

EASY LEADERBOARD (Facebook + PlayFab)

Manual de instruções, versão 2.0.0

Alterações

- Novas cenas de exemplo (otimizadas para mobile, totalmente redesenhada);
- Funcionalidade de Invite removida (o Facebook deprecou a funcionalidade de Convites de Aplicativo).

A proposta do Easy Leaderboard é facilitar a implementação de rankings no seu jogo, contando com dois serviços muito utilizados pelos desenvolvedores: Facebook e PlayFab.

Por que usar o Easy Leaderboard?

- Implementação rápida: crie um ranking mundial no seu jogo em questão de minutos;
- Foto: mostra a foto de perfil dos jogadores;
- Busca incremental: lógica otimizada para retornar registros do PlayFab de forma incremental;
- Busca flexível: você define quantos registros serão retornados em cada incremento;
- Sem limite de estatísticas: manipule quantas estatísticas forem necessárias;
- Independente de UI: o script não tem nenhuma dependência de componentes de UI, o que deixa tudo mais flexível e mais fácil de se adaptar ao seu jogo;
- Consiga mais jogadores: divulgue seu jogo por meio do compartilhamento do Facebook!

1. PRÉ-REQUISITOS PARA SEU PROJETO

Assim que você importa o Easy Leaderboard no seu projeto aparecerão vários erros. Isso ocorre porque ele depende de classes que são disponibilizadas pelo Facebook e PlayFab por meio de seus respectivos kits de desenvolvimento (SDKs) e esses kits não estão incluídos no projeto. Nesta seção veremos não só como resolver os erros na Unity mas também como deixar seu aplicativo configurado corretamente e pronto para uso no Facebook e PlayFab.

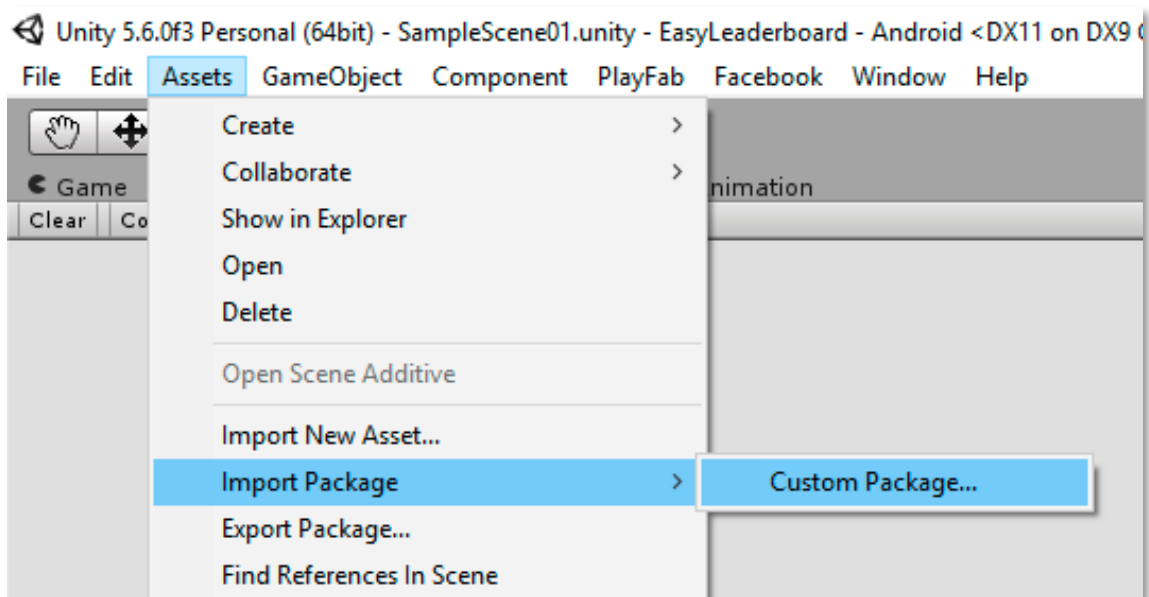
1.1 Facebook

1.1.1 Importação do SDK

- a) Acesse o link <https://developers.facebook.com/docs/unity/downloads> e faça o download do SDK do Facebook.

ATENÇÃO: este Asset foi desenvolvido e testado utilizando a SDK do Facebook na versão 7.15.0, então para que não ocorra problemas desconhecidos recomendamos que você utilize a mesma versão.

- b) Na Unity, clique em Assets > Import Package > Custom Package...



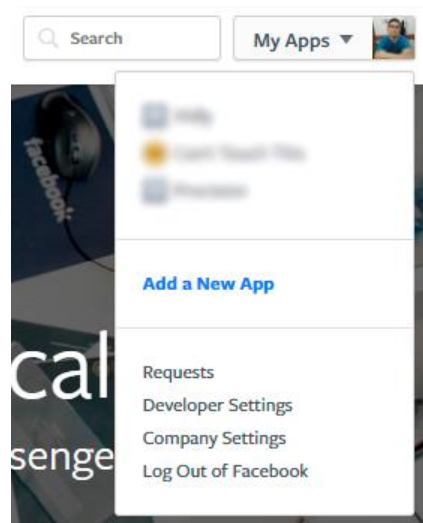
- c) Procure pelo arquivo com a extensão *.unitypackage* dentro do pacote que você acabou de baixar e clique em Import (geralmente fica dentro da pasta FacebookSDK).

1.1.2 Conta de desenvolvedor

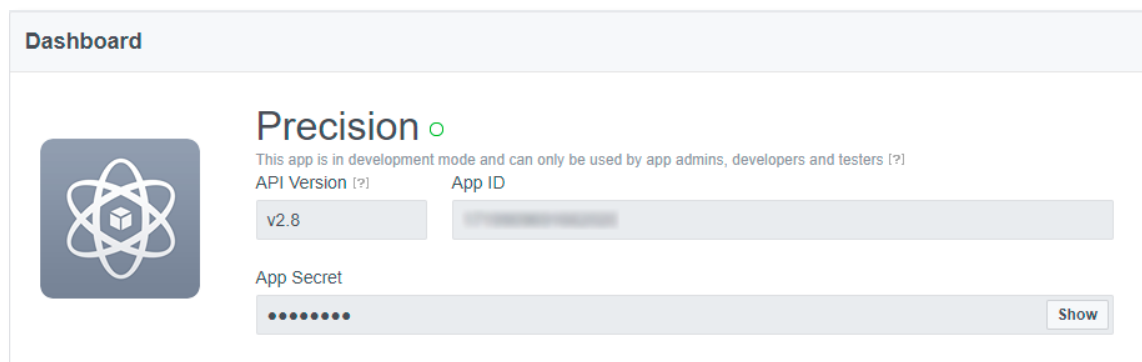
Se você ainda não possui uma conta de desenvolvedor no Facebook, crie uma seguindo as instruções fornecidas pelo site <https://developers.facebook.com>.

1.1.3 Configuração do aplicativo no Facebook

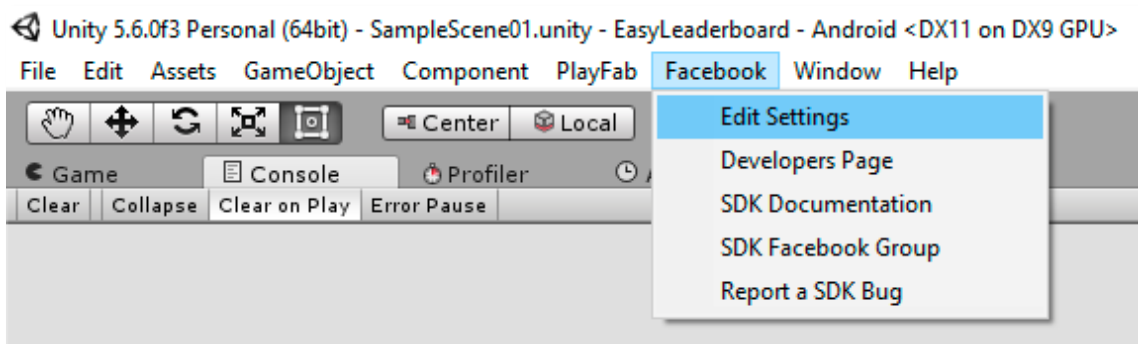
a) No canto superior direito, clique em *Adicionar um novo aplicativo*:



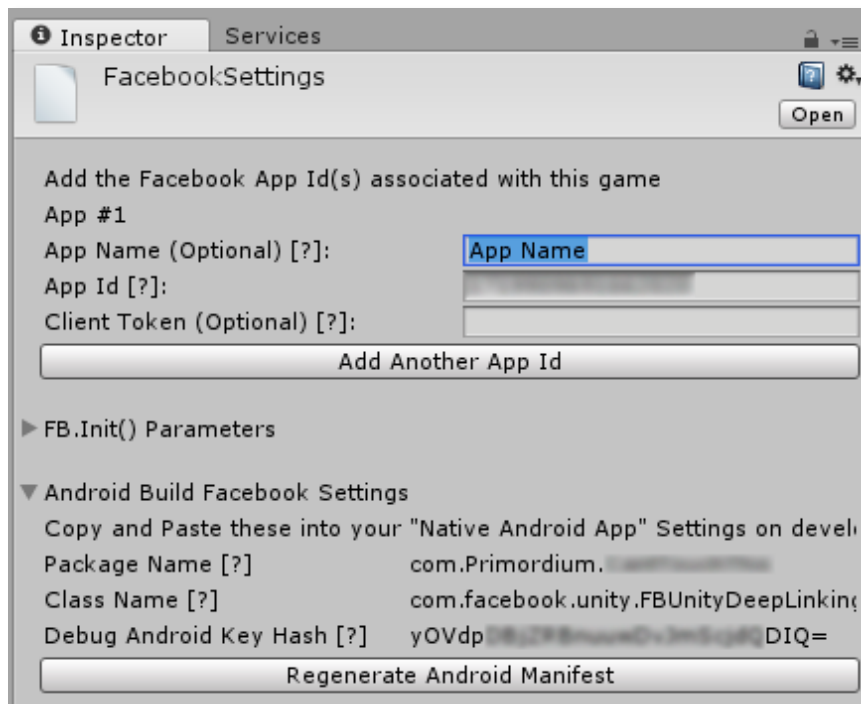
b) Será exibida uma tela como essa:



- c) Copie o conteúdo do campo *ID do Aplicativo* (usaremos essa informação posteriormente).
- d) No canto superior esquerdo, clique em Configurações. Depois clique em *Adicionar plataforma* (vamos utilizar Android no exemplo).
- e) Para preencher os campos *Nome do pacote do Google Play*, *Nome da Classe* e *Hashes chave*, vamos precisar voltar ao Unity.
- f) Clique em Facebook > Edit Settings (se esse menu não aparecer, reinicie o Unity).



- g) Após clicar nessa opção, deverá aparecer as seguintes configurações no Editor do Unity:



- h) Preencha o campo *App Name* com o nome do seu aplicativo e cole o ID do Aplicativo (a informação que copiamos anteriormente) no campo *App Id*.
- i) Agora copie o conteúdo dos 3 próximos campos, cole nos 3 campos do console de desenvolvedor do Facebook (respectivamente) e clique em *Salvar alterações*.

PS: quando você for testar o aplicativo pela primeira vez no seu dispositivo aparecerá uma mensagem dizendo que a Key Hash é inválida. Digite a Key Hash informada, adicione-a ao campo Key Hashes e salve as alterações. Dessa forma o Facebook será capaz de identificar seu aplicativo e tudo funcionará corretamente.

1.2 PlayFab

1.2.1 Importação do SDK

- a) Acesse o link <https://api.playfab.com/sdks/unity> e faça o download do SDK do PlayFab.
- b) Repita o passo 1.1.1b, porém importando agora a SDK do PlayFab.

1.2.2 Conta de desenvolvedor

Se você ainda não possui uma conta de desenvolvedor no PlayFab, crie uma conta seguindo as instruções fornecidas pelo site <https://developer.playfab.com>.

1.2.3 Configuração do aplicativo no PlayFab

- a) Para criar um novo aplicativo no PlayFab, clique em New Game, preencha as informações do seu aplicativo e salve as alterações.
- b) Do lado esquerdo da tela, clique em Settings e depois na aba API.
- c) Copie o *Title ID* e marque a opção *Allow client to post player statistics* e salve as alterações. Essa opção possibilita que seu aplicativo manipule as estatísticas do jogador.

2. CONFIGURANDO O EASY LEADERBOARD

2.1 Uso recomendado

- a) Arraste o prefab EasyLeaderboardManager para a hierarquia da primeira cena do seu jogo. O script anexo ao prefab (FacebookAndPlayFabManager) é implementado utilizando o padrão Singleton e utiliza o método DontDestroyOnLoad da classe MonoBehaviour. Isso significa que, assim que carregado, você poderá acessar as funcionalidades do script FacebookAndPlayFabManager em qualquer ponto do seu código, independente da cena.
- b) Preencha o campo *PlayFab Title Id* com o *Title ID* que você copiou do console de desenvolvedor do PlayFab.
- c) Os campos da seção *Facebook Share Info* são necessários somente se você for utilizar a opção de compartilhamento do seu jogo.

3. UTILIZANDO O EASY LEADERBOARD

3.1 Métodos públicos

Os métodos públicos da classe FacebookAndPlayFabManager são do tipo void e podem ser chamados em todos os lugares do código.

3.1.1 LogOnFacebook: efetua o login do jogador no Facebook.

Parâmetros:

- Action<ILoginResult> *successCallback* (opcional): ação a ser executada caso o processo seja realizado corretamente.
- Action<ILoginResult> *errorCallback* (opcional): ação a ser executada caso o processo falhe.

OBS: Se o login for efetuado corretamente, o método se responsabiliza não só por executar a ação passada no parâmetro *successCallback*, mas também por efetuar login no PlayFab e atribuir valor às seguintes propriedades da classe:

- bool *IsLoggedInOnFacebook*: o jogador está logado no Facebook?
- bool *IsLoggedInOnPlayFab*: o jogador está logado no PlayFab?
- string *FacebookUserName*: nome do perfil do Facebook do jogador.
- Sprite *FacebookUserPictureSprite*: foto do perfil do Facebook do jogador.

As propriedades são públicas e somente leitura, portanto você pode acessar os valores de qualquer ponto do seu código, assim como os métodos.

3.1.2 LogOutFacebook: efetua o logout do jogador do Facebook.

3.1.3 GetFacebookUserName: busca o nome do jogador no Facebook.

Parâmetros:

- string *id*: identificador único de um perfil no Facebook.
- Action<IGraphResult> *successCallback* (opcional): ação a ser executada caso o processo seja realizado corretamente.
- Action<IGraphResult> *errorCallback* (opcional): ação a ser executada caso o processo falhe.

3.1.4 GetFacebookUserPicture: busca a foto do perfil do jogador no Facebook.

Parâmetros:

- string *id*: identificador único de um perfil no Facebook.
- int *width*: largura que a imagem de retorno deve ter.
- int *height*: altura que a imagem de retorno deve ter.
- Action<IGraphResult> *successCallback* (opcional): ação a ser executada caso o processo seja realizado corretamente.
- Action<IGraphResult> *errorCallback* (opcional): ação a ser executada caso o processo falhe.

Algumas versões mais recentes da Unity não são compatíveis com a forma padrão de obtenção da foto do perfil do jogador. Se você estiver com problemas para obter a foto do perfil do jogador, utilize o método abaixo.

3.1.4 (Alternativo) GetFacebookUserPictureFromUrl: método alternativo para busca da foto do perfil do jogador no Facebook.

Parâmetros:

- string *id*: identificador único de um perfil no Facebook.
- int *width*: largura que a imagem de retorno deve ter.
- int *height*: altura que a imagem de retorno deve ter.
- Action<IGraphResult> *successCallback* (opcional): ação a ser executada caso o processo seja realizado corretamente.
- Action<IGraphResult> *errorCallback* (opcional): ação a ser executada caso o processo falhe.

3.1.5 GetLeaderboard: busca os valores de uma determinada estatística no PlayFab.

Parâmetros:

- string *statisticName*: nome da estatística a ser retornada.
- bool *friendsOnly*: buscar informações somente dos seus amigos do Facebook?
- int *maxResultsCount*: quantidade máxima de registros a retornar.
- Action<GetLeaderboardResult> *successCallback*: ação a ser executada caso o processo seja realizado corretamente.
- int *startPosition* (opcional, valor padrão = 0): ponto de partida da busca pelos registros.

3.1.6 UpdateStat: atualiza o valor de uma determinada estatística no PlayFab.

Parâmetros:

- string *statisticName*: nome da estatística a ser atualizada.
- int *value*: valor que a estatística deverá receber.

- Action<UpdatePlayerStatisticsResult> *successCallback* (opcional): ação a ser executada caso o processo seja realizado corretamente.

OBS: é recomendado criar constantes com os nomes das estatísticas que serão manipuladas (veja a classe PlayFabStatConstants em EasyLeaderboard > Scripts > Utility).

3.1.7 ShareOnFacebook: realiza um compartilhamento no Facebook.

Parâmetros:

- Action<IShareResult> *successCallback* (opcional): ação a ser executada caso o processo seja realizado corretamente.

- Action<IShareResult> *errorCallback* (opcional): ação a ser executada caso o processo falhe.

OBS: preencher os campos *Content Url*, *Content Title*, *Content Description* e *Picture Url*.

4. CONTATO

Em caso de dúvidas entre em contato com o suporte pelo e-mail primordiumlabs@gmail.com (favor definir o assunto como Easy Leaderboard). Seu e-mail será respondido o mais rápido possível!

Por favor nos encoraje a melhorar avaliando o Easy Leaderboard na Unity Asset Store.

Obrigado!

**PRIMORDIUM
GAME STUDIO**