PULSE: An Infrastructure for Collection of Audience Heartbeat for Visualization
Mandep Shetty  Advisor: Dr. Minseok Kwon

INTRODUCTION

Hearbeats and music have an established correlation. There are subtle variations during crescendo-like sections that is potentially more pronounced in concert settings.

Gathering heartbeat & visualizing it will make for an immersive and interactive experience for the audience.

This project aims to build a system that can gather heartbeat information from a small Arduino coupled sensor and pipe it to a server using an Android phone as an intermediary.

The system should provide easy extensibility and control over all aspects of data flow with minimum network overhead.

EVENT FLOW

1. The app checks with server if it is already registered by giving its unique device ID.

2. If registered, the app waits for the event to start.

3. If not, the user registers with name, seat no. and a sensor code. The sensor code is obtained by scanning a QR code on the sensor.

4. The server registers the user by binding unique Android device ID with the sensor ID.

5. The app then pairs with the correct sensor over Bluetooth and waits for the event to start.

6. The server signals the start of the event and the app starts sending heart beat data in BPM to the server.

7. Server stores data in a MySQL database.

8. Visualization software fetches data from the database to create visuals.

SYSTEM DESIGN

Server controls registration, events, filters etc.

MySQL DB to store BPM, user profile, device info etc.

FEATURES

The app is subscribed with both fanout and direct exchange. Fanouts enable the reception of broadcasts. Direct exchange delivers messages based on binding keys. Consumers get only those messages with the message key equal to the key they use to connect to the exchange.

Server can construct and send any filter. Can instruct all or a subset of devices to start or stop sending data or vary the rate at which it sends data.

Arduino uses adaptive thresholds and on-board accelerometer to remove outlier readings and motion noise.

Arduino only accepts connections presenting valid pin code.

RESULTS

Observed elevated heart rate when music playing. Android app uses slightly more battery than screen on standby. Framework generic enough to be easily extended or modified.

FUTURE WORK

Gather fine-grained data with more accurate sensors and perform analysis with different styles of music against a baseline heartbeat.

Use in real world concert scenario to see potentially stronger correlation due to the immersive conditions.

Perform power measurements on the Arduino under different data rates to find ideal sample rate to prolong batteries.