

イベント時におけるイベント開始前
ユーザー参加型コンテンツの開発

Development of user participation type contents before the event starts.

松 澤 衛

Mamoru MATSUZAWA

北翔大学教育文化学部研究紀要
第4号 2019

イベント時におけるイベント開始前 ユーザー参加型コンテンツの開発

Development of user participation type contents before the event starts.

松 澤 衛
Mamoru MATSUZAWA

アブストラクト

本研究の目的は、イベント開催時におけるメインイベント開始前の待ち時間を有効に過ごしてもらうため参加者が持参しているスマートフォンなどを活用し、普段使用しているSNSを使って、会場にあるスクリーンにアクションを反映させることをプログラムを開発することを目的とした。

各個人のスマートフォンとTwitter、あらかじめ指定したハッシュタグでツイートしてもらい、会場に設置したパソコンでハッシュタグを検索し、新規にツイートとされたもの出れば、会場のスクリーンにアクションを投影した。

背景と目的

平成29年度の総務省の統計によるとスマートフォンの普及率は、71.8%と4人に3人は持っている計算になる。また、20代94.2%、30代90.4%、13才～19才81.1%と全体の普及率を大きく上回っている。イベント開始の待ち時間を旨くスマートフォンを活用できないかと考えた。イベント開始前に、案内を出す方法として、スクリーンを使用したり、指定したWebサイトを閲覧してもらうことで開始までの時間を過ごしてもらう方法が一般的であるが、イベント開始までの間に期待感等を演出することと参加している自覚も必要である。本研究は、ユーザーの持っているスマートフォン（端末）を使って何か出来ないかと考えたのが発端である。

当初、ゲームアプリの開発やWebサイトを構築し、ユーザー参加型のコンテンツを作成することを考えた。しかし、ゲームアプリとなると事前にアプリのダウンロード、インストール、使用方法の把握などが必要であり、また、すべての端末に対してアプリを開発するには、コストが非常にかかる。すべての端末に対しての対処として、Webサイトを利用する方法もあるが、URLの周知方法や使用方法などへのフォローなど問題が残る。また、Web上で、3次元などを扱うときは各端末へのフォローの問題も起きてくる。

参加には、ユーザーにあまり負担をかけないことが必要であると考え、新しいアプリや新しいサイトへの接続をさせるよりは、多くのユーザーが利用しているSNSを考え、2017年12月の調査（株式会社ジャストシステム「モバイル&ソーシャルメディア 月次定点調査（2017年総集編）」）で、15～19才が、76%のユーザーが使用しているTwitterを使うことにした。普段利用しているため、Twitterの細かなレクチャーは必要なく、決められたハッシュタグでツイートをして欲しいということを伝えるだけで参加することが出来る。イベント毎に、ハッシュタグを決め、そのハッシュタグを入れてツイートしてもらうことをきっかけに会場のスクリーンに変化を起こすことを目的とした。

方法1

開発環境は、当初Processingで行い、ツイートをきっかけに花火が出る仕組みを考えた。開発の第2段階では、ゲームエディタであるUnityを使用し、3次元表現を可能にした。

当初、開発環境は、オープンソースであるProcessingを使用した、メディアアートとビジュアルデザインを手軽にするために開発された開発環境で、イメージやアイデアを形にしやすい言語である。

開発環境およびイベント当日再生マシンには、Apple MacBook Proを使用した。検証はしていないが、Windowsでの開発でも可能である。当日の再生マシンは、プロジェクションマッピング用のソフトウェアがWindows版がないために再生は出来ない。

今回、SNSには、若者の利用が多いTwitterを利用することにして、ProcessingよりTwitter APIのJavaラッパーのTwitter4JとProcessingのライブラリであるSimpleTweetを使用した。

```
import gohai.simpletweet.*;
import twitter4j.Query;
import twitter4j.QueryResult;
import twitter4j.Status;
import twitter4j.TwitterException;
import twitter4j.User;
```

Twitter APIとは、Twitterとプログラムレベルで連携が取れるもので利用するには、利用者登録が必要で、Twitterより4つのTwitter API Keyを取得する。取得したKeyはsimpletweetの命令を使用して、以下のソースのkeyの部分に挿入する。

```
simpletweet.setOAuthConsumerKey ("key");
simpletweet.setOAuthConsumerSecret ("key");
simpletweet.setOAuthAccessToken ("key");
```

```
simpletweet.setOAuthAccessTokenSecret ("key");
```

また、このプログラムは、プロジェクタを使ってスクリーンに投影することが前提になるので、プロジェクションマッピング用ソフトウェアであるMilluminでデータが取得出来る様にSyphonでの出力も考える。

Syphonの処理は以下のようになる。

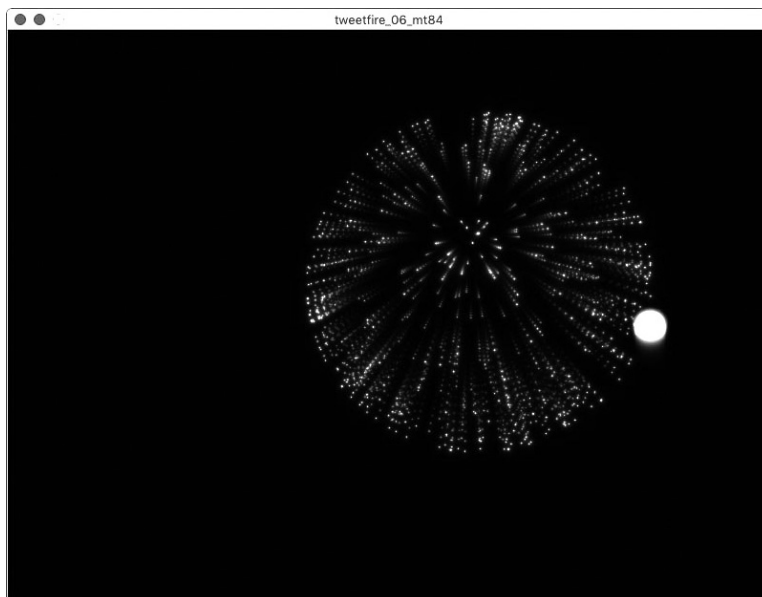
```
import codeanticode.syphon.*;
SyphonServer server;

void setup () {
  初期化处理
  server = new SyphonServer (this, "Processing Syphon");
}

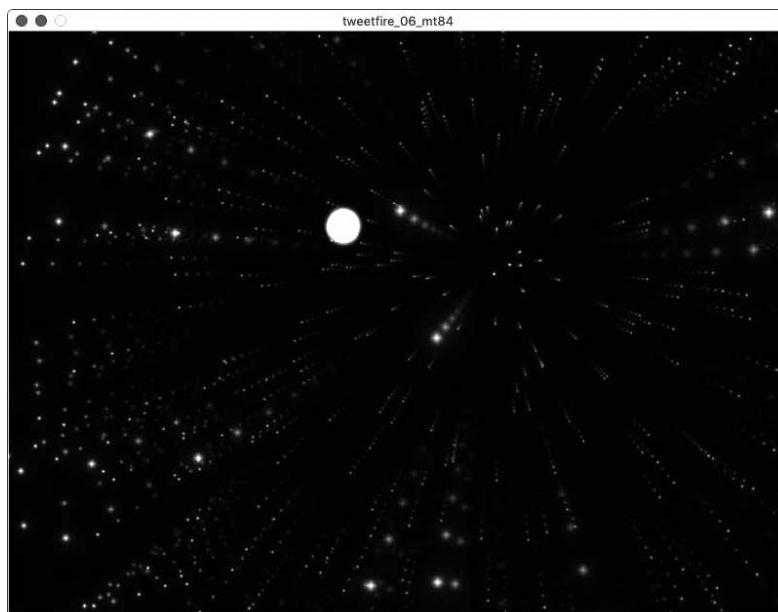
void draw () {
  描画ブロック
  server.sendScreen ();
}
```

ツイートの検索と花火の描画は、非同期のスレッド処理を行う。スレッド処理を行わないとツイートの検索時に花火の描画が止まることになりスムーズな描画が出来ない。

thread (“ツイート処理を書いた関数”);を呼び出すことで花火の描画と別スレッドで処理することが出来る。



画像1



画像2

Twitter APIには、制限があり検索に関しては、15分に180リクエストという上限がある。検索をするのは、5秒に1回で、リアルタイムにツイートと花火を同期させることは不可能である。

実施 1

毎回オープンキャンパスのスタート時の映像を担当していて、当初より開始までの待ち時間を退屈しないように工夫が出来ないかと考えていたが、担当者と打ち合わせをして、8月のオープンキャンパスで初実施をした。10時から10時55分まで、1時間程度会場案内の背景に花火を描画する。会場のユーザーには、紙でハッシュタグ等のお知らせを行う。参加者は高校生のために最初は、書き込み等を遠慮していたが、一人が書き込みを行うと数人が続いた。ユーザーアイコン、ユーザー名を画面上に表示した。どの花火が、どのユーザーが書き込んだかわかる仕組みにした。

運用に関しては問題なく動作し、書き込みをしたユーザーはスクリーンを確認していた。

また、JAPAN DRONE RACING GT チュプチニカ杯 in 北海道にて、プロジェクションマッピングを行う前段として、実施した。このイベントでは、会場の人だけ知っているハッシュタグではなくドローンレースをネット中継で見ているユーザーにも知らせて会場の花火画像を表示した。

実際のイベントに対して、実施をしたが、自分がどの程度関与して、スクリーン上の表示に影響を与えているかがわかりずらいために積極的な参加が少ない。方法2では、開発環境を変更する。

方法2

Processingの作成を経て、Processingでは、物理計算や3次元での映像表現に限界があり、開発環境をゲームエンジンであるUnityに変更し、C#で開発することにした。Unityは、統合開発環境を内蔵し、複数の端末に対応するゲームエンジンである。UnityでのSyphon対応は、KlakSyphonというライブラリを使用する。



Unity 開発画面



3次元が可能のためによりリアリティーのある表現が可能になった。

Twitter API用のUnityライブラリは、Twityを利用した。まだbeta版だがAPIで利用する部分が少ないために動作に問題はない。

認証処理

```
public class EventHandler : MonoBehaviour {
    void Start () {
        Twity.Oauth.consumerKey      = "key";
        Twity.Oauth.consumerSecret   = "key";
        Twity.Oauth.accessToken      = "key";
        Twity.Oauth.accessTokenSecret = "key";
    }
}
```

APIを使用するための初期化を行う。Processingと同じものになる。

Processingでのスレッドにあたるものは、Unityでは、コルーチンを使用する。

検索処理

```
parameters["q"] = "指定したハッシュタグ";
StartCoroutine (Twity.Client.Get ("search/tweets", parameters, Callback));

void Callback (bool success, string response)
{
}
```

```

if (success)
{
    SearchTweetsResponse Response =
JsonUtility.FromJson<SearchTweetsResponse> (response);
    foreach (var status in Response.statuses)
    {
        twname = status.user.screen_name;
    }
}
else
{
}
}
}

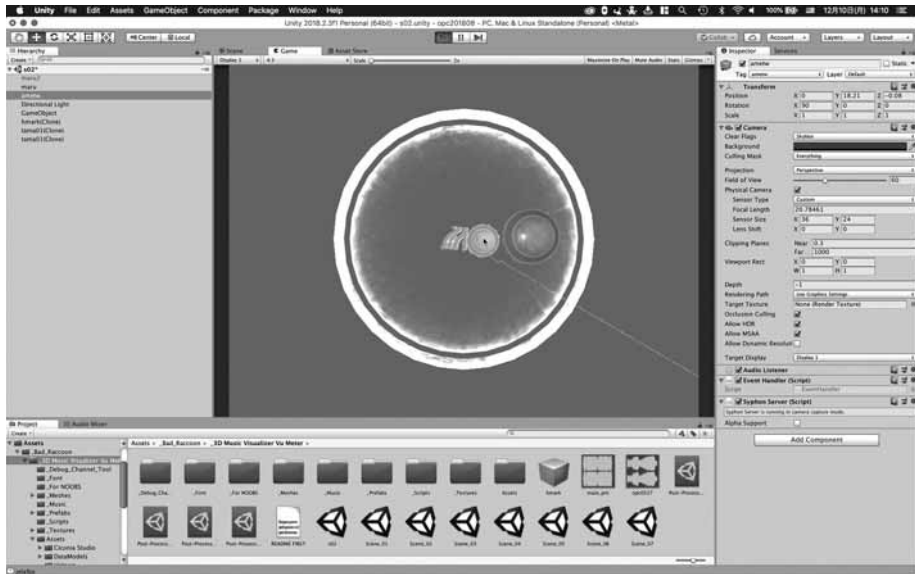
```

検索するハッシュタグは、parametersに格納する。

作成する画像は、北翔大学のマークを立体にしたものと、ガラスの球体が、ツイートによりガラスのボールに落ちてくるという表現にしました。落ちてくる物体の色は、ランダムに変化する。画像は、Processingに比べリアリティーが増した。



実際の画面 1



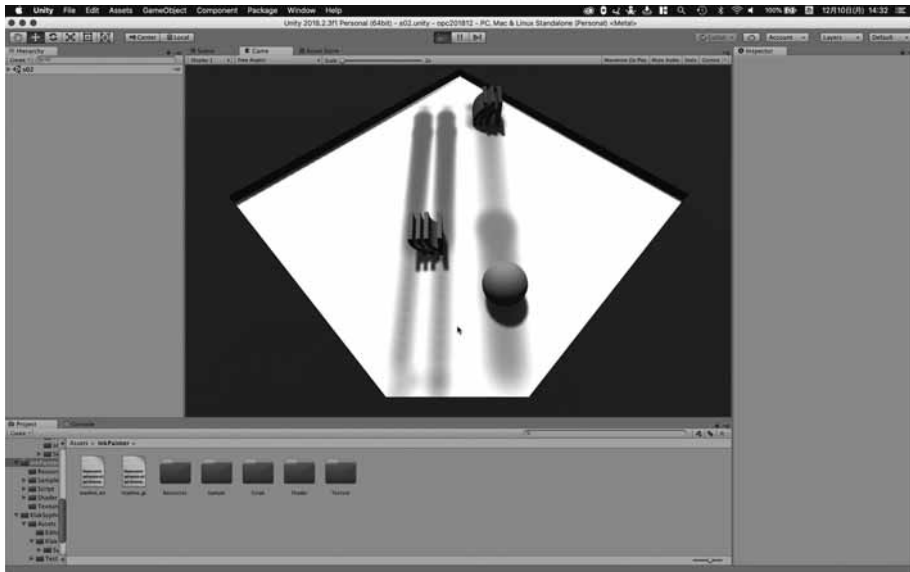
実際の画面2

実施2

9月のオープンキャンパスで、実施したが、前回よりもツイート数が少なかった。出てくる画像が、何で球体なのかというのが、感じられ花火より画質は上がったが、華やかさがなくなった。また、Twitterのユーザー名やユーザーアイコンなどを外したこともありユーザーの操作感がなくなった。

方法3

前回の球体が、透明なボールに落ちるものから、色を持った球体が、白いキャンバスに自由に絵を描くという表現に変更した。UnityのライブラリInk Painterを使用して実現している。リアルタイムにInk Painterを適応した物体のテクスチャマッピングを行うことで実現している。



実際の画面 3



実際の画面 4

実施 3

12月2日ミニオープンキャンパスにて、実施した。今回のオープンキャンパスは、通常より参加者が少ないためにテストケースとした。実際は、ほとんどツイートされることがなかった。落ちてくる物体が動く奇跡が、ペイントとして残るので、目で見ている飽きが来ないが、ツイートしたユーザーへのフィードバックが少ない。

結果

開発環境を変え、3回実施をした。当初予定していた画像は表示でき、担当者レベルでは良好であった。実施時にエラーもなく問題が発生することもなかった。しかし、参加者の少なさは致命的であり、開発には成功したが、利用には失敗した。参加をいかにさせるか、参加するメリットを感じるようなコンテンツの開発が必要である。ツイート、ユーザーアイコンなど、実際のTwitterに関連したものを表示しないと感心がわからない。次回は、ツイートしたユーザーの情報をいかに出すのかを考える必要がある。

考察

イベント開始前ユーザー参加型コンテンツの開発を行った、コンテンツを使用してもらうには、まだまだ時間がかかると感じた。実際にイベント開始前に行うサービスとして提供したが、このようなサービスを今まで行ってないためか、思ったように参加をするユーザーが少なかった。また、自分のツイートがどのようにスクリーンに影響を与えるかがわかるコンテンツが必要である。技術的な問題として15分180回というTwitter APIの制限の中でいかに、ユーザーに操作感を持たせ、各ユーザーのツイートに対してのリアルタイムの反応をどう表現するかが今後の課題である。

参考文献

- 1) UnityからTwitter APIを叩くライブラリをつくっている
<https://qiita.com/toofu/items/075efed6ab1f23e94388>
- 2) モバイル&ソーシャルメディア 月次定点調査(2017年 総集編) 株式会社ジャストシステム
https://www.justsystems.com/jp/download/contents/fastask/biz/report/fa_report-monthly-20180124.pdf?_ga=2.250200709.208155832.1518572799-1761877588.1518572799
- 3) ソーシャルメディアラボ
<https://gaiax-socialmedialab.jp/post-30833/>
- 4) Processing
<https://processing.org>
- 5) Unity
<https://unity3d.com/jp>
- 6) KlakSyphon
<https://github.com/keijiro/KlakSyphon>

7) Ink Painter

<https://github.com/EsProgram/InkPainter>

