THE SOUNDS OF THE DISCUSSION OF SOUNDS

Matt Bethancourt

CUNY Hostos

Media Design Programs

450 Grand Concourse, C Building, Bronx, New York 10451

matt@mouseandthebillionaire.com

ABSTRACT

The Sound of the Discussion of Sounds is a real-time composition and performance computer program built in PHP, MySQL, Max/MSP and Ableton Live that allows listeners to musically hear the subtle changes that occur over time for trending and popular musical artists on Twitter.

1. INTRODUCTION

This is a piece of music that explores the discussion of music. Public opinion is often describes as fickle, but how fleeting is this popularity? Graphs, charts, and datasets give us a glimpse into the short lived nature of fame, but for the most part are limited to two dimensions at a fixed point in time. By making music with data gleaned from social media (Twitter, in this instance), the very nature of the ongoing discussion about musicians and their craft is built into and experienced through a piece of music. This piece will last for as long as the artists it is referencing remain in the popular discussion. As artists lose their hold in the public's eye the piece will become increasingly sparse, until finally there is silence. This may take days, weeks or perhaps even months, and during the process we will be able to hear these changes take place in a way that is both beautiful and thought provoking.

2. PARSING THE JSON DATA

The trending and daily JSON data sets provided by Twitter Music Trends are each parsed with two different sets of PHP programs [Appendix 1-4]. As the program is launched, this PHP code scrapes the top fifty trending artists and the top sixteen artists of the day and inserts each set into a MySQL database. This database is then updated every fifteen seconds, computing the recent change in popularity of each artist, the overall change in popularity of the artist since the program was launched, and the total change of all artist popularity.

Each of the top fifty currently trending artists is also assigned an initial number from one to fifty, that will relate to their musical note in the Max/MSP code. Finally, all of the data and various calculations are published into an HTML page that allows Max/MSP to more effectively parse the data (goo.gl/PJuvZ & goo.gl/mcDAe)

3. MUSICAL ALGORITHMS

3.1. Daily Chord

When the Max/MSP application is launched a root chord is created from the top sixteen artists of the day in the key of A major (from MIDI value 57 to 74). The program continually refers to the parsed JSON data, and every time there is a change in the popularity of any artist, it triggers the note to be played via Ableton Live. The popularity change of each artist increases the velocity of their particular note. In this way, the changes in velocity over time creates subtle changes in the tonal quality of the chord. Meanwhile, the program locates the artist with the highest positive change amount since the last chord was played and applies a short frequency modulation to their corresponding note, further highlighting the movement of the discussion in a musical way.

3.2. Trending Lead

Each of the top fifty trending artists is assigned a sequential note from MIDI value 27 to 77 (D#1 to F5). The current top sixteen notes are sounded in order of popularity at eighth note intervals. When the song first begins this is an ascending D# chromatic scale, but as the program continues to update itself, and the popularity rankings of artists change, interesting patterns and unexpected musical phrases emerge. In addition, the frequency of change in popularity for each artist is directly applied to the quality of the sound of their corresponding note, so that the more change occurs over the course of the piece the less pure the note becomes. Through these methods, the nature of the collective conversation about musicians and their craft shapes the music we are hearing.

4. EXAMPLES AND DOCUMENTS

[1] Project Website: http://www.mouseandthebillionaire.com/icad/
[2] MP3 recording of the first 5 minutes of the program running: goo.gl/P1HpP
[3] The Max/MSP Code can be found here: goo.gl/uxmLm.
[4] Ableton Live project can be found here: goo.gl/guPFE
[5] Pertinent PHP code included in Appendix

5. APPENDIX

1. Daily PHP Initial

```php
<?php

// PHP connection code omitted

$json = file_get_contents("http://
twittermusictrends.com/daily.json");
$json_a = json_decode($json, true);

// Clear table of any existing information

mysql_query('TRUNCATE TABLE tbl_daily;');

// Insert Artists into Database

$artist = 0;
while ($artist <= 49){
  $artistName = $json_a["daily"][$artist]
["name"];
  $artistName_fixed =
mysql_real_escape_string($artistName);
  $artistMBid = $json_a["daily"][$artist]
["mbid"];
  $artistInitialScore = $json_a["daily"]
[$artist]["score"];
  $query_insertArtists = "INSERT INTO
tbl_daily (artistID, artistName,
artistMBid, artistInitialScore,
artistPreviousScore, artistCurrentScore)
VALUES ('$artist', '$artistName_fixed',
'$artistMBid', '$artistInitialScore',
'$artistInitialScore',
'$artistInitialScore');";
    mysql_query($query_insertArtists) or
die ("Error in query:
$query_insertArtists");
  $artist++;
}?>
```

2. Daily PHP Updater

```php
<?php

// PHP connection code omitted

$json = file_get_contents("http://
twittermusictrends.com/daily.json");
$json_a = json_decode($json, true);

// Find new score and insert into database

for ($artist=0; $artist<=49; $artist++){
  // Get the artist
  $mbid = $json_a["daily"][$artist]
["mbid"];
  // Get new scores
  $tempNewScore = $json_a["daily"]
[$artist]["score"];
```

```php
  $newScore = round($tempNewScore, 6);
  // Get their record
  $query_rs_artistCheck = "SELECT
artistID, `artistName`, `artistMBid`,
artistInitialScore, artistCurrentScore
FROM tbl_daily WHERE artistMBid='".
$json_a["daily"][$artist]["mbid"]."' ORDER
BY artistID ASC";
  $rs_artistCheck =
mysql_query($query_rs_artistCheck,
$conn_icad) or die(mysql_error());
  $row_rs_artistCheck =
mysql_fetch_assoc($rs_artistCheck);
  $totalRows_rs_artistCheck =
mysql_num_rows($rs_artistCheck);
  // Get Previous Score
  $previousScore =
$row_rs_artistCheck['artistCurrentScore'];
  if (isset($previousScore)) {
    $previousScore = $previousScore;
  } else {
    $previousScore = 0;
  }
  // Calculate score difference
  $scoreChange = $newScore -
$previousScore;
    // Assign score direction identifier
  if ($scoreChange == 0) {
    $changeDirection = 0; // none
  } elseif ($scoreChange < 0) {
    $changeDirection = 1; // down
  } else {
    $changeDirection = 2; // up
  }
  // And add them to the database
  $query_updateArtist = "UPDATE tbl_daily
SET artistPreviousScore='".
$previousScore."', artistCurrentScore='".
$newScore."', scoreChangeAmmount='".
$scoreChange."', scoreChangeDirection='".
$changeDirection."' WHERE artistMBid='".
$mbid."'";
  $updateArtist =
mysql_query($query_updateArtist);
}

// Query updated Artists in the Database

$query_rs_artists = "SELECT artistID,
`artistName`, `artistMBid`,
artistInitialScore, artistPreviousScore,
artistCurrentScore, scoreChangeAmmount,
scoreChangeDirection FROM tbl_daily ORDER
BY artistID ASC LIMIT 16";
$rs_artists =
mysql_query($query_rs_artists, $conn_icad)
or die(mysql_error());
$row_rs_artists =
mysql_fetch_assoc($rs_artists);
$totalRows_rs_artists =
mysql_num_rows($rs_artists);
?>

// HTML displaying results omitted
```

3. Trending PHP Initial

```php
<?php

// PHP connection code omitted

date_default_timezone_set('America/
New_York');

$json = file_get_contents("http://
twittermusictrends.com/trending.json");
$json_a = json_decode($json, true);

// Clear table of any existing information

mysql_query('TRUNCATE TABLE
tbl_trending;');

// Insert Artists into Database

$artist = 0;
while ($artist <= 49){
  $artistName = $json_a["trending"]
[$artist]["name"];
  $artistName_fixed =
mysql_real_escape_string($artistName);
  $artistMBid = $json_a["trending"]
[$artist]["mbid"];
  $artistInitialScore =
$json_a["trending"][$artist]["score"];
  $query_insertArtists = "INSERT INTO
tbl_trending (artistID, artistName,
artistMBid, artistInitialScore,
artistPreviousScore, artistCurrentScore)
VALUES ('$artist', '$artistName_fixed',
'$artistMBid', '$artistInitialScore',
'$artistInitialScore',
'$artistInitialScore');";
    mysql_query($query_insertArtists) or
die ("Error in query:
$query_insertArtists");
  $artist++;
}?>
```

4. Trending PHP Updater

```php
<?php

// PHP connection code omitted

$json = file_get_contents("http://
twittermusictrends.com/trending.json");
$json_a = json_decode($json, true);

// Find new score and insert into database

for ($artist=0; $artist<=49; $artist++){
  // Get the artist
  $mbid = $json_a["trending"][$artist]
["mbid"];
  // Get the new scores
  $newScore = $json_a["trending"][$artist]
["score"];
  // Get their record
  $query_rs_artistCheck = "SELECT
artistID, `artistName`, `artistMBid`,
artistInitialScore, artistCurrentScore
FROM tbl_trending WHERE artistMBid='".
$json_a["trending"][$artist]["mbid"]."'
ORDER BY artistID ASC";
  $rs_artistCheck =
mysql_query($query_rs_artistCheck,
$conn_icad) or die(mysql_error());
  $row_rs_artistCheck =
mysql_fetch_assoc($rs_artistCheck);
  $totalRows_rs_artistCheck =
mysql_num_rows($rs_artistCheck);
  // Get Previous Score
  $previousScore =
$row_rs_artistCheck['artistCurrentScore'];
  // Calculate score difference
  $scoreChange = $newScore -
$previousScore;
  // Assign score direction identifier
  if ($scoreChange < 0) {
    $changeDirection = 0; // none
  } else if ($scoreChange = 0) {
    $changeDirection = 1; // down
  } else {
    $changeDirection = 2; // up
  }
  // And add them to the database
  $query_updateArtist = "UPDATE
tbl_trending SET artistPreviousScore='".
$previousScore."', artistCurrentScore='".
$newScore."', scoreChangeAmmount='".
$scoreChange."', scoreChangeDirection='".
$changeDirection."' WHERE artistMBid='".
$mbid."'";
  $updateArtist =
mysql_query($query_updateArtist);
}

// Query updated Artists in the Database

$query_rs_artists = "SELECT artistID,
`artistName`, `artistMBid`,
artistInitialScore, artistPreviousScore,
artistCurrentScore, scoreChangeAmmount,
scoreChangeDirection FROM tbl_trending
ORDER BY artistCurrentScore DESC LIMIT
50";
$rs_artists =
mysql_query($query_rs_artists, $conn_icad)
or die(mysql_error());
$row_rs_artists =
mysql_fetch_assoc($rs_artists);
$totalRows_rs_artists =
mysql_num_rows($rs_artists);

?>

// HTML displaying results omitted
```