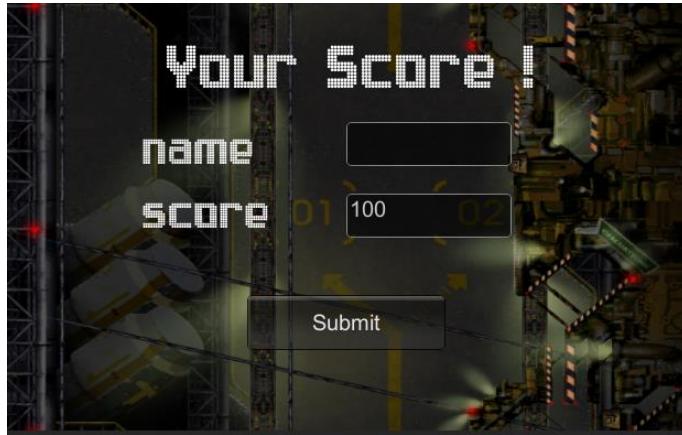


# Unityゲーム× ニフティクラウドmobile backend 体験会



## オンラインランキングを実装する



Unityチュートリアルの「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。  
5分～15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ひいたします。



- 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との接続
- スコア保存実装の流れ
- ランキング実装の流れ

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！

# ニフティクラウドmobile backendとは？

NIFTY Cloud  
ニフティ クラウド



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



プッシュ通知



会員管理・認証



SNS連携



データストア



ファイルストア



位置情報検索

# Unityに導入することで実現できること

## UnitySDKで提供中の機能



Android・iOS各プラットフォームへの  
プッシュ通知配信



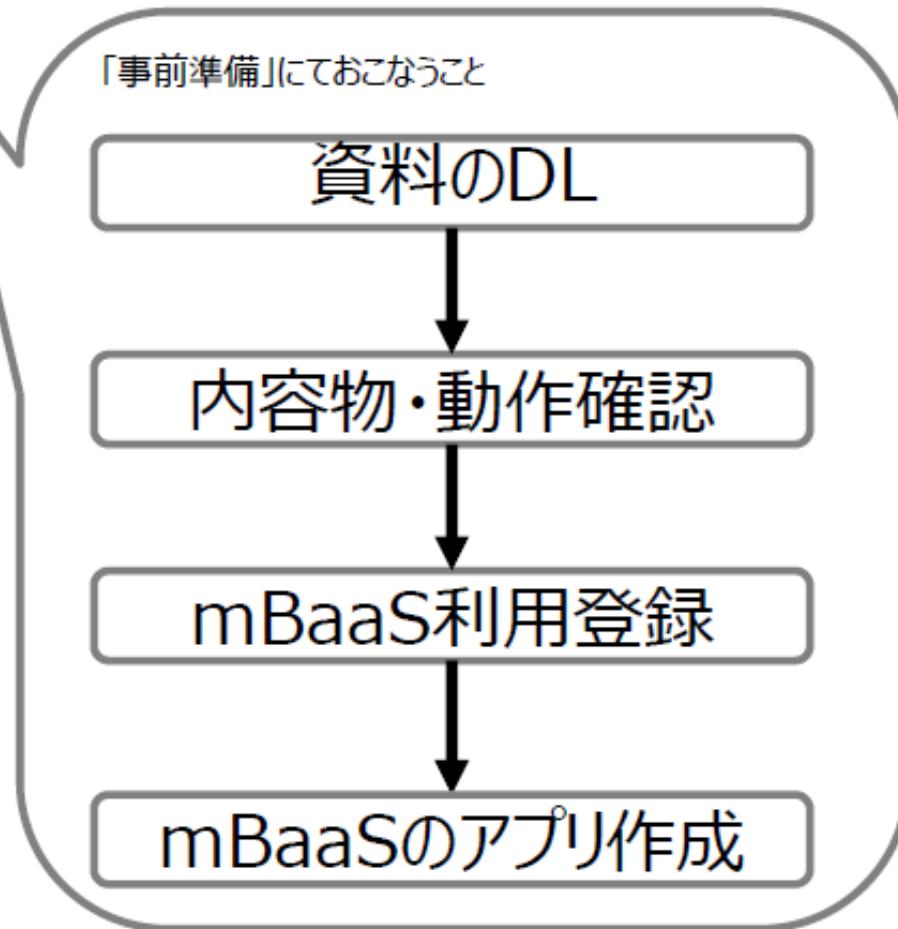
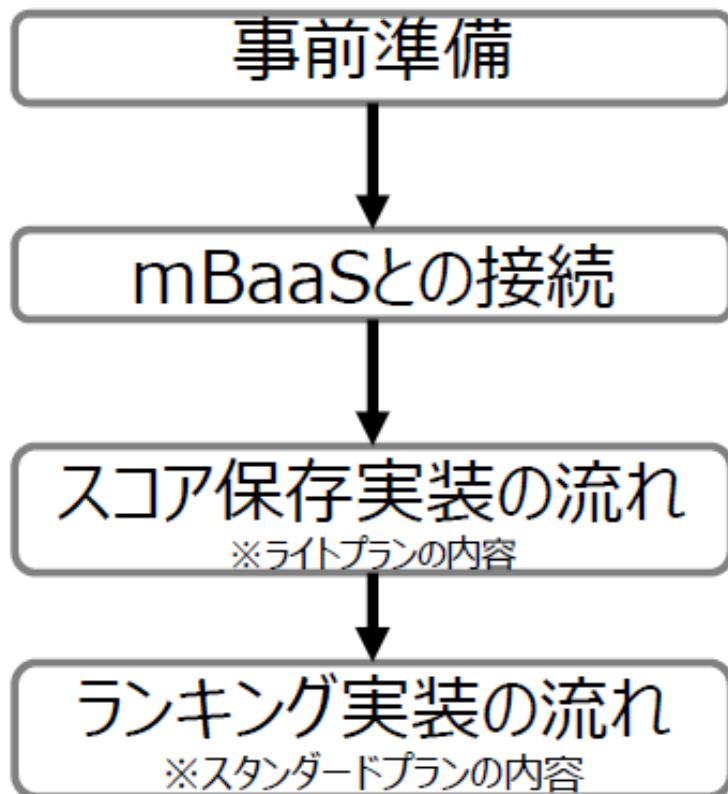
## ユーザーログイン認証



## オンラインランキング



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。



※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\\_2DShooting](https://github.com/hounenhounen/NCMB_2DShooting)

hounenhounen / NCMB\_BitSummit\_2DShooting

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

hounenhounen authored an hour ago latest commit 9d40c5d07f

complete Commit2 an hour ago

handson Commit2 an hour ago

NCMB\_BitSummit\_2DShooting.pptx Commit2 an hour ago

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・動作確認

## 内容物確認

DLファイルフォルダには、以下の2つが含まれています。



## 動作確認

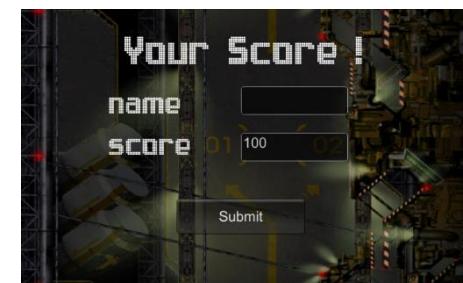
体験会で使うフォルダの中にAsset> Scenes Stage.sceneをダブルクリックしてゲームを実行してください 実行時とゲームオーバー時に下記の画面が出ればひとまず大丈夫です。



ゲーム実行時の画面



ゲームオーバー時の画面



## 注意

1. 本資料を置くフォルダのパスに全角文字は使用しないでください。エラーが表示されることがあります。
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が必要です。

# mBaaSの利用登録 1/2

NIFTY Cloud  
ニフティクラウド

まず、下記URLよりmBaaSのサービスサイトにアクセスしてください。

<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。



## 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

法人会員として

ここをクリック

ニフティクラウド mobile backendをご利用いただくには、  
@nifty会員登録が必要になります。登録料金は無料です。

@nifty会員の登録（無料）

すでに@nifty会員の方はSTEP2へお進みください。

ニフティの法人会員としてご契約  
mobile backend利用料金を企業  
Webまたは郵送にて@nifty法人ID  
い。なお、法人としてご契約いた  
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」  
行ってください。なお、申込書

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を試験してください！

基本情報の入力

ユーザー名  入力時における注意  
[半角6~24文字]

パスワード  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  入力されるメールアドレスについて

個人情報の取り扱い及び@nifty会員規約に同意し  
 登録

利用規約を  
確認後、同意して  
利用開始！

発者向けドキュメント  @nifty

ニフティクラウド mobile backend利用規約

第1条（利用規約）  
1. 本利用規約は、ニフティ株式会社（以下「ニフティ」といいます。）が提供する「ニフティクラウドmobile backend」（以下「本サービス」といいます。）の利用にかかる一切に適用されます。  
2. ニフティがユーザーに通知する本サービスの説明、案内、利用上の注意等（以下「説明等」といいます。）は、各項目の欄にかかるらず本利用規約の一部を構成するものとします。  
3. ニフティは、ユーザーのアカウントを従うことなく本利用規約を隨時変更することができるものとします。変更後の本利用規約は、ニフティが本サービスのホームページ上に掲載することでユーザーに通知した時点より効力が生じるものとします。

第2条（仕様）  
 以上の規約に同意する

[@nifty Top](#) [@nifty会員登録](#) [お問い合わせ](#) [利用規約](#) [個人情報保護ポリシー](#)

ここをクリック



@nifty

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

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。)

# mBaaSのアプリ作成

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。

「kif\_2dst」  
と入力してください

アプリの新規作成

アプリ名

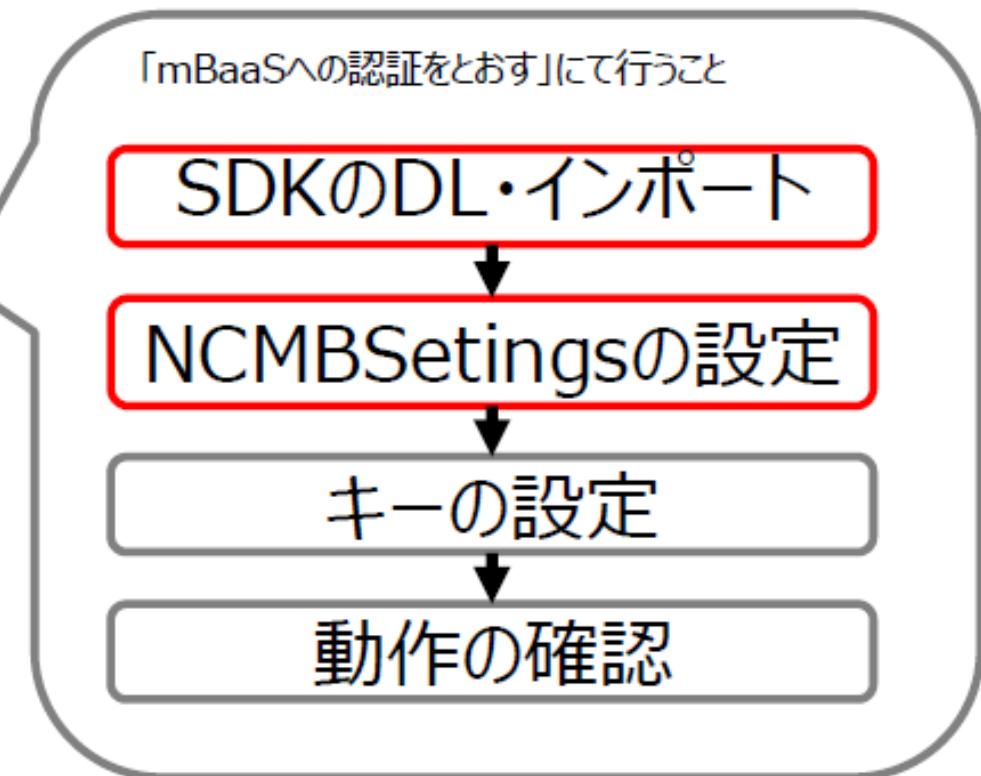
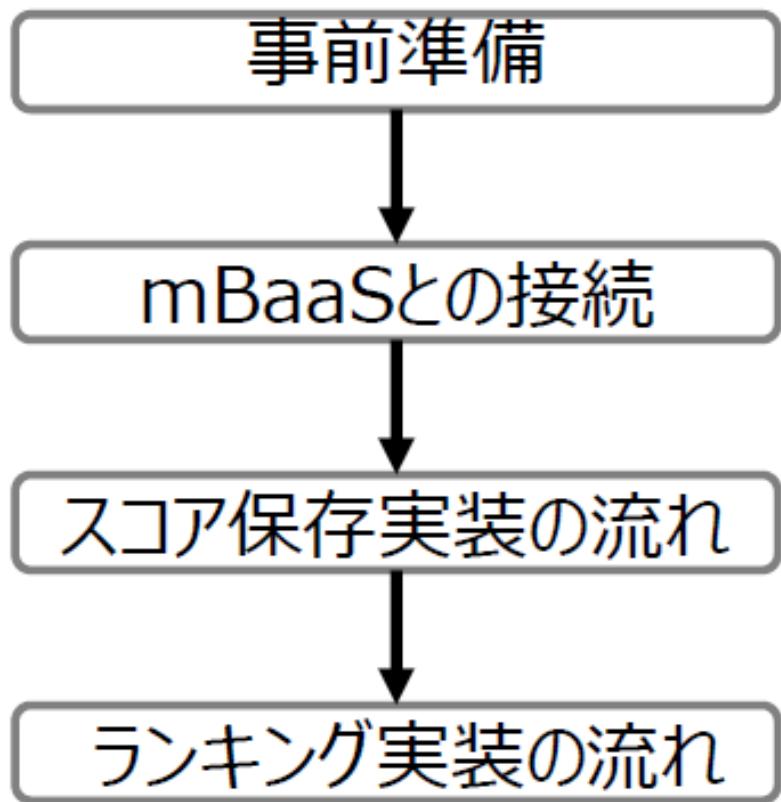
半角英数字もしくはアンダースコア

◀戻る 新規作成



mBaaSは、「アプリケーションキー」  
「クライアントキー」の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。



赤枠部分は、すでに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。  
また本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。

※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](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

ncmbsdk 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**

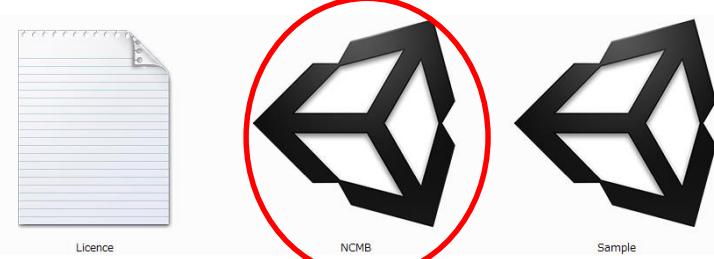
**クリックするとDLできます**

[NCMB1.1.4.zip](#)

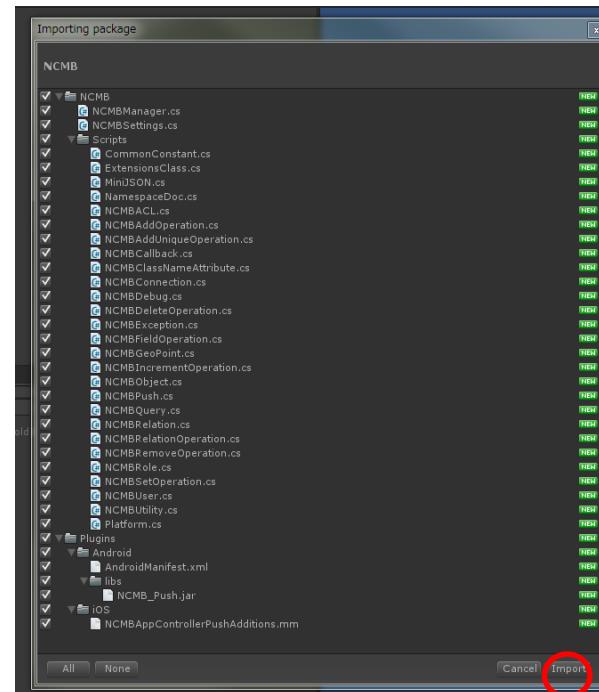
[Source code \(zip\)](#)

[Source code \(tar.gz\)](#)

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）

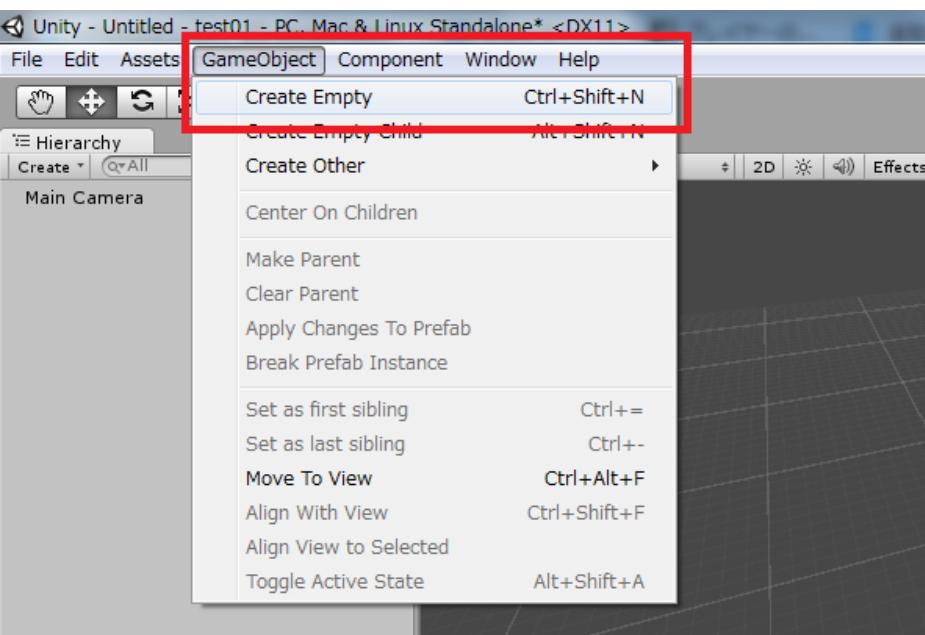
NIFTY Cloud  
ニフティ クラウド

※今回は本ページの作業を  
行う必要はありません。

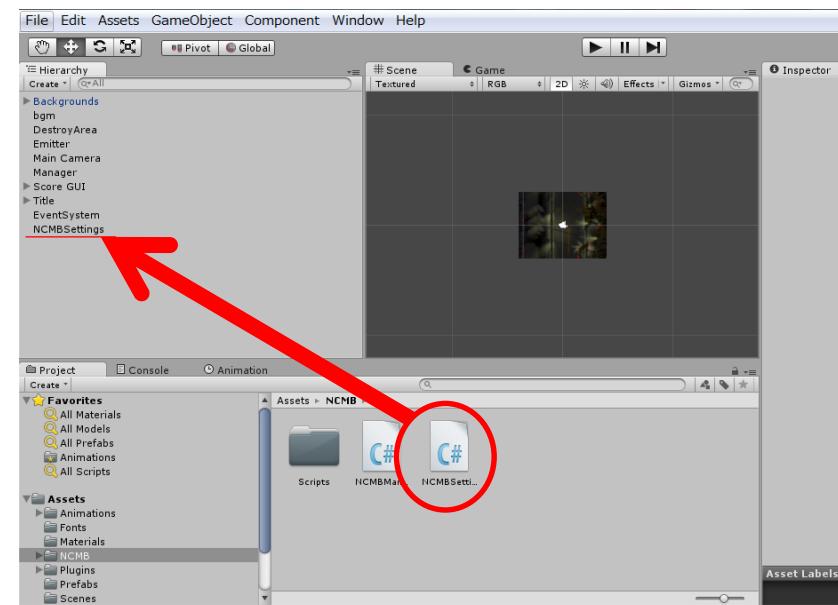
mBaaSへの認証のための設定を行う

NCMBSettingsスクリプトを利用するための作業を行います。

Asset>Scenesの「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。

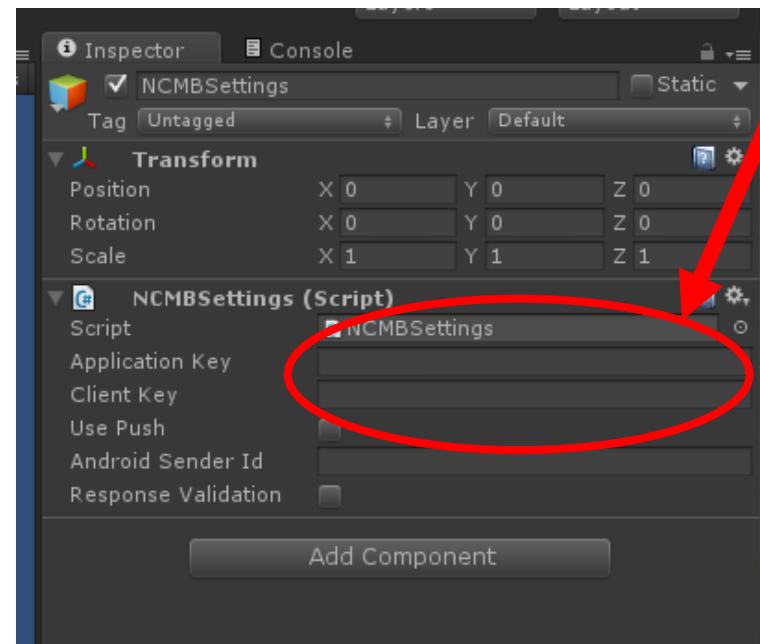


# キーの設定

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。



次にキーの設定についてです。  
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。



# 動作確認

スコアの保存は一度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。

下記のスコア保存画面で「Submit」ボタンを押下後、mBaaSをご確認ください

ゲームオーバー後  
管理画面よりデータストアを  
ご確認いただき下記のように  
データが入っていれば成功です。



mBaaSを  
確認

	objectId	name	score
	FPaz1OkWreNdUcxF	test1	2100

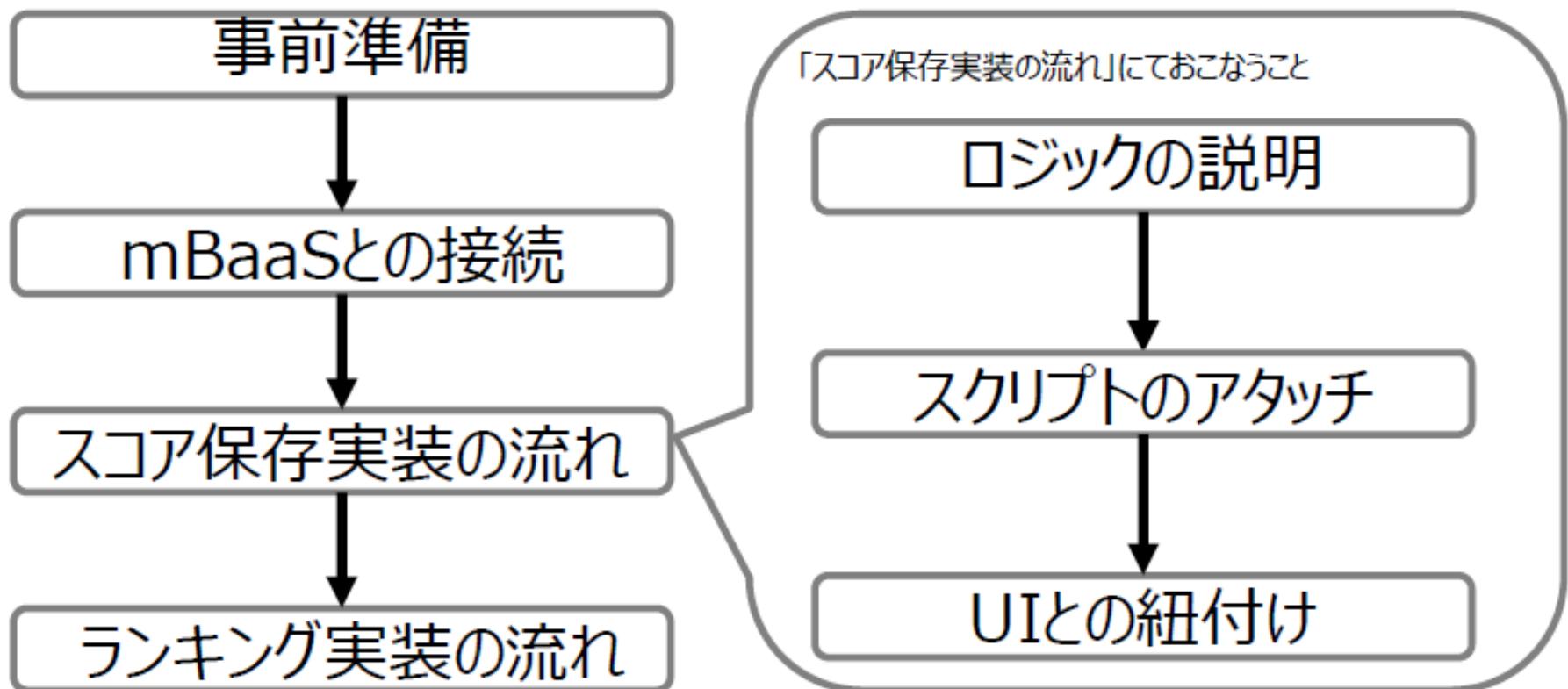
ランキングの確認は  
スタート時の画面の左上「Leader Board」ボタンを押下し、  
ランキングが表示されることを確認してください。



ランキングに  
遷移



上記のように名前とスコアが表示されたら成功です！



# ロジックの説明

Asset>Scriptsの「SaveScore.cs」にてmBaaSへのスコア保存を行っています。  
下記のコードが、スコア保存に関するコードになります、ご参照ください。

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

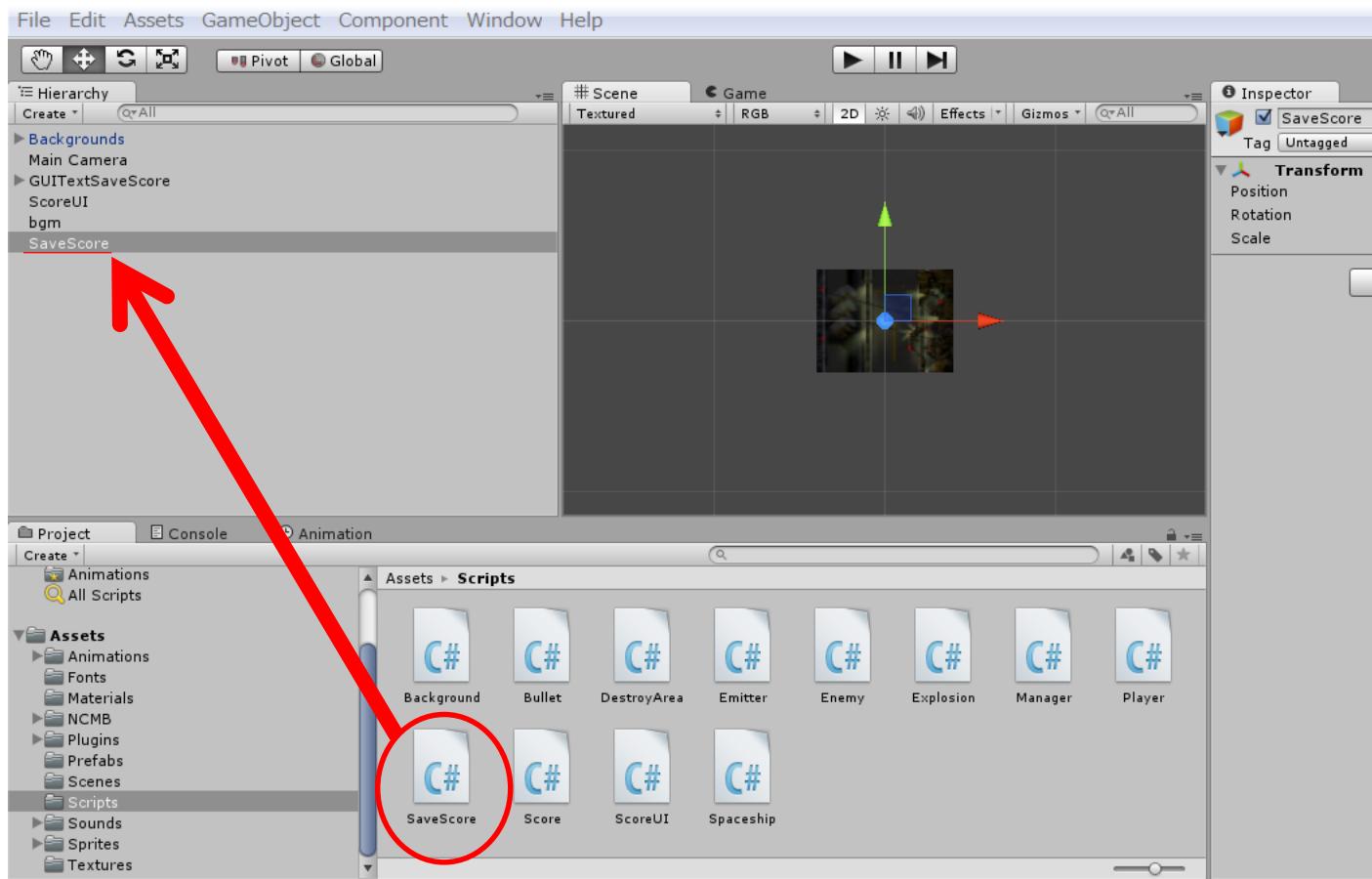
public class SaveScore : MonoBehaviour {
    // mobile backendに接続-----
    public void save( string name, int score ) {
        NCMBObject obj = new NCMBObject ("Score");
        obj ["name"] = name; ← オブジェクトに名前とスコアを設定
        obj ["score"] = score;
        obj.SaveAsync ();
    }
}
```

データストアに  
保存する操作

更新するクラスを指定

# スクリプトのアタッチ

Asset>Scenesの「SaveScore」シーンの「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います。



# UIとの紐付け

## UIとの紐付け

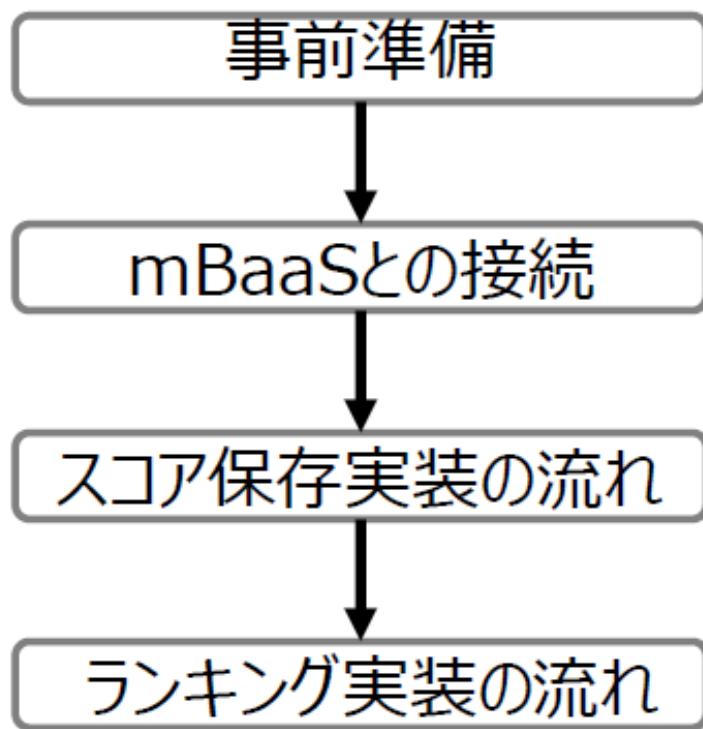
Asset>Scriptの「ScoreUI.cs」のOnGUIでは右のゲームオーバー画面の「Submit」ボタン押した際にスコア保存ロジックが駆動するようになっています。スコアと一緒に任意の名前も設定できます。



```
void OnGUI () {
    drawSaveScore();
    // ログインボタンが押されたら
    if (SubmitButton) {
        //SaveScoreを読み込み、メソッドを呼び出す
        FindObjectOfType<SaveScore> ().save (name, score);
        //ステージシーンを読み込む
        Application.LoadLevel ("Stage");
    }
}
```

以上にてライトプランは終了です！  
ご参加いただきありがとうございました！

# 体験会の流れ



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

構成要素の確認

ロジックの解説

ロジックの呼び出し方

※本ページ以降は  
スタンダートプランの内容になります。

# 構成要素の確認

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、LeadeBoard.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「Score」 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<NCMBOObject> query = new NCMBQuery<NCMBOObject> ("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

# ロジックの説明 2/2

```
query.FindAsync<NCMBOBJECT> ((List<NCMBOBJECT> 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 (NCMBOBJECT 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

# ロジックの呼び出し方

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は下記で解説いたします。

```
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。

以上にてスタンダートプランも終了です！  
ご参加いただきありがとうございました！

@nifty



ありがとう 5周年

Going  
Next  
Stage ↑

ニフティとなら、きっとかなう。  
With Us, You Can.

NIFTY Cloud  
ニフティクラウド