

PhaserH5试玩游戏

1 简介

Phaser是一个HTML5游戏框架，它的目的是辅助开发者真正快速的制作强大的、跨浏览器的HTML5游戏

- 优点：功能强大
- 缺点：平台对试玩游戏有空间大小限制，素材过多将超过平台限制，而phaser api 占用1MB，可以考虑使用原生js开发

2 试玩小游戏制作流程

创意→素材制作→确定实际游戏中素材大小(压缩图片，音频，视频)→编码→压缩代码

所需基础：css, js，编码过程中参考官方示例和查阅文档

资源加载具有跨域问题，需先将资源文件转换成base64数据，再读出使用

3 API对接

3.1 抖音头条

- 1将下面代码放入 body 中，且在开发者自己的 js 代码之前
- 2调用window.playableSDK.openAppStore();

3. 文件传输说明

最终文件请以zip的格式传输给AD后台。

zip文件请按以下素材规格需求 list 检验通过后，再上传广告平台

1. 试玩素材应为 zip 格式
2. 试玩广告素材大小应小于 3 MB（整包压缩后大小）
3. 一级目录中主 html 文件，请命名为 index.html
4. 试玩广告需将点击跳转操作交由js-sdk完成，所以需要正确接入前端 js-sdk
5. 一级目录里包含 config.json 文件，文件内容为 json 格式，包含 playable_orientation 字段，代表试玩素材的横竖适配，取值为 0、1、2，含义：0（横竖均可），1（只支持竖屏），2（只支持横屏）{"playable_orientation": 0}
6. 建议把素材拆分小块，支持加速并行下载
7. 试玩广告素材需要支持多种设备类型
8. 试玩广告素材不应使用 mraid.js 格式
9. 不允许通过外部网络加载动态素材
10. 试玩 HTML5 素材中不应包含任何 JS 重定向
11. 试玩广告素材不应发出任何 HTTP 请求
12. 试玩素材不支持存在内容为空的文件
13. 文件名仅支持大小写字母、数字、减号和下划线

3.2 facebook

试玩 HTML5 素材

- 试玩游戏应为 HTML5 格式（扩展名为“html”或“htm”）。
- 文件大小应小于 2 MB。
- 试玩广告素材不应使用 mraid.js 格式。
- 试玩广告素材不应发出任何 HTTP 请求，例如：“XMLHttpRequest”。
- 试玩广告素材应为纵向模式。
- 试玩广告素材应采用响应式设计，因为它需要支持多种设备类型，而设备的分辨率各不相同。
- 包含所有素材的试玩广告的单个或多个文件都可作为单个 HTML 文件和素材的一部分，而且应采用 URI 压缩数据格式（Javascript、CSS、图片、声音）。
- 试玩广告代码应在用户与广告中的行动号召互动时使用 Javascript 函数 `FbPlayableAd.onCTAClick()`。Facebook 使用该代码以导航至应用正确的应用商店页面。
- 试玩 HTML5 素材中不应包含任何 JavaScript 重定向。
- 不允许通过外部网络加载动态素材。
- 如要合并为一个文件：将图像编码为 base64 字符串，将 js 集成到 `index.html` 中。

压缩文件

- 当前，所有 iOS 版和 Android 版广告版位均支持压缩文件格式的试玩广告。
- 压缩归档文件应包含位于根目录的 `index.html`，例如：

```
./index.html
```

注意：只要是相对于 `index.html` 来引用资源，资源可位于目录结构中的任意位置。例如，以下文件：

```
./assets/splash.png
```

必须引用为：

```
<image src="assets/splash.png"/>
```

- 压缩归档文件中的所有 html 文件和 JavaScript 文件应存在相同的限制，但只需在一个位置调用 `FbPlayableAd.onCTAClick()`。
- 上传文件的总大小不得超过 5MB。
- 捆绑包内的文件数量不得超过 100 个。
- `index.html` 文件的大小不得超过 2MB。

- 只需添加 `FbPlayableAd.onCTAClick();`

3.3 google

HTML5/试玩广告尺寸和格式

在制作应用广告系列之前，请参考以下有关制作 HTML5 素材资源的提示：

- Google Ads 建议您使用 [Google Web Designer](#) 来制作 HTML5 素材资源。
- 目前接受的 HTML5 尺寸包括：
 - **推荐尺寸**（您的素材资源在大多数设备上都以全屏广告的形式显示）
 - 320x480（纵向）
 - 480x320（横向）
 - **其他尺寸**（您的素材资源将显示为固定尺寸的横幅广告，不会根据各种屏幕尺寸加以调整）
 - 300x250
 - 320x50
- 要在上传前验证 HTML5 素材资源，请通过 [Google Ads HTML5 验证工具](#) 运行 .ZIP 文件。
- 以 .ZIP 文件的格式上传素材资源，最大不超过 1MB，文件数量不超过 40 个。
- 您最多可以为每个广告系列上传 20 个 .ZIP 文件。
- .ZIP 文件可以包含以下格式：HTML、CSS、JS、GIF、PNG、JPG、JPEG、SVG。
- 必须对非 ASCII 字符使用 UTF-8 编码。
- “声音”选项不适用于应用广告系列的 HTML5 广告。

HTML 要求	<p>HTML5 素材资源必须包含：</p> <ul style="list-style-type: none">• <!DOCTYPE html> 声明• <html> 标记• <body> 标记• <head> 标记中的广告格式尺寸元标记。例如：<meta name="ad.size" content="width=300,height=250"> <p>明确的结束标记（不接受内嵌的结束标记）：</p> <ul style="list-style-type: none">• 有效的结束标记示例：<path></path>• 无效的结束标记示例：<path> 或 <path /> <p>或者，您可以将内嵌的 svg 提取到单独的文件中。</p>
上传要求	<p>所有代码和素材资源都必须使用指向 .ZIP 文件中所含资源的相对路径进行引用。</p> <p>除对以下对象的引用外，不得使用任何外部引用：</p> <ul style="list-style-type: none">• Google Fonts• Google 托管的 jQuery• Google 托管的 Greensock（文件应可正常使用，无论版本如何）<ul style="list-style-type: none">• tweenlite• tweenmax• cssplugin• easepack• timelinelite• timelinemax• Google 托管的 CreateJS<ul style="list-style-type: none">• tweenjs• easeljs• createjs
可点击的按钮	<p>如果您想确保用户只能点击您的号召性用语按钮（例如“下载”），请按照以下说明添加 exitapi.js 脚本。如果您不添加 exitapi.js 脚本，Google Ads 会将您的整个素材资源都设为可点击。</p> <p>操作说明</p> <ol style="list-style-type: none">1. 请在 HTML <head> 标记中添加以下脚本：<script type="text/javascript" src="https://tpc.googlesyndication.com/pagead/gadgets/html5/api/exitapi.js"></script>2. 然后执行以下 JavaScript 调用以启用最终到达网址：ExitApi.exit()<ol style="list-style-type: none">a. 例如，要将用户引导至最终到达网址，请使用标准锚标记：<ol style="list-style-type: none">i. 了解详情

- 1.HTML 标记中添加以下脚本
- 2.然后执行以下 JavaScript 调用以启用最终到达网址：ExitApi.exit()

4 案例模板

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <title>示例模板</title>
  <meta name="viewport" content="width=device-width,initial-scale=1, minimum-scale=1, maximum-scale=1 user-scalable=no"/>
  <meta name="apple-mobile-web-app-capable" content="yes"/>
  <meta name="full-screen" content="true"/>
  <meta name="screen-orientation" content="portrait"/>
  <meta name="x5-fullscreen" content="true"/>
```

```

<meta name="360-fullscreen" content="true"/>
<style type="text/css">
    body{
        width: 100%;
        height: 100%;
        margin: 0;
        background-color: rgb(0, 0, 0);
    }
    .parent{
        width: 100%;
        height: 100%;
        position: absolute;
        z-index: 0;
        background-color: rgb(0, 0, 0);
    }
    /* view1 */
    #logoStart{
        height: 16%;
        margin-top: 50%;
        z-index: 50;
        position: relative;
        display: block;
        margin-bottom: 2%;
    }
    /* view2 */
    #phaser-example{
        width: 100%;
        height: 100%;
        position: absolute;
        z-index: 2;
        display: none;
    }
    #view2{
        display: none;
    }
</style>
<!-- 资源base64文件,如:
var data_asset01 = '...'; -->
<script src="js/assetdata.js"></script>
<!-- phaser 本地api, 下载链接: http://phaser.io/download/stable -->
<script src="js/phaser.min.js"></script>
<!-- 3.使用google api时,需用添加下方脚本 -->
<!-- <scripttype="text/javascript"
src="https://tpc.googlesyndication.com/pagead/gadgets/html5/api/exitapi.js"> </script> -->
</head>
<!-- 1.使用抖音头条api时,需用添加下方脚本 -->
<!-- <script src="https://sf3-ttcdn-tos.pstatp.com/obj/union-fe-nc/playable/sdk/playable-
sdk.js" data-tomark-pass data-tomark-pass ></script> -->

<body onload="myload()">
    <div class='parent' id='view1' align="center">
        <img id='logoStart'>
    </div>
    <div class='parent' id='view2' align="center">
        <div id='phaser-example'></div>
    </div>
    <script type="text/javascript">
        function myload(){
            console.log('加载页面');
            //示例: js设置图片
            document.getElementById("logoStart").setAttribute("src", data_asset01);
        }
        function callAdFuc() {
            try {
                window.playableSDK.openAppStore();//1.调用抖音头条api(还需再index.html同目录下添加
config.json,json内容添加{"playable_orientation": 0 }", 0横竖)
                // FbPlayableAd.onCTAClick(); //2.facebook api
                // ExitApi.exit(); //3.google api
            } catch (error) {

```

```

        console.log(error);
    }
}

//##region boot 资源加载
class Boot extends Phaser.Scene{
    constructor(){
        super('boot');
    }
    preload(){
        //示例: 加载视频资源
        this.load.video('video1', data_video1, 'loadeddata', false, false);
    }
    create(){
        //添加初始化事件监听, 只执行一次
        this.events.once('initial', this.initial, this);
        this.loadAssets();
    }
    update(){
        //监听资源加载完成, 执行初始化
        if(this.loadIndex < this.loadNums){
            return;
        }else{
            //加载完成后, 执行初始化
            this.events.emit('initial');
        }
    }
}

/**
 * 总加载图片和音频资源
 */
loadAssets(){
    this.loadIndex = 0;
    this.loadNums = 2;//下面有几条加载就是几
    this.loadAsset_Audios();//加载所有音频
    this.loadAsset_Image(data_asset01, 'asset01');//加载图片
    this.loadAsset_Image(data_asset02, 'asset02');//加载图片
}

/**
 * 加载资源--图片
 * @param {*} imgData base64数据图片
 * @param {*} key 创建名为key的texture>>>获取this.textures.get(key); 创建this[key+_dom']的HtmlImageDom节点
 */
loadAsset_Image(imgData, key, spriteSheet = 'false'){
    let img = new Image();
    img.src = imgData;
    //图片加载完成执行
    img.onload = ()=>{//图片加载完成执行
        //添加Texture
        this.textures.addImage(key, img);
        //添加HTMLImageElement
        if(spriteSheet){
            this[key + '_dom'] = img;
        }
        ++this.loadIndex; //资源加载完成数+1
    };
}

/**
 * 加载资源--音频
 */
loadAsset_Audio(data, key){
    this.sound.once('decoded', ()=>{//音频加载完成执行
        ++this.loadIndex;//资源加载完成数+1
    });
    this.sound.decodeAudio(key, data);
}

loadAsset_Audios(){
    let audioFiles = new Array(
        { key: 'audio_01', data: data_asset_audio01 },

```

```

        { key: 'audio_02', data: data_asset_audio02 }
    );
    this.sound.once('decodedall', ()=>{//音频加载完成执行
        ++this.loadIndex;//资源加载完成数+1
    });
    this.sound.decodeAudio(audioFiles);
}
/**
 * 资源加载完成后执行一次
 */
initial(){
    this.time.addEvent({ delay: 200, callback: ()=>{
        console.log('-----loadAsset success!-----');
        this.scene.start('game');//关卡跳转
    }, callbackScope: this, repeat: 0 });
}
}
}
//#endregion

//#region game
class Game extends Phaser.Scene{
    constructor(){
        super('game');
    }
    create(){
        console.log("gaming");
        document.getElementById("view1").style['display']= 'none';
        document.getElementById("view2").style['display']= 'block';
        document.getElementById("phaser-example").style['display']= 'block';

    }
}
//#endregion

//#region main
var config = {
    type: Phaser.AUTO,
    backgroundColor: '#0x000000',
    scale: {
        mode: Phaser.Scale.FIT,
        parent: 'phaser-example',
        width: 900,
        height: 1600
    },
    scene:[Boot, Game]
};
var game = new Phaser.Game(config);
//#endregion
</script>
</body>
</html>

```

5 资源链接

- 资源转化base64: <https://base64.us/>
- js代码压缩: <https://jsmin.bmcx.com/>
- 官方文档: <https://photonstorm.github.io/phaser3-docs/>
- 官方案例: <http://phaser.io/examples>

