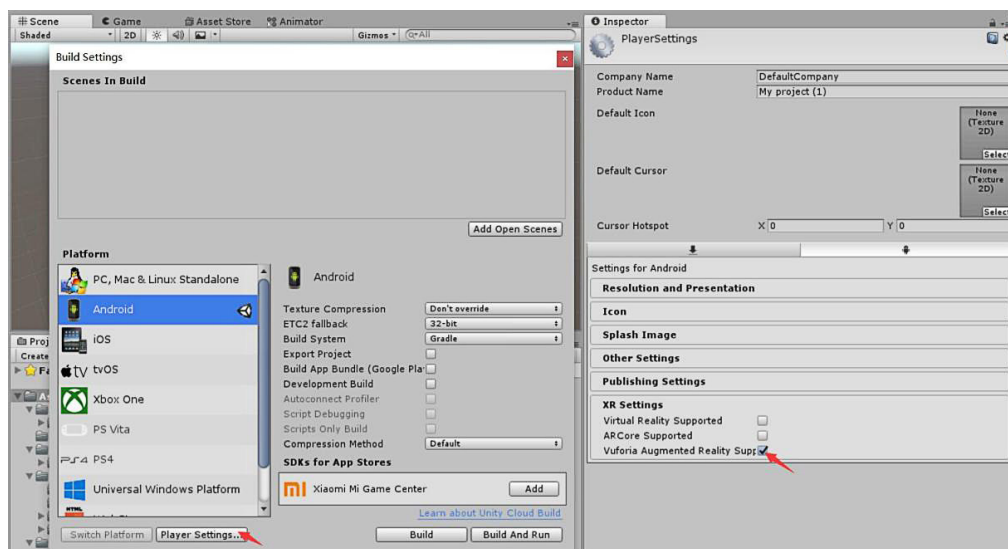


图 1-8

在“Hierarchy”面板中，选择“AR Camera”对象，在其“Inspector”属性面板中点击“Open Vuforia Configuration”选项，进行Vuforia的设置：在“App License Key”中，将已申请好的App license复制到输入框中；在“Datasets”设置中，勾选之前导入的识别图片对象，如图1-9所示。

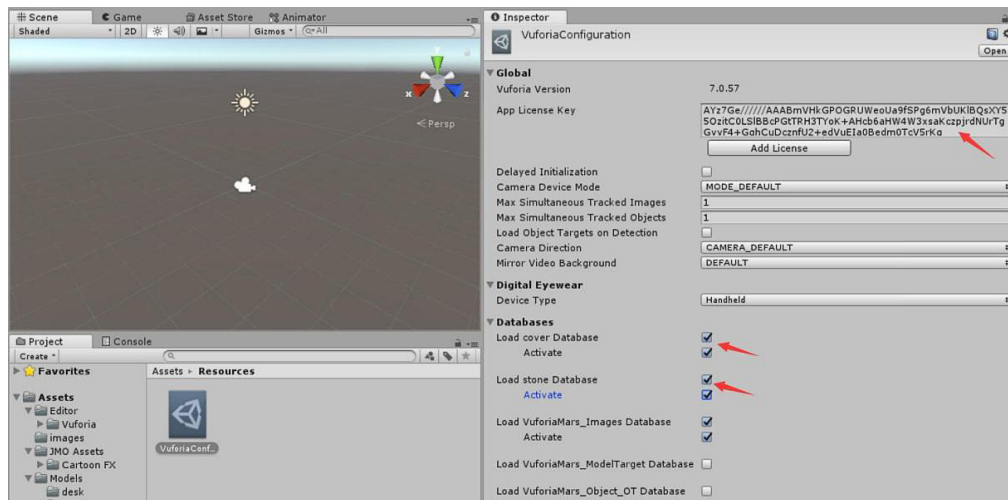


图 1-9

### 1.2.3 功能实现

#### 1. UI界面与交互

首先创建一个UI控件，在菜单栏“GameObject>UI”中选择“Button”，创建后在“Hierarchy”面板中将创建出来的Canvas重命名为“huanglashiUI”，层级下的Button重命名为



UI 界面与交互

“jianjie”，Button层级下的“Text”删除，完成后如图1-10所示。

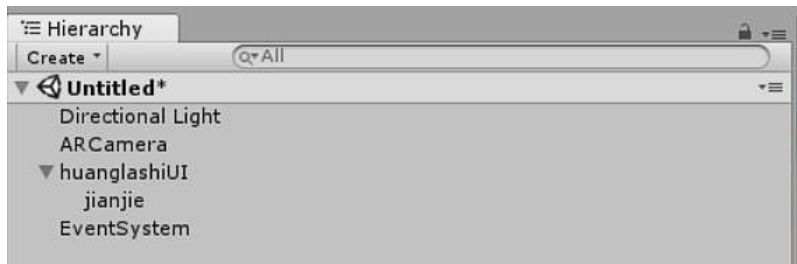


图1-10

在“Hierarchy”面板中选择“jianjie”对象，在其“Inspector”属性面板中设置位置和宽高，具体数值如图1-11所示。

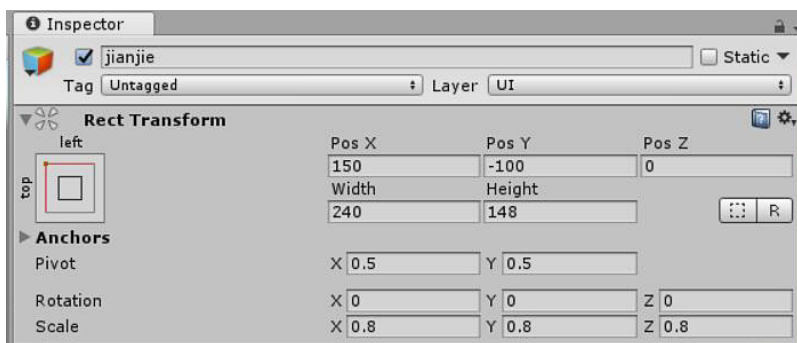


图1-11

在“Project”面板找到“1111”文件，将其拖曳至“Source Image”处，如图1-12所示。由于UI的素材图片为半透明，所以第一次使用需要在图片的“Inspector”属性面板中修改Texture Type为“Sprite(2D and UI)”。

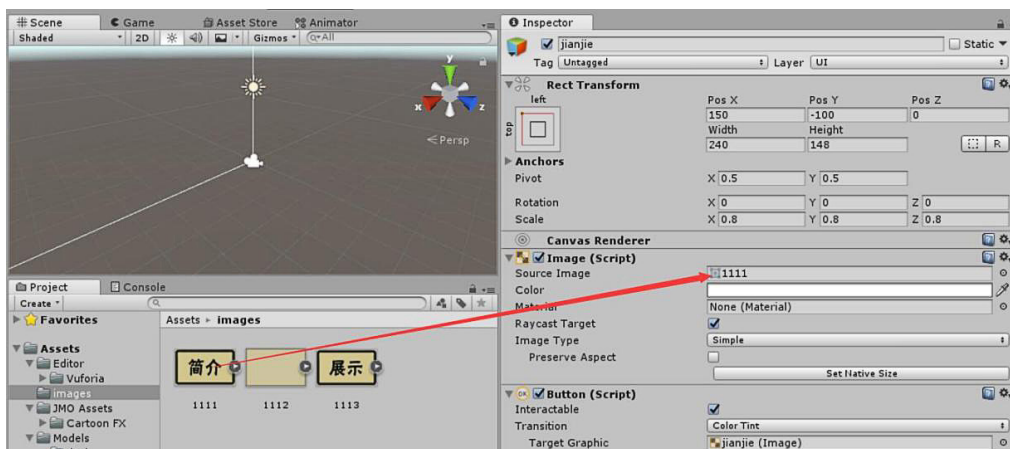


图1-12

在“Hierarchy”面板中选择“huanglashiUI”对象，鼠标右键，选择“UI”——“Image”，创建一个Image，将其重命名为“xiangqing”，如图1-13所示。

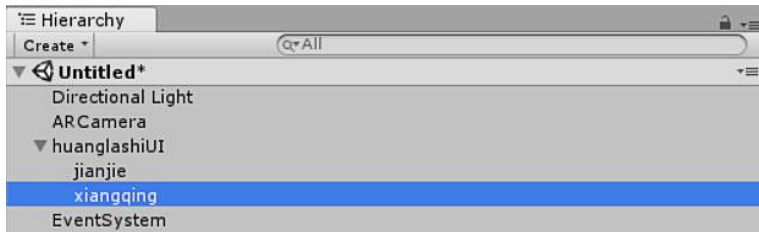


图1-13

在“Hierarchy”面板中选择“xiangqing”对象，修改其属性中的宽高，设置为650×400，在“Project”面板找到“1112”文件，将其拖曳至“Source Image”处，如图1-14所示。

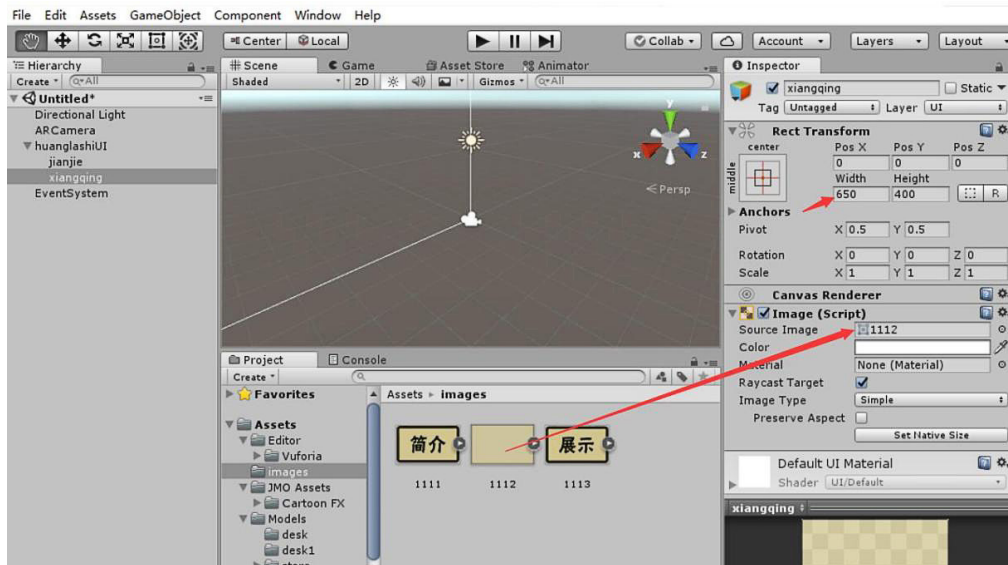


图1-14

创建文本对象。在“Hierarchy”面板中，选择“xiangqing”对象，鼠标右键，选择“UI”，选择“Text”，创建一个文本对象，如图1-15所示。

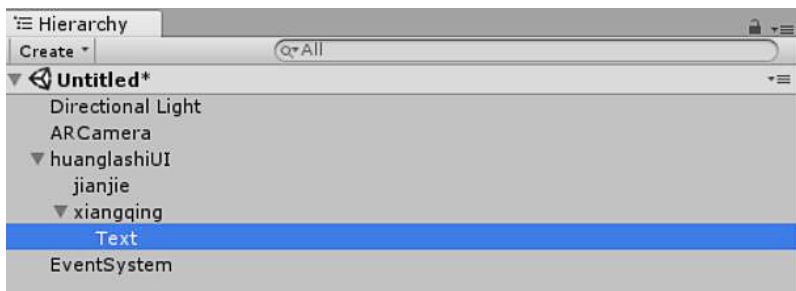


图1-15



在“Hierarchy”面板中选择“Text”对象，修改其位置和宽高，将文本内容（黄蜡石是观赏石，具备了“漏、透、瘦、皱”传统赏石要素。特别是黄蜡石中的晶蜡，石表凹凸不平，纹路纵横交错，有如“筋骨裸露”，观赏价值高，深受赏石爱好者欢迎。黄蜡石是水冲石，水洗度高，表皮光滑，蜡质感强，多数有一层温润的包浆，特别是致密度高的籽料手感好，用手抚摸犹如婴儿的皮肤，所以黄蜡石也有“玩皮”一说。纹理是观赏石的鉴评要素之一。）输入至“Text”处，字体大小设置为“25”，行间距“1.5”，字体颜色选择“RGB（16，12，0）”，完成后如图1-16所示。

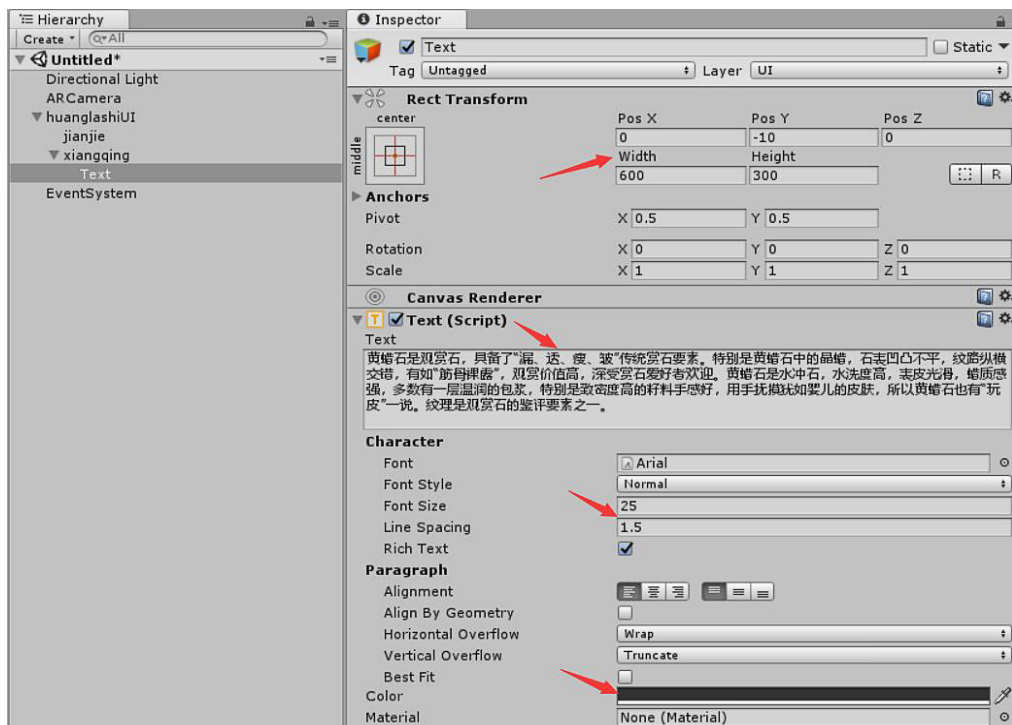


图1-16

制作另外三个奇石的简介UI。在“Hierarchy”面板中，选择“huanglashiUI”，复制三份，分别重命名为“laibinshiUI”“dahuashiUI”“caitaoshiUI”，如图1-17所示。

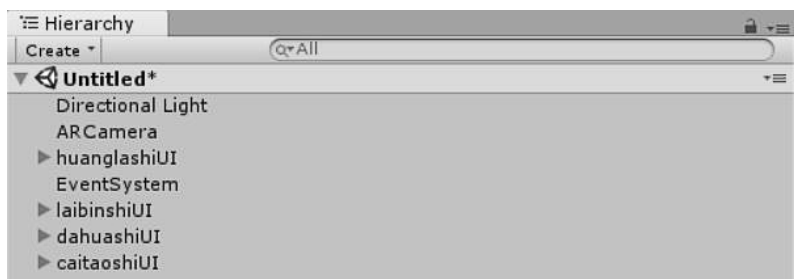


图1-17

在“Hierarchy”面板中，选择“laibinshiUI”对象，修改其文本内容。选择“Text”，将其文本内容修改为“来宾黑石主要产于广西红水河河段。红水河从西向东横贯兴宾区境，共流经11个乡镇（占兴宾区总乡镇的二分之一）长达162公里。沿河是溶岩地带，河床中沉积着大量各种岩石砾块。红水河来宾河段，河床狭窄，弯多滩险，水落差大，流急砂多，对河床中的岩石砾块不断地搬运、翻滚、撞击、磨蚀，从而造就了各种各样的奇石。”完成后如图1-18所示。

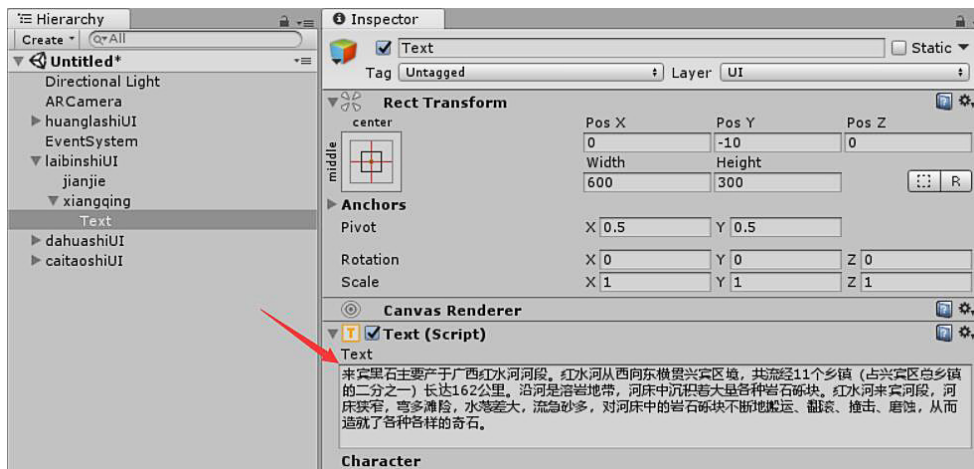


图1-18

在“Hierarchy”面板中，选择“dahuashiUI”对象，修改其文本内容。选择“Text”，将其文本内容修改为“大化石产于广西大化县境内的岩滩水电站附近河段，从开发出来的大化石看，无论大到一二十吨的巨石，或小到二三十克的小石子，无不都具有石质坚硬，硅化或玉化程度高；石形奇特，千姿百态；花纹图案变化无穷；色彩艳丽和谐悦目等特色。由此可见大化石的风雅、气质、神韵都达到了非凡的境地，它一露面便轰动广西，誉满中华，影响遍及全球。”完成后如图1-19所示。

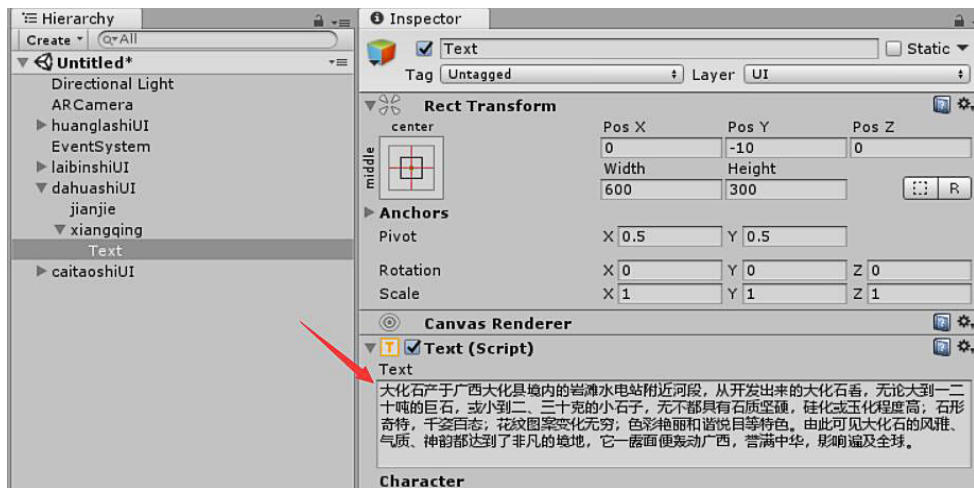


图1-19



在“Hierarchy”面板中，选择“caitaoshiUI”对象，修改其文本内容。选择“Text”，将其文本内容修改为“彩陶石产于红水河的下游，此石的产地很狭窄。当红水河流经此地时，由于受到长达几公里的暗礁阻击，长年累月暗礁右侧便冲出一条很深的河道，暗礁的左侧，则形成一条三百多米长的回水湾，红河石就卧躺在这一条回水湾中。这是由于地壳的变化，河滩上青色的岩层被挤压出条条裂纹。雨季到来，大水淹没暗礁，把河滩上这些带有裂纹的青石头冲进了水湾。”完成后如图1-20所示。

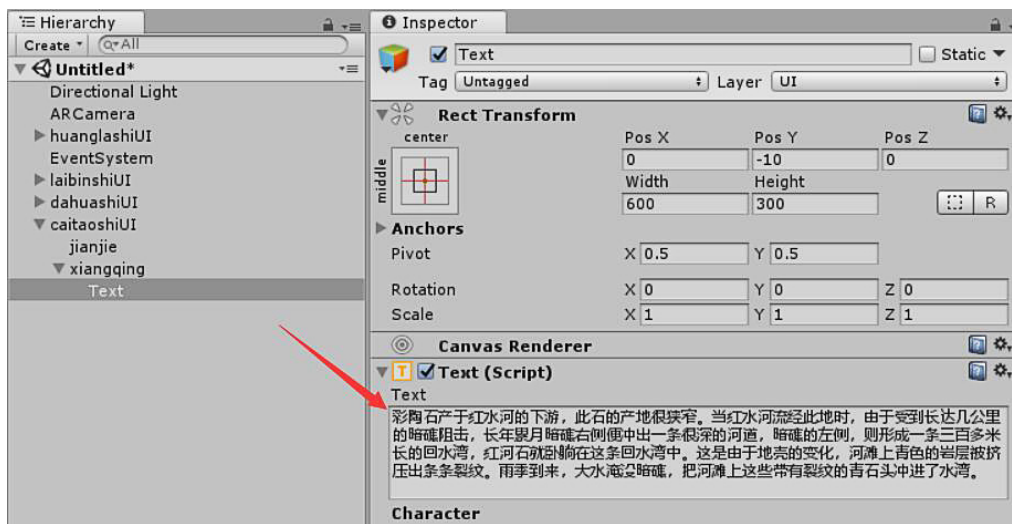


图1-20

UI脚本制作与绑定在“Project”面板中，选择“Vuforia”“Scripts”文件夹，在里面新建一个脚本，重命名为“UI.cs”，如图1-21所示。

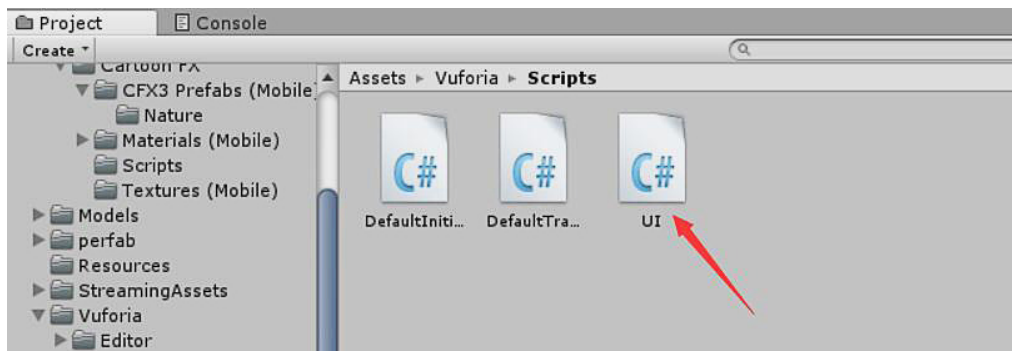


图1-21

选择脚本“UI.cs”，双击打开，添加具体代码如下：

```

using UnityEngine; using UnityEngine.UI;
public class UI : MonoBehaviour
{
    private Button jianjie; //简介按钮
    Private Image Xiangqing; //详情图像
    //调用场景中的成员
    private void Awake() { //从场景中获取简介按钮
        jianjie = transform.Find("jianjie").GetComponent<Button>();
    //从场景中获取详情图像
        Xiangqing =transform.Find("xiangqing").GetComponent<Image>();
    //监听简介按钮事件 jianjie.onClick.AddListener(() => {
        //如果点击了简介按钮，出现详情图像，缩放由0到1
        Xiangqing.transform.localScale = Vector3.one; }); }}
    
```

在“Project”面板中，将制作好的UI脚本文件挂载到四个UI对象上。打开“Vuforia” — “Scripts”文件夹，选择脚本“UI.cs”，将其分别拖曳给“Hierarchy”面板中“huanglashiUI” “laibinshiUI” “dahuashiUI” “caitaoshiUI”四个对象，实现UI脚本的挂载，如图1-22所示。

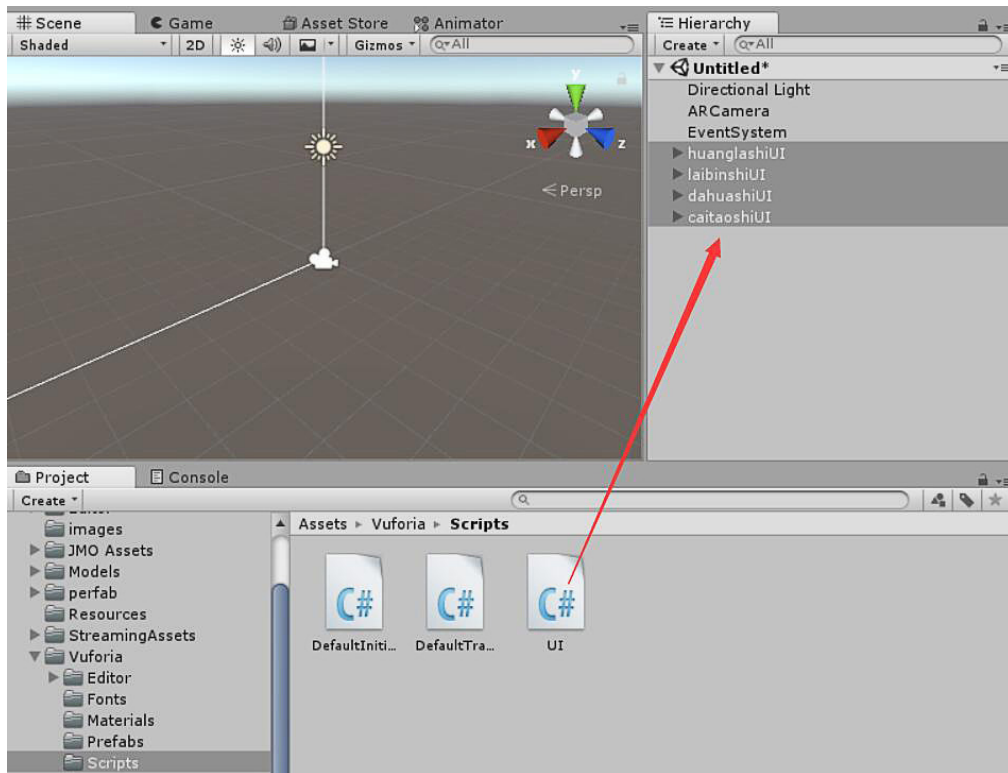


图1-22

UI预制体制作在“Hierarchy”面板中，选择之前做好的“huanglashiUI”“laibinshiUI”“dahuashiUI”“caitaoshiUI”四个对象，将其分别拖曳至“Project”面板中的“perfab”文件夹中，创建四个UI预制体，如图1-23所示。

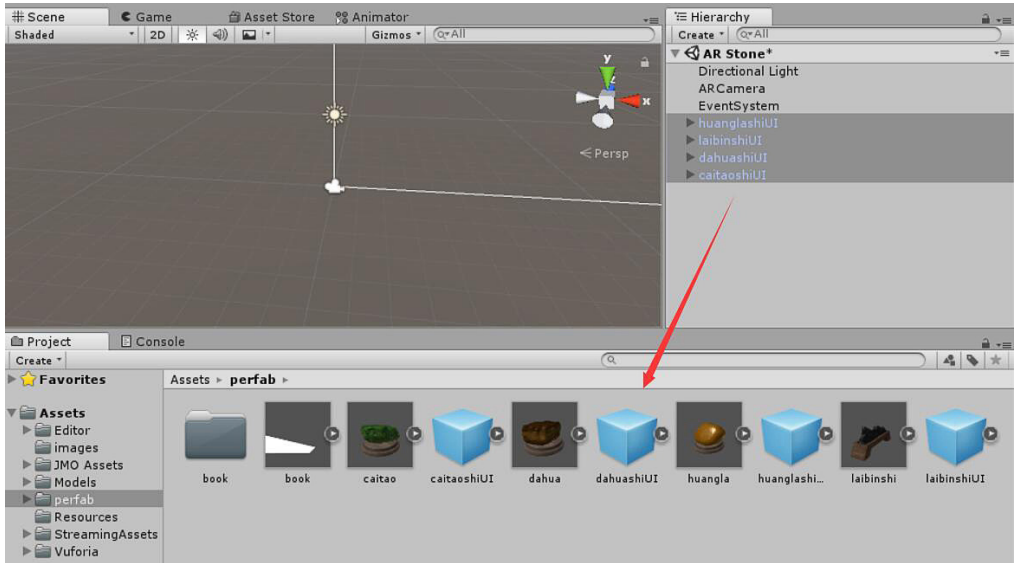


图1-23

## 2. 动画

制作识别封面图像出现书翻页动画。首先需要创建一个Images对象，在“GameObject”—“Vuforia”中选择“Image”，创建Images对象，用于存放识别图像，如图1-24所示。



动画

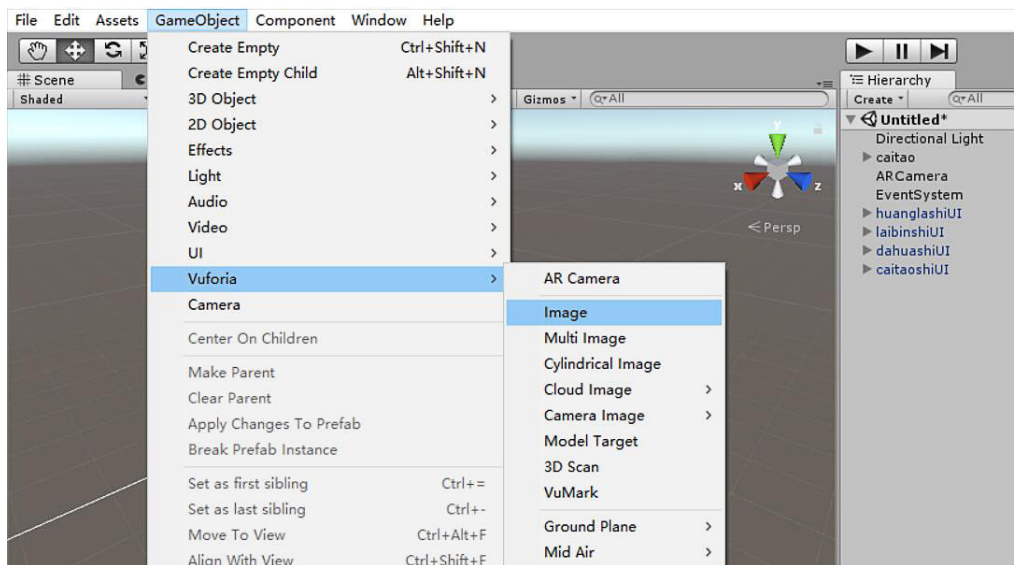


图1-24

在“Hierarchy”面板中，选择“ImageTarget”，在其“Inspector”属性面板中重命名为“fengmian”，Database和Image Target分别选择之前设置好的数据库“cover”和图片对象“fengmian”，完成后如图1-25所示。

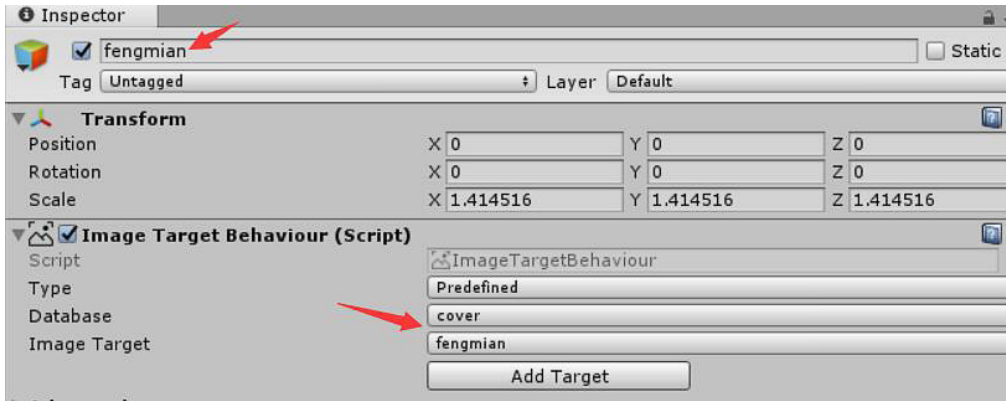


图1-25

选择“fengmian”，在其“Inspector”属性面板中，点击脚本“Default Trackable Event Handler.cs”，会在项目面板里显示其文件所在位置，如图1-26所示。

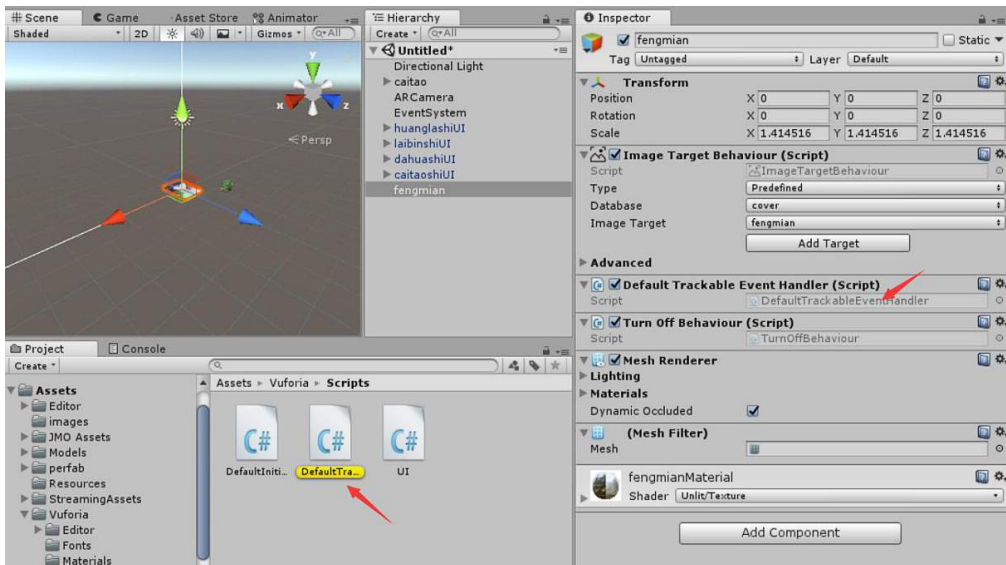


图1-26

在“Project”面板中，选择脚本“DefaultTrackableEventHandler.cs”，按Ctrl+D复制一份，重命名为“MyDefaultTrackableEventHandler”，如图1-27所示。

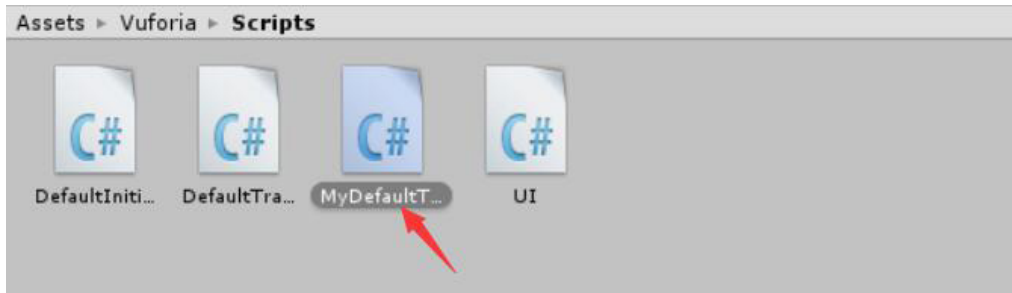


图1-27

双击打开脚本“MyDefaultTrackableEventHandler.cs”，在代码中将其类名修改为“MyDefaultTrackableEventHandler”，具体修改代码如下：

```
public class MyDefaultTrackableEventHandler : MonoBehaviour,
ITrackableEventHandler
{
    .....
}
```

在“Hierarchy”面板中，选择“fengmian”对象，在其“Inspector”属性面板中，选择脚本“DefaultTrackableEventHandler”，鼠标右键，选择“Remove component”移除组件，移除脚本组件“DefaultTrackableEventHandler”，如图1-28所示。

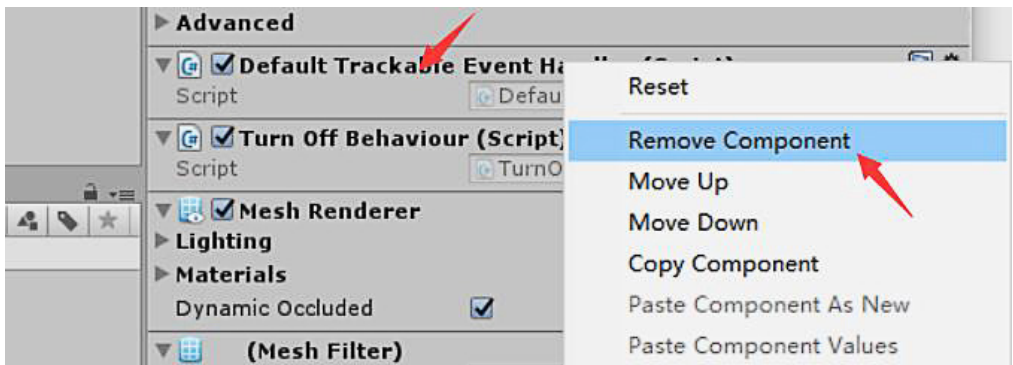


图1-28

在“Project”面板中，打开“Vuforia” — “Scripts”文件夹，选择其中的脚本“MyDefaultTrackableEventHandler.cs”，将其拖曳给“Hierarchy”面板中的“fengmian”对象，最终会在“fengmian”的“Inspector”属性面板中显示该脚本，如图1-29所示。

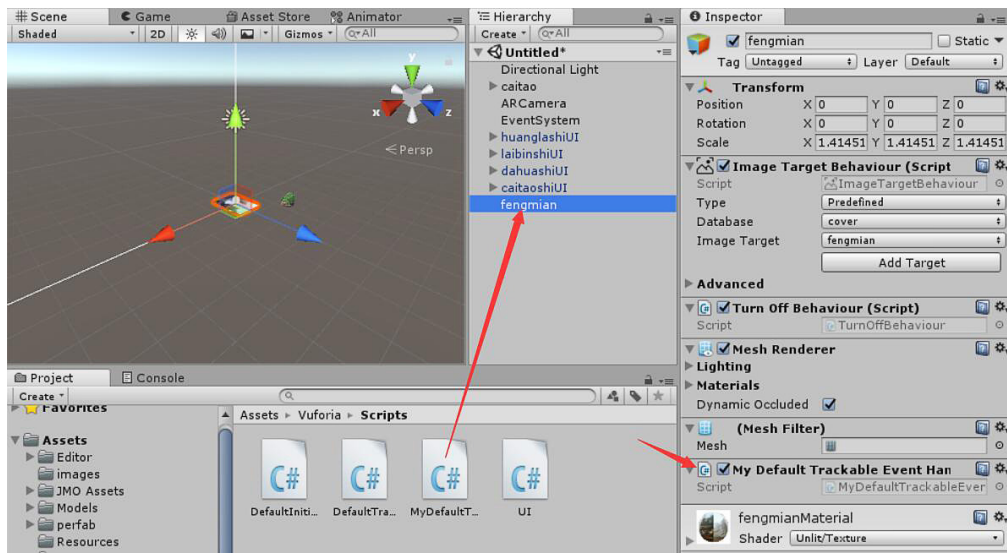


图1-29

在“Project”面板中，双击打开脚本“MyDefaultTrackableEventHandler.cs”，定义变量，具体代码如下：

```
public class MyDefaultTrackableEventHandler : MonoBehaviour,
ITrackableEventHandler
{
    public GameObject modelPrefab; //定义模型
    .....
}
```

在“Project”面板中，打开“perfab”文件夹，选择“book”动画预制体，将其拖曳给“Hierarchy”面板中的“fengmian”对象和“fengmian”挂载的脚本“MyDefaultTrackableEventHandler”的变量“Model Prefab”处，实现动画的挂载，如图1-30所示。

在“Hierarchy”面板中，选择“book”对象，在其属性面板修改位置及大小，点击“Apply”同步，并取消勾选，让它场景中隐藏不显示，完成如图1-31所示。



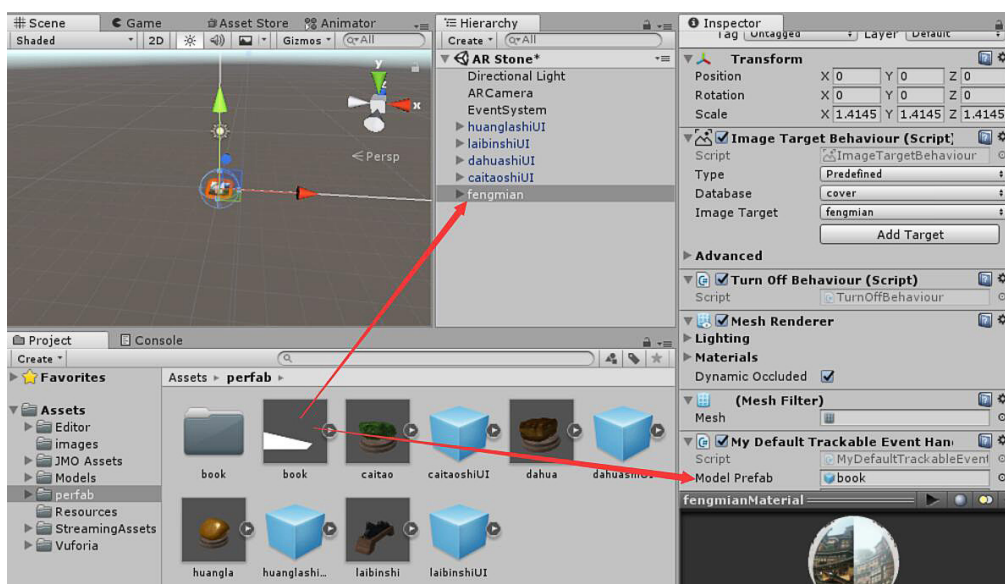


图1-30

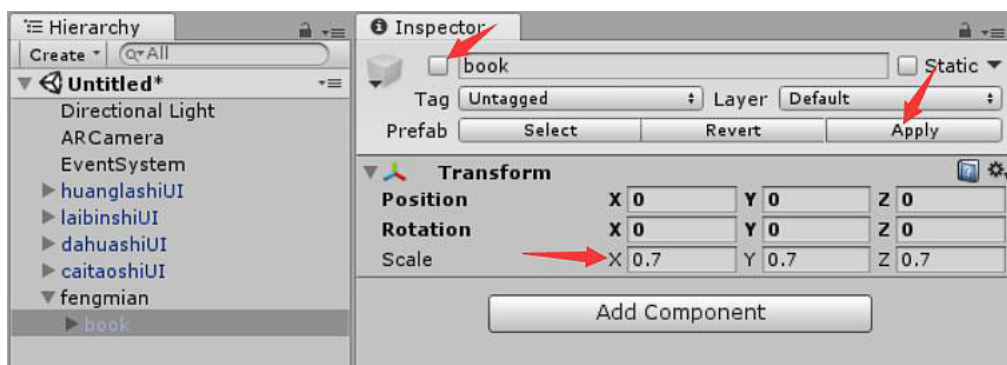


图1-31

### 3. 特效

在“Project”面板中，选择脚本“MyDefaultTrackableEvent Handler.cs”，双击打开，定义一个特效变量，添加具体代码如下：



特效

```
public class MyDefaultTrackableEventHandler : MonoBehaviour,
ITrackableEventHandler
{
    public GameObject modelPrefab; //定义模型
    public GameObject textPrefab; //定义特效
}
```

选择“fengmian”对象，然后在“Project”面板中，选择特效预制体“CFXM3\_FallingLeaves”，将其拖曳至“fengmian”对象的脚本变量“Texiao Prefab”处。如图1-32所示。

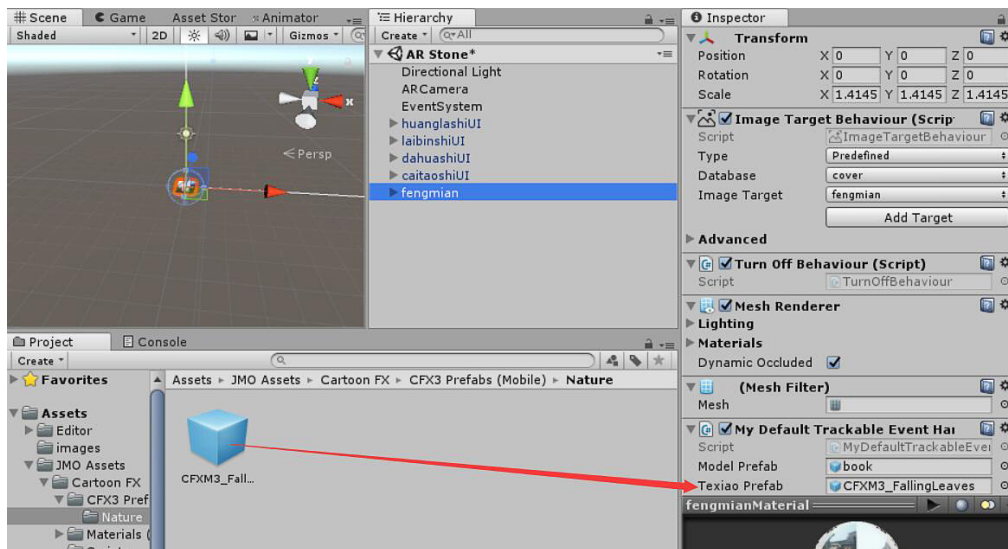


图1-32

#### 4. 音频播放

在“Hierarchy”面板中，选择“fengmian”对象，在其属性面板下添加音效组件，如图1-33所示。



音频播放

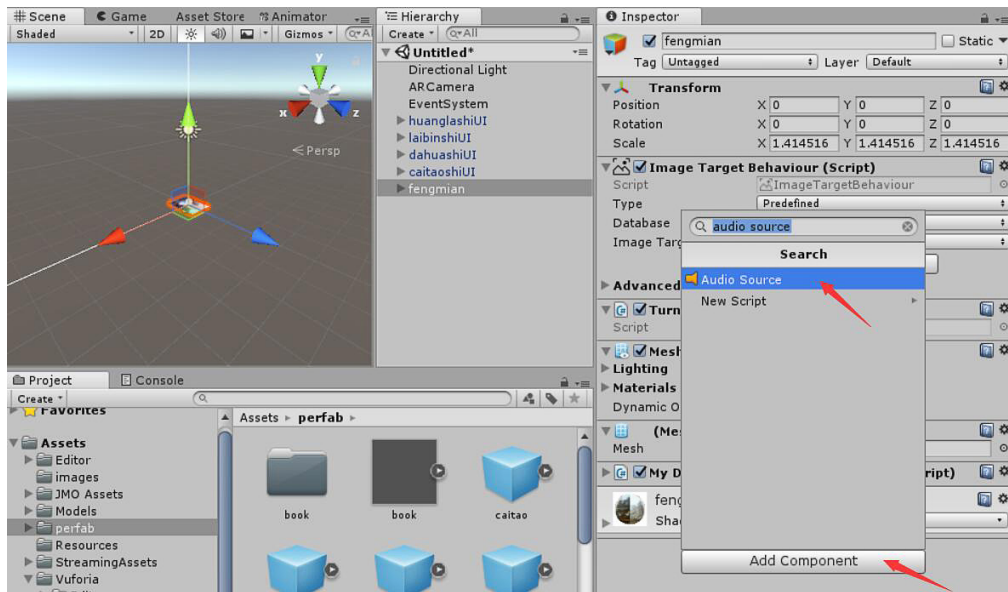


图1-33



在“Project”面板中，选择音效“The Fading Stories-不再年轻的村庄”，将其拖曳至音效组件处，并取消勾选自动播放，如图1-34所示。

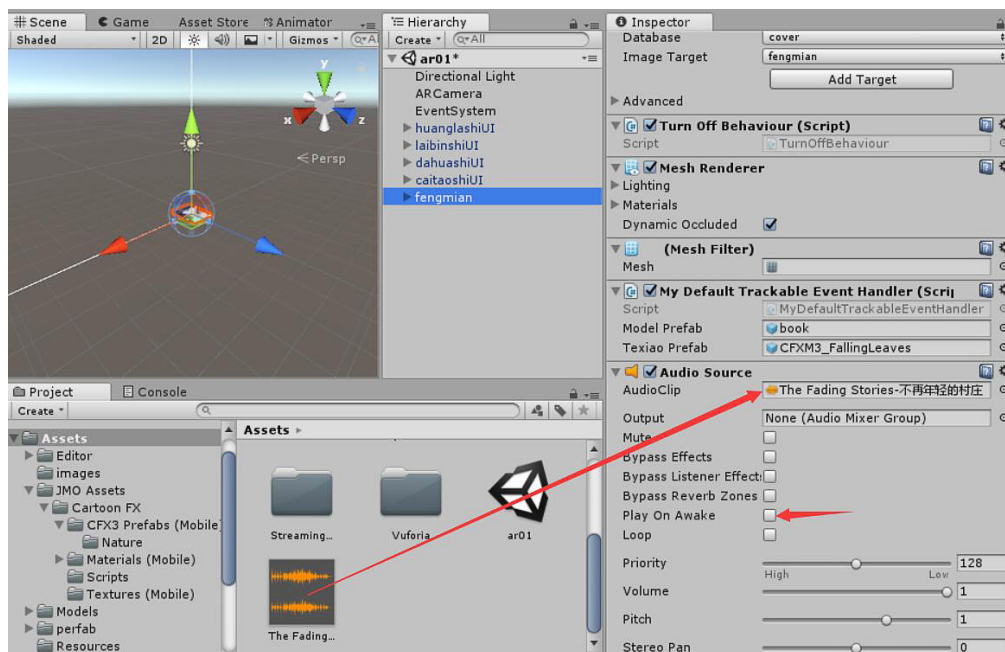


图1-34

添加音效组件后，在脚本“MyDefaultTrackableEventHandler”中继承这个音频组件，打开脚本“MyDefaultTrackableEventHandler”，实现音频组件继承，具体代码如下：

```
public class MyDefaultTrackableEventHandler : MonoBehaviour,
ITrackableEventHandler
{
    public GameObject modelPrefab; //定义模型
    public GameObject texiaoPrefab; //定义特效
    private AudioSource audio; //音频成员
    .....
    protected virtual void Start()
    {
        mTrackableBehaviour
        GetComponent<TrackableBehaviour>();
        if (mTrackableBehaviour)
            mTrackableBehaviour.RegisterTrackableEventHandler(this);
    }
}
```

```

//继承项目中的音频组件
    audio = this.GetComponent<AudioSource>();
}
}
识别到图像时播放背景音效。打开脚本
MyDefaultTrackableEventHandler, 具体代码如下:
protected virtual void OnTrackingFound()
{
    if (!audio.isPlaying) //判断音频是否播放
    {
        audio.Play(); //播放音频
    }
    .....
}

```

### 5. 生成与销毁

首先制作第一张奇石识别图像——黄蜡石。在“Hierarchy”面板中，选择“fengmian”对象，Ctrl+D复制，重命名为“huanglashi”，将层级下的“book”预制体删除，如图1-35所示。

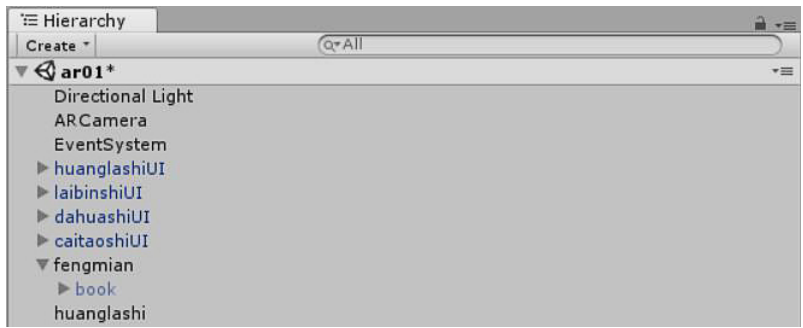


图1-35

在“Inspector”面板中，修改“huanglashi”对象属性面板里的数据库和识别图片，Database选择“stone”，Image Target选择“huangla2”，并且移除音频组件“Audio Source”。

在“Project”面板中找到“huangla”预制体模型，将其分别拖曳至“Hierarchy”面板的“huanglashi”对象处和脚本变量“Model Prefab”处，最终完成如图1-36所示。

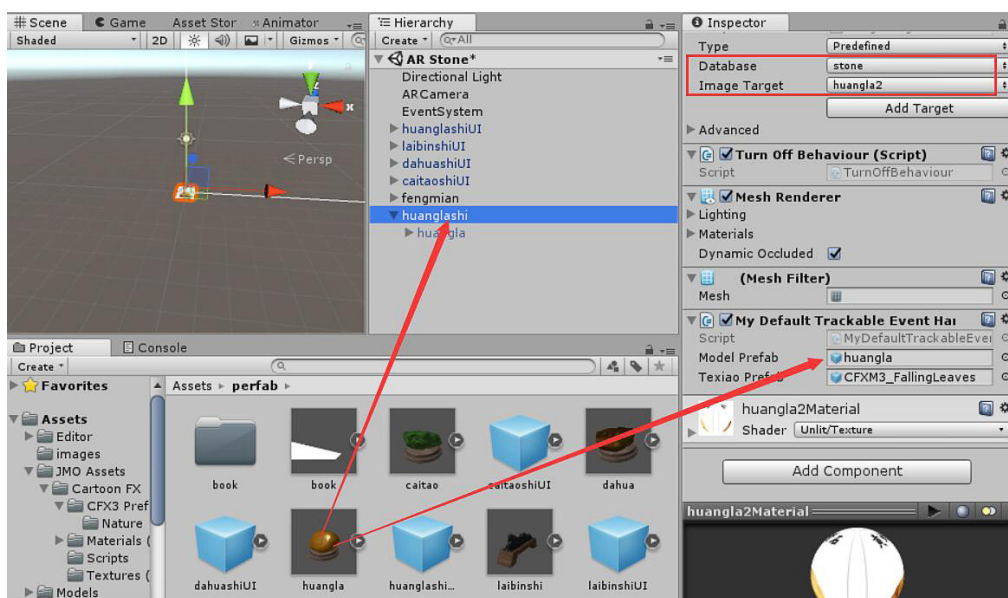


图1-36

隐藏场景中的“huangla”模型。在“Hierarchy”面板中，选择“huanglashi”对象层级下的“huangla”模型，在其属性面板取消勾选，如图1-37所示。

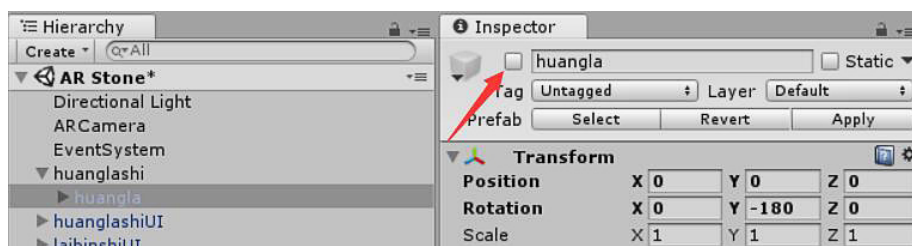


图1-37

制作第二张识别图像——来宾石。在“Hierarchy”面板中，选择“huanglashi”对象，Ctrl+D复制，重命名为“laibinshi”，将其层级下的“huangla”预制体删掉，如图1-38所示。

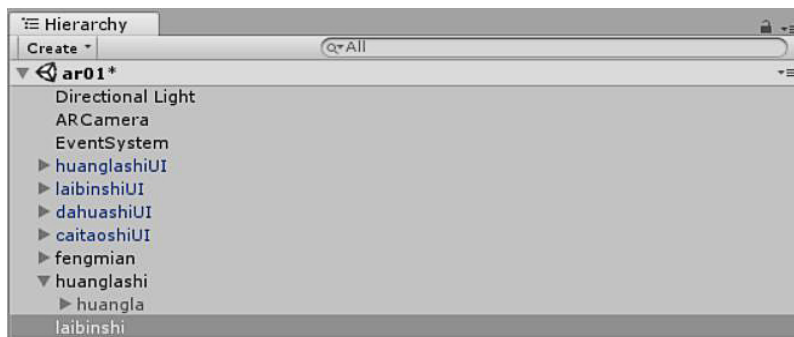


图1-38

在“Inspector”面板中，更换“laibinshi”对象的识别图片，Image Target选择“laibin2”。

在“Project”面板中找到“laibinshi”预制体模型，将其分别拖曳给“Hierarchy”面板的“laibinshi”对象以及“laibinshi”对象所绑定的脚本变量“Model Prefab”处，最终完成如图1-39所示。

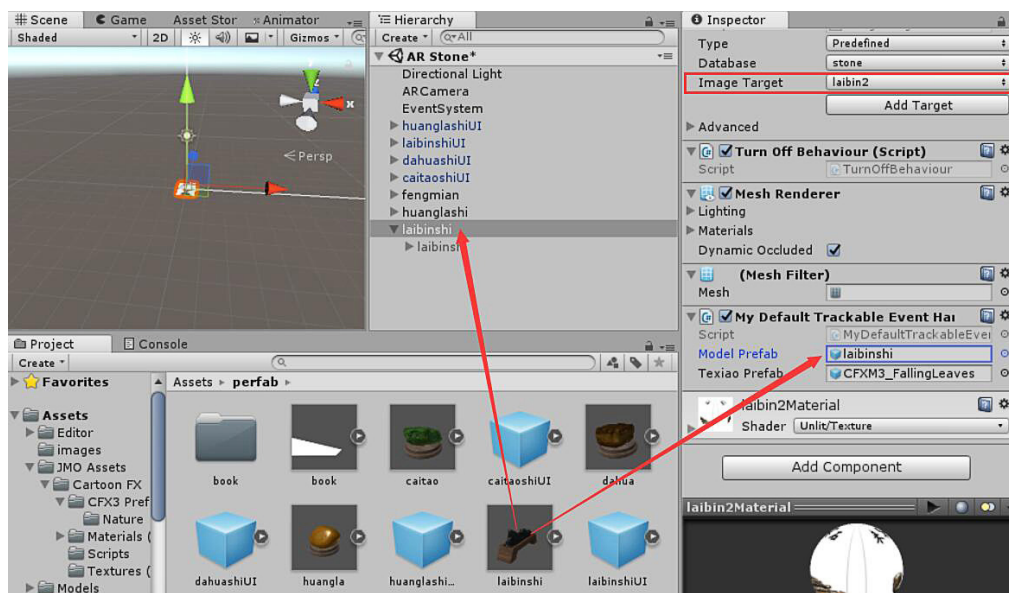


图1-39

隐藏场景中的“laibinshi”模型。在“Hierarchy”面板中，选择“laibinshi”对象层级下的“laibinshi”模型，在其属性面板中取消勾选，如图1-40所示。

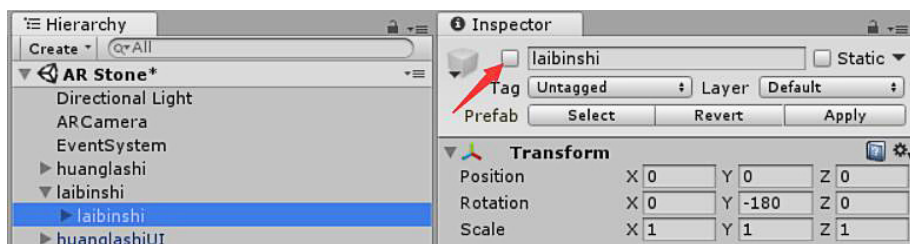


图1-40 隐藏laibinshi图

制作第三张奇石识别图像——大化石。在“Hierarchy”面板中，选择“laibinshi”对象，Ctrl+D复制，重命名为“dahuashi”，将其层级下的“laibinshi”预制体删掉，如图1-41所示。

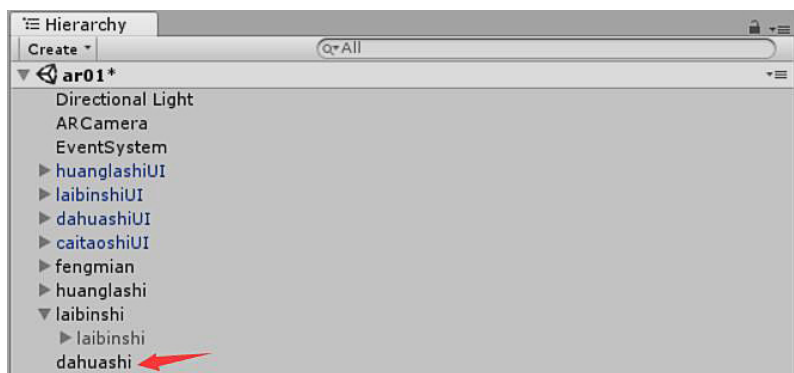


图1-41

在“Project”面板中找到“dahua”预制体模型，将其分别拖曳至“Hierarchy”面板的“dahuashi”处和脚本“Model Prefab”处。

在“Inspector”面板中，修改“dahuashi”对象属性面板中的识别图片，Image Target选择“dahua2”，最终完成后如图1-42所示。

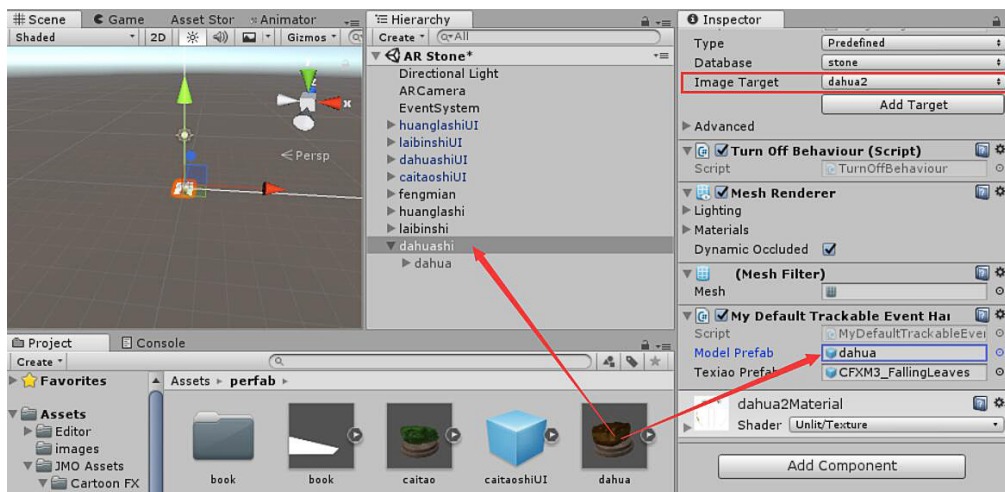


图1-42

隐藏场景中的“dahua”模型。在“Hierarchy”面板中，选择“dahuashi”对象层级下的“dahua”模型，在其属性面板中取消勾选，如图1-43所示。

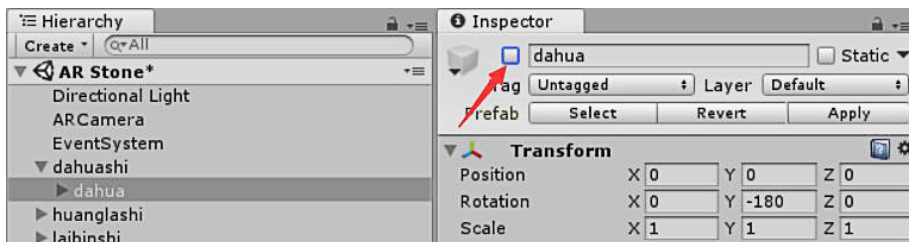


图1-43

制作第四张奇石识别图像——彩陶石。在“Hierarchy”面板中，选择“dahuashi”对象，Ctrl+D复制，重命名为“caitaoshi”，将其层级下的“dahua”预制体删掉，如图1-44所示。

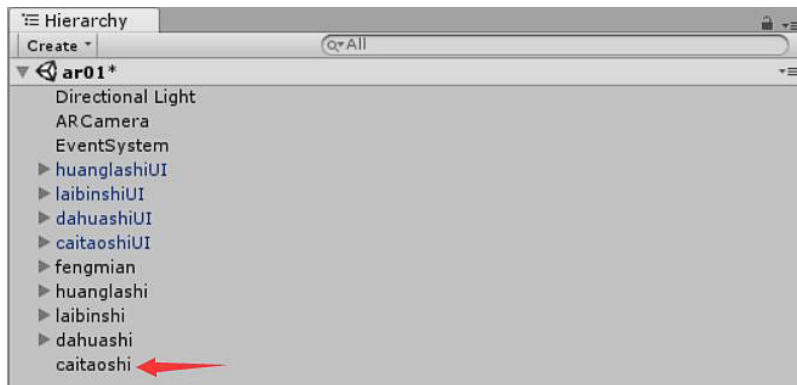


图1-44

在“Project”面板中，选择“caitao”预制体模型，将其分别拖曳至“Hierarchy”面板的“caitaoshi”对象处和“caitaoshi”对象所绑定的脚本变量“Model Prefab”处。

在“Inspector”面板中，修改“caitaoshi”对象属性面板中的识别图片，Image Target选择“caitao2”，最终完成后如图1-45所示。

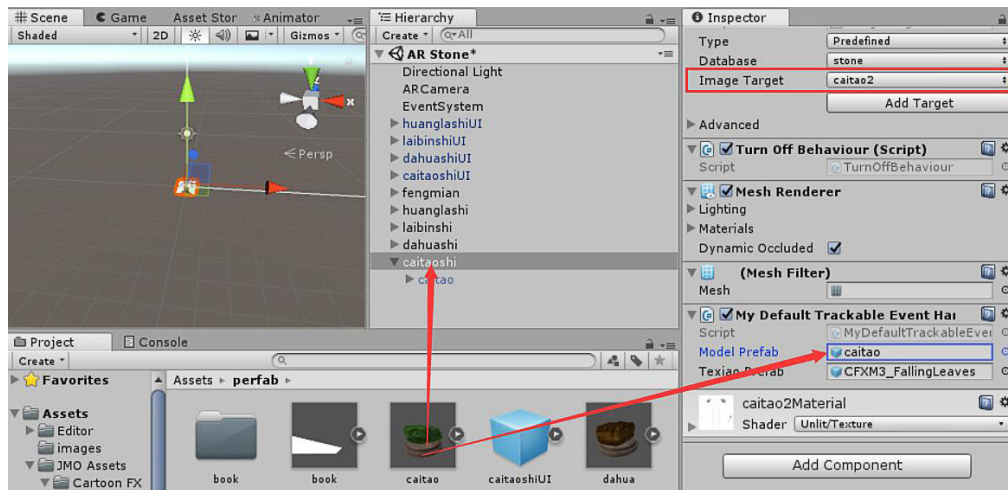


图1-45

隐藏场景中的“caitao”模型。在“Hierarchy”面板中，选择“caitaoshi”对象层级下的“caitao”模型，在其属性面板中取消勾选，如图1-46所示。



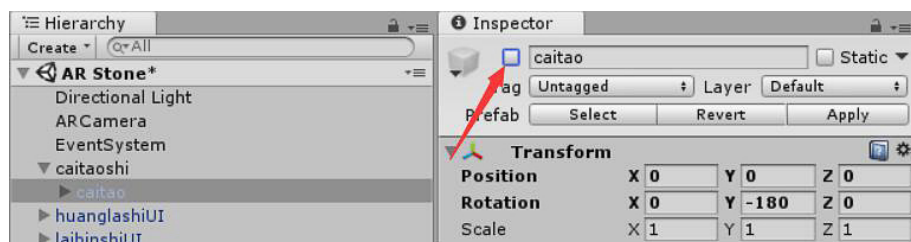


图1-46

在识别图片上挂载UI。在“Project”面板中，打开“Vuforia” — “Scripts”文件夹，打开脚本“MyDefaultTrackableEventHandler.cs”，定义一个UI变量，用于放置UI预制体，具体代码如下：

```
public class MyDefaultTrackableEventHandler : MonoBehaviour,
ITrackableEventHandler
{
    public GameObject modelPrefab; //定义模型
    public GameObject texiaoPrefab; //定义特效
    private AudioSource audio; //私有成员
    public GameObject thisui; //定义UI
    .....
}
```

在代码的最前面需对使用UI进行声明，添加代码如下：

```
using UnityEngine;
using Vuforia;
//声明支持UI
using UnityEngine.UI;
```

在“Project”面板中，打开“prefab”文件夹，选择之前做好的四个UI预制体，分别拖曳给“Hierarchy”面板中的“huanglashi”“laibinshi”“dahuashi”“caitao shi”四个对象所挂在的脚本变量“Thisui”处，其中“huanglashi”的UI对象拖曳如图1-47所示。

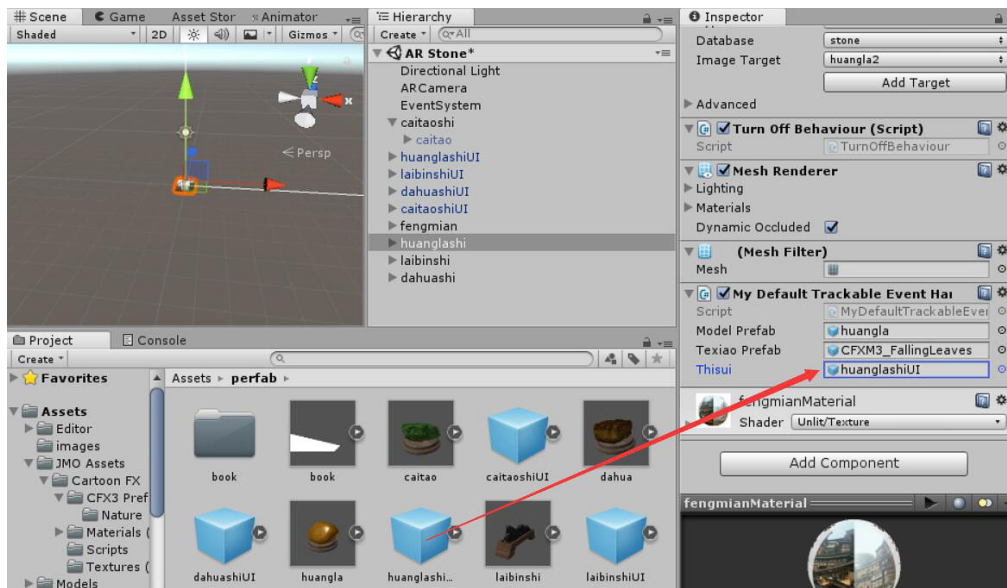


图1-47

实现模型、UI、动画特效的生成与销毁。双击打开脚本“MyDefaultTrackableEventHandler.cs”，删除“OnTrackingFound()”函数和“OnTrackingLost()”函数原有的代码，实现模型、UI、动画特效的生成与销毁，具体代码如下，注意销毁代码里的文本应与识别图像后生成的模型的名称一致。

```
protected virtual void OnTrackingFound()
{
    if (!audio.isPlaying)
    {
        audio.Play();
    }
    //识别到图片时生成模型
    //生成奇石模型
    GameObject model = GameObject.Instantiate(modelPrefab, transform.
position - new Vector3(0f, 0f, 0f), transform.rotation);
    model.transform.parent = this.transform;
    model.transform.rotation = Quaternion.Euler(0, -180, 0);
    //生成特效
    GameObject texiao = GameObject.Instantiate(texiaoPrefab,
transform.position, Quaternion.identity);
    texiao.transform.parent = this.transform;
```



```

//生成UI
GameObject ui = GameObject.Instantiate(thisui);
ui.transform.parent = this.transform;
}
protected virtual void OnTrackingLost()
{
//未识别到图片时销毁已生成的模型
//销毁奇石模型
Destroy(GameObject.Find("book(Clone)"));
Destroy(GameObject.Find("huangla(Clone)"));
Destroy(GameObject.Find("laibinshi(Clone)"));
Destroy(GameObject.Find("dahua(Clone)"));
Destroy(GameObject.Find("caitao(Clone)"));
//销毁特效
Destroy(GameObject.Find("CFXM3_FallingLeaves(Clone)"));
//销毁UI
Destroy(GameObject.Find("huanglashiUI(Clone)"));
Destroy(GameObject.Find("laibinshiUI(Clone)"));
Destroy(GameObject.Find("dahuashiUI(Clone)"));
Destroy(GameObject.Find("caitaoshiUI(Clone)"));
}

```

## 6. 旋转

在“Project”面板中，选择“Scripts”文件夹，鼠标右键，选择“Create” — “C#Script”，创建一个新脚本，重命名为“Rotate”，如图1-48所示。



旋转与缩放

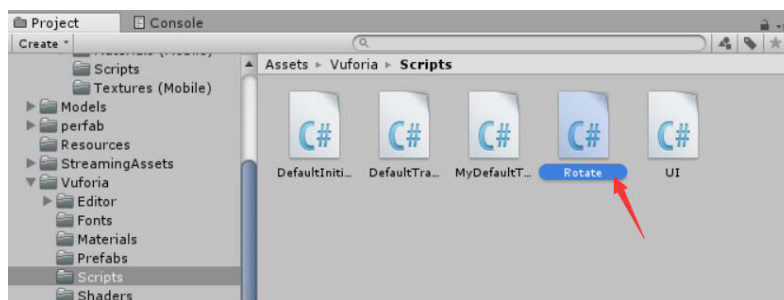


图1-48

双击打开“Rotate”脚本，具体实现代码如下：

```

public class Rotate : MonoBehaviour
{
    float xSpeed = 150f; //水平速度
    float ySpeed = 50f; //垂直速度
    void Update ()
    {
        if(Input.GetMouseButton(0)) //如果触摸了屏幕
        {
            //判断是几个手指触摸
            if(Input.touchCount==1)
            {
                //第一个手指触摸， phase状态为Moved
                滑动
                if (Input.GetTouch(0).phase == TouchPhase.Moved)
                {
                    //以手指横向移动的值绕
                    世界坐标的Y轴旋转
                    transform.Rotate(Vector3.up * Input.GetAxis("Mouse X") * xSpeed * Time.
                    deltaTime, Space.World);
                    //以手指纵向移动的值绕
                    世界坐标的X轴旋转
                    transform.Rotate(Vector3.left * Input.GetAxis("Mouse Y") * ySpeed * Time.
                    deltaTime, Space.World);
                }
            }
        }
    }
}

```

在“Project”面板中，打开“Vuforia” — “Scripts”文件夹，将制作好的“Rotate”脚本文件挂载到“Hierarchy”面板中的“huangla” “laibinshi” “dahua” “caitao”四个对象模型上，如图1-49所示完成脚本挂载。

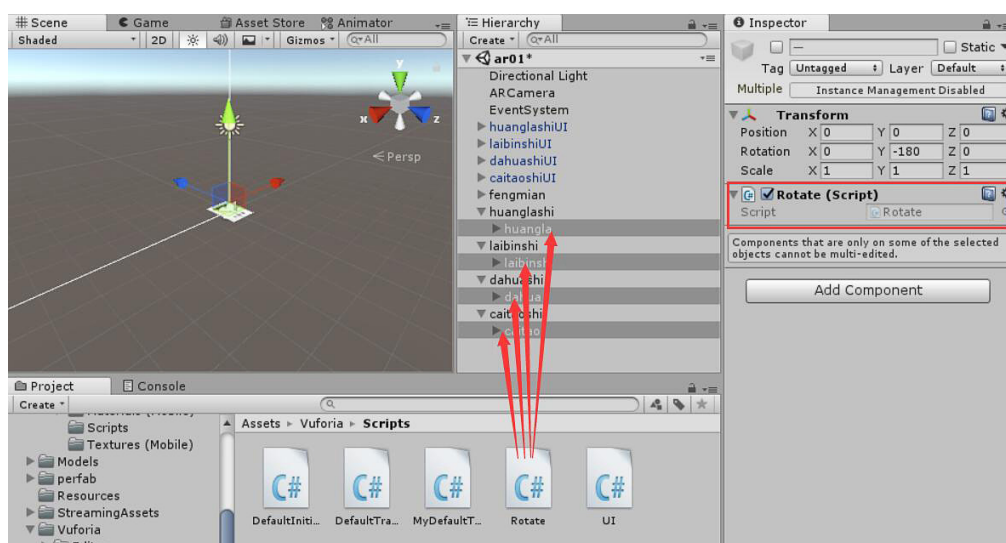


图1-49

## 7. 缩放

实现奇石模型放大缩小功能。在“Project”面板中，打开“Vuforia” — “Scripts”文件夹，新建一个脚本，重命名为“EnLarge.cs”，如图1-50所示。

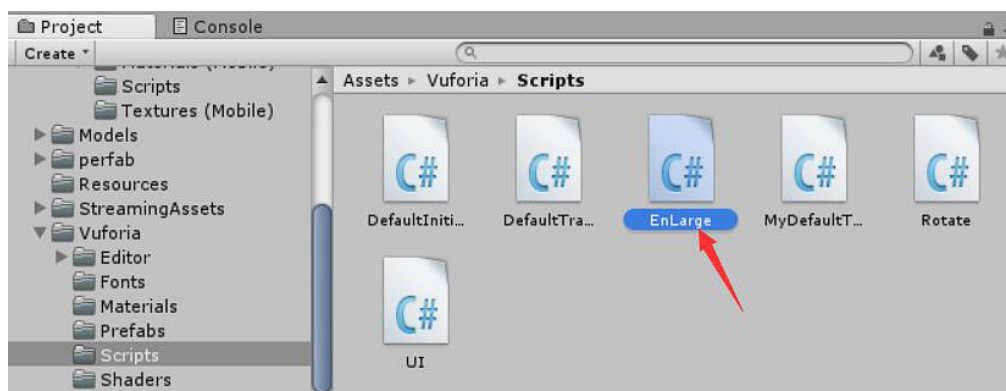


图1-50

打开脚本“EnLarge.cs”，具体实现代码如下：

```
public class EnLarge : MonoBehaviour
{
    Vector2 oldPos1; //第一个手指
    Vector2 oldPos2; //第二个手指
    void Update ()
    {
```

```

        //如果有两个手指
        if(Input.touchCount == 2)
        {
            //如果第一个手指位置是移动的或者第二个手指
            //位置是移动的
            if (Input.GetTouch(0).phase == TouchPhase.Moved || Input.
            GetTouch(1).phase == TouchPhase.Moved)
            {
                //第一个手指位置
                Vector2 temPos1 = Input.GetTouch(0).position;
                //第二个手指位置
                Vector2 temPos2 = Input.GetTouch(1).position;
                if(isEnLarge(oldPos1, oldPos2, temPos1, temPos2))
                {
                    float oldScale = transform.localScale.x; //原始大小
                    float newScale = oldScale * 1.025f; //放大的倍数
                    transform.localScale = new Vector3(newScale,newScale,
                    newScale); //整体放大
                }
                else
                {
                    float oldScale = transform.localScale.x;
                    float newScale = oldScale / 1.025f; //缩小的倍数
                    transform.localScale = new Vector3(newScale, newScale,
                    newScale); //整体缩小
                }
                oldPos1 = temPos1;
                oldPos2 = temPos2;
            }
        }
    }
    //判断手势
    bool isEnLarge(Vector2 oP1,Vector2 oP2,Vector2 nP1,Vector2 nP2)
    {
        float length1 = Mathf.Sqrt((oP1.x - oP2.x) * (oP1.x - oP2.x)+ (oP1.y
        - oP2.y) * (oP1.y - oP2.y));
    }
}

```

```

float length2 = Mathf.Sqrt((nP1.x - nP2.x) * (nP1.x - nP2.x) + (nP1.y
- nP2.y) * (nP1.y - nP2.y));
//放大
if(length1 < length2)
{
return true;
}
//缩小
else
{
return false;
}
}

```

在“Project”面板中，打开“Vuforia” — “Scripts”文件夹，将制作好的脚本“EnLarge.cs”文件绑定到“Hierarchy”面板中的“huangla” “labinshi” “dahua” “caitao”四个模型对象上，如图1-51所示。

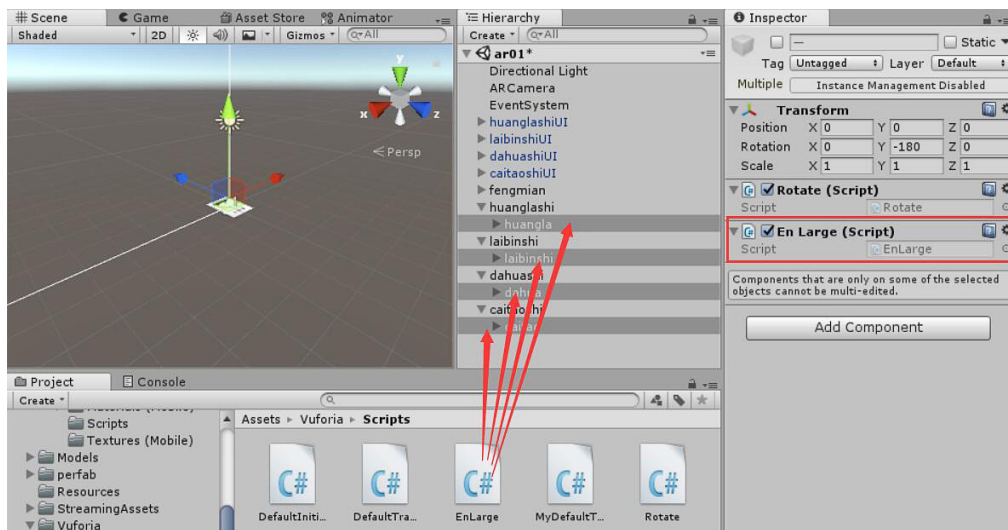


图1-51

### 1.2.4 案例发布

在菜单栏中点击“File” — “Build Settings”，点击“PlayerSettings”在PC端勾选“Vuforia Augmented Realit”，最后点击“Build”即可导出App，如图1-52所示。



案例发布

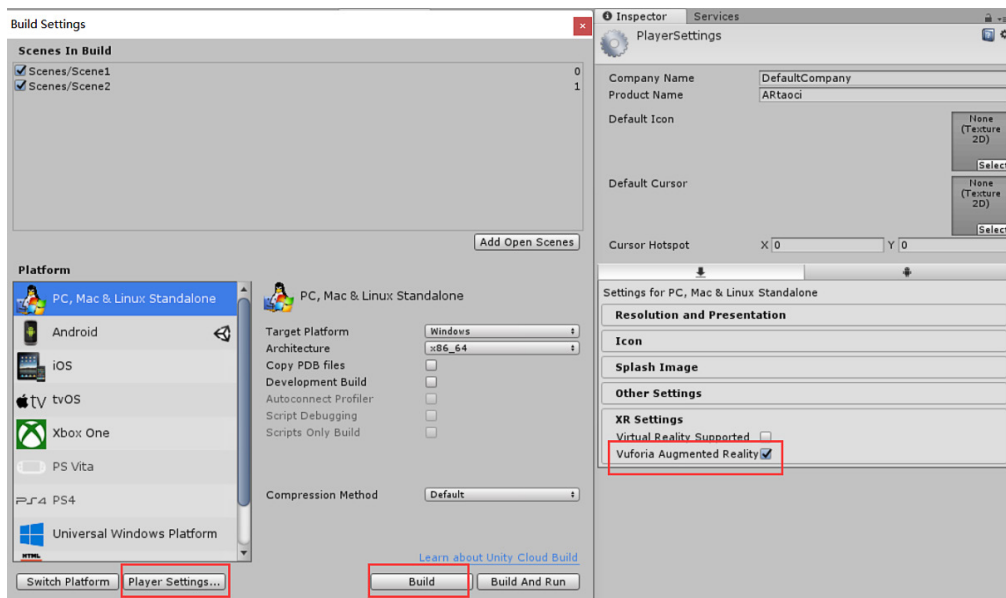


图1-52

案例发布效果图如图1-53、1-54 所示。

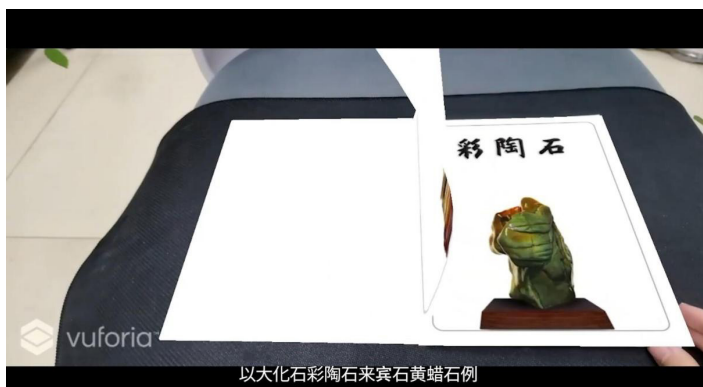


图1-53

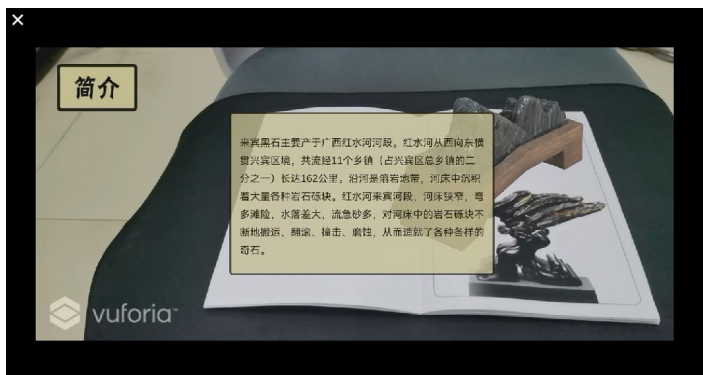


图1-54