

Unityゲーム × ニフティクラウドmobile backend 体験会

- ごあいさつ
- ニフティクラウドmobile backendについて
- 体験会について
- 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。
本体験会はニフティクラウドmobile backendを
使ってUnityゲームにサーバー機能を
「**素早く 無料**」で導入する体験会です。

本体験会では下記の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



SDKで
できる
こと

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



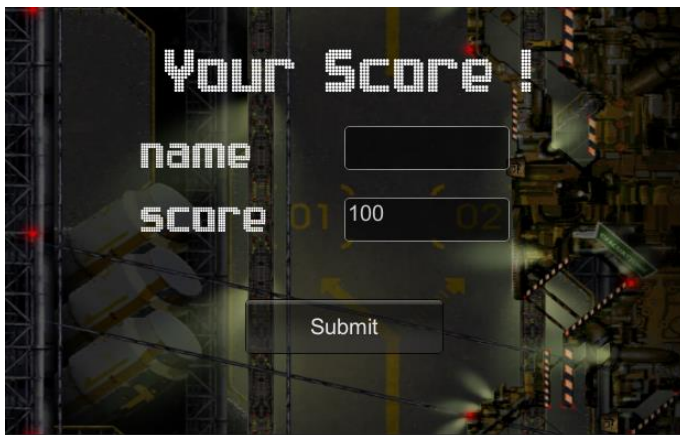
リッチプッシュ通知を開く



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



Unityチュートリアル「2Dシューティング」に
ニフティクラウドmobile backend(mBaaS)を導入します。

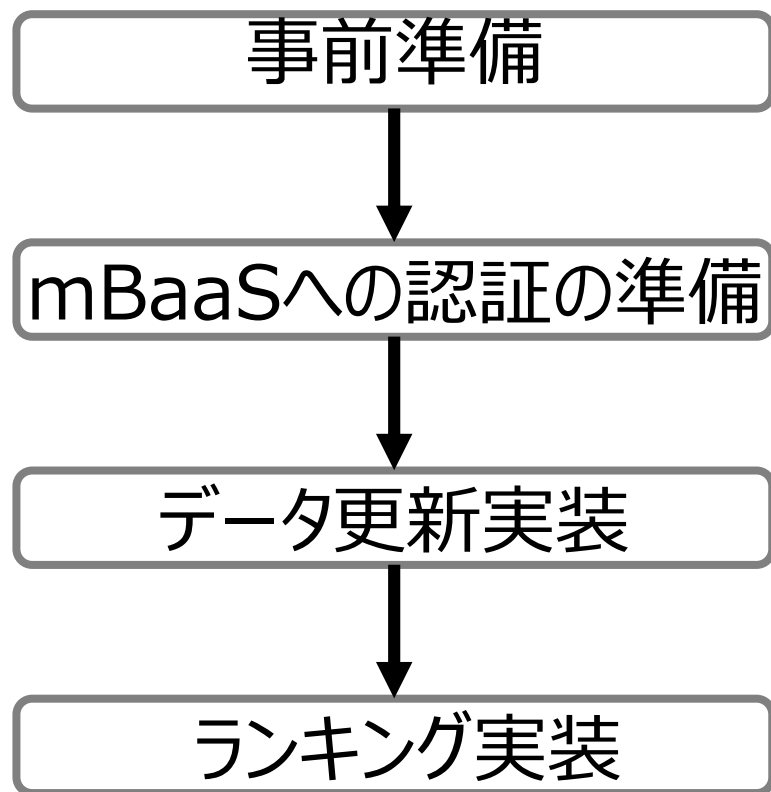
この体験会では下記の二つのことが体験できます

- ・ゲームスコアの保存
- ・ランキングの実装

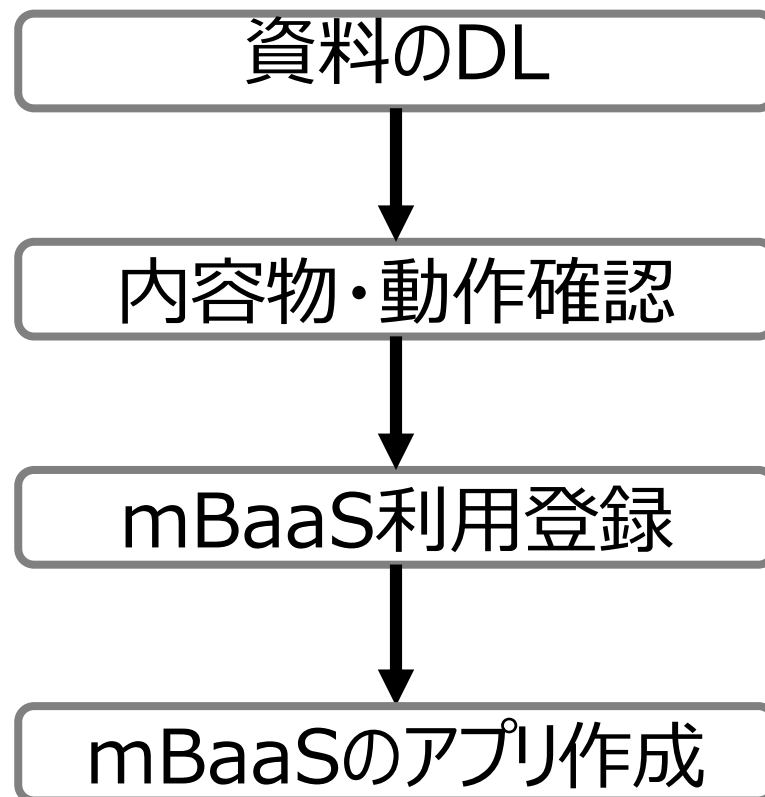
どちらもアーケード形式を想定してご案内いたします。
30分ほどのお時間をいただくかと思いますが
なにとぞよろしくお願いいたします。



本日の体験会は大きく分けて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_BitSummit_2DShooting

hounenhounen / NCMB_BitSummit_2DShooting

Unwatch 1 Star 0 Fork 0

Description Website

Short description of this repository Website for this repository (optional) Save or Cancel

2 commits 1 branch 0 releases 1 contributor

branch: master NCMB_BitSummit_2DShooting / +

Commit2	Commit2	Commit2
hounenhounen authored an hour ago	latest commit 9d40c5d97f	
complete	Commit2	an hour ago
handson	Commit2	an hour ago
NCMB_BitSummit_2DShooting.pptx	Commit2	an hour ago

Help people interested in this repository understand your project by adding a README! Add a README

HTTPS clone URL
https://github.com/hounenhounen/NCMB_BitSummit_2DShooting

You can clone with HTTPS, SSH, or Subversion

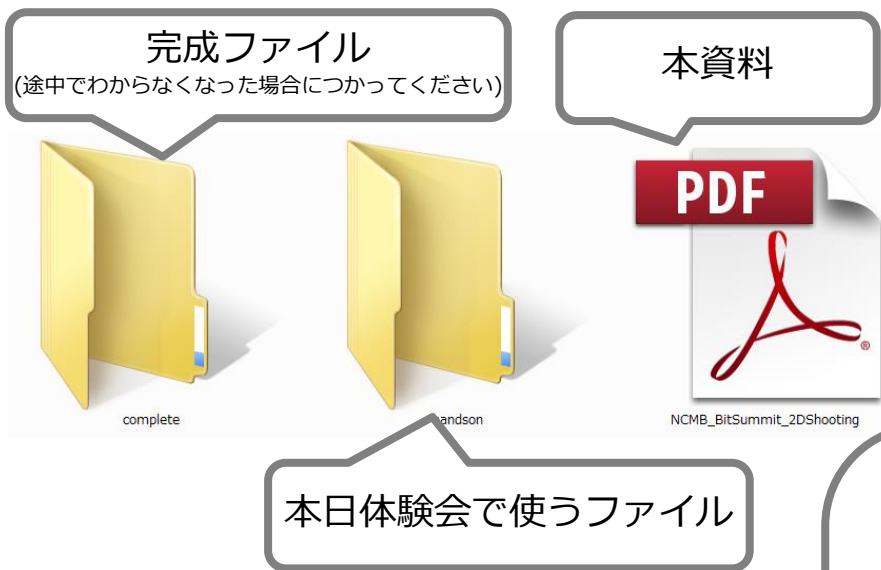
Clone in Desktop

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



動作確認

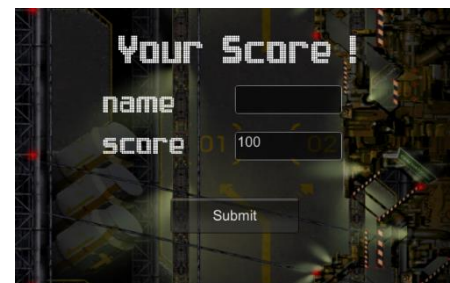
体験会で使うフォルダの中にAsset> Scenes Stage.sceneをダブルクリックしてゲームを実行してください 実行時とゲームオーバー時に下記の画面が出ればひとまず大丈夫です

実行・ゲームオーバー時に
下記の画面が出ればOK

ゲーム実行時の画面



ゲームオーバー時の画面



注意

1. 本資料をおくフォルダのパスに全角文字は使用なさないでください 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

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

@nifty
@nifty会員登録

[step1]
登録内容の入力

[登録の流れ]
[step1] 登録内容の入力
[step2] 登録内容の確認
[step3] 登録完了

@nifty会員へご登録いただきますと、@niftyのサービスがご利用できます。
登録料や会費はもちろん無料です！ご利用料金は有料サービスのご利用時のみ必要になります。

基本情報の入力

希望する @niftyユーザー名 ※必須項目	@niftyのサービスへログインする時に使用します。 例) amori-2210 [半角文字] <input type="text"/> ユーザー名が使用できるか調べる	・登録後の変更はできません。 ・使用できる文字は、半角英数字、ハイフン(-)、アンダーバー(_)、ピリオド(.)です。 ・ その他入力時における注意
パスワード ※必須項目	@niftyのサービスへログインする時に使用します。 [半角6～24文字] <input type="password"/> パスワードの安全性 <div></div> 確認のため、もう一度入力してください。 <input type="password"/>	・使用できる文字 ・英大文字と英小文字が区別されます。 ・@niftyユーザー名と同じものは使用できません。 ・ 【ご注意】パスワードの不正利用を防ぐために
連絡先メールアドレス ※いずれか必須項目	【パソコンメールアドレス】 <input type="text"/> 【携帯メールアドレス】 <input type="text"/> @ <input type="text"/>	・携帯メールアドレスを登録する人は、 メール指 定受信の設定を確認してください 。 ・スマートフォンをご利用の場合は、パソコンメールアドレス欄にパソコンでも使用可能なメールアドレスを入力してください。

[@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および@nifty会員規約に同意し【登録内容の確認へ】

※SSL通信により暗号化されます。

[@nifty会員規約印刷用](#)
2011年4月21日
< @nifty 会員規約 >

@nifty
@nifty会員登録

[step2]
登録内容の確認

[登録の流れ]
[step1] 登録内容の入力
[step2] 登録内容の確認
[step3] 登録完了

以下の内容をお客様の@nifty会員情報として登録いたします。

@niftyユーザー名

登録後の変更はできませんのでご注意ください。

パスワード

連絡先メールアドレス
【パソコンメールアドレス】

【携帯メールアドレス】

@nifty Top

@nifty
@nifty会員登録

[step3]
登録完了

[登録の流れ]
[step1] 登録内容の入力
[step2] 登録内容の確認
[step3] 登録完了

ご登録ありがとうございます。

@niftyユーザー名

パスワード

@nifty ID

[@nifty IDとは？](#)

連絡先メールアドレス
【パソコンメールアドレス】

メールを送りました。

メールに記録されているアドレスへアクセスし、メール確認手続きを行ってください。
メール確認が済みでない場合は、@niftyユーザーズポイント、コミュニティサービスを利用することができません。

[ご利用中のサービスへ戻る](#)
(このウィンドウを閉じる)

@niftyユーザー名とパスワードは@niftyのサービスにログインする際に使用します。あふれにならないようご注意ください。

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。

「BitSummit_2DShooting」
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てみましょう！
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。

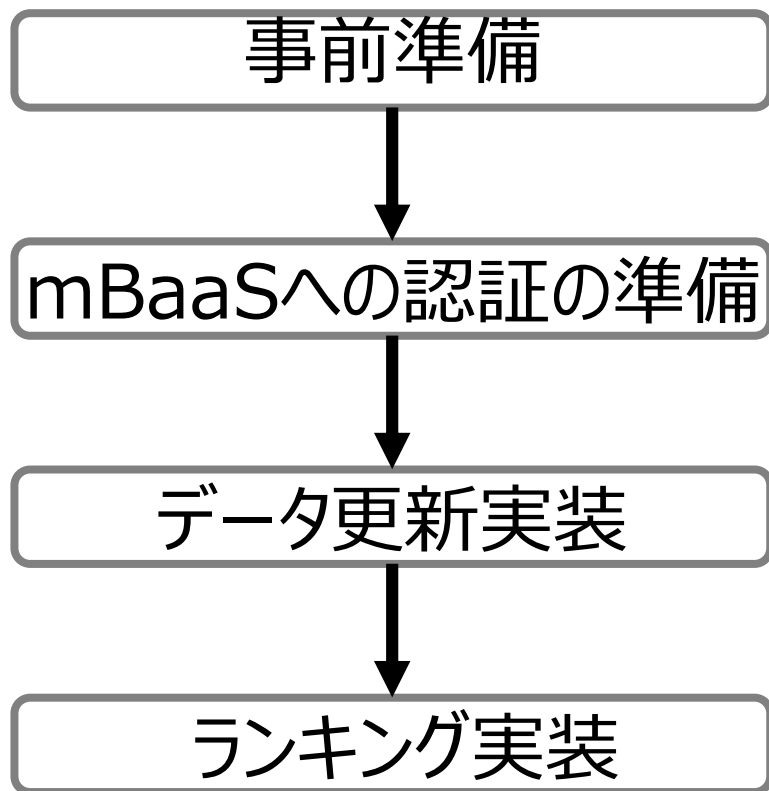
OK

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への認証をとおす」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

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

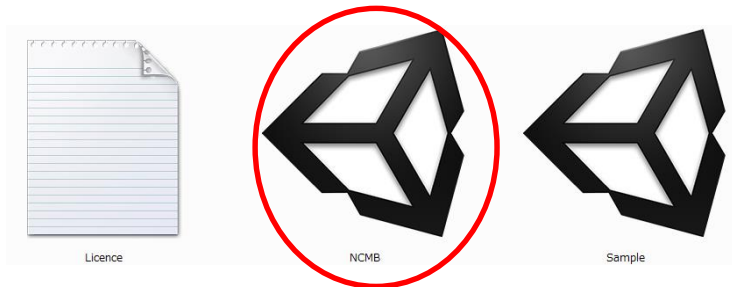
Source code (zip)

Source code (tar.gz)

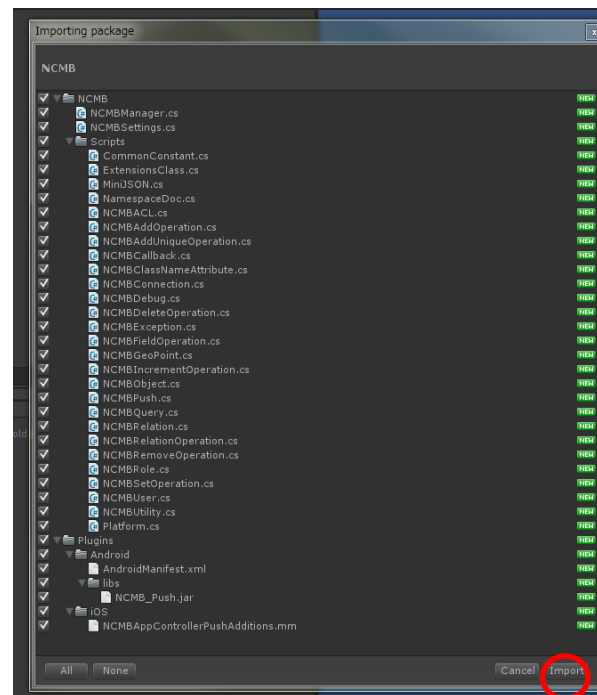
クリックすると
DLできます

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

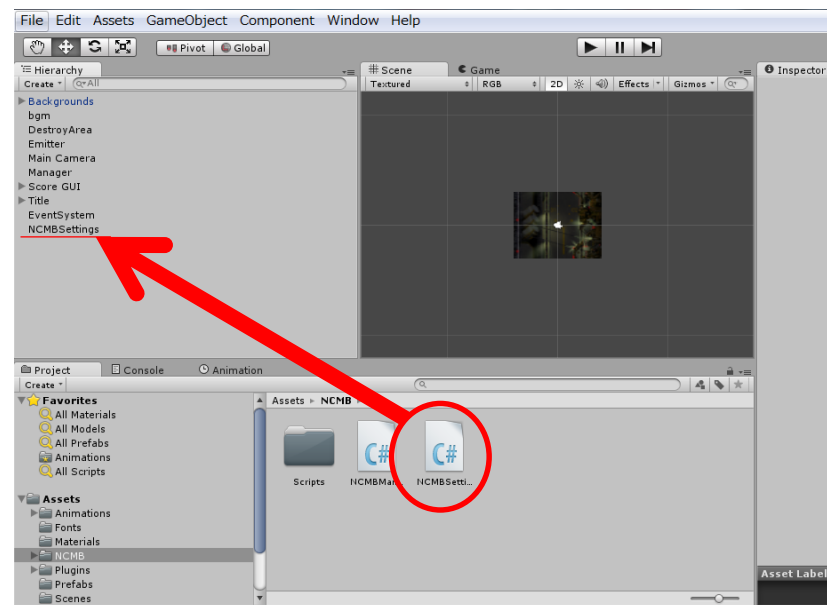
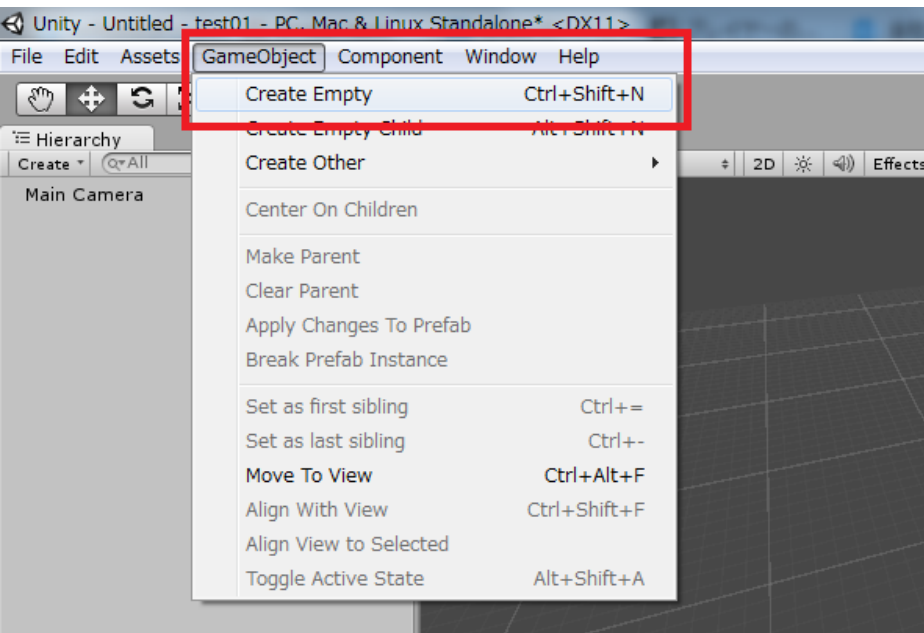


表示された
画面の「Import」
をクリックすると
Import可能

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「Stage」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 (ダッシュボード) for the application "IoTBlogApp" displays 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 graph (グラフ) for APIリクエスト数 is shown for the period from 2015/05/03 to 2015/06/02.

A callout bubble 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. The left sidebar lists various settings categories: ダッシュボード, プッシュ通知, データ・ファイルストア, コラボレータ, 会員認証設定, SNS連携, エクスポート, アプリ設定 (highlighted), and 連携サービス.

The "アプリ基本" (Application Basic) section contains the following fields:

- アプリ名 (Application Name): IoTBlogApp
- App Store URL
- Google Play URL
- メモ (Memo)

A green "保存する" (Save) button is located below these fields.

A callout bubble with the text "ここにあります" (It is here) points to the "APIキー" (API Key) section, which is highlighted with a green box. This section contains:

- 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]

The footer of the interface includes links for @nifty Top, @nifty 会費に賛同, お問い合わせ, 利用規約, 個人情報保護ポリシー, and the copyright notice ©NIFTY Corporation.

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 composite of two screenshots. The top screenshot is from the Unity Inspector, displaying the 'NCMBSettings (Script)' component. The '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 screenshot is from the NIFTY Cloud console, showing 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. These fields contain long alphanumeric strings and are also highlighted with a red oval. A 'コピー' (Copy) button is visible next to each key.

Inspector Console

NCMBSettings

Tag Untagged Layer Default

Transform

Position X 0 Y 0 Z 0

Rotation X 0 Y 0 Z 0

Scale X 1 Y 1 Z 1

NCMBSettings (Script)

Script NCMBSettings

Application Key

Client Key

Use Push

Android Sender Id

Response Validation

Add Component

2013年9月

プッシュ通知 0回

ストレージ 0.00MB

残り49,999,921回

残り9,999,998回

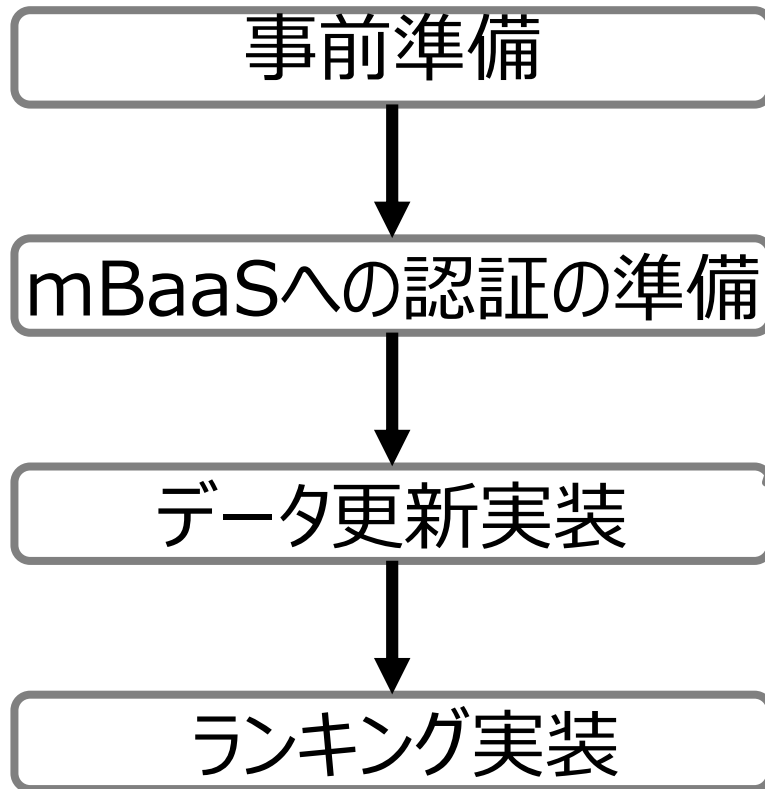
空き102,399.72MB

アプリケーションキー ca57501b099fb9d8a6c2fb32fc41a79fa22858fb514102459a684; コピー

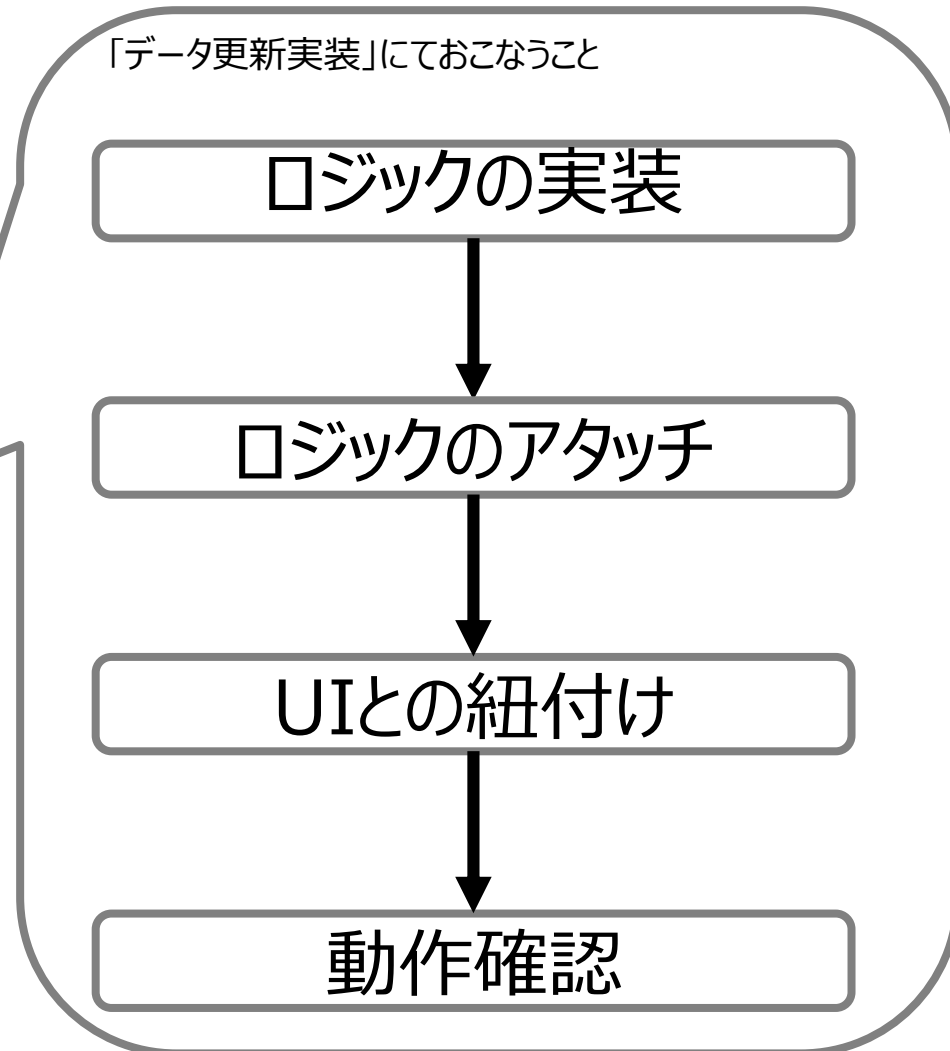
クライアントキー 3863cbc8884b80f047bce327d92ecc5c11a8288c9e1b7f6b5e6b2 コピー

コピー＆ペーストで貼り付けて設定が行えます

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



Asset>Scenesの「 SaveScore 」 シーンを開きAssets>Scriptに空の「C# Script」を作り
SaveScoreとリネームし下記のコードを実装してください

```
using NCMB; //mobile backendのSDKを読み込む
using UnityEngine;
using System.Collections;
using System.Collections.Generic;

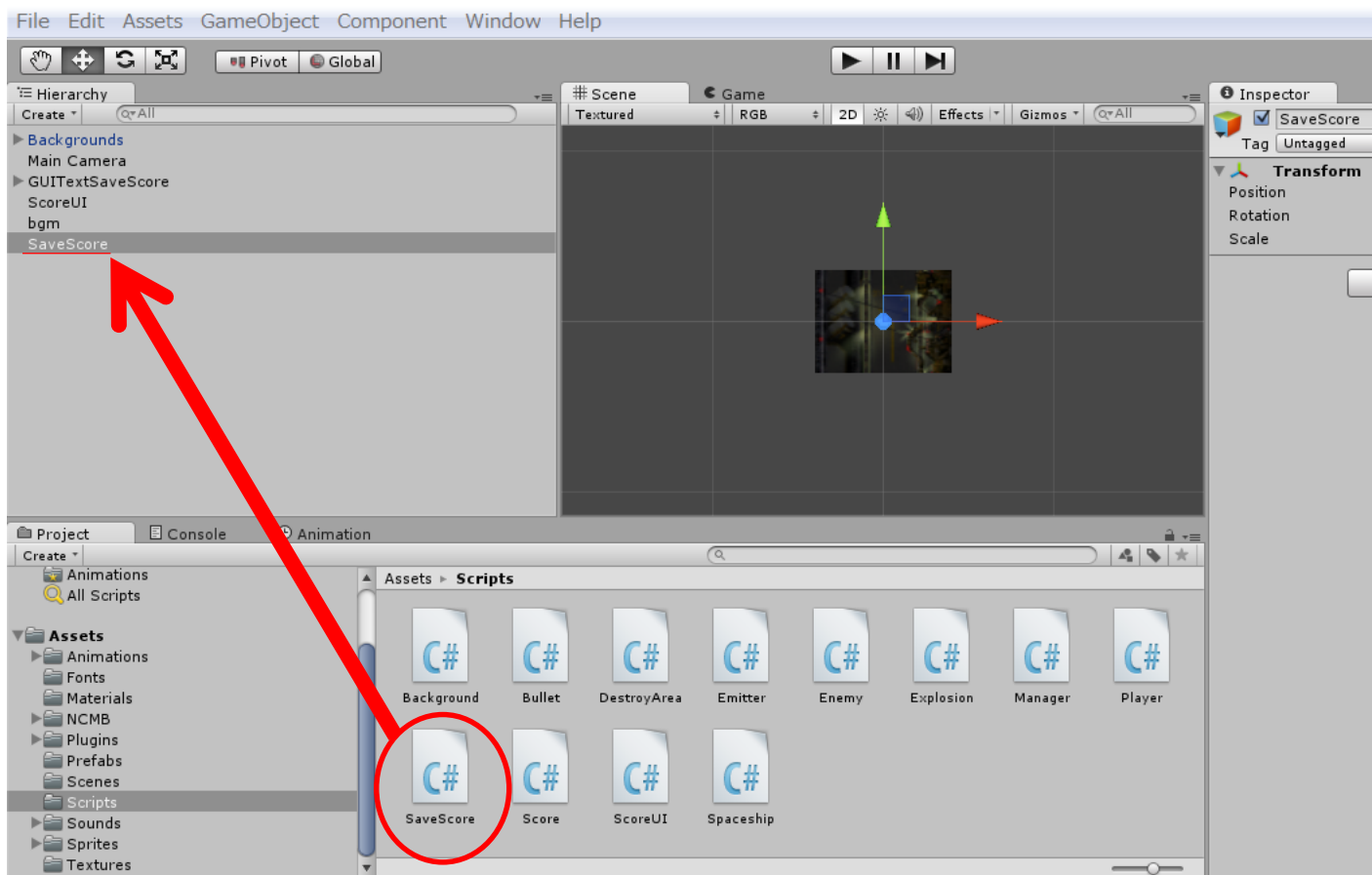
public class SaveScore : MonoBehaviour {
    // mobile backendに接続-----
    public void save( string name, int score ) {
        NCMBObject obj = new NCMBObject ("Score");
        obj.Add ("name", name);
        obj.Add ("score", score);
        obj.SaveAsync ((NCMBException e) => { //非同期でサーバーに書き込む
            if (e != null) {
                //エラー処理
            } else {
                //成功時の処理
            }
        });
    }
}
```

更新するクラスを指定

オブジェクトに名前とスコアを設定

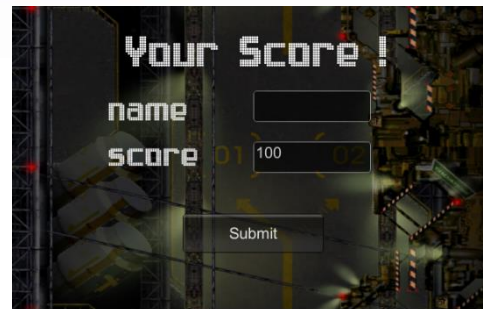
データストアに
保存する操作

Asset>Scenesの「 SaveScore 」 シーンに「 SaveScore」 Game Objectを作成し
先ほどの「 SaveScore.cs」 スクリプトをアタッチしてください



UIとの紐付け

Asset>Scriptの「ScoreUI.cs」の
OnGUIのコメントアウト(赤丸部分)を消去すると
左のゲームオーバー画面の
「Submit」ボタン押した際に
入力した名前とスコアを保存できるようになります



```
void OnGUI () {  
    drawSaveScore();  
    // ログインボタンが押されたら  
    if (SubmitButton) {  
        //FindObjectOfType<SaveScore> ().save (name, score);  
        Application.LoadLevel ("Stage");  
    }  
}
```

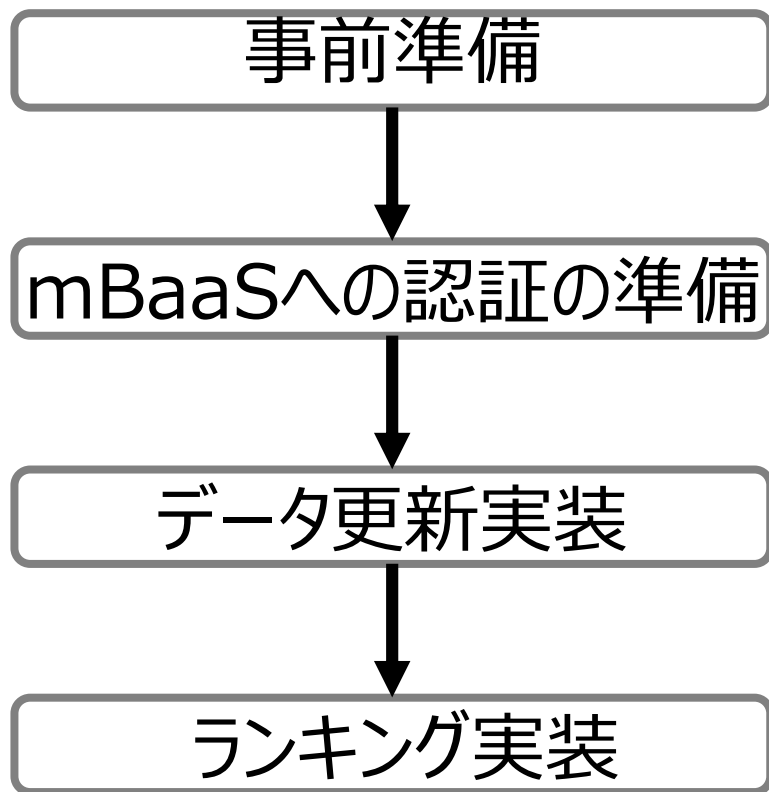
動作確認

一度Asset>Scenesの「 Stage 」シーンに
戻っていただきゲームを実行してください

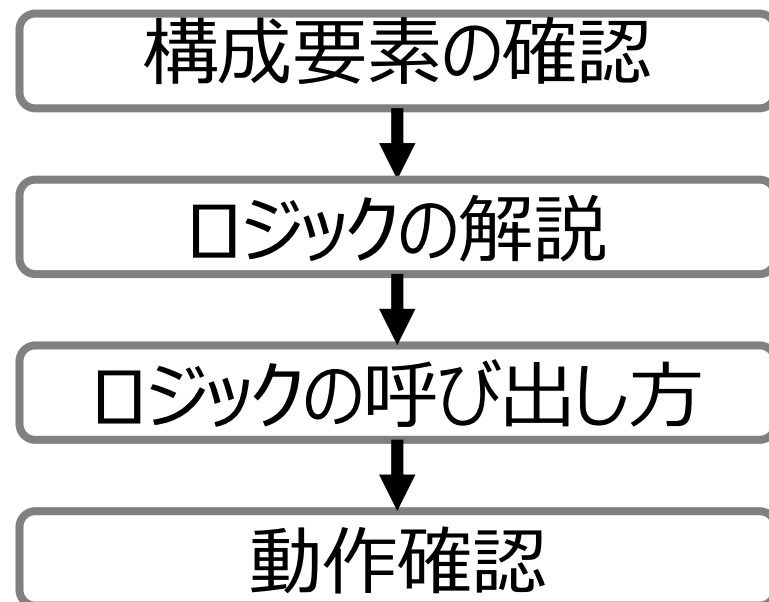
ゲームオーバー後
管理画面よりデータストアを
ご確認ください左記のように
データが入っていれば成功です

クラス		Score		+新しいフィールド		削除	
+作成		+新しいレコード					
installation		objectId		name		score	
role		FPaz1OkWreNdUcxF		test1		2100	
Score						createDate	
						2015-03-24T16:	

データストアの取得を応用し、ランキングを作成します。本体験会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>Scenesの「LeaderBoard」シーンで生成されています。
下記にその構成要素と各役割を記します。

- LeaderBoardシーンには以下のスクリプトが含まれています
 - LeaderBoard.cs
 - mobile backendと接続してスコアと名前を引き出す
 - LeaderBoardManager.cs
 - ランキングの表示部、LeaderBoard.csを呼び
スコアと名前を引き出しランキングを描画
 - Rankers.cs
 - 引き出したスコアと名前を一時的に保存するもの
- スクリプト以外のものは以下になります
 - 各種見出しGUITextオブジェクト
 - 順位とプレイヤー名、スコアを表示する
GUITextオブジェクト

mBaaSデータを取得するロジックはAsset>Scripts 「LeaderBoard.cs」に記載されています。

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

```
// データストアの「HighScore」クラスから検索
```

スコアを保存しているScoreクラスを
操作するクエリの作成

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

```
query.OrderByDescending ("score");
```

scoreカラムを降順に並び替えて
引き出すよう設定

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

5個だけ取り出すように設定

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

データストアからスコアと名前を引き出す操作

```
    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 (NCMBObj obj in objList) {
```

引き出したオブジェクトからscoreだけを取り出す

```
            Int  s = System.Convert.ToInt32(obj["score"]);
```

引き出したオブジェクトからnameだけを取り出す

```
            string n = System.Convert.ToString(obj["name"]);  
            list.Add( new Rankers( s, n ) );
```

----以下省略----

引き出した
スコアと名前を
Rankersクラスに保存

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
実装したいこともあるかと思います。
その場合のメソッドの呼び出し方についても下記に解説いたします。

```
LeaderBoard IBoard;
```

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

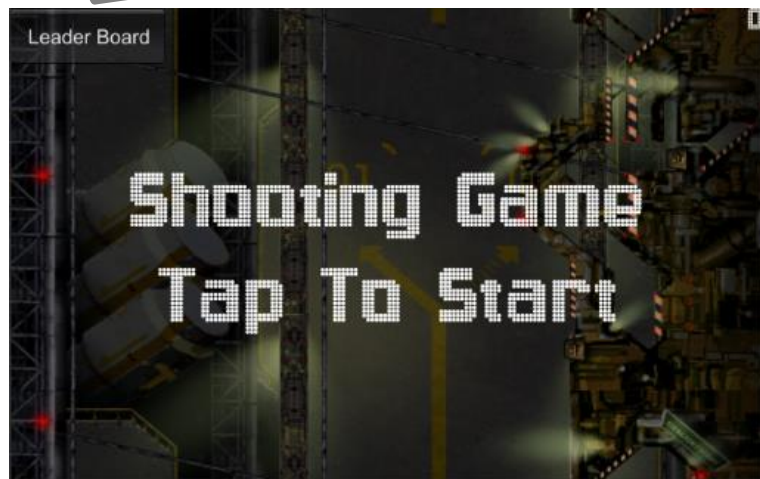
```
IBoard.fetchTopRankers(); //メソッドの呼び出し
```

なお、上記のコードはAssets>Scripts の「LeaderBoardManager.cs」にて記載されています

一度Asset>Scenesの「 Stage 」 シーンに戻っていただきゲームを実行してください

スタート時の画面の左上
「Leader Board」 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

ここをクリック



ランキングに
遷移



上記のように名前とスコアが表示されたら成功です！

体験会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、体験会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。

ニフティとなら、きっとかなう。
With Us, You Can.