

Twitch Chat Analysis Process
July-August 2019
Let's Play
Game: Street Fighter

PART I. TWITCH CHAT DOWNLOAD

1. Download "Twitch Chat Downloader"

```
pip install tcd
```

2. Download target let's play video using Twitch video ID.

```
# Download chat from VODs by video id
```

```
tcd --video ['insert twitch ID here'] --format irc --output ~/Downloads
```

```
# Download chat from the first 10 VODs from multiple streamers
```

```
tcd --channel ['insert twitch channel names here'] --first=10
```

output file: fort_01.txt

PART II. TEXT PROCESSING / CLEANING AND DATABASE CREATION

3. Convert chat .txt data to csv, data cleaning and processing

[Code] [data_processing.py](#)

input file: fort_01.txt

output file: fort_01.csv

PART III. E-SPORTS HIGH REACTION EVENT ANALYSIS

4. Getting chat count per second

[Code] [chatfrequency.py](#)

input file: fort_01_time.txt

output file: fort_01_freq.csv (chat count per second)

5. Getting frequency sum for 10 seconds

[Code] [excited_time.py](#)

input file: fort_01_freq.csv

output file: fort_01_freqsum.csv

6. Create .csv file [manually] with the start and end times of chat times with highest number of chat per second.

input file: fort_01_freqsum.csv

output file: fort_01_startendtime.csv

7. Get the unique times of high reaction events

[Code] [chatttime_extractor.py](#)

input file: fort_01_startendtime.csv

output file: fort_01_uniquetimes.csv

8. Creating the high reaction events chat database

[Code] [chat_dictionary.py](#)

input file: fort01.csv & fort_01_uniquetimes.csv

output file: fort_01_excitedchat.csv

PART IV. STREET FIGHTER CHAT ANALYSIS

1. Download Chat [PART I. 1-2]

output file: street_04.txt

2. Unsupervised Clustering via Kmeans

[Code] [kmeans_clustering2.py](#) #Try various k numbers

input file: street_04.txt #Try looking at elbow curve method graph

output file: street_04.csv

3. Analyze Kmeans keywords and clusters, merge similar clusters [manual]

4. Label each chat message according to determined clusters [manual]

5. Watch video and label each chat message according to event in video [manual]

output file: street_04.csv #renamed to database.csv

6. Conduct exploratory data analysis (eda) and user personalities

input file: database.csv

[Code] [testing.py](#) #finding user personalities

[Code] [unigrams_bigrams.py](#) #getting the unigrams, bigrams, and trigrams

[Code] [eda.py](#) #several analysis / processes

[Code] [eda_streetfighter.py](#) #boxplots and percentages for each chat type

[Code] [personalities_visualization.py](#) #show percentages of chat type for each personality

[Code] [personalities_event_chatmessage.py](#) #get database for each user personality

7. Round Analysis

#Manual creation of 4 database (Round 1 – Win, Round 1 – Lose, Round 2 – Win, and Round 2 – Lose)

input file: database.csv

output files: r1_win_groups.csv / r1_lose_groups.csv / r2_win_groups.csv /
r2_lose_groups.csv

#Get round percentages

[Code] [round_percentage.py](#)

input files: r1_win_groups.csv / r1_lose_groups.csv / r2_win_groups.csv /
r2_lose_groups.csv

output file: r1_w_percentage.csv / r1_l_percentage.csv /
r2_w_percentage.csv / r2_l_percentage.csv