

防沉迷系统开发文档

1.说明

该文档为Unity防沉迷系统的开发文档，为开发者提供防沉迷系统的功能。

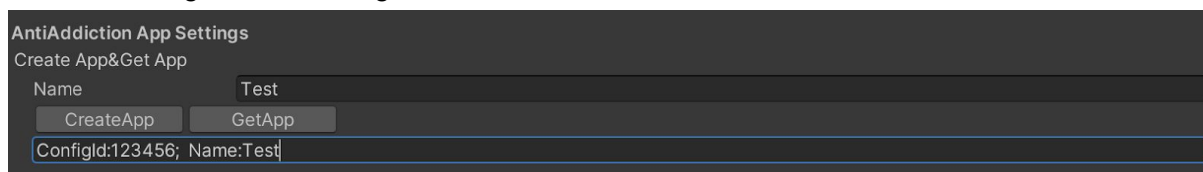
2.环境配置

2.1 Unity PlayerIdentity(CN)

开发者需在Package Manager中下载名为Player Identity(CN)的package，并根据document配置好相关设置。

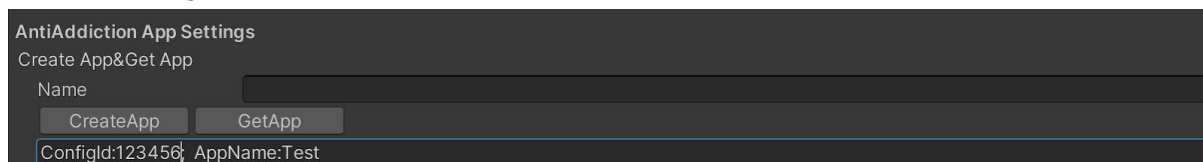
2.2 创建防沉迷APP

打开菜单Edit/ProjectSettings，找到Player Identity/Backends/Unity UserAuth,在AntiAddiction App Settings字段中的Name，填入想要创建的APP名称，点击CreateApp可在下方获得configId，注意configId和APP名称绑定，为后续使用防沉迷功能所必须的属性。



2.3 查询防沉迷APP

打开菜单Edit/ProjectSettings，找到Player Identity/Backends/Unity UserAuth,在AntiAddiction App Settings字段中点击GetApp即可查询当前idDomain下创建的所有APP名称以及对应configId。



3.防沉迷系统开发流程

3.1 防沉迷系统说明

防沉迷系统主要分为三个模块，登录模块，计时模块以及充值模块。

在登录模块，玩家用户通过Unity PlayerId账号来进行登录。对于未实名认证的用户将自动打开浏览器跳转到实名认证界面引导用户进行实名认证。此时游戏内将强制下线，用户需要重新登录。对于已实名认证的用户防沉迷系统将根据认证信息来判定用户的适用规则。

在计时模块，需要开发者定时主动调用计时接口，向防沉迷服务器报告用户是否仍在游玩。服务器将根据用户的游玩时长给出对应的指令或消息。

在充值模块，开发者需要调用充值查询接口，此接口将返回当前用户所处年龄段信息，开发者可依据该信息自行决定充值限制。

此外，根据防沉迷系统的不同指令，开发者需要注册对应的回调事件以保证防沉迷系统能够正常运作，在执行完对应的指令后，开发者还需要调用上报执行接口向防沉迷服务器上报告对应的指令已执行。具体的注册事件见3.2，对应的接口见3.3

3.2 注册事件

事件名称	传入参数	事件说明
OnMessage	<string,string,string>	传入三个string类型的参数，分别为title, message和context。通过该事件来获取防沉迷系统返回的消息。正常情况下，title和message会返回防沉迷文案，context则是调用方法是传入的内容。特殊情况和异常处理也会通过该事件传入，具体见3.4
OnKickOff	<string,string>	传入两个string类型的参数，通过该事件执行强制下线操作，传入的两个参数用于执行强制下线操作后调用ReportExecution时所需的参数，第一个string类型的参数为流水号，第二个为规则名称，作为用于上报执行请求时的参数。见3.3
OnJudgePay	<int>	传入一个int类型的参数，该参数代表用户实名认证后所处的年龄段，年龄段共分为四段[0,8)对应传入的参数为1，[8,16)对应传入的参数为8，[16,18)对应传入的参数为16，大于等于18传入的参数为模糊年龄（即传入的年龄一定大于18但并不代表用户的真实年龄）

3.3 防沉迷系统方法API

方法名	传入参数	方法说明
void ContinueHeartbeat	int duration, string context	该方法用于持续计时请求，在上一次计时有效期内调用即可，不必等到上一次计时有效期完全结束后再调用。防沉迷系统将根据当前账号的游玩时长给出相应的指令，。此外，在用户成功登录后，将自动调用一次该接口，因此开发者第一次调用只需在有效时间五分钟内即可。duration为下一次计时有效时长，以秒为单位，默认300秒。context同上。
void StopHeartbeat		该方法用于结束计时请求，在用户下线，强制下线或者退出游戏时调用。所有参数同上。

void JudgePay	string context	该方法用于判定用户所处年龄段，在用户发起充值请求时调用，防沉迷系统将返回用户所处年龄段，由开发者依据该信息决定相应的充值限制。context同上。
void ReportExecution	string traceId, string ruleName, int execTime, string context	在执行完对应的指令之后，需要上报该指令已执行，不需要等待回包。traceId和ruleName均从事件中获取直接传入，execTime为指令的实际执行时间，UTC+08的时间戳，context同上。

3.3 防沉迷系统方法调用流程说明

调用防沉迷系统方法需要在Player Identity(CN)的UserAuthPrefab的Inspector中配置AntiAddictionClientApi的configId，该属性可在2.2中进行获取。然后需要对AntiAddictionClientApi中的三个事件进行注册，参考3.1，防沉迷系统的所有方法将会执行相应的事件。

在用户登录时，防沉迷系统将自动调用登录决策请求，防沉迷系统将根据用户的状态执行指令：允许进入游戏，无法进入游戏，打开网页进行实名认证。若需要实名认证，将执行打开浏览器进行实名认证操作，此时用户在游戏内将被强制退出账号，并执行OnKickOff，用户需完成实名认证后重新登录。若玩家受到防沉迷限制，也将被强制下线。在登录成功后，系统将自动调用一次有效期为300秒的计时请求。在到达有效期之前，若用户仍在游玩，需要调用持续计时请求。（开发者不用等到计时完全结束再调用该方法，可提前调用，不影响实际记录时间）当用户退出账号/游戏时，需要调用结束计时请求。该方法会触发受防沉迷系统限制的用户游玩状态。在用户触发防沉迷系统的游玩限制时，防沉迷系统将通过OnKickOff来强制用户下线限制用户继续游玩，在完成强制下线后，需要调用上报执行方法，将传入的流水号和指令类型直接填入即可（实际算法流程实现中需要先上报执行，再将用户下线，参考Demo）。

当用户进行充值操作时，需要调用充值决策方法，防沉迷系统将根据当前用户的状态返回用户所处年龄段，开发者自行决定充值限制。

回调事件的具体实现方法可参考demo。

3.4 异常情况处理

调用3.3中的接口时，存在因上传数据格式错误，网络状况不良或服务器问题而返回错误异常信息，此时错误信息会通过OnMessage事件传入，title为AntiAddictionServerError，msg为具体的错误信息，context同3.2.对于上传数据格式错误的情况，msg中会指出格式有误的target，开发者需要在开发阶段解决对应的错误，确保在玩家用户实际使用阶段不会产生数据

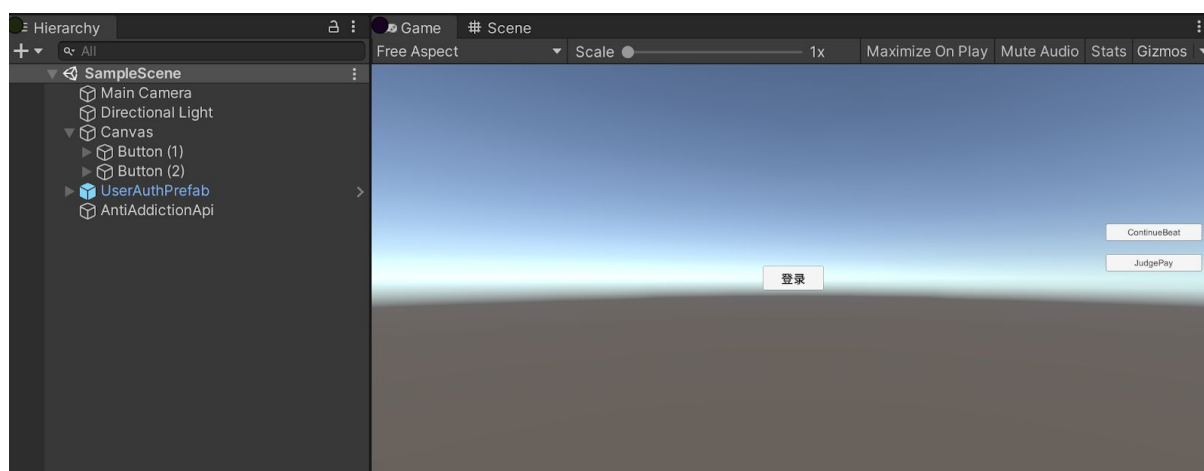
格式错误。对于因网络状况不良或服务器问题返回的错误信息，开发者可以不用关注，防沉迷系统默认不执行任何操作。

3.5 测试环境与正式环境说明

防沉迷系统分测试环境与正式环境，两个环境的数据互不相通，建议开发者先在测试环境中进行测试，没有问题再到正式环境测试。注意，测试环境仅用于测试使用，线上真实用户的请求数据都要走正式环境（含体验服）。如何切换正式服与测试服见demo。

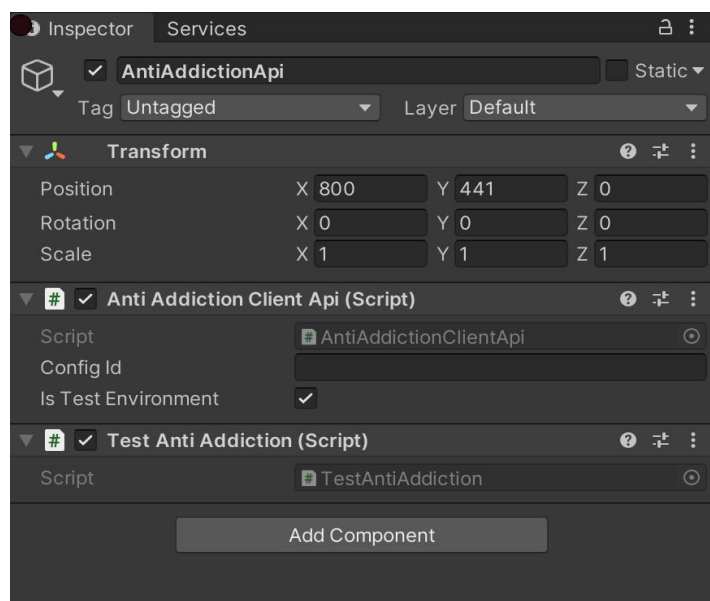
4. Demo使用说明

打开SampleScene,可以看到如下场景，中间是Unity PlayerId的登录控件，玩家用户通过该组件登录unity账号。右边是两个button，分别代表调用 ContinueHeartbeat和JudgePay接口。要使用Demo，需要完成如下几个步骤。



配置ConfigId：在环境配置完成之后，选中物体AntiAddictionApi，将得到的ConfigId填入Inspector上的脚本AntiAddictionClientApi中。

测试环境与正式环境切换：勾选IsTestEnvironment代表开启测试环境，取消勾选代表使用正式环境。



3.2中所提到的注册事件已经在TestAntiAddiction脚本中完成，因此在完成上述步骤后即可直接运行。运行后首先登录unity账号，第一次登录账号将拉起实名认证网页，此时Console面板也将显示对应的log信息。完成实名认证之后重新登录，此时即可通过右边的两个button进行计时和查询用户年龄段的操作，此外TestAntiAddiction中也会固定一段时间调用计时方法。若使用未成年账号登录，在一小时以及两小时后，分别会通过OnMessage收到相应的防沉迷信息提示，以及超过当日游玩上限进行强制下线的操作。查询用户年龄段则可直接点击JudgePay的button，便可获得用户所处年龄段信息。开发者可参考Demo实现相关的功能。

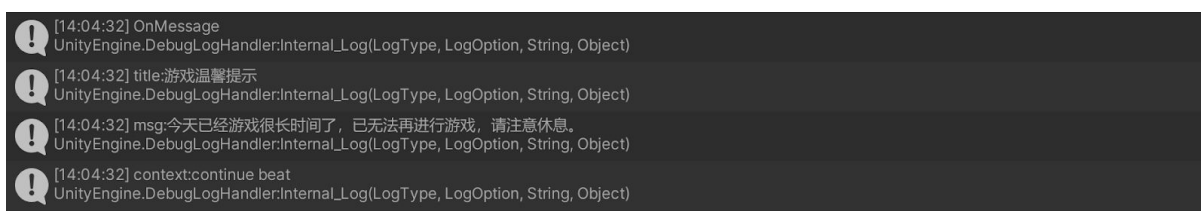
Demo效果展示：



未成年用户登录触发宵禁（白天）提示并强制下线



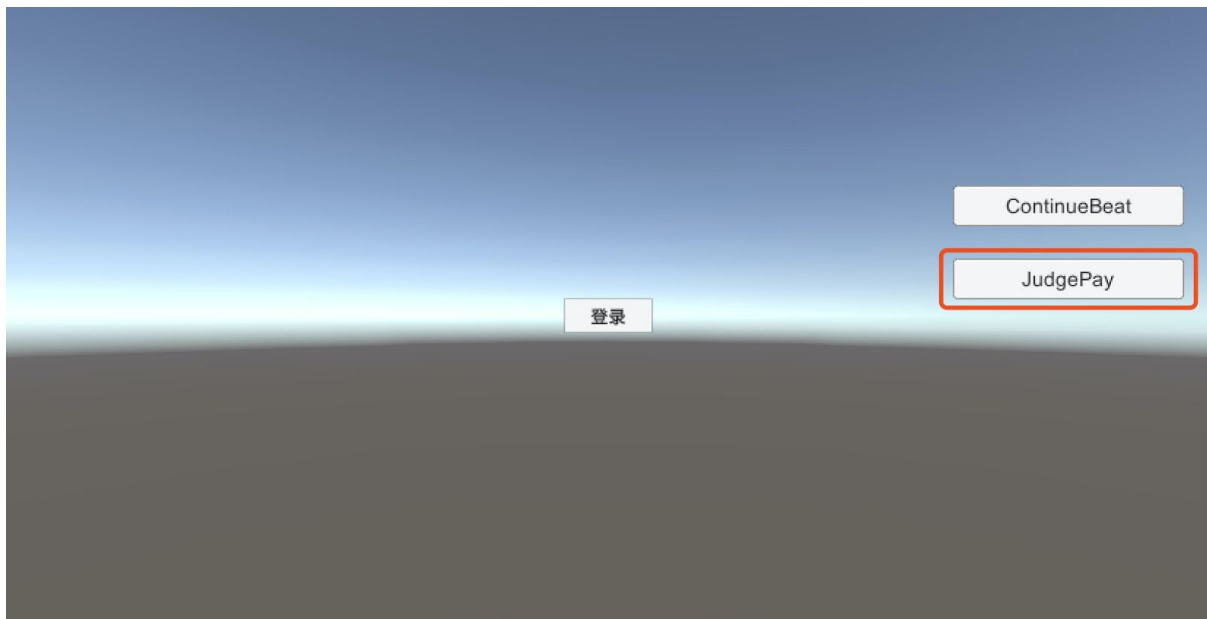
用户第一次登录游戏，自动打开浏览器进行实名认证



未成年用户当日游玩超过一定时长，触发提示并强制下线

```
[17:15:55] OnPayAmount
UnityEngine.DebugLogHandler:Internal_Log(LogType, LogOption, String, Object)
[17:15:55] age:1
UnityEngine.DebugLogHandler:Internal_Log(LogType, LogOption, String, Object)
[17:15:55] user's age is in [0,8)
UnityEngine.DebugLogHandler:Internal_Log(LogType, LogOption, String, Object)
```

用户调用充值查询接口，返回用户所处年龄段



```
[18:46:25] OnJudgeTime
UnityEngine.Debug:Log(Object)
[18:46:25] time: 0h12min16s
UnityEngine.Debug:Log(Object)
```