

【ハコスコ対応】VRゲームに ニフティクラウドmobile backend(mBaaS)を使って オンラインランキングを実装しよう

- ごあいさつ
- mBaaSについて
- ハンズオンについて
- 事前準備
- mBaaSへの認証の準備
- データ更新実装
- ランキング実装
- まとめ

本日はニフティブースに
お越しいただきありがとうございます。
ハンズオンはmBaaSを使ってUnityゲームにサー
バー機能を

「**素早く 無料**」で導入するハンズオンです。
ハンズオンでは下記の2つのことが体験できます

- ・ ゲームスコアの保存
- ・ ランキングの実装

ぜひお楽しみください！



スマートフォンアプリの**バックエンド機能**が
開発不要になる**クラウドサービス**

ニフティクラウドmobile backendとは
mBaaS(mobile backend as a Service)と呼ばれる、
クラウドサービスのジャンルのひとつで
スマートフォンアプリでよく利用される汎用的な機能を
クラウドから提供するサービスです。

クラウド上に用意された機能を
API・SDKで呼び出すだけで利用できるので、
サーバー開発・運用不要でよりリッチなバックエンド機能を
アプリに実装することができ、工数削減による
コストカット・スピードアップに貢献します。

※SDKはUnity,iOS ,Android,JavaScriptの4つを提供しております

提供中の機能



UnitySDKで提供中の機能



ユーザーログイン認証



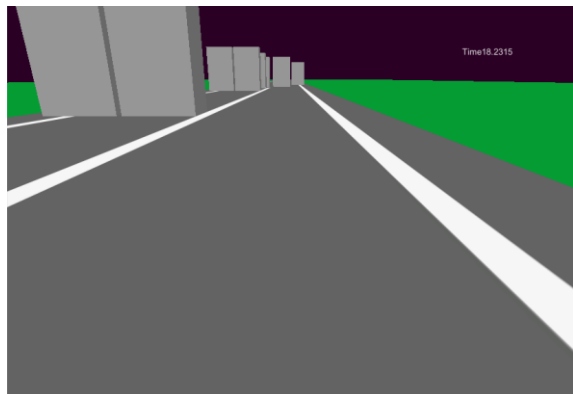
Android・iOS各プラットフォームへの プッシュ通知配信



オンラインランキング



データストアの更新・取得を利用した スコア保存・ランキングの実装

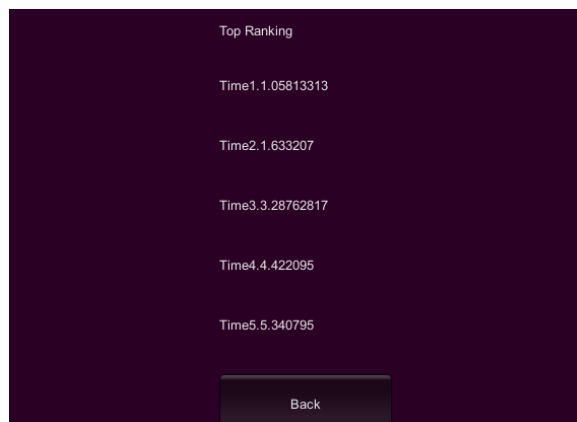


VRレーシングゲームに
ニフティクラウドmobile backendを導入します。

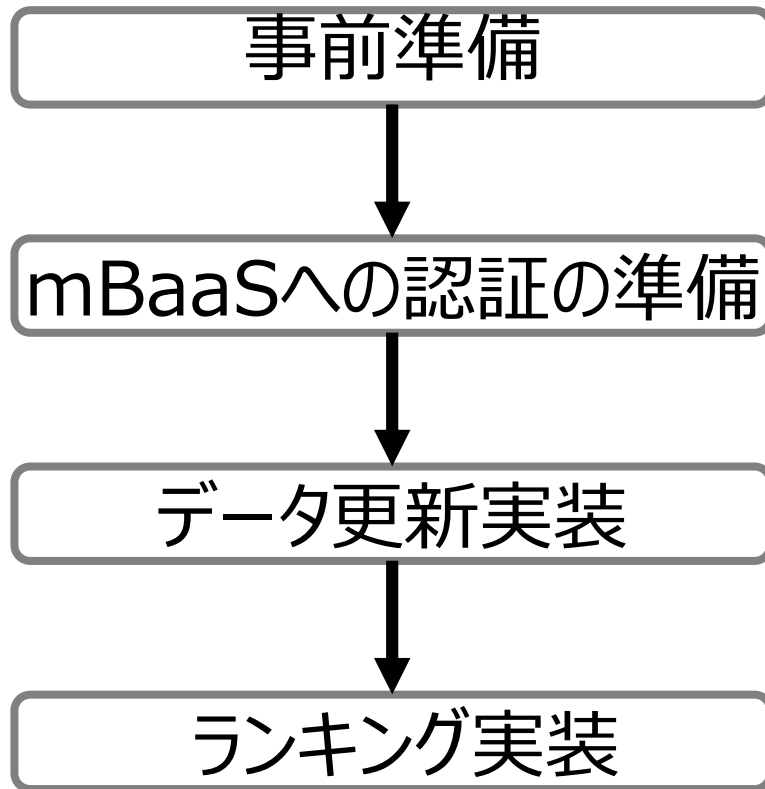
このハンズオンでは下記の二つのことが体験できます

- ・ゲームスコアの保存
- ・ランキングの実装

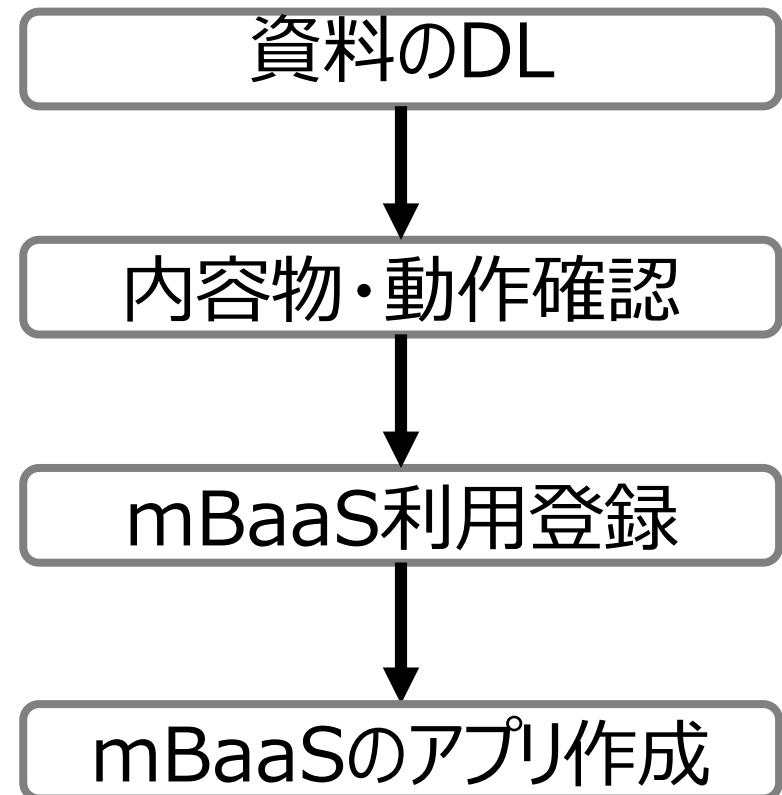
どちらもアーケード形式を想定してご案内いたします。
なにとぞよろしくお願いいたします。



本日のハンズオンは大きく分けて4つのフローで行います。まずは事前準備からです。。



「事前準備」にておこなうこと



※mBaaSのアプリ作成とは
アプリ用のサーバー環境を作っています

資料が ↓ にあります

https://github.com/hounenhounen/NCMB_CEDEC_hacoscon

branch: master NCMB_BitSummit_hacoscon / +

commit2

hounenhounen authored 34 minutes ago latest commit 8d84e25768

hacosco	commit2	34 minutes ago
gitattributes	Added .gitattributes & .gitignore files	2 hours ago
.gitignore	Added .gitattributes & .gitignore files	2 hours ago
NCMB_BitSummit_hacosco.pdf	commit1	39 minutes ago

Help people interested in this repository understand your project by adding a README! [Add a README](#)

HTTPS clone URL
<https://github.com>

You can clone with [HTTPS](#), [SSH](#), or [Subversion](#).

[Clone in Desktop](#)

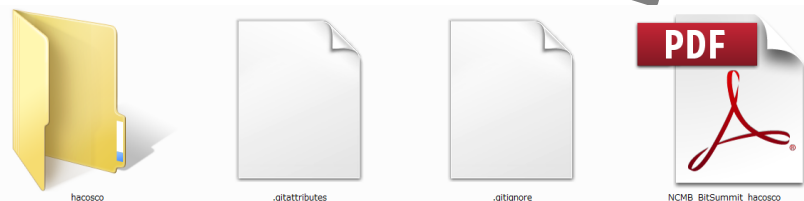
[Download ZIP](#)

ここよりDLしてください

内容物確認

DLファイルフォルダには以下の3つが含まれています

本資料



ハンズオンで使うファイル

注意

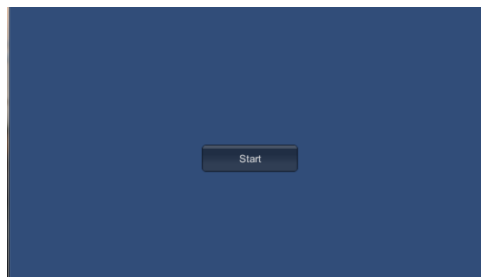
1. 本資料をおくフォルダのパスに全角文字は使用しないでください
場合によってはエラーが表示されます
2. 本ファイルはUnity4系で作成しています5系を利用の方はUpgradeが必要となります

動作確認

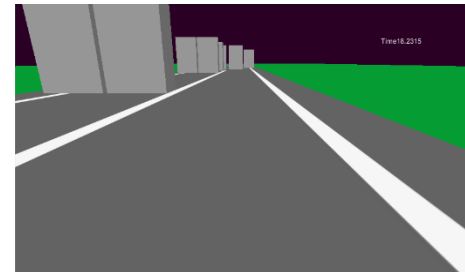
ハンズオンで使うフォルダhacoscoの中に
Asset> Scenes Start.sceneをダブルクリックしてゲームを実行してください
スタート時とプレイ時時に下記の画面が出ればひとまず大丈夫です

実行・ゲームオーバー時に
下記の画面が出ればOK

ゲームスタート時の画面



ゲームプレイ時の画面



まず、下記URLよりmBaaSのサービスサイトにアクセスしていただきます。

<http://mb.cloud.nifty.com/>

※右クリックして新しいタブで開くと便利です

ここをクリック



無料登録をしていただくと、@nifty会員登録を行うフローに移ります。
@nifty会員登録をしていただいた後、mBaaSにログインしていただくと
mBaaSの利用登録が行えます。以下上記のフローの説明を行います。

NIFTY Cloud
mobile backend

@nifty

mBaaSとは 機能 料金 事例紹介 利用ケース eBook ブログ サポート FAQ コミュニティ ドキュメント ログイン 資料請求 無料登録

トップ > ID無料登録 ニフティクラウド mobile backendはWebから簡単に申し込み

ID登録後、プッシュ通知配信画面や会員管理画面など、コントロールパネルの中をご覧ください。

ID登録 (即時・無料)

IDは即時・無料で発行いただけます。

サインアップ

個人会員として登録ご希望の方

ここをクリック

STEP1
ニフティクラウド mobile backendをご利用いただくには、@nifty会員の登録が必要です (登録料/会費は無料)

@nifty会員の登録 (無料)

@nifty会員の方はSTEP2へお進みください。

法人会員として登録ご希望の方

ニフティの法人会員としてご契約いただくと、ニフティクラウド mobile backend利用料金を企業・団体に決済いただけます。Webまたは郵送にて@nifty法人IDの取得手続きを行ってください。なお、法人としてご契約いただけるのは、企業/各種法人/各種団体・グループ単位となります。

STEP1 @nifty法人ID取得
「@nifty法人ID取得フォーム」から、法人ID取得手続きを行ってください。なお、申込書 (郵送) での申し込みも可

必要事項を入力して
@nifty会員登録してください

無料でID登録

まずは無料から、ニフティクラウド mobile backendをお試しください！

基本情報の入力

ユーザー名 ※	<input type="text"/>	・ 入力時における注意
パスワード※	<div>[半角6～24文字]</div> <div><input type="password"/></div> <div><input type="password"/></div>	・ 使用できる文字 【ご注意】パスワードの不正利用を防ぐために
メールアドレス ※	<input type="text"/>	・ 入力されるメールアドレスについて

[@nifty会員規約](#) [個人情報の取り扱いについて](#)

個人情報の取り扱い及び@nifty会員規約に同意し

登 録

ニフティクラウドmobile backendの利用登録を行います。
<http://mb.cloud.nifty.com/>

ご登録いただいた
@nifty IDで
ログイン



ここをクリック

ログイン

【お知らせ】SSL3.0脆弱性への対応に伴うブラウザ

@niftyユーザー名または、@nifty IDをお持ちの方

@niftyユーザー名または@nifty ID

@niftyユーザー名または@nifty ID

パスワード

パスワード

セキュリティキーボード [ヘルプ](#)

☐ ログインしたままにする(共用のパソコンではチェックを外してください。)

ログイン

利用規約を
確認後、同意して
利用開始!



利用登録が終わると
アプリの新規作成画面が表示されます。
アプリ名を入力して新規作成してください。

「ced_hacosco」
と入力してください

アプリの新規作成

アプリ名

半角英数字もしくはアンダースコア

戻る

新規作成



✓ 新しいアプリが作成されました

IoTBlogAppアプリが作成されました。

SDKを利用して、mobile backendと連携したアプリを開発しましょう！
詳しくはクイックスタートをご覧ください ([iOS](#) / [Android](#) / [Javascript](#) / [Unity](#))

APIキー

アプリケーションキー 2417fd23d90bdc35ba48f5b0b48164ec6a4021c139e3419 コピー

クライアントキー b4cc5b1c55ad012f7c91b2ddc4e2a65f75c0fa29d447d1a5 コピー

APIキーは[アプリ設定](#)からいつでも参照できます。

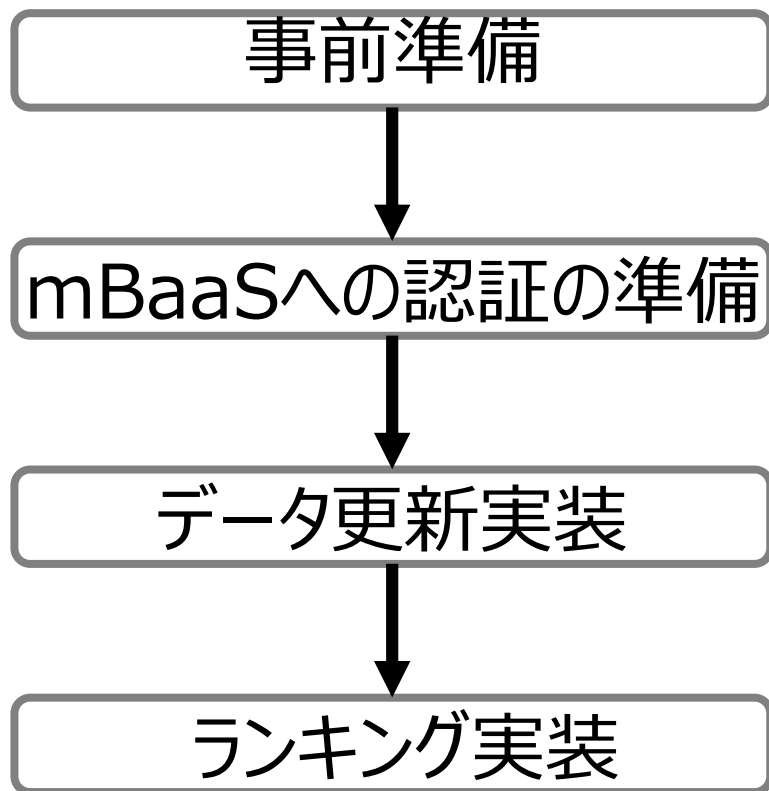
OK

2つのキーは
後で使います

OKをクリックすると
管理画面が表示されます

mBaaSはアプリケーションキー
クライアントキー2つのキーを使いサーバー
接続の認証を行っています
その2つのキーがアプリ作成時に生成されます

事前準備が終わりましたので次は自分のmBaaS環境とUnityプロジェクトが通信できるようにします



「mBaaSへの認証の準備」にておこなうこと

SDKのDL・インポート

NCMBSetingsの設定

キーの設定

赤枠で囲んでいるところに関しては、もう作業を済ませていますので最後のキーの設定のみを行ってください
また資料には赤枠部分の説明もございます
※NCMBとはNifty Cloud Mobile Backendの略称です

※参考資料※SDKのDL・インポート（作業済）

※今回は本ページの作業を行う必要はありません。

SDKのDL

https://github.com/NIFTYCloud-mbaas/ncmb_unity/releases

のページを開くと下記のページが開かれます
赤丸の部分をクリックしてSDKをDLします

NIFTYCloud-mbaas / ncmb_unity

Releases Tags

Latest release

v1.1.4

ncmb sdk released this on 31 Mar

- NCMBPushのプッシュ通知登録に以下の設定項目を追加
 - Category
 - DeliveryExpirationDate
 - DeliveryExpirationTime

Downloads

NCMB1.1.4.zip

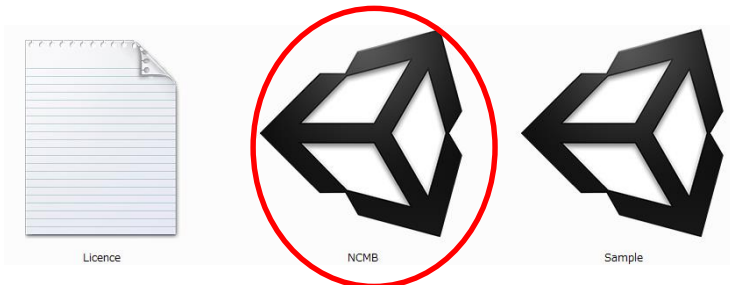
Source code (zip)

Source code (tar.gz)

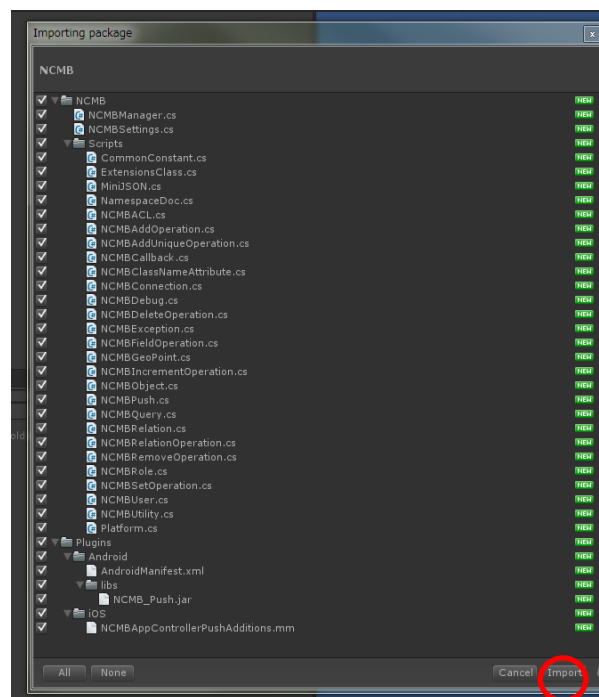
クリックすると
DLできます

DL終了後
インポート
へ

SDKのファイルを展開すると
下記のような構成になっております



先ほどDLしたサンプルゲームを開いた状態で
NCMB_latestのUnity package file「NCMB」を
ダブルクリックすると下記↓の画面が表示されます



表示された
画面の「Import」
をクリックすると
Import可能

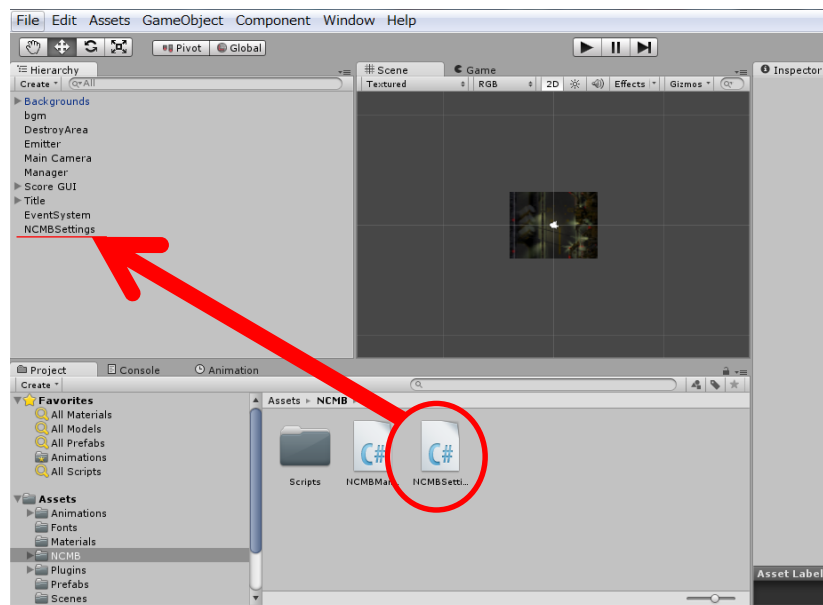
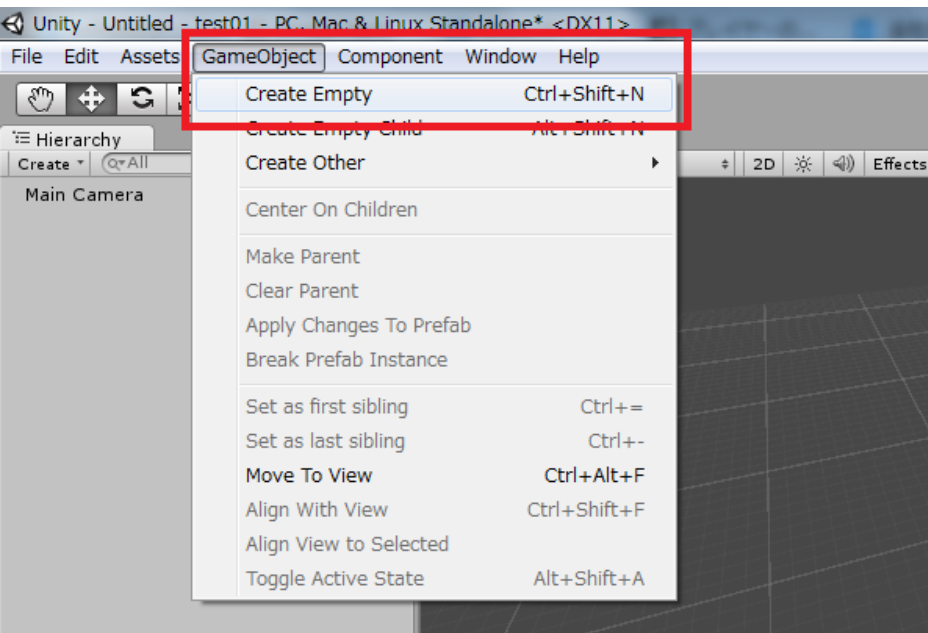
※参考資料※ NCMBSettingsの設定(作業済)

※今回は本ページの作業を行う必要はありません。

mBaaSへの認証のための設定を行う
NCMBSettingsスクリプトを利用するための作業を行います

Asset>Scenesの「Start」シーンを開いて
空のGame Objectを作成し
「NCMBSettings」にリネームします

Asset>NCMBの「NCMBSettings.cs」を
「NCMBSettings」のゲームオブジェクト
にアタッチします



まず、mBaaSのアプリ作成で管理画面を開いた方にキーの見方をご案内します

The image shows a two-step process for finding the API key in the NIFTY Cloud mobile backend interface.

Step 1: Dashboard Overview

The dashboard for 'mobile backend' (IoTBlogApp) shows the following status for June 2015:

- 登録会員数 (Registered Members): 0人
- APIリクエスト (API Requests): 0回 (Limit: 残り49,997,161回)
- プッシュ通知 (Push Notifications): 0回 (Limit: 残り10,000,000回)
- ストレージ (Storage): 0.00MB (Limit: 残り102,291.48MB)

A callout box with the text "アプリ設定をクリック" (Click Application Settings) points to the "アプリ設定" (Application Settings) icon in the left sidebar.

Step 2: Application Settings

The "アプリ設定: 基本" (Application Settings: Basic) page is shown. A callout box with the text "ここにあります" (It is here) points to the "APIキー" (API Key) section.

API Key Details:

- アプリケーションキー (Application Key): 2417fd23d90bdc35ba48f5b0b48164ec6a4021c139e3419 [コピー]
- クライアントキー (Client Key): b4cc5b1c55ad012f7c91b2ddc4e2a65f75c0fa29d447d1a5 [コピー]

UnityのHierarchy配下にあるNCMBSettingsをクリックしそのインスペクター部分にアプリケーションキー、クライアントキーを設定します

The image shows a two-part process for setting up NIFTY Cloud keys in Unity. The top part is a screenshot of the Unity Inspector window. The 'NCMBSettings (Script)' component is selected, and its 'Application Key' and 'Client Key' fields are highlighted with a red oval. A red arrow points from this oval to a text box that says 'コピー＆ペーストで貼り付けて設定が行えます' (You can set it by copying and pasting). The bottom part is a screenshot of the NIFTY Cloud console. It shows the 'アプリケーションキー' (Application Key) and 'クライアントキー' (Client Key) fields, each with a 'コピー' (Copy) button. These fields and buttons are circled in red. The console also displays statistics for 'プッシュ通知' (Push Notifications) and 'ストレージ' (Storage).

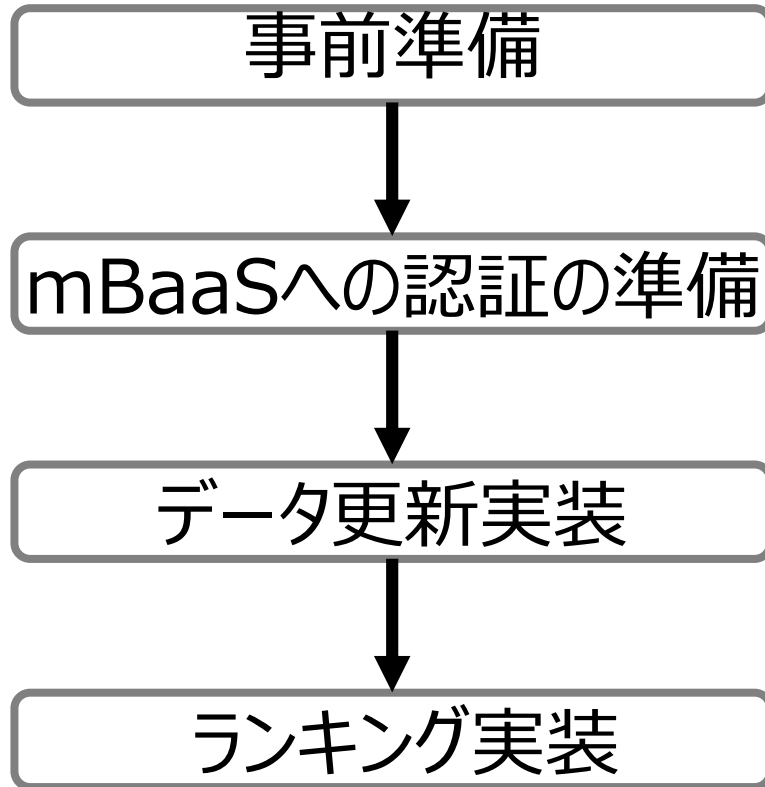
Inspector Window:

- NCMBSettings (Script)
- Application Key
- Client Key
- Use Push
- Android Sender Id
- Response Validation

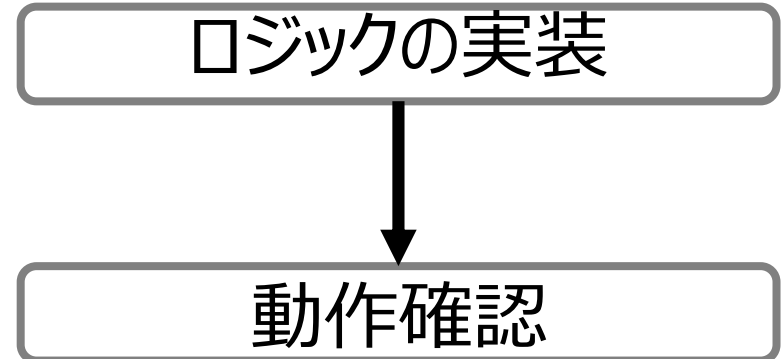
NIFTY Cloud Console:

- 2013年9月
- プッシュ通知: 0回
- ストレージ: 0.00MB
- アプリケーションキー: ca57501b099fb9d8a6c2fb32fc41a79fa22858fb514102459a684; [コピー]
- クライアントキー: 3863cbc8884b80f047bce327d92ecc5c11a8288c9e1b7f6b5e6b2 [コピー]

mBaaSへの認証の準備が終わりましたのでいよいよ
mBaaSのSDKを本格的に利用する段階に移ります



「データ更新実装」にておこなうこと

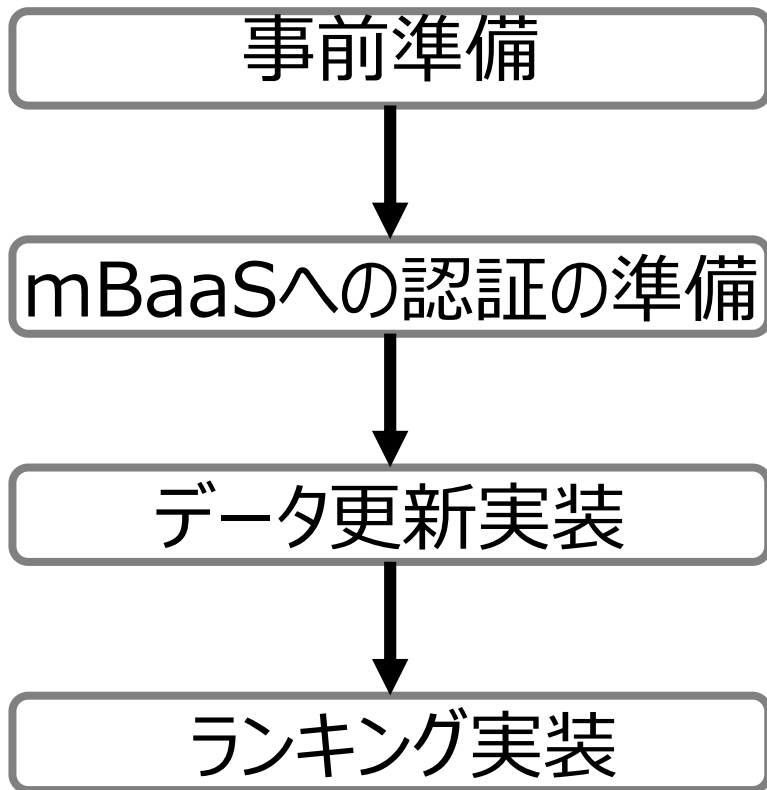


Assets>Scripts>Runner内にあるスクリプト「Timer.cs」の77行目からはじまるOnGoalに下記の黒文字で書かれているコードを実装してください。

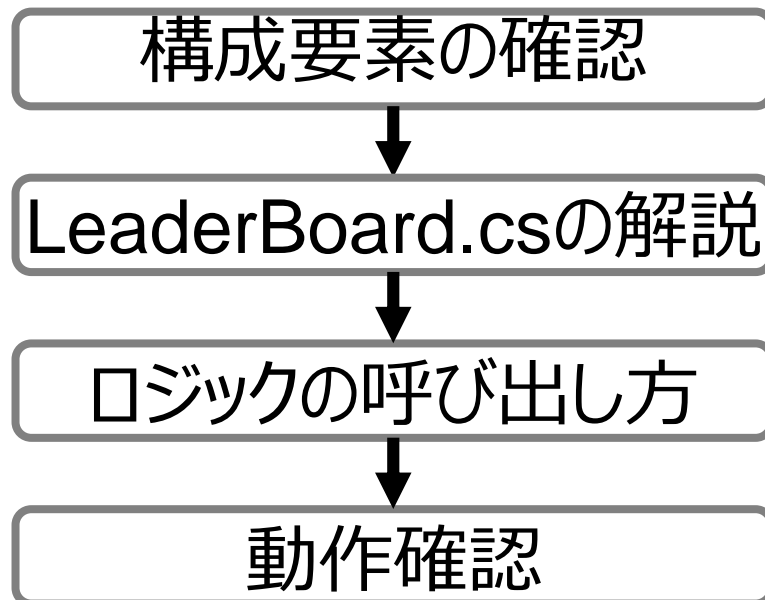
```
using UnityEngine;
using System.Collections;
using NCMB;
.
. (省略)
.
// Goal到着が検知されたとき
void OnGoal(){
    goal = true;
    //クラスの指定
    NCMBObject timeClass = new NCMBObject("Time");
    //カラムと挿入するデータの指定
    timeClass["time"] = lapTime;
    //非同期でのアップロード
    timeClass.SaveAsync();
}
```

実際にゲームを実行し、データがクラウドに保存されるか確認してください。

データストアの取得を応用し、ランキングを作成します。ハンズオンではコーディングはせず、ランキングの構成要素とロジックの解説を行います。



「ランキング実装」にておこなうこと



ランキングはAsset> Scripts > LeaderBoard のスクリプトで取得を、描画をTimer.csで行っています。
構成要素を整理して確認します。

- LeaderBoardフォルダには以下のスクリプトが含まれています
 - LeaderBoard.cs
 - mobile backendと接続して経過時間を引き出す
 - Rankers.cs
 - 引き出したスコアと名前を一時的に保存するもの
- ランキングの描画は以下のメソッドで行っております
 - Timer.csの55行目から71行目

Assets > Scripts > LeaderBoard内にあるスクリプト「LeaderBoard.cs」を参照ください

```
public void fetchTopRankers(){
```

```
// データストアの「HighScore」クラスから検索
```

時間を保存しているTimeクラスを操作する
クエリの作成

```
NCMBQuery<NCMBObject> query = new NCMBQuery<NCMBObject>  
("Time");
```

```
query.OrderByAscending ("time");
```

timeカラムを昇順に並び替えて
引き出すよう設定

```
query.Limit = 5;
```

5個だけ取り出すように設定

Assets > Scripts > LeaderBoard内にあるスクリプト「LeaderBoard.cs」を参照ください

```
query.FindAsync ((List<NCMBObject> objList ,NCMBException e) => {
```

時間を引き出す操作

```
    if (e != null) { //検索失敗時の処理  
    } else { //検索成功時の処理
```

```
        List<NCMB.Rankers> list = new  
List<NCMB.Rankers>();  
        foreach (NCMBObject obj in objList) {
```

引き出したオブジェクトからtimeだけを取り出す

引き出した
時間をRankers
クラスに保存

```
            string t = System.Convert.ToString(obj[" time "]);  
            list.Add( new Rankers( t ) );
```

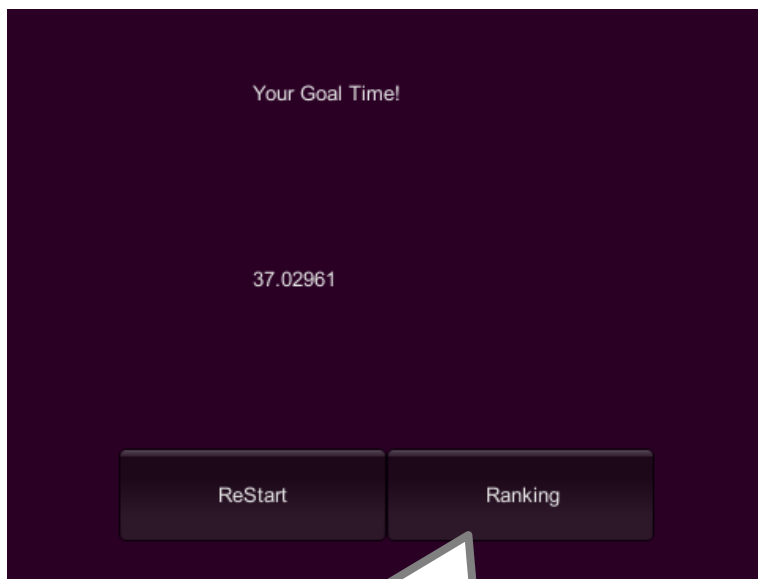
今回、LeaderBoard.csはUnityに依存しない形で実装しています。
開発状態によってはこのようにUnityに依存しない形で実装することもあるかと思うしますので
その場合のメソッドの呼び出し方についても記載しておきます

```
LeaderBoard lBoard;
```

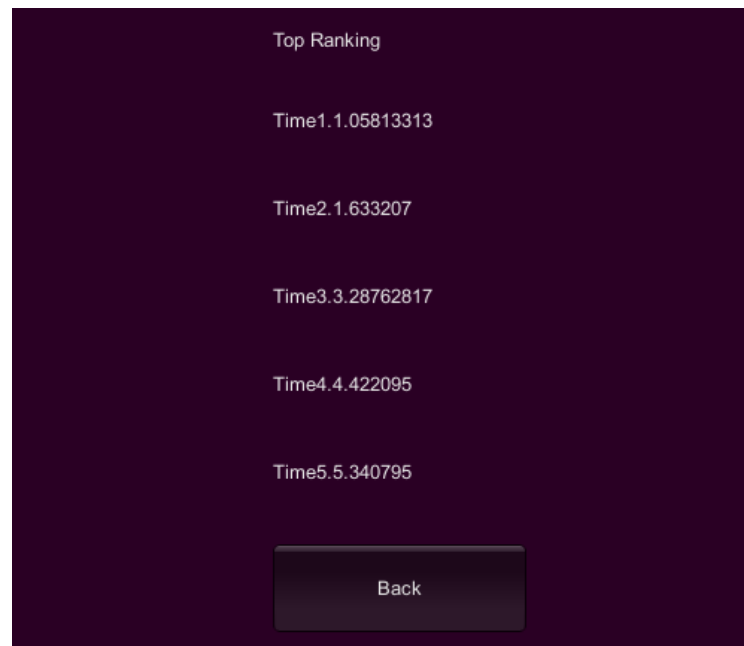
```
lBoard = new LeaderBoard(); // LeaderBoard クラスのインスタンス生成
```

```
lBoard.fetchTopRankers(); //メソッドの呼び出し
```

一度Asset>Scenesの「Start」シーンに戻っていただきゲームを実行してゴールしてください
ゴール時の画面の下部「Ranking」ボタンをクリックしランキングが表示されることを確認してください



ランキングに
遷移



ここをクリック

上記のようにランキングが表示されたら成功です！

ゴール後管理画面よりデータストアをご確認いただき左記のようにデータが入っていることもご確認ください

 ダッシュボード 会員管理 データストア	クラス	+作成 ▼	Time	+新しいレコード	+新しいフィールド	削除
	installation					
	role					
	Time					
	<input type="checkbox"/>	objectId	time	createDate ▼		
	<input type="checkbox"/>	dhra58NARCandohh	35.0738	2015-07-02T00:		
	<input type="checkbox"/>	15djei7tV8Tb1WnY	15.7273827	2015-06-29T21:		
	<input type="checkbox"/>	HGTyfuPQhstSjZvH	58.31019	2015-06-29T21:		

ハンズオンは以上にて終了です。お疲れ様でした！
本日は以下のことを体験していただきました。

- ・ゲームスコアの保存
- ・ランキングの実装

またmBaaSはデータストアだけでなく、プッシュ通知の配信なども行えます。
プッシュ通知の実装に関するドキュメントもございますのでこちらもぜひ一度お試しください。

プッシュ通知の解説ドキュメント

<http://mb.cloud.nifty.com/doc/sdkguide/unity/push.html>

この他にもなにかわからないことがありましたら
ユーザーコミュニティもご用意していますのでそちらもぜひご利用ください！

ユーザーコミュニティ

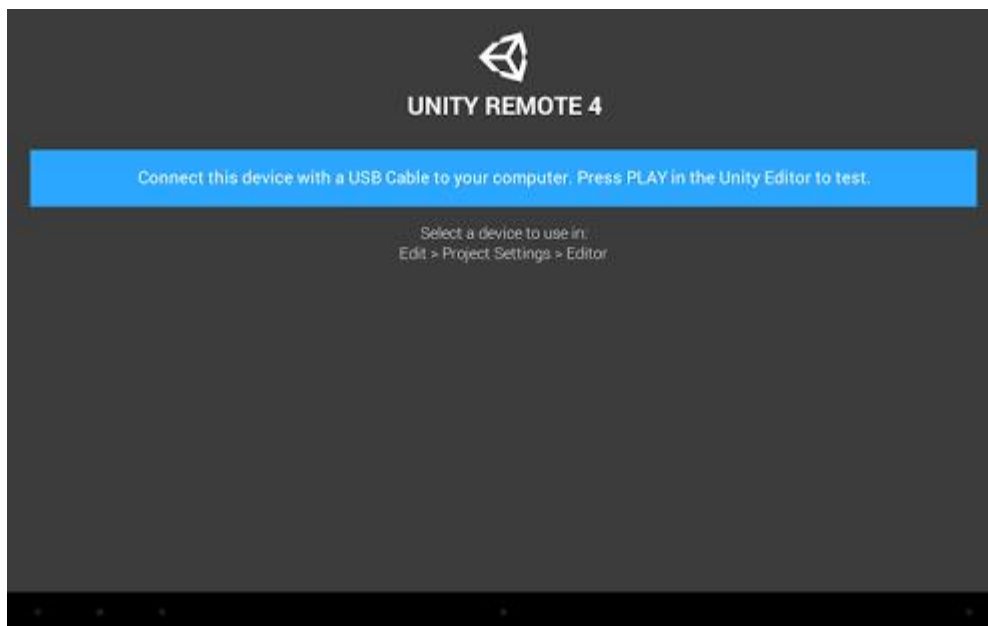
<https://github.com/NIFTYCloud-mbaas/UserCommunity>

本日はお忙しい中、ハンズオンにお越しいただきありがとうございました。
今後ともニフティクラウドmobile backendをなにとぞよろしくお願いいたします。

このハコスコアプリをUnityRemote4でデバッグできます。

デバッグの方法は下記のUnityRemote4のドキュメントに書かれています。

<http://docs.unity3d.com/ja/current/Manual/UnityRemote4.html>



ビルドをしなくても
スマホでデバッグできます。

加速度などのセンサの値も
取得できるのでハコスコなどの
実機動作確認を行う際に使えます

Android:

<https://play.google.com/store/apps/details?id=com.unity3d.genericremote>

iOS:

<https://itunes.apple.com/us/app/unity-remote-4/id871767552>

今回障害物のプロジェクトは
多くのUnityチュートリアルを執筆されている
hiyotamaさんブログのcreatescript

<http://hiyotama.hatenablog.com/entry/2015/05/09/080000>

を参考にさせていただいています。

素晴らしいUnityの解説がたくさん載っています
ので皆さん、ぜひお読みください。

hiyotamaさんありがとうございます！

ニフティとなら、きっとかなう。
With Us, You Can.