

OptimusQ

Gestore ed Elimina Code con Display e client di avanzamento per la gestione centralizzata di una coda

- Websocket Server: realizzato in NodeJS gestisce una lista e condivide ad ogni istante il suo stato.
- Client Cassa: scritto in Python implementa l'interfaccia grafica con Qt e comunica con il WS Server
- Client Display: è un programma Python che esegue un Browser basato su Chromium in modalità Kiosk. Il browser implementa Javascript e ha il consenso alla riproduzione di audio.

Personalizzazione

I parametri del software vengono personalizzati tramite i file .conf di ogni componente

I config dei software Cassa e Display vengono letti riga per riga dal software Python pertanto non sono ammesse variazioni di sintassi.

NOTA: Durante lo sviluppo è importante fare `push` nel repository salvando i file con gestione del ritorno a capo di Windows tramite il comando: `git config --global core.autocrlf false`

Server

Display

Config: %appdata%\EliminaCodeDisplay\display.conf

Layout predefiniti

Sono implementati due layout impostabili dal parametro **layout** - Verticale: 1 - Orizzontale: 2 È necessario modificare la dimensione della finestra manualmente

Sezione Finestra

- `anchor_x= destra|sinistra|no`

Porta la finestra al lato destro o sinistro dello schermo

Se `no` verranno utilizzati i parametri `dim` e `pos`

- `anchor_y= sopra|sotto`

Porta la finestra al lato superiore o inferiore dello schermo

Per disabilitare impostare `anchor_x: no`

Verranno quindi utilizzati i parametri `dim` e `pos`

- `width_ratio & heigth_ratio`

Da usare insieme a `anchor_x` o `anchor_y`

Indica la larghezza/altezza della finestra in rapporto alla larghezza/altezza del display

Esempio: `display_width/width_ratio = 1280/4 = 320 = width della finestra`

- `dim` Dimensioni manuali: larghezza X altezza

Esempio: `dim=400x150`

- `pos` Posizione manuale riferite all'angolo in alto a sinistra

- È possibile usare valori negativi che si riferiranno alla differenza tra la lunghezza/altezza effettiva del display e sottraendo il valore assegnati

Esempio: `pos=-100;-1`

Sezione Grafica

Si tratta di un insieme dati in formato JSON in cui sono elencate e descritte con commenti le opzioni grafiche della pagina.

→ È possibile in **fase di installazione** posizionare un pulsante nella pagina che fa il refresh della pagina al fine di verificare rapidamente le impostazioni inserite.

Suoni Display

Directory: %appdata%\EliminaCodeDisplay\assets\

In questa directory vanno inseriti i suoni nel caso si voglia far pronunciare i nomi delle casse. Si inserisce un file `.mp3` con nome del file il nome della cassa. La denominazione è **case insensitive**.

Esempio: `salumeria.mp3`

Sviluppo

Python

Versione: *3.6.8*

```
pip install pyqt5 pyqtwebengine websocket-client * PyQt5 * PyQtWebEngine * websocket-client
```

Librerie NodeJS

[Base SocketServer](#) Realizzato con versione 14

Eseguire il comando nella directory del sorgente: `npm install`

Build

È possibile eventualmente compilare i sorgenti per distribuirli su installazioni senza necessità di installare gli interpreti Python e NodeJS.

Javascript

Libreria

```
npm install --global nexex
```

Comando

```
nexex .\server.js -t x86-8.0.0
```

Python

Libreria

```
pip install pyinstaller
```

Comando

```
pyinstaller.exe .\display.py pyinstaller.exe .\cassa.py
```

Inclusione .dll mancanti

Durante la compilazione del **display** è necessario includere risorse che devono essere presenti nella root con l'eseguibile `display.exe` * `QtWebEngineProcess.exe` da `.\PyQt5\Qt\bin` * tutti i file da `.\PyQt5\Qt\resources` * i file `index.html` e `app.js` e la dir `assets\`