

CS305 Programming Assignment 2

Use python to implement a simple だんま(danmaku) system.

Introduction

Danmaku is a popular form of comment. When watching videos or live shows, you can send a danmaku on the certain timestamp or in real-time. Generally, you will see your danmaku flying from one side of the screen to the other side (e.g., right to left).

In this assignment, you will need to implement a simple danmaku system of **TWO VERSIONS**, using HTTP protocol and WebSocket protocol (WS in short). respectively. You will also need to compare the two protocols.

For more details, click www.bilibili.com/video/BV1GJ411x7h7 to find more.

NOTES

(1) We have provided a skeleton for both versions. You can complete your codes based on them. However, you are free to design your own frameworks.

(2) In WS version, your submissions may include **some python files and other types of files**. Please name it clearly so that we can understand them. We recommend to **run your server on '127.0.0.1' and port is 8765**, which does not conflict with other process.

(3) In HTTP version, **we recommend to run your server on '127.0.0.1' and port is 8765**, so that you can connect to the server by accessing "127.0.0.1:8765" on your web browser.

(4) You **MUST** provide sufficient comments for us to understand your programs.

(5) Please upload your programs together with a **REPORT** that describes **your designs, running result screenshots, and analysis conclusions**.

Environment

- Python 3.8
- Recommended extra libraries are HTTP and websockets
- You may use other libraries, but you should ensure that we can also install them

Implementation details

1. In WS version, the danmaku server runs on port 8765 and IP address is 127.0.0.1 as recommended, and it should be able to:
 - Listen and accept WS messages.
 - Receive danmakus from clients, and send them to all clients and display them on their screens. We will test your program with at least 3 clients, i.e., on at least 3 browser tabs.
2. In HTTP version, the danmaku server also runs on port 8765 and IP address is 127.0.0.1 as recommended, it should:
 - Listen on 8765 and accept HTTP requests, including POST and GET requests. Please see the skeleton we provide.
 - Receive danmakus from POST requests. Your clients should send GET requests at regular intervals to get new danmakus and show them on the screen. Make sure that all danmakus not retrieved are sent correctly to each clients.

3. In the given demo, we use a python file as a server and a html file as a client for WS. You can run `server.py` and open `html` file in a browser to see the base part and create your code on it. For HTTP version, a Python skeleton is given and you can run it and open '127.0.0.1:8765' on your browser to see the constructed page. Just focus on the codes which handle GET and POST requests. You are allowed to implement the systems without using the skeletons, as long as you describe it clearly in your report.

4. Danmakus on videos and live stream (20 as bonus).

- Design a better system which can **fit videos and live stream**. For the bonus points, you can implement it with either WS or HTTP protocol.
- For videos, you need to **record the timestamp** so that next we watch the video we can find our danmakus again.

【官方 MV】Never Gonna Give You Up - Rick Astley

5366.2万播放 · 总弹幕数7.4万 2020-01-01 07:43:23



Figure 1 Videos mode (example)

For live stream, you need to implement a danmaku list to see what is talking about now. You don't need to focus much on users' name and live contents, just **distinguish different sockets**.



Figure 2 Live mode (example)

5. For videos, using a list to record received danmakus, a local file is recommended.

Grading policy

1. Tasks

- Basic part (90, 50% for HTTP version and 50% for WS version)
 - Report (10)
 - Danmakus on videos and live stream (20 as bonus)

2. Report contents

- The designs of your system, a **system diagram** is encouraged.
 - The running results, **not limited to screenshots**.
 - Analytical conclusions about the difference of the two protocols. You can analyze from the design diagram, network traffic, upgrade efficiency, and any other aspects.

Test details

1. Correctness: we will test your program to see:

- o Whether danmakus are correctly sent and appear simultaneously on each client.
 - o Whether HTTP and WS messages are captured.

2. Validity: we will look through your program to make sure that you use the corresponding protocols.

3. For bonus points, we will check whether all danmaku were stored. If you implement video and live mode, show them clearly so that we can recognize them.

What to submit

You should submit **your programs and report**. Any other files needed should also be zipped for submission. Put the two versions of systems in two separated file folders.

Please comment in your code to explain different parts of your code. You need to at least illustrate in your implementation about every part.

