

【Unity問題集】 オンラインランキング機能を作 ってみよう！ 「シューティングゲーム」

2017/05/17作成 (2017/05/18修正)



GitHub

<https://github.com/natsumo/UnityFirstApp>

コンテンツ概要

- ニフティクラウド mobile backend の機能『データストア』を学習するための問題集です
- ニフティクラウド mobile backend の利用登録（無料）が必要です
- 問題用プロジェクトにはオンラインランキング機能が実装されていない状態の「シューティングゲーム」です
- 既に実装済みのニフティクラウド mobile backend を利用するための準備（SDK導入など）方法の詳細はこちらをご覧ください

http://mb.cloud.nifty.com/doc/current/introduction/quickstart_unity.html

問題について

- 問題は2問あります
- 2問クリアすると「シューティングゲーム」にオンラインランキング機能を実装したアプリが完成します
- 問題に取り組む上で必要な開発環境は以下です
 - Unity 3Dが動作するWindows PC もしくはMac PC
 - Unity5.3以降

問題に取り組む前の準備

プロジェクトのダウンロード

▼問題用プロジェクト▼

<https://github.com/natsumo/UnityFirstApp/archive/Question.zip>

1. 上記リンクからzipファイル取得します
2. ローカルに保存し、解凍します

「シューティングゲーム」の操作方法

1. 【ゲーム(Game)ビュー】から初期画面(Tap To Start)をタッチします
2. 矢印で方向を変えながら、弾丸やレーザーなどの飛び道具を利用して敵機を撃ち落とす、単純なゲームです
3. GameOverになるとゲームプレイヤーの名前を入力するアラートが表示されますので、名前を入力し「Submit」をクリックします
4. ゲームプレイヤーの名前とスコアが保存され、初期画面に戻ります

※ **注意**：問題に取り組む前の状態では「LeaderBoard」(ランキングを見る)ボタンをタップしてもランキングは表示されません

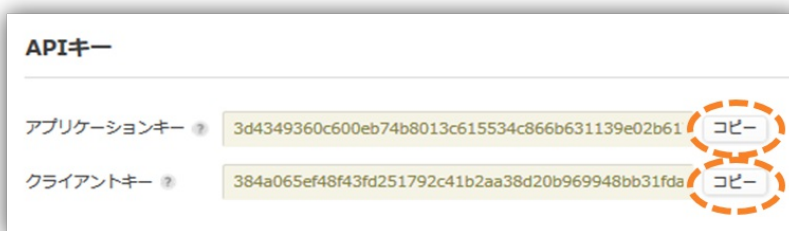
アプリの新規作成とAPIキーの設定

mBaaS ダッシュボード

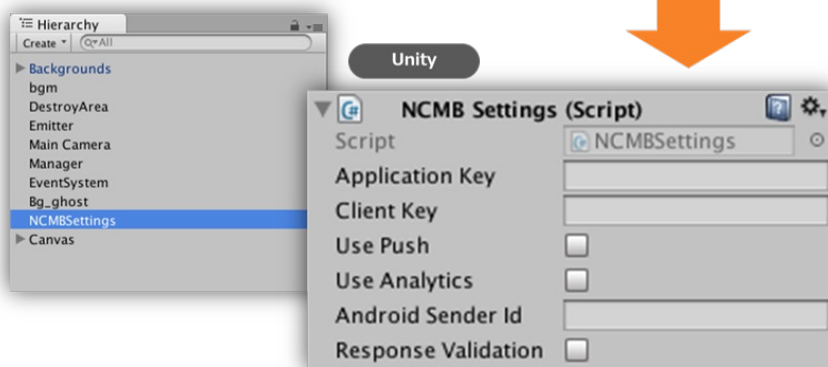
- ニフティクラウド mobile backend にログインしアプリの新規作成を行います
- アプリ名はわかりやすいものにしましょう 例)
「UnityShootGame」
- アプリが作成されるとAPIキーが2種類（アプリケーションキーとクライアントキー）発行されます
- 次で使います

Unity

- 【ヒエラルキー(Hierarchy)ビュー】から **NCMBSettings** を編集します
- 先程ニフティクラウド mobile backend のダッシュボード上で確認したAPIキーを入力します



重要
APPLICATION_KEYと
CLIENT_KEYは
mobile backendの
管理画面から
コピーボタンで
コピーして使います！



- 【インスペクター(Inspector)ビュー】から「NCMB Settings」欄の Application Keyと Client Keyの入力部分に各APIキーを入力します

【問題 1】

名前とスコアの保存を試みよう！

/Assets/Scripts/saveScore.cs を開きます。下図の **saveScore** メソッドを編集し、引数の **name**（アラートで入力した名前）と **score**（シューティングゲームのスコア）の値をニフティクラウド mobile backend に保存する処理をコーディングしてください。

```
6 // 【mBaaS】データの保存
7 public class SaveScore : MonoBehaviour {
8     public void save( string name, int score ) {
9         // *****【問題1】名前とスコアを保存しよう！*****
10
11
12
13
14
15
16
17
18
19
20         // *****
21     }
22 }
```

- データストアに保存先クラスを作成します
- クラス名は「**GameScore**」としてください
- **name** を保存するフィールドを「**name**」、**score** を保存するフィールドを「**score**」として保存してください

ヒント

- ニフティクラウド mobile backend のUnityドキュメントをご参考ください

http://mb.cloud.nifty.com/doc/current/datastore/basic_usage_unity.html

コーディング後の作業

問題1のコーディングが完了したら、下記の作業を行います

【作業1-1】

それぞれ該当する箇所に以下の処理を追記して、再生時にUnity Console上にログを表示できるようにします

- 保存に失敗した場合の処理を行う箇所に追記

```
// 保存に失敗した場合の処理  
Debug.Log("保存に失敗しました。エラーコード:"+e.ErrorCode);
```

- 保存に成功した場合の処理を行う箇所に追記

```
// 保存に成功した場合の処理  
Debug.Log("保存に成功しました。objectId:"+ obj.ObjectId);
```

【作業1-2】

「再生」をクリックし、ゲームをします

- ゲームオーバーになったら名前を入力し、「OK」がクリックされると【問題1】で作成した `saveScore` メソッドが呼ばれ、データが保存されます
- このとき下記のいずれかのログが出力されます
 - 保存成功時：「保存に成功しました。objectId:*****」
 - 保存失敗時：「保存に失敗しました。エラーコード:*****」

※ エラーコードが出た場合はこちらで確認できます

http://mb.cloud.nifty.com/doc/current/rest/common/error.html#REST_APIのエラーコードについて

【問題1】 答え合わせ

ニフティクラウドmobile backend上での確認

mBaaS ダッシュボード

- 保存されたデータを確認しましょう
 - 「データストア」をクリックすると、「GameScore」クラスにデータが登録されていることが確認できます。

The screenshot shows the Nifty Cloud mBaaS dashboard. The top navigation bar includes links for 'アプリ一覧' (App List), 'ダッシュボード' (Dashboard), 'ドキュメント' (Documents), '開発TIPS' (Development Tips), 'コミュニティ' (Community), and '連携サービス' (Linked Services). The main header has a '+新しいアプリ' (New App) button and a '更新' (Refresh) button. The left sidebar contains a 'メニューを開じる' (Open Menu) button and links to 'ダッシュボード', '会員管理' (Member Management), and 'データストア' (Data Store). The 'データストア' section is active, showing a list of classes: 'installation', 'role', and 'GameScore'. The 'GameScore' class is selected, displaying a table of data. The table has columns for 'objectId', 'name', 'score', and 'createDate'. A single record is shown with 'objectId' 'jbBzPTAvBW7jiCC1', 'name' 'name', 'score' '300', and 'createDate' '2016-08-16T15:27:50.336+0'.

	objectId	name	score	createDate
	jbBzPTAvBW7jiCC1	name	300	2016-08-16T15:27:50.336+0'

コードの答え合わせ

- 模範解答は以下です

```
// *****【問題1】名前とスコアを保存しよう!*****  
// 保存先クラスを作成  
NCMBObject obj = new NCMBObject ("GameScore");  
// 値を設定  
obj ["name"] = name;  
obj ["score"] = score;  
// 保存を実施  
obj.SaveAsync ((NCMBException e) => {  
    if (e != null)  
    {  
        // 保存に失敗した場合の処理  
        Debug.Log("保存に失敗しました。エラーコード:"+e.ErrorCode);  
    }  
    else  
    {  
        // 保存に成功した場合の処理  
        Debug.Log("保存に成功しました。objectId:"+ obj.ObjectId);  
    }  
});  
// *****
```


【問題2】 ランキングを表示しよう！

/Assets/Scripts/LeaderBoard.cs を開きます。下図の fetchTopRankers メソッドを編集し、データストアの GameScore クラスに保存した name と score のデータを score の降順（スコアの高い順）で検索・取得する処理をコーディングしてください。

```
11 // 【mBaaS】保存したデータの検索と取得
12 public void fetchTopRankers() {
13     // ***** 【問題2】ランキング表示しよう！ *****
14
15
16
17
18
19
20
21
22
23
24 // *****
25 }
```

- 検索データ件数は5件とします

ヒント

- ニフティクラウド mobile backend のUnityドキュメントをご参考ください

http://mb.cloud.nifty.com/doc/current/datastore/ranking_unity.html

コーディング後の作業

問題2のコーディングが完了したら、下記の作業を行います。

【作業2-1】

該当する箇所に以下の処理を追記して、再生時にUnity Console上にログを表示できるようにします

- 検索に失敗した場合の処理を行う箇所に追記

```
//検索に失敗した場合の処理  
Debug.Log("検索に失敗しました。エラーコード:" + e.ErrorCode);
```

- 検索に成功した場合の処理を行う箇所に追記

```
//検索に成功した場合の処理  
Debug.Log("検索に成功しました。");
```

【作業2-2】

Unityから「再生」し、「LeaderBoard」(ランキングを見る)ボタンをタップします

- 画面起動後、`fetchTopRankers` メソッドが呼ばれ、【問題1】で保存されたデータが検索・取得されます
- このとき下記のいずれかのログが出力されます
 - 検索成功時：「`検索に成功しました。`」
 - 検索失敗時：「`検索に失敗しました。エラーコード：*****`」

※ エラーコードが出た場合はこちらで確認できます

http://mb.cloud.nifty.com/doc/current/rest/common/error.html#REST_APIのエラーコードについて

【作業2-3】

検索に成功したら、該当する箇所に以下の処理を追記して、取得した値から必要なデータを取り出し、ランキング画面へ反映させます

- 検索に成功した場合の処理を行う箇所に追記

```
// 取得したデータをリストに設定
List<NCMB.Rankers> list = new List<NCMB.Rankers>();
foreach (NCMBObject obj in objList) {
    int    s = System.Convert.ToInt32(obj["score"]);
    string n = System.Convert.ToString(obj["name"]);
    list.Add( new Rankers( s, n ) );
}
topRankers = list;
```

【作業2-4】

Unityから「再生」し、「LeaderBoard」(ランキングを見る)ボタンをタップします

- 先ほどのスコアが表示されれば完成です！おめでとうございます★

【問題2】 答え合わせ

ランキング画面の確認

- ランキング画面を確認しましょう
 - 「LeaderBoard」 (ランキングを見る) ボタンをタップすると以下のようにランキングが表示されます。



- 上図はランキングが表示されることを確認しましょう！

コードの答え合わせ

- 模範解答は以下です

```
// *****【問題2】ランキングを表示しよう!*****
// GameScoreクラスを検索するクエリを作成
NCMBQuery<NCMBObject> query = new NCMBQuery<NCMBObject> ("GameScore");
// Scoreを降順でデータを取得するように設定
query.OrderByDescending ("score");
// 検案件数を設定
query.Limit = 5;
// データストアを検索
query.FindAsync ((List<NCMBObject> objList ,NCMBException e) => {
    if (e != null) {
        //検索に失敗した場合の処理
        Debug.Log("検索に失敗しました。エラーコード:" + e.ErrorCode);
    } else {
        //検索に成功した場合の処理
        Debug.Log("検索に成功しました。");
        // 取得したデータをリストに設定
        List<NCMB.Rankers> list = new List<NCMB.Rankers>();
        foreach (NCMBObject obj in objList) {
            int s = System.Convert.ToInt32(obj["score"]);
            string n = System.Convert.ToString(obj["name"]);
            list.Add( new Rankers( s, n ) );
        }
        topRankers = list;
    }
});
// *****
```

参考

- 問題の回答を実装した完全なプロジェクトをご用意しています

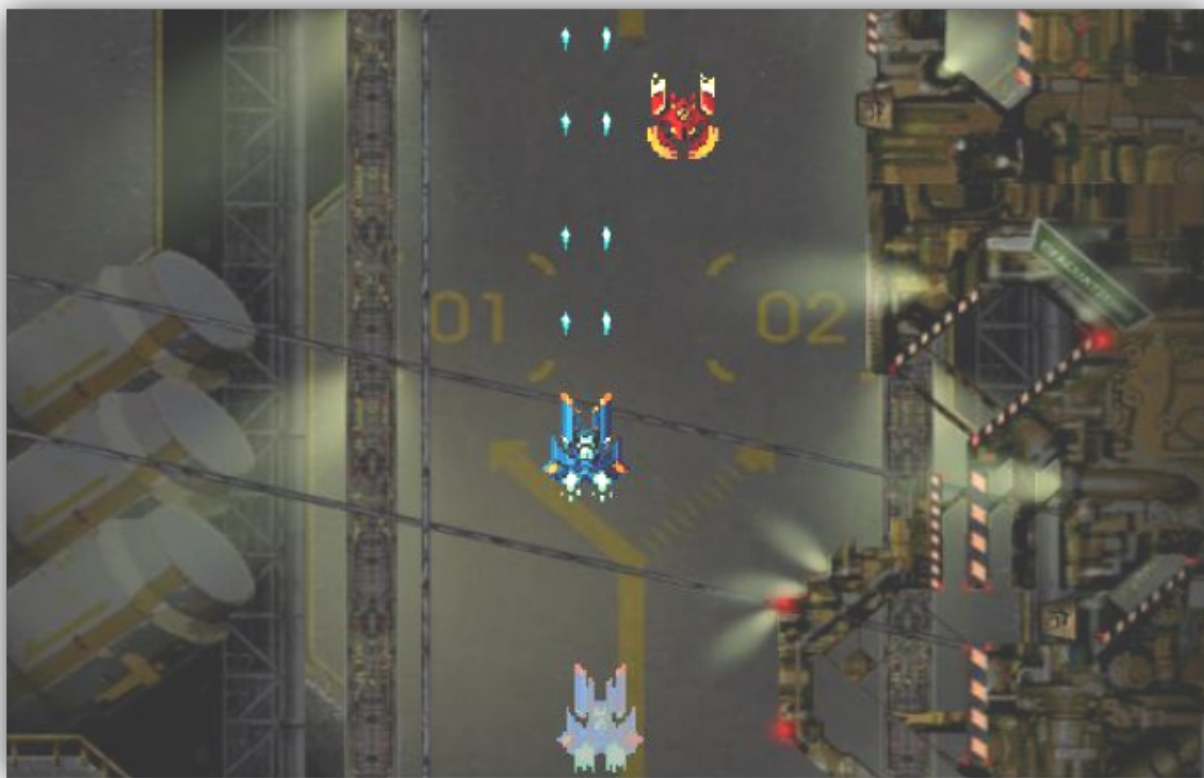
▼完成版プロジェクト▼

<https://github.com/natsumo/UnityFirstApp/archive/AnswerProject.zip>

- APIキーを設定してご利用ください

おまけ

- ニフティクラウド mobile backend を使って、シューティングゲームに「ゴースト機能」を実装することができます！



- 興味がある方はこちらをご覧ください

<https://github.com/natsumo/UnityFirstApp/blob/master/Ghost.md>