



## 5.1 基于 Kivicube 的 WebAR 应用开发

Kivicube 是成都弥知科技有限公司推出的国内首款免费的 WebXR 在线制作平台,于 2018 年 10 月上线,帮助用户零门槛制作 AR 场景,支持如图像跟踪与实物跟踪、空间 SLAM 等场景使用,持续为品牌提供 AR 制作能力支持。Kivicube 可以在线制作 AR、VR 与 3D 场景,制作的 3D 场景可运行在通用的 Web 平台上。用户可以在其官网(网址请参考本书配套资源)注册账号并进行项目的制作,在大部分情况下用户无须编写代码。由于使用 Kivicube 开发 WebXR 项目无须用户在本地搭建复杂的开发环境,只需要在有网络的情况下就可以进行访问和制作,因此可以认为 Kivicube 是一种 SaaS(软件即服务)平台。

### 5.1.1 项目创建

#### 1. 注册并登录 Kivicube 平台

在浏览器(推荐使用 Windows Edge、火狐或者 Google 浏览器)中访问 Kivicube 官方网站,默认弹出注册和登录界面,如图 5.1 所示。可以选择使用手机号、邮箱或者用户名的方式来注册登录 Kivicube,在后续的使用过程中,每次访问 Kivicube 都需要使用已经注册好的账户进行登录。

注册登录成功之后,即可进入 Kivicube 的后台主页面,如图 5.2 所示。在该页面的上方,可以选择为创建好的项目添加 AR 场景、3D 场景,或者进行 AR 场景定制;在下方,可以单击 ALL 按钮来显示当前用户所拥有的所有项目。单击“+”按钮可以进行项目的新建。在左下角可以通过切换标签选项卡,来显示当前用户的项目、素材或者用户中心。

在用户中心可以看到当前用户的权限信息,可以看出当前用户为免费账号,可以创建的场景数最大为 2 个,如图 5.3 所示。

#### 2. 场景创建

单击后台主页下方的“+”按钮新建项目,如图 5.4 所示。

在弹出的“新建项目”界面中,需要填写项目信息,如图 5.5 所示,包括项目名称、项目描



图 5.1 Kivcube 注册和登录界面



图 5.2 Kivcube 后台主页

述、项目 Logo 和项目类型。其中,项目名称和项目 Logo 为必选项,上传一个封面图片作为项目 Logo,默认的项目类型为 Web3D,然后单击“保存”按钮,这样一个 WebXR 项目就创建成功了。



图 5.3 用户中心



图 5.4 新建项目

### 3. 项目设置

默认情况下,新建的项目为 Web3D 项目,即在 Web 浏览器中展示三维模型。应用此默认选项后,项目入口没有任何效果,用户无法通过扫描打开项目下的场景。如果设置项目为“云识别”,则项目下所有的云识别场景都可以通过项目入口扫描识别图进入。如果设置项目为“图像检测与跟踪”,则场景也可以通过项目入口扫描识别图进入。

这里需要特别说明的是,如果项目同时含有 Web3D、云识别、图像检测与跟踪 3 类场景,此时当项目被设置为云识别,那么项目入口就只能扫描到云识别场景,图像检测与跟踪场景同理。

## 5.1.2 场景创建

### 1. 选择需要创建的场景类型

可以为当前已经创建好的项目添加 AR 场景、3D 场景以及定制化的 AR 场景,如图 5.6 所示。

- (1) AR 场景: AR 场景是指在实时摄像头画面上呈现 AR 内容的场景。
- (2) 3D 场景: 指普通的 3D 场景,并且体验时不会呈现摄像头的画面。
- (3) AR 场景定制: 指创建适用于各个流量平台的场景,包括支付宝、淘宝、天猫、京东等购物类网站、Facebook、Snapchat 等。该案例中选择场景的类型为 3D 场景。

图 5.5 项目信息填写



图 5.6 新建场景

## 2. 创建 3D 场景

基于 3D 内容的 3D 互动体验,广泛地运用于智慧电商、3D 广告、3D 传媒等,能够广泛支持所有安卓版浏览器与带浏览器功能的 App、安卓微信、安卓微信小程序、所有 iOS 浏览器与带浏览器功能的 App,以及 iOS 微信小程序。创建此类场景时首先要填写 3D 互动场景信息,如图 5.7 所示;选择当前场景所属的项目名称,并且填写场景名称与场景描述,单击“立即制作”按钮,即可完成 3D 场景的创建。

在这里输入的场景名称会显示在展示端的主页,如图 5.8 所示。

## 3. 创建 AR 场景

创建 AR 场景,用户可以通过扫描现实世界中的平面图像(如杂志封面、照片与名片等),然后呈现出与之对应的场景内容与交互,并实现实时位置匹配与跟踪。

选择场景类型为 AR 场景后,需要填写图像检测与跟踪场景信息,如图 5.9 所示。同样包括所属项目的名称和场景名称,同时需要上传待识别的图片,然后单击“立即制作”按钮完成创建。

图 5.7 创建 3D 场景



图 5.8 3D 场景的预览效果



图 5.9 创建 AR 场景

例如,创建一个 AR 场景,添加 Babylon.js 的 Logo 作为识别图,单击“立即制作”按钮,如图 5.10 所示。



(a) 创建场景并上传识别图

(b) 场景中显示识别图

图 5.10 添加识别图

可以为一个场景添加多张识别图,一方面用户通过扫描不同的识别图进入同一个场景体验,另一方面也可以完成对实物的识别。AR 体验场景还支持空间位置定位与跟踪,用户通过扫描现实环境中的平面,然后在平面上呈现与之对应的场景内容与交互,并实现实时位置匹配与跟踪。

同样,系统支持陀螺仪。陀螺仪是基于移动设备的陀螺仪传感器,用户可以转动设备查看四周的场景内容,并与之交互。云识别基于云端图像识别与匹配,能够支持 10 万张图片的识别。

#### 4. 创建流量平台 AR 场景

平台支持支付宝、淘宝/天猫、Facebook、Snapchat、京东等平台,可以轻松地在上述平台中进行数字化建模,并将虚拟的商品放置在地板、桌面等现实世界中。其中支付宝能够支持图像检测与跟踪、平面检测与跟踪、人脸检测与跟踪、微笑检测、人体姿态检测、手势检测等功能。

### 5.1.3 创建场景内容

#### 1. 打开场景

单击打开在上述的操作中已经创建好的 AR 场景,如图 5.11 所示。



图 5.11 打开 AR 场景

#### 2. 上传素材

Kivicube 目前支持图片、模型、AR 视频、音频、透明视频、全景图 6 种类型的素材,其中“AR 场景”→“图像检测与跟踪”不支持全景图。如图 5.12 所示,这里选择上传的素材类型为 3D 模型,然后单击“上传素材”按钮。

这里要求上传的模型文件类型为 zip 格式,建议将模型从 3ds Max/Maya/Blender 等三维建模软件中导出,然后将其压缩为 zip 格式压缩包并拖入指定的区域后,模型开始上传,当上传进度为 100% 或者上传进度提示信息消失时,单击“完成”按钮,如图 5.13 所示。



(a) 选择素材类型为3D模型

(b) 单击“上传素材”按钮

图 5.12 选择素材的类型并上传



(a) 上传模型界面

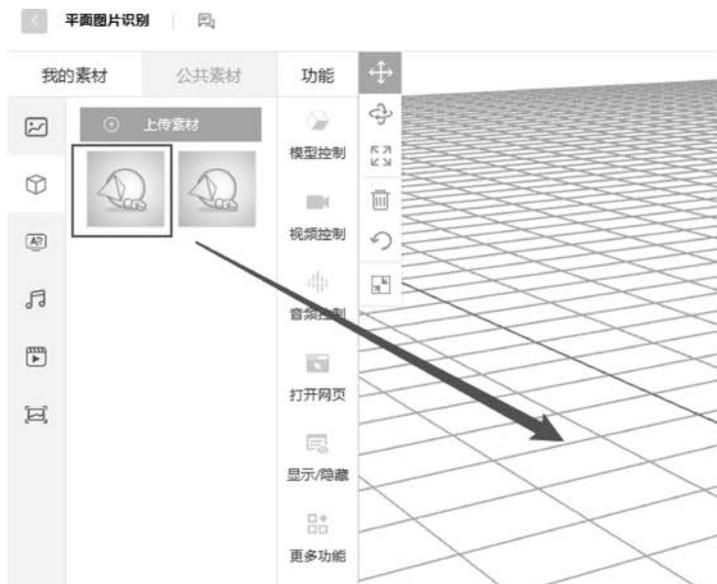
(b) 模型上传成功

图 5.13 上传模型素材

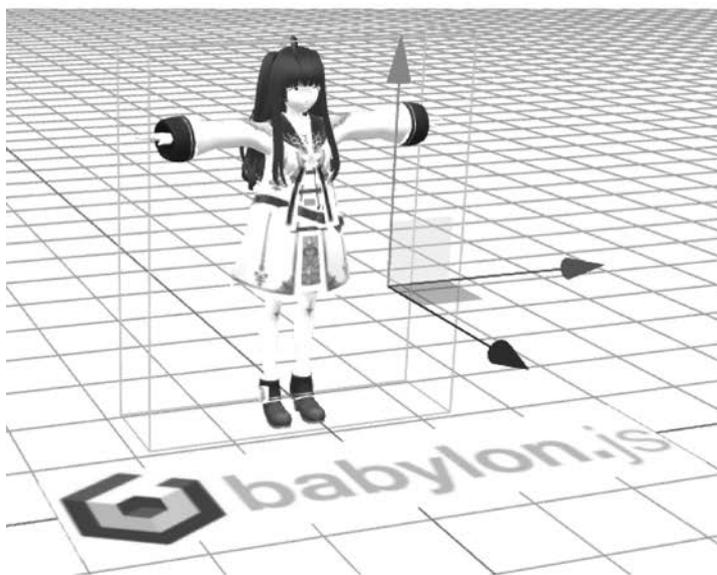
### 3. 将素材添加至场景

上传后的模型素材会显示在左侧的素材列表中,直接将需要的模型拖入场景,如图 5.14 所示。

素材添加完成后,就能从编辑器右边的“场景结构”选项卡中查看已添加的素材内容了。



(a) 将模型拖入场景



(b) 调整模型的位置以及坐标

图 5.14 将模型素材添加至场景

## 5.1.4 创建场景交互

### 1. 添加自动循环播放的音频

按照上述方法,还可添加音频类素材,如图 5.15 所示。添加完成后,将音频素材同样拖入场景中,然后在素材列表中选择该音频素材,单击“音频控制”按钮,在音频控制界面可以设置该音频文件的播放模式为“循环播放”(Loop),并且设置音频播放的触发条件为“扫描识别到”。



图 5.15 音频控制

### 2. 添加单击图片打开网页的交互(可选)

如果要为图片添加网页交互,那么可以选择图片素材,然后单击“打开网页”按钮。在弹出的打开网页设置界面中,设置功能的名称,添加将要打开的网址,并且设置触发条件为“扫描识别到”,如图 5.16 所示。

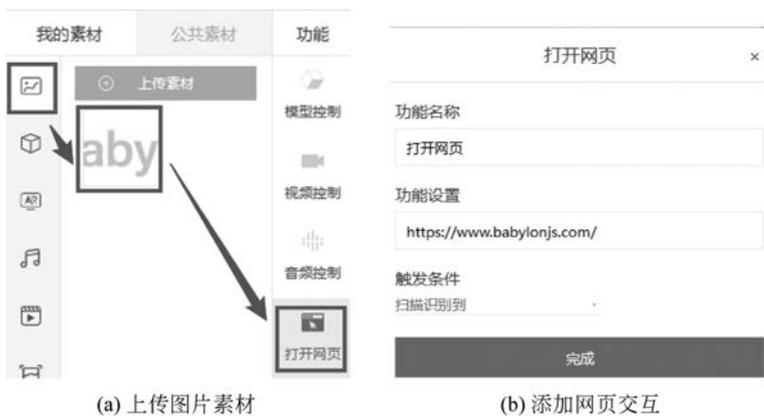


图 5.16 添加网页设置

### 3. 添加模型素材显示或隐藏的交互

选择模型素材,单击“显示/隐藏”按钮,在弹出的“显示/隐藏”设置界面选择具体的功能设置,这里选中当前的3D模型名称(3DPlayer)。然后选择要执行的操作为“显示”或者“隐藏”,并且选择具体的触发条件,例如“扫描识别到”,如图5.17所示。

#### 5.1.5 场景保存与分享

上述步骤完成后,就可以保存场景并分享体验了。首先选择右上角的“场景结构”选项卡,确认场景中添加的素材节点(包括模型、音频、图片等)以及相关的控制设置(例如,音频控制、模型控制等)符合要求,如图5.18(a)所示。然后选择“场景设置”选项卡,确认当前场景的设置信息,或者对已经存在的设置信息进行重新修改,如图5.18(b)所示。最后单击“保存”按钮保存场景。



图 5.17 显示/隐藏设置



图 5.18 项目设置与保存

项目保存完毕之后,可以单击“分享”按钮进行项目的发布,如图5.19(a)所示。在场景发布界面,可以设置分享的Logo图片、分享的名称以及分享的描述信息。同时,还会自动生成分享二维码以及网址链接,如图5.19(b)所示。二维码可以直接扫描并进行体验,而网址链接可以在移动端或者固定端的浏览器中进行访问。就便捷性而言,二维码方式的分享

体验更好一些。



图 5.19 场景分享

最终的场景效果请读者结合本书配套资源中的素材资源,亲自动手实践进行体验。

## 5.2 基于 EasyAR 的 WebAR 应用开发

### 5.2.1 EasyAR WebAR 简介

EasyAR WebAR 是以 Web 平台来集成 AR 技术,区别于原生 AR 应用,具有模式轻、部署快、传播能力强等特点,可以轻松地运行在 Android、iOS、Windows、macOS 等系统的 Web 浏览器上,无需 App,真正实现跨平台。EasyAR WebAR 提供在 Web 端嵌入 AR 技术的整套服务和解决方案,后续也会向开发者发布 WebAR SDK。

EasyAR WebAR 由 Web 前端和 EasyAR 云服务两部分组成,支持平面图片识别、云识别、3D 渲染、复杂互动等功能,主要以 URL 的格式来传播,符合微信等社交媒体信息流动的基本技术要求,配合创意策划方案,可以形成爆炸式的营销效果。

### 5.2.2 EasyAR WebAR 快速入门

使用 EasyAR 开发 WebAR 应用之前,需要做好如下准备工作。

#### 1. EasyAR WebAR 授权

访问 EasyAR 官方网站(网址请参考本书配套资源),在图 5.20 右侧进行 EasyAR 账户的注册和登录,详细的操作步骤与 Kivcube 相同,此处不再赘述。

通过邮箱注册之后,可以在邮件中通过链接激活 EasyAR 账户,然后使用新账户登录就能够访问图 5.20 中的开发中心了。初次进入开发中心时,系统会要求补全用户信息,如图 5.21 所示,可以选择下次再填写,将当前窗口关闭。



图 5.20 EasyAR 官网首页

为了获得更好的产品与服务, 请您填写以下信息

---

\* 国家

\* 手机号

\* 验证码

\* 身份  企业  个人开发者  学生

\* 行业  文旅  教育  汽车  游戏/娱乐  工业  军事  
 医疗  营销/广告  其他

\* 您想使用或正在使用的EasyAR服务和功能

- EasyAR Sense (SDK)
  - 稀疏空间地图
  - 稠密空间地图
  - 运动跟踪
  - 表面跟踪
  - 3D物体跟踪
  - 平面图像跟踪
  - 录屏
- 云识别服务 (CRS)
- 云定位服务 (CLS)
- WebAR/小程序AR
- 姿势/手势识别服务
- EasyAR Mega
- 其他

图 5.21 EasyAR 官网服务使用情况调查

EasyAR 开发之前首先需要为项目申请 Sense 授权。EasyAR Sense 能够实现 Mega 云定位服务、AR 图像识别、稠密空间地图、稀疏空间地图、3D 物体跟踪、运动跟踪、平面图像跟踪、表面跟踪、录屏等多种功能。默认情况下, 开发中心是没有 Sense 授权的, 此时可以单

击“Sense 授权管理”下的“我需要一个新的 Sense 许可证密钥”按钮来申请新的授权,如图 5.22 所示。



图 5.22 Sense 授权申请

在展开的 Sense 订阅界面可以看出,支持的授权类型有 3 种,分别是个人版、专业版和经典版,如图 5.23 所示。其中专业版和经典版都是需要付费进行使用的,不同的是专业版是按月付费订阅,而经典版是一次性付费永久使用。这里的案例仅用于学习和研究,开发的应用并不用于商业目的,因此申请个人版免费授权即可。

#### 订阅Sense

<b>Sense类型</b>	EasyAR Sense 4.0
	<div style="border: 1px solid black; padding: 5px;"> <p><a href="#">查看Sense功能介绍</a></p> <p><input type="radio"/> EasyAR Sense 4.0 个人版 免费,不可商用,有水印</p> <p><input type="radio"/> EasyAR Sense 4.0 专业版 按月付费,可商用,无水印</p> <p><input type="radio"/> EasyAR Sense 4.0 经典版 一次性付费永久使用,可商用,去水印,包含专业版所有功能</p> </div>
<b>创建应用</b>	<p><b>应用名称</b> <input type="text" value="请输入应用名称"/> 可修改</p> <p><b>Bundle ID</b> iOS <input type="text" value="请输入Bundle ID"/> 可修改, iOS平台Sense License KEY需要与Bundle ID对应使用</p> <p><b>Package Name</b> Android <input type="text" value="请输入Package Name"/> 可修改, Android平台Sense License KEY需要与PackageName对应使用</p> <p><b>支持平台</b> <input checked="" type="checkbox"/> iOS <input checked="" type="checkbox"/> Android <input checked="" type="checkbox"/> Windows <input checked="" type="checkbox"/> macOS</p> <p><input type="button" value="确认"/></p>

图 5.23 Sense 授权类型

选择授权类型为个人版之后,选择所需要的授权功能。如果需要使用稀疏空间地图,则可以选择“是”,否则选择“否”即可。在“创建应用”中分别设置 AR 应用的名称以及在不同的移动端系统中的参数。这里如果是 iOS 系统,那么需要绑定 Bundle ID;如果是 Android 系统,那么需要填写打包时的 Package Name,这里将二者设置为一致即可。AR 应用的有

效期可以视情况而定,这里设置为无限期,即永久生效,然后单击“确认”按钮,就将应用的名称与 Sense 授权进行绑定了,如图 5.24 所示。

授权功能  稠密空间地图  3D物体跟踪  平面图像跟踪  支持云识别  Mega云定位服务  
 已默认选中不支持更改  稀疏空间地图  运动跟踪  表面跟踪  录屏

是否使用稀疏空间地图:  是  否 [查看稀疏空间地图功能介绍](#)

说明!  
 创建一个SpatialMap库以便使用EasyAR Sense的稀疏空间地图功能。  
 如果您现在还不确定是否会用到这个功能,也可以选择“否”,并在日后需要时再创建。

创建应用

应用名称   
 可修改

Bundle ID   
 iOS  
 可修改, iOS平台Sense License KEY需要与Bundle ID对应使用

Package Name   
 Android  
 可修改, Android平台Sense License KEY需要与PackageName对应使用

支持平台  iOS  Android  Windows  macOS

期限(月):

费用: ¥0元

图 5.24 申请个人版 Sense 授权

申请后的授权列表如图 5.25 所示。

应用名称	Sense类型	创建时间	操作
ARTest	4.0 个人版	2022-12-13 22:49	<a href="#">查看</a> <a href="#">升级</a> <a href="#">创建SpatialMap库</a>

图 5.25 申请后的授权列表

如果开发的 EasyAR 应用是微信小程序,那么接下来在授权管理中,可以创建 Mega 微信小程序许可证密钥。这里选择授权类型为测试版,在“创建应用”中需要填写小程序的名称以及小程序的 APP ID,然后单击“确认”按钮,如图 5.26 所示。

如果要使用 EasyAR 云服务,那么首先需要创建一个云服务,这里选择“云识别管理”,然后创建一个图库,名为 ARImages,如图 5.27 所示。图库分为按调用次数计费和按日活计费两种方式,这里选择默认的“按调用次数”进行计费,并且可以默认拥有 500 次/日的试用次数。

订阅小程序授权管理

---

**Licence类型** EasyAR Mega 微信小程序1.0

[查看小程序功能介绍](#)

EasyAR Mega微信小程序测试版  
开发测试专用，不可商用，有水印，仅能创建一个

EasyAR Mega微信小程序专业版  
按年付费，可商用，无水印

EasyAR Mega微信小程序经典版  
一次性付费永久使用，可商用，去水印，包含专业版所有功能

**授权功能**  Mega云定位服务

**创建应用**

**小程序名称**   
可修改

**APP ID**   
可修改，小程序License KEY需要与APP ID对应使用

**支持平台**  微信小程序

**期限 (年) :**

**费用:**

图 5.26 申请 Mega 微信小程序授权

接下来需要申请云服务 API Key,如图 5.28 所示,将应用名称与云服务的类型进行绑定。这里创建名为 ARTTest 的应用,并且选择云服务类型为“云识别”。

## 2. Web 服务器

用于存储 HTML 等静态网页内容。可以通过部署 Windows 下的 IIS 服务器或者是在 Linux 中安装 Web 服务器来提供静态网页内容的访问。

## 3. 支持 HTTPS 的域名

在浏览器上使用摄像头,需要 HTTPS 协议支持。

具体的开发步骤可以通过如下代码实现,基本的代码还是遵循了 HTML + CSS + JavaScript 的编程风格。



图 5.27 创建云识别图库



图 5.28 申请云识别 API Key

## 1) 打开摄像头

```

1. // 更多参数请查看
2. //https://developer.mozilla.org/en-US/docs/Web/API/MediaStreamConstraints
3. const constraints = {
4.   audio: false,
5.   video: {facingMode: {exact: 'environment'}}
6.   //video: {facingMode: {exact: 'user'}}
7.   //video: true
8. };
9.
10. navigator.mediaDevices.getUserMedia(constraints)
11.   .then((stream) => {
12.     // videoElement 为 video 元素,将摄像头视频流绑定到 video 上实时预览
13.     videoElement.srcObject = stream;
14.     videoElement.style.display = 'block';
15.     videoElement.play();
16.     resolve(true);
17.   })
18.   .catch((err) => {
19.     reject(err);
20.   });

```

## 2) 截取摄像头图像

```

1. // canvasElement 为 canvas 元素
2. // canvasContext 为 canvas 的 context 2D 对象
3. // videoSetting 为 video 元素的宽、高
4. canvasContext.drawImage(videoElement, 0, 0, videoSetting.width, videoSetting.height);
5. const image = canvasElement.toDataURL('image/jpeg', 0.5).split('base64,')[1];

```

## 3) 将图像发送到服务器识别

在 EasyAR 开发中心中选择“授权”→“云服务 API Key”，查看云服务 API Key，如图 5.29 所示。通过上述操作已经为 ARTest 应用申请了 API Key，此时单击后面的“管理”按钮。



应用名称	API Key	API Secret	创建时间	操作
ARTest	03c0a88c43271243b7765e3e90e648 复制	c51695755907289c6a69a420396922 bd0d9a5d4dc21a5a97a2e7385c65782 复制 d2	2022-12-20 16:26	管理

图 5.29 API Key 列表

选择 Token 有效期的天数后，单击“生成 Token”按钮，如图 5.30 所示。可以看出，生成的 Token 能够支持为期 5 天的云识别服务，并且将新建的图库名称进行了绑定。



图 5.30 生成 Token

Token 会过期,支持动态刷新。Token 是写在 JavaScript 文件中的,适合在安全要求较低的环境中使用,建议调用接口动态获取 Token。

在“云服务”的“云识别管理”中,可以看到之前申请的云识别图库,如图 5.31 所示。



图 5.31 云识别图库列表

单击该图库后面的“管理”按钮,可以查看云识别库的相关信息,如图 5.32 所示。其中,在“识别图”标签中,可以看到当前没有识别图,可根据具体的应用开发,单击“上传识别图”按钮,将需要识别的图片进行上传。



图 5.32 云识别库详细信息

然后在“密钥”标签中可以看到实现识别所需要的相关密钥信息,如图 5.33 所示,包括 CRS AppId、API KEY 以及 Cloud URLs 三项内容。

**云识别库信息**

图库名称: ARImages 修改

图库类型: 按调用次数      图库大小: 100000      使用率: 0/100000

区域: 中国一区      创建日期: 2022-12-20      到期日期: 2023-01-04 续费

计费方式: 500次/日(试用)      超限处理方式: 超出后暂停服务

识别图    **密钥**    统计分析

Sense4.1及其之后版本使用    Sense3.1.0至4.0.1版本使用    Sense3.0.1及其之前版本使用    WebAR/小程序AR使用

**注意!**  
基于Sense3.1及之后版本(包括Sense4.x)使用云识别服务时,只支持APPID+API KEY认证机制,不再使用Key+Secret的认证机制。

**CRS AppId**    8d8b15af3cb34e4ae1e8c65eae800e05    复制

**API KEY:**     (若列表中没有可用API KEY,您可以创建一个支持CRS服务的API KEY,查看API KEY的使用文档)

**Cloud URLs**    Client-end (Target Recognition) URL: https://a855f6a5a737d5dbec69d672ab1c3923.cn1.crs.easyar.com:8443    复制

图 5.33 云识别库密钥

单击图 5.33 中的“API KEY 列表”按钮,弹出如图 5.34 所示的基于应用名称的 API Key 列表,然后根据实际情况进行选择。

**创建API KEY**

注意! 您可以使用API KEY调用相关云服务。APIKey 文档

应用名称	API Key	API Secret	创建时间	操作
ARTest	03c0a88c43271243b7765e3fe90e848 复制	c51695755907289cfa69a420396922 bd0d9a5d4dc21a5a97a2e7385c65782 复制 d2	2022-12-20 18:26	管理

图 5.34 API Key 列表

接下来,将获取的上述 3 项内容,补充到如下代码中,就能够完成识别代码的编写。

```

1. // 云图库的 Client - end URL
2. const clientendUrl = '您云图库的 Client - end URL/search';
3. // 云图库的 Cloud Token
4. const token = '这里是云图库的 Cloud Token';
5. // 云图库的 CRS AppId
6. const appId = '这里是云图库的 CRS AppId';
7.
8. // 可以使用 jQuery 或 axios 等发送网络请求
9. const http = new XMLHttpRequest();
10. http.onload = () => {
11.     try {

```

```
12.     const msg = JSON.parse(http.responseText);
13.     if (http.status === 200) {
14.         if (msg.statusCode === 0) {
15.             resolve(msg.result);
16.         } else {
17.             reject(msg);
18.         }
19.     } else {
20.         reject(msg);
21.     }
22. } catch (err) {
23.     reject(err);
24. }
25. };
26. http.onerror = (err) => {
27.     reject(err);
28. };
29.
30. // 发送到云图库服务器
31. http.open('POST', clientendUrl);
32. http.setRequestHeader('Content - Type', 'application/json;Charset = UTF - 8');
33.
34. // 将 Cloud Token 写在请求头中
35. http.setRequestHeader('Authorization', token);
36.
37. // image 为截取的摄像头图片数据,如:{image: '/9j/4AAQSkZJRgA...', appId: appId}
38. http.send(JSON.stringify(image));
```

#### 4. 检测识别结果

如果未识别到内容,则继续识别,否则停止识别,将识别的结果(如 targetId、meta 等)信息取出处理。通过 URL“https://云图库的客户端 URL/search”进行目标识别时,若未识别到目标,则 HTTP 状态码会变为 404。

### 5.2.3 EasyAR Web3D 模型动画要求

EasyAR 能够提供优秀的 3D 模型展示,通过 Web 方式展示 3D 模型动画,比平面 HTML5 页面具有更强的感染力。例如,在电商活动中,通过 Web 3D 方式展示产品模型,为用户提供直观感受产品的视觉入口,增强活动传播效果。EasyAR 对于 3D 模型的展示具有如下要求。

- (1) 动画模型需要单个 mesh(可编辑网格),当模型没有动画时无此要求。
- (2) 动画模型骨骼只可以有一套,骨骼权重要和骨骼保持一致,蒙皮权重必须要完整,可以限制网格顶点,受控制骨骼数目最好不超过 4 个。
- (3) 骨骼层级要遵循命名规范,目的在于当动画出现问题时可以追溯。
- (4) 避免使用曲线网格。

(5) 动画模型纹理需要使用 JPG/PNG 格式,不能使用 psd 作为纹理源文件,材质最好不使用透明通道,若使用透明通道则需要进行代码调试。

(6) Three.js 支持的动画文件格式只有 FBX、DAE 和 JSON 三种,FBX 格式包括 ASCII 和二进制两种文件类型,最好可以获得源工程文件,在使用 3dsMax/Maya/Blender 导入导出时有不兼容的地方需要手动调整。

(7) 模型面数限制在 10000 以下。

(8) 在动画模型中为了方便绘制角色的运动轨迹,通常会为角色添加一个根节点,这个节点没有对应信息的绑定,会导致导出模型的初始位置出错。这里需要找到根节点并选择根节点的下级节点,再断开其与根节点的链接。

(9) 不支持动画中存在虚拟节点和控制器。

(10) 所有节点不能有负数。

(11) 模型缩放为 1,模型只能是一个 mesh,单个模型独立对应一种纹理并独立对应一套骨骼。对于绑定过的模型,不要二次返修。

(12) 模型制作使用 polygon 多边形建模,避免使用 NURBS 等其他类型建模。制作完成的模型统一导出格式为 FBX,导出之前需要将模型历史记录及无关节点清空,模型坐标位于坐标点中心,并将坐标数据清零,冻结变换。

### 5.3 基于开源的 AR.js 应用开发

AR.js 是一个用于网络增强现实(WebAR)的轻量级库,具有图像跟踪、基于位置的 AR 和基于标记的 AR 等功能,它是完全开源的,非常适合平时做科研使用。图 5.35 展示了 AR.js 的一些使用案例。

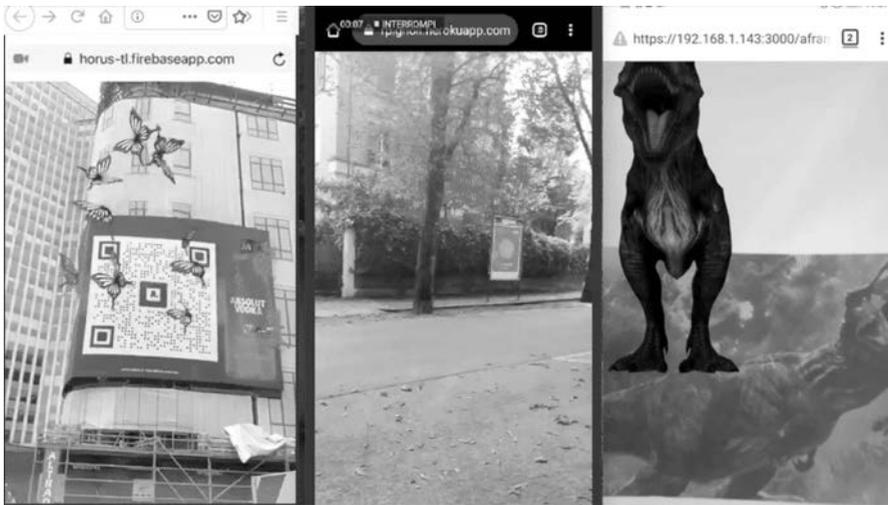


图 5.35 AR.js 使用案例

该项目现在属于 Github 组织,可以在 AR.js 官网(网址请参考本书配套资源)找到并免费申请加入。下面介绍 AR.js 的三大功能。

### 1. 图像跟踪

当相机发现 2D 图像时,可以在其顶部或附近显示某种内容。其内容可以是 2D 图像、GIF、3D 模型(也是动画)和 2D 视频,应用范围包括增强艺术、学习(增强型书籍)、增强型传单或广告等。

### 2. 基于位置的 AR

这种 AR 使用真实世界的地理位置,以便在用户设备上显示增强现实内容。用户可以移动(最好是户外),并且可以通过智能手机看到现实世界中的 AR 内容。四处移动和旋转手机将使 AR 内容根据用户的位置和旋转而变化(因此地点被“锚定”在其真实位置,并根据它们与用户距离的远近而变小或变大)。借助此解决方案,可以构建诸如导游的交互式支持、探索新城市时的帮助、查找建筑物/博物馆/餐馆/酒店等体验;还可以构建诸如寻宝、生物学或历史学习游戏等体验,或者将这种技术用于情境艺术(与特定现实世界坐标绑定的视觉艺术体验)。

### 3. 基于标记的 AR

当相机找到标记时,可以显示一些内容(与图像跟踪相同)。标记非常稳定,但形状、颜色和大小有限,使用示例包括增强型书籍、增强型传单或广告等。

## 5.3.1 图像跟踪案例

创建新的项目并添加如下代码:

```
1. <script src = "https://cdn.jsdelivr.net/gh/aframevr/aframe@1c2407b26c61958baa93967b54124
87cd94b290b/dist/aframe - master.min.js">
2. </script>
3. <script src = "https://raw.githubusercontent.com/AR - js - org/AR.js/master/aframe/build/aframe -
ar - nft.js">
4. </script>
5. <style>
6.   .arjs - loader {
7.     height: 100%;
8.     width: 100%;
9.     position: absolute;
10.    top: 0;
11.    left: 0;
12.    background-color: rgba(0, 0, 0, 0.8);
13.    z-index: 9999;
14.    display: flex;
15.    justify-content: center;
16.    align-items: center;
17.  }
18.
19.  .arjs - loader div {
```

```

20.     text-align: center;
21.     font-size: 1.25em;
22.     color: white;
23.   }
24. </style>
25.
26. <body style="margin: 0px; overflow: hidden;">
27.   <!-- 在加载图像描述符之前显示的最小加载程序 -->
28.   <div class="arjs-loader">
29.     <div>Loading, please wait...</div>
30.   </div>
31.   <a-scene
32.     vr-mode-ui="enabled: false;"
33.     renderer="logarithmicDepthBuffer: true;"
34.     embedded
35.     arjs="trackingMethod: best; sourceType: webcam; debugUIEnabled: false;"
36.   >
37.     <!-- 使用 cors 代理来避免跨源问题 -->
38.     <a-nft
39.       type="nft"
40.       url="https://arjs-cors-proxy.herokuapp.com/https://raw.githubusercontent.com/AR-js-org/AR.js/master/aframe/examples/image-tracking/nft/trex/trex-image/trex"
41.       smooth="true"
42.       smoothCount="10"
43.       smoothTolerance=".01"
44.       smoothThreshold="5"
45.     >
46.       <a-entity
47.         gltf-model="https://arjs-cors-proxy.herokuapp.com/https://raw.githubusercontent.com/AR-js-org/AR.js/master/aframe/examples/image-tracking/nft/trex/scene.gltf"
48.         scale="5 5 5"
49.         position="50 150 0"
50.       >
51.         </a-entity>
52.       </a-nft>
53.     <a-entity camera></a-entity>
54.   </a-scene>
55. </body>

```

将上述代码部署至已经搭建好的 Web 服务器中,可以使用基于 Windows 的 IIS 服务器或者是基于 Linux 的 Apache/Nginx 服务器。

在手机浏览器中访问 Web 服务器提供的 URL,对上述配置的站点进行访问。

扫描如图 5.36 所示的图像,就能够在手机中体验 AR 应用了。



图 5.36 AR 效果实验图(一)

### 5.3.2 基于位置的 AR 案例

使用以下代码创建一个项目,将第 22 行代码中的 `add-your-latitude` 和 `add-your-longitude` 属性更改为实际的纬度和经度,注意不带“<>”。

```
1. <!DOCTYPE html >
2. <html >
3.   <head >
4.     <meta charset = "utf - 8" />
5.     <meta http - equiv = "X - UA - Compatible" content = "IE = edge" />
6.     <title> GeoAR. js demo </title >
7.     <script src = "https://aframe.io/releases/1.0.4/aframe.min.js"></script >
8.     <script src = "https://unpkg.com/aframe - look - at - component@0.8.0/dist/aframe -
look - at - component.min.js"></script >
9.     <script src = "https://raw.githack.com/AR - js - org/AR. js/master/aframe/build/
aframe - ar - nft.js"></script >
10.   </head >
11.
12.   <body style = "margin: 0; overflow: hidden;" >
13.     <a - scene
14.       vr - mode - ui = "enabled: false"
15.       embedded
16.       arjs = "sourceType: webcam; debugUIEnabled: false;"
17.     >
18.       <a - text
19.         value = "This content will always face you. "
20.         look - at = "[gps - camera]"
21.         scale = "120 120 120"
22.         gps - entity - place = "latitude: < add - your - latitude >; longitude: < add - your -
longitude >;"
23.       ></a - text >
24.       <a - camera gps - camera rotation - reader ></a - camera >
25.     </a - scene >
26.   </body >
27. </html >
```

将上述代码部署至已经搭建好的 Web 服务器中,可以使用基于 Windows 的 IIS 服务器或者是基于 Linux 的 Apache/Nginx 服务器。

在手机上激活 GPS 并导航到示例 URL。即使环顾四周并移动手机,也应该看到文本正对着用户,出现在请求的位置。

### 5.3.3 基于标记的 AR 案例

使用下列代码创建项目:

```
1. <!DOCTYPE html >
2. <html >
```

```
3. <script src = "https://aframe.io/releases/1.0.4/aframe.min.js"></script>
4. <!-- 导入 ARjs 版本, 不带 NFT, 但支持标记 + 位置 -->
5. <script src = "https://raw.githubusercontent.com/AR-js-org/AR.js/master/aframe/build/aframe-ar.js"></script>
6. <body style = "margin : 0px; overflow: hidden;">
7.   <a - scene embedded arjs >
8.     <a - marker preset = "hiro">
9.       <a - entity
10.         position = "0 0 0"
11.         scale = "0.05 0.05 0.05"
12.         gltf - model = "https://arjs - cors - proxy. herokuapp. com/https://raw. github. com/AR - js - org/AR. js/master/aframe/examples/image - tracking/nft/trex/scene. gltf"
13.       ></a - entity >
14.     </a - marker >
15.   <a - entity camera ></a - entity >
16. </a - scene >
17. </body >
18. </html >
```

将上述代码部署至已经搭建好的 Web 服务器中,可以使用基于 Windows 的 IIS 服务器或者是基于 Linux 的 Apache/Nginx 服务器。

在手机浏览器中访问 Web 服务器提供的 URL 网址,对上述配置的站点进行访问。扫描如图 5.37 所示的图像,就能够在手机中体验 AR 应用了。



图 5.37 AR 效果实验图(二)