Unity转快游戏使用说明文档

转换工具链接

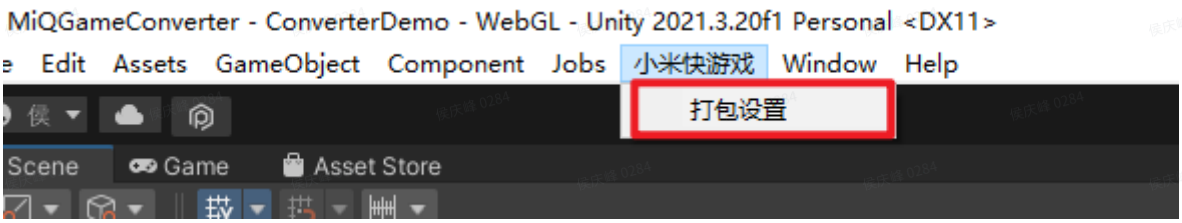
http://tennews.cn/MiQGameConverter.unitypackage

支持的unity版本**2021、2020，2019**。

结构：

使用步骤

1. 点击菜单栏 小米快游戏/打包设置



2. 弹出操作窗口

参数说明：

a. 游戏名称：小米快游戏名称。

b. 包名：在小米快游戏中心申请的包名，一定要填写正确的包名，否则会导致加载失败。

c. 版本号：版本号。

d. 版本code：版本code。

e. 输出目录：打包输出目录，此目录每次打包都会清空，请注意选择合适的目录。

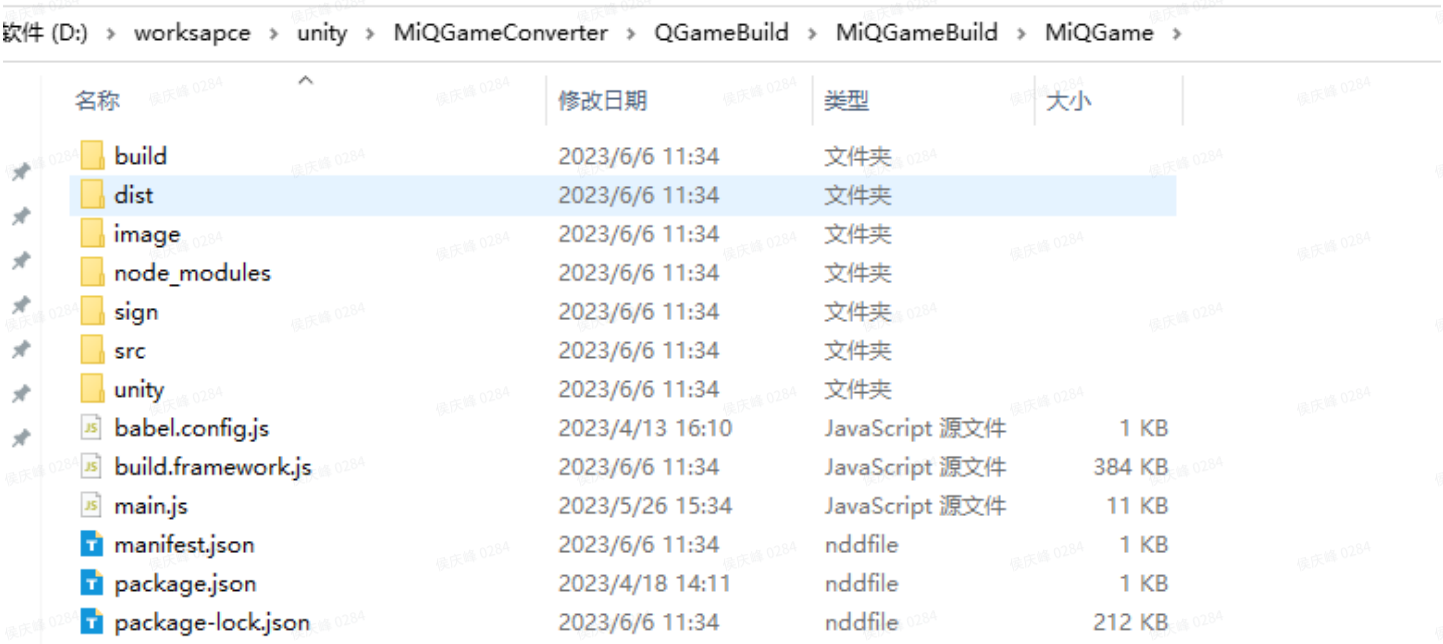
f. 游戏方向：横屏或者竖屏。

g. 包类型：debug或者release包。release包需要release签名文件，请设定release签名文件所在的目录。

h. 包体超限：由于快游戏包体大小限定，unity游戏资源较多的情况下，可以使用此项，把打出来资源放在服务器上，使用远程资源加载。

i. 打包：执行打包操作。

3. 打包完成会打开打包目录。如下所示：



MQGame即为**小米快游戏工程**。

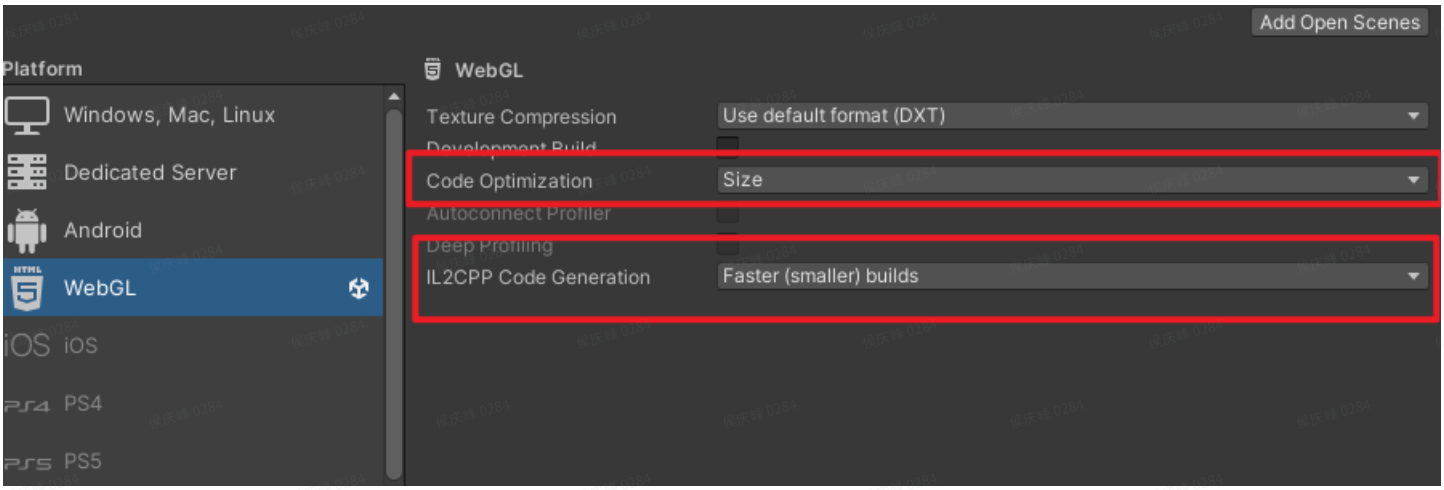
转换完成会在dist目录下生成快游包：

在此目录下，也可以手动执行npm命令生成快游戏包。

1. npm install命令安装快游戏依赖包；
2. npm run build 或者npm run release生成对应的debug包和release包。
3. npm run server，可以生成启动调试服务器，使用快游戏调试器（下载地址：https://www.quickapp.cn/docCenter/post/69），扫描二维安装即可启动游戏。注意：需要把调试器环境切换为：com.miui.hybrid环境！！！

优化建议

1. 包体过大，或者遇见out of memory，可以把这两两项改为如下选项，优化包体和内存。

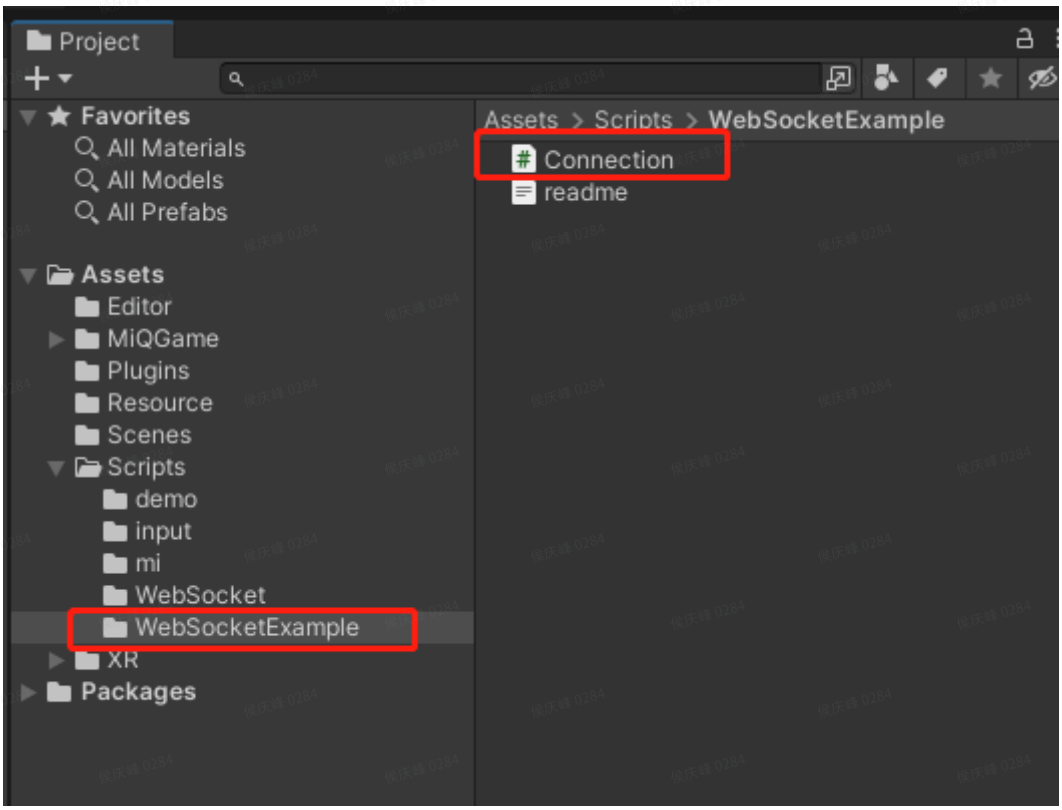


注意！！！Unity基础功能使用说明

unity转快游戏，有些unity功能是不能直接在快游戏里面使用的，需要在unity层使用的时候做一些适配工作。

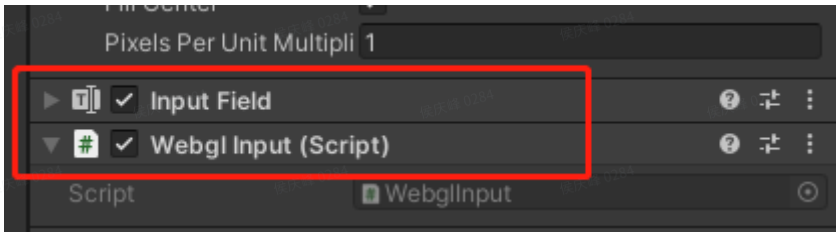
1. Websocket（可选）

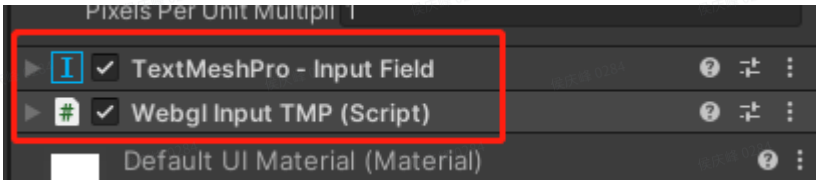
示例：WebSocket服务器搭建完成后，在客户端场景物体上挂载示例脚本Connection，并修改Connection脚本中的IP地址即可。



2. InputField（必须）

在挂载输入框组件的物体上，需手动挂载一个脚本。 如果使用的是 InputField (Legacy) 组件则手动挂载 WebglInput 脚本，使用的是 InputField (TMP)组件则手动挂载 WebglInputTMP 脚本。





快游戏功能接入

转换工具提供的快游戏功能有：登录、支付、广告、文件读写、创建游戏图标，key-value本地存储等功能。

1、 初始化

初始化需要在调用sdk功能之前调用，确保初始化完成，才能使用SDK功能。

例子：

```
//初始化
MiBridge.Instance.Init();
```

2、 登录和获取用户信息

```
//快游戏小米账号登录
MiBridge.Instance.Login((accountId, session) => {
    MiBridge.Instance.QGLog("login account id={0}, session={1}", accountId, sess
    MiBridge.Instance.GetUserInfo((nickName, avatarUrl, gender) =>
    {
        MiBridge.Instance.QGLog("get user info nickName={0}, avatar={1}, gender=
    }, (code, msg) =>
    {
        MiBridge.Instance.QGLog("get user info error code={0}, msg={1}", code, m
    });
}, (code, msg) => {
    MiBridge.Instance.QGLog("login error code={0}, msg={1}", code, msg);
});
```

3、支付

支付接入需要服务器接入，具体服务器接入方式请查看小米开发者文档：https://dev.mi.com/distribute/doc/details?pId=1109。

```
//订单信息
OrderInfo orderInfo = new OrderInfo
{
    appId = "1",                // 游戏唯一ID
    appAccountId = 12,          // 与登录接口返回的appAccountId一致
    session = "1",              // 与登录接口返回的session一致
    cpOrderId = "1",            // 游戏订单号
    cpUserInfo = "",            // cp透传信息（非空）
    displayName = "1",          // 支付的时候显示的商品名称
    feeValue = 100,             // 价格 单位分
    sign = "1",                 // 签名 用于校验 具体生成方式，请查看：https://dev.mi
};

//支付接口
MiBridge.Instance.Pay(orderInfo, (success, code, msg) => {
    MiBridge.Instance.QGLog("pay result success={0}, code={1}, msg={2}", success
});
```

4、广告

4.1、Banner广告

```
/// <summary>
/// 创建banner广告
/// </summary>
/// <param name="adId">广告id</param>
/// <param name="listener">监听器</param>
/// <returns></returns>
 MiBridge.Instance.CreateBannerAd("81e6cbe35e56b53eebbc547fd1bc5614", new AdEven
    onAdError = (code, msg) => {
        MiBridge.Instance.QGLog("c# ad error");
    },

    onAdClose = (isEnd) => {
        MiBridge.Instance.QGLog("c# ad close");
    },

    onAdLoad = (info) => {
        MiBridge.Instance.QGLog("c# ad load");
    },
});
```

```
20
21 /// <summary>
22 /// 隐藏广告
23 /// </summary>
24 /// <param name="adId"></param>
25 MiBridge.Instance.HideAd("81e6cbe35e56b53eebbc547fd1bc5614");
26
27 /// <summary>
28 /// 销毁广告，销毁后，如需重新显示，请先调用创建广告
29 /// </summary>
30 /// <param name="adId"></param>
31 MiBridge.Instance.DestroyAd("81e6cbe35e56b53eebbc547fd1bc5614");
```

4.2、插屏广告

```
1  /// <summary>
2  /// 创建插屏广告
3  /// </summary>
4  /// <param name="adId">广告id</param>
5  /// <param name="listener">监听器</param>
6  /// <returns></returns>
7  MiBridge.Instance.CreateInterstitialAd("f54f3dfc0ba63cf3dbf582816ee069d7", new A
8  {
9      onAdError = (code, msg) => {
10         MiBridge.Instance.QGLog("c# insert ad error");
11     },
12
13     onAdClose = (end) => {
14         MiBridge.Instance.QGLog("c# insert ad close");
15     },
16
17     onAdLoad = (info) => {
18         MiBridge.Instance.QGLog("c# insert ad load");
19     },
20 });
21
22 /// <summary>
23 /// 展示插屏广告测试
24 /// </summary>
25 MiBridge.Instance.ShowAd("f54f3dfc0ba63cf3dbf582816ee069d7");
26
27 /// <summary>
28 /// 销毁插屏广告测试
29 /// </summary>
30 MiBridge.Instance.DestroyAd("f54f3dfc0ba63cf3dbf582816ee069d7");
```

4.3、Native广告

```
/// <summary>
/// 创建原生广告
/// </summary>
/// <param name="adId">广告id</param>
/// <param name="listener">监听器</param>
/// <returns></returns>
MiBridge.Instance.CreateNativeAd("da11b7e8c582ee7d1acf16a627ea6b34", new AdEvent
{
    onAdError = (code, msg) => {
        MiBridge.Instance.QGLog("c# native ad error");
    },

    onAdClose = (end) => {
        MiBridge.Instance.QGLog("c# native ad close");
    },

    onAdLoad = (info) => {
        MiBridge.Instance.QGLog("c# native ad load");
    },
});

/// <summary>
/// 加载广告
/// </summary>
/// <param name="adId"></param>
MiBridge.Instance.LoadAd("da11b7e8c582ee7d1acf16a627ea6b34");

/// <summary>
/// 展示广告
/// </summary>
/// <param name="adId"></param>
MiBridge.Instance.ShowAd("da11b7e8c582ee7d1acf16a627ea6b34");

/// <summary>
/// 销毁广告，销毁后，如需重新显示，请先调用创建广告
/// </summary>
/// <param name="adId"></param>
MiBridge.Instance.DestroyAd("da11b7e8c582ee7d1acf16a627ea6b34");
```

4.4、激励视频广告

```
/// <summary>
/// 创建激励视频广告
/// </summary>
/// <param name="adId">广告id</param>
/// <param name="listener">监听器</param>
/// <returns></returns>
MiBridge.Instance.CreateRewardedVideoAd("77295ab0558fa54fc5cc9ed6a28b6da7", new
{
    onAdError = (code, msg) => {
        MiBridge.Instance.QGLog("c# reward ad error");
    },

    onAdClose = (end) => {
        string info = "success";
        if (!end) {
            // end == true 说明激励视频播放完毕，是正常结束
            info = "failure";
        }
        MiBridge.Instance.QGLog($"c# reward ad close end={info}");
    },

    onAdLoad = (info) => {
        MiBridge.Instance.QGLog("c# reward ad load");
    },
});

/// <summary>
/// 加载广告
/// </summary>
/// <param name="adId"></param>
MiBridge.Instance.LoadAd("77295ab0558fa54fc5cc9ed6a28b6da7");

/// <summary>
/// 展示广告
/// </summary>
/// <param name="adId"></param>
MiBridge.Instance.ShowAd("77295ab0558fa54fc5cc9ed6a28b6da7");
```

4.5、互推盒子广告

```
/// <summary>
/// 展示互动盒子广告
/// 互动盒子广告介绍：https://dev.mi.com/distribute/doc/details?pId=1442
/// </summary>
/// <param name="adId">广告id</param>
```

```
/// <param name="type">
/// 广告类型:
/// 1. type=100,     互推盒子-九宫格
/// 2. type=120,     互推盒子-横幅
/// 3. type=130/140, 互推盒子-抽屉,     白色背景为130，黑色背景为140
/// 4. type=150,     互推盒子-悬浮球
/// </param>
/// <param name="listener">监听器</param>
/// <returns></returns>
MiBridge.Instance.ShowRecommendAd("da11b7e8c582ee7d1acf16a627ea6b34", 100, (succ
{

});

/// <summary>
/// 关闭互动盒子广告
/// 互动盒子广告介绍: https://dev.mi.com/distribute/doc/details?pId=1442
/// </summary>
/// <param name="adId">广告id</param>
/// <param name="type">
/// 广告类型:
/// 1. type=100,     互推盒子-九宫格
/// 2. type=120,     互推盒子-横幅
/// 3. type=130/140, 互推盒子-抽屉,     白色背景为130，黑色背景为140
/// 4. type=150,     互推盒子-悬浮球
/// </param>
/// <param name="listener">监听器</param>
MiBridge.Instance.CloseRecommendAd("da11b7e8c582ee7d1acf16a627ea6b34", 100, (suc
{

});
```

5、文件操作

5.1、读文件

```
MiBridge.Instance.ReadFile("log.txt", (data, len) =>
{
    string test = System.Text.Encoding.ASCII.GetString(data);
    MiBridge.Instance.QGLog("MiBridge.Instance.ReadFile success {0}, {1}", len,
}, (msg) =>
{
    MiBridge.Instance.QGLog("read file failure {0}", msg);
});
```

5.2、写文件

```
string myString = "Hello world!!!";
byte[] data = Encoding.UTF8.GetBytes(myString);
MiBridge.Instance.WriteFile("log.txt", data, false, (success, msg) =>
{
    MiBridge.Instance.QGLog("write file {0}, {1}", success, msg);
});
```

5.3、删除文件

```
MiBridge.Instance.DeleteFile("log.txt", (success) => {
});
```

6、游戏图标

```
MiBridge.Instance.HasShortcut( has => {
        if (has)
        {
            MiBridge.Instance.QGLog("has short cut");
        }
        else
        {
            MiBridge.Instance.QGLog("has not short cut");
            MiBridge.Instance.CreateShortcut("便于打开游戏", (success, code, msg)
                MiBridge.Instance.QGLog("create short cut {0}, {1}, {2}", succes
        });
    }
});
```

7、KV存储

支持四种类型存储：int，float，double和string类型。

存和读操作：

```
MiBridge.Instance.SetKVInt("int", 1);
var kvInt = MiBridge.Instance.GetKVInt("int");
MiBridge.Instance.QGLog("get int kv={0}", kvInt);

```

```
MiBridge.Instance.SetKVFloat("float", 2.0522222f);
var kvFloat = MiBridge.Instance.GetKVFloat("float");
MiBridge.Instance.QGLog("get float kv={0}", kvFloat);

MiBridge.Instance.SetKVDouble("dobule", 3.2656453453d);
var kvDouble = MiBridge.Instance.GetKVDouble("dobule");
MiBridge.Instance.QGLog("get dobule={0}", kvDouble);

MiBridge.Instance.SetKVString("string", " fsdfsdf ");
var kvString = MiBridge.Instance.GetKVString("string");
MiBridge.Instance.QGLog("get string={0}", kvString);
```

删除操作：

```
MiBridge.Instance.DeleteKV("int");
```

8、日志打印到快游戏平台

```
MiBridge.Instance.QGLog("hello world! ");
```