

カスタマイズ可能なゲームのスコアランキング表の開発

埼玉県立川越南高等学校 かわばそ 大本和輝

概要

Vue.jsやGoogleが提供するNoSQLデータベースであるFirebase Cloud Firestoreを利用して、ブラウザから閲覧可能な、自作ゲームのスコアのランキング表を開発した。

改良前

<スコア送信>

部員がUnity(C#)やJavaで制作したゲームは、ゲームクリア/オーバー時にスコアなどの各種データをGoogleスプレッドシートへPOSTする。

<スコア表示>

Flask(Python)で作成されたサイトにアクセスすると、スコアなどをGoogleスプレッドシートから取得してランキング形式で表示され、30秒ごとに更新される。

Rank	Score	UserName	Date	Clear	GameName
1	173502.0	stgUser2	Sun Sep 09 2018 14:23:26 GMT+0900 (JST)	False	STG
2	167324.0	stgUser2	Sun Sep 09 2018 14:06:17 GMT+0900 (JST)	False	STG
3	143884.0	stgUser2	Sun Sep 09 2018 13:55:24 GMT+0900 (JST)	False	STG

図1 改良前のランキング表(一部)

問題点

表示できるゲーム名などが固定であり、他のゲームに対応させることができない。また、ゲームクリアの可否やクリアできたステージ数などのスコアに付随するデータが増えた際にそれらのデータを表示(列を追加)することができない。

解決方法

列の概念にとらわれないNoSQLのデータベースであるCloud Firestoreを採用し、ランキング表ページへのアクセス時のURLに応じて、表示対象のゲームやデータを変える。

結果(改良後)

<スコア送信>

ゲームからCloud FirestoreのAPIにスコアなどをPOSTする(図2)。

<スコア表示>

1. ランキングサイトのJSが、URLから取得したパラメータ(ゲーム名など)を用いてCloud Firestoreへデータをリクエストし取得する(図3 A)。

```
db.collection(self.$route.params.gameTitle) // ゲーム名を指定
  .orderBy('Score', 'desc')
  .limit(Number(self.$route.params.rankLimit)) // 順位の下限を指定
  .onSnapshot(function(querySnapshot) {...(略)})
```

2. 取得したデータをそれぞれのリストに追加する(図3 B)。付随する各種データに関しては、連想配列(辞書)を利用して管理した。

```
...
querySnapshot.forEach(function(doc) {
  self.nameList.splice(rankCount, 1, doc.data()['Name'])
  ...
})
```

```
p = {param1: [100, 200], param2: [50, 90]} // 例
```

<今後も使いやすく>

改良後は、表示するゲーム名や項目などをURLで制御できるようになった。したがって、新しいゲームに対応する際にプログラムを書き換える必要がなくなり、ビルドして生成されたファイルを使い回せるようになった。



図2 スコアの送信

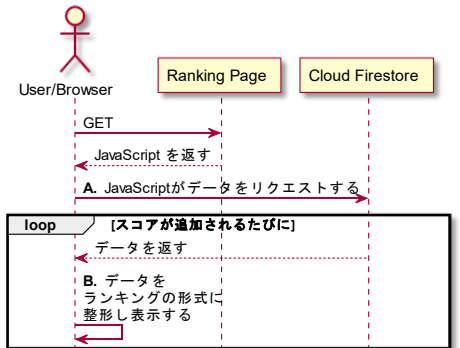


図3 スコアの表示

Rank	Name	Score	Date	clear	stage
1	名無し	111.111	2019/1/1 00:00:00	true	5
2	名無し	99.999	2019/1/1 00:00:00	true	6
3	名無し	88.888	2019/1/1 00:00:00	true	5
4	名無し	77.777	2019/1/1 00:00:00	false	4
5	名無し	66.666	2019/1/1 00:00:00	false	3

図4 ランキング表の様子

“/JumpGame/5?param=clear,stage”
clearとstage列が追加されている

これから

スコアの重複がほとんど無いゲームのため、同じ順位のスコアが発生したときの処理を実装していない。また、上位5位を取得する処理を行うと100回の読み取り処理が行われるなどの、スコアの取得クエリに無駄がある。今後はこれらの課題を解決していきたい。

参考文献

Cloud Firestore ドキュメント : <https://firebase.google.com/docs/firestore>