

REALIZZAZIONE  
IL TEAM F.S.C. PRESENTA:



---

# STRUMENTALMENTE

---

**Il team:**

Alessandro Annese  
Davide De Salvo  
Andrea Esposito  
Graziano Montanaro  
Regina Zaccaria

Informatica e Comunicazione Digitale - (TA)  
A.A. 2018/19



***F.S.C.*** — *Five Students of Computer Science*

Copyright © 2019 F.S.C.

Concesso in licenza secondo i termini della Licenza Apache, versione 2.0 (la "Licenza"); è proibito usare questo file se non in conformità alla Licenza. Una copia della Licenza è disponibile all'indirizzo:

<http://www.apache.org/licenses/LICENSE-2.0>

Se non richiesto dalla legislazione vigente o concordato per iscritto, il software distribuito nei termini della Licenza è distribuito "COSÌ COM'È", SENZA GARANZIE O CONDIZIONI DI ALCUN TIPO, esplicite o implicite. Consultare la Licenza per il testo specifico che regola le autorizzazioni e le limitazioni previste dalla medesima.

# Indice

<b>1</b>	<b>Le tecnologie utilizzate</b>	<b>5</b>
1.1	I linguaggi . . . . .	5
1.2	Il codice . . . . .	5
<b>2</b>	<b>Avviare dal codice sorgente</b>	<b>7</b>
2.1	Il Makefile . . . . .	7
<b>3</b>	<b>Documentazione del codice</b>	<b>9</b>
3.1	Funzioni . . . . .	9
3.1.1	openChildWindow(pageUrl, [windowIcon]) . . . . .	10
3.1.2	createWindow() . . . . .	10
3.1.3	promptModal(parentWindow, [options], callback) . . . . .	10
3.1.4	openMobileNavigation() . . . . .	10
3.1.5	drop(name, [defaultLinkClass]) . . . . .	11
3.1.6	setLinks(links) . . . . .	11
3.1.7	initialize(initial, base) . . . . .	12
3.1.8	changeTopic(topicName, [base]) . . . . .	12
3.1.9	initializeQuiz() . . . . .	13
3.1.10	changeQuizSlide(finalSlide) . . . . .	13
3.1.11	checkQuiz() . . . . .	13
3.1.12	playStopAudio(audioTagId, buttonRef, stopButtonId) . . . . .	13
3.1.13	showExitDialog() . . . . .	14
3.1.14	openInBrowser(link) . . . . .	14
3.1.15	openModal(content, [options], [windowIcon]) . . . . .	14
3.1.16	openOnKeyboardShortcut(shortcut, content, [openAsModal]) . . . . .	15



# 1 Le tecnologie utilizzate

## 1.1 I linguaggi

Il sistema StrumentalMente è scritto utilizzando l'**HTML5** (il che porta all'utilizzo del **CSS3** per definire le varie regole di stile) e il **JavaScript**. L'intera applicazione è basata su *Node.js*, principalmente per il pacchetto **Electron**, che consente di utilizzare i precedenti linguaggi, in combinazione con il motore di rendering *Chromium*, per creare delle applicazioni *desktop*.

## 1.2 Il codice

Il codice sorgente di StrumentalMente è fondamentalmente suddiviso in tre file principali (escludendo i file HTML e CSS utilizzati per creare l'aspetto grafico dell'applicazione). Tali file sono:

**app.js** È il file principale dell'applicazione. Contiene il processo principale (“*main process*”) di Electron e gestisce, quindi, tutti gli eventi principali di StrumentalMente, tra cui:

- L'apertura dell'applicazione e il *rendering* della prima finestra
- La chiusura dell'applicazione e le relative peculiarità di alcuni sistemi operativi (si pensi alla possibilità di ricreare la finestra appena chiusa su MacOS)
- L'apertura di finestre di dialogo
- L'apertura di finestra secondarie

**render.js** È il file principale del *rendering* dell'applicazione. Contiene tutte le funzioni da eseguire nel “*rendering process*” (processo di *rendering*) di Electron.

**main.js** Contiene tutte le funzioni che servono a dare dinamicità alle pagine HTML dell'applicazione.



## 2 Avviare dal codice sorgente

Affinchè si possa avviare StrumentalMente dal codice sorgente è necessario che siano installati `npm` e `Node.js`. Successivamente, è necessario installare le dipendenze del progetto, eseguendo i seguenti comandi nella *directory* in cui è situato il codice sorgente (cartella `src/` della *repository*):

```
npm install --save-dev electron
npm install --save-dev mousetrap
npm install --save-dev node-localstorage
```

È possibile installare *Electron Packager* per poter compilare un eseguibile (non è necessario per avviare l'applicazione) utilizzando il comando:

```
npm install -g electron-packager
```

Dopo aver installato tutte le dipendenze, è possibile avviare l'applicazione utilizzando il comando seguente all'interno della *directory* del codice sorgente:

```
npm start
```

### 2.1 Il Makefile

Per comodità, si è creato un **Makefile** per eseguire facilmente tutte le operazioni sul codice. Il comando **make** è da eseguire all'interno della *directory* principale della *repository*.

Il file **Makefile** contiene le seguenti regole:

**(all)** Avvia StrumentalMente.

**install** Installa le dipendenze del progetto ed *Electron Packager*.

**start** Avvia StrumentalMente.

**deploy** Compila StrumentalMente in un eseguibile per varie piattaforme. Al momento le piattaforme per cui la generazione è inclusa in questa regola sono:

- Windows (x64)

**docs** Ricompila tutta la documentazione partendo dai sorgenti  $\text{\LaTeX}$ .

**jsdoc** Ricompila la documentazione del codice in un file  $\text{\LaTeX}$ .

**bib2html** Converte la bibliografia in formato `BIBTEX` in HTML e la importa automaticamente nell'apposita sezione della pagina "about".



## 3 Documentazione del codice

La presente documentazione è stata generata in modo semi-automatico da *JSDoc*, *JSDoc to Markdown* e *Pandoc*.

### 3.1 Funzioni

**openChildWindow(pageUrl, [windowIcon])** Apre una finestra “figlia” e modale.

**createWindow()** Crea la finestra principale.

**promptModal(parentWindow, [options], callback)** Creazione della finestra di dialogo.

**openMobileNavigation()** Apre la navbar in modalità “mobile”. Questa funzione è mantenuta solo per consentire un eventuale eccessivo ridimensionamento della finestra.

**drop(name, [defaultLinkClass])** Permette, alla pressione di un bottone, di aprire una sottolista della navbar.

**setLinks(links)** Cambia i link e i nomi dell’argomento precedente e quello successivo a quello attuale

**initialize(initial, base)** Funzione che, al caricamento della pagina, si occupa di impostare il numero di tag section presenti all’interno della pagina nella memoria locale del browser, di impostare come sezione visibile corrente la prima (sempre all’interno della memoria locale) e di nascondere tutti i tag section successivi al primo.

**changeTopic(topicName, [base])** Cambia l’argomento correntemente mostrato.

**initializeQuiz()** Inizializza la pagina del quiz.

**changeQuizSlide(finalSlide)** Cambia la slide del quiz attualmente mostrata.

**checkQuiz()** Calcola i risultati del quiz.

**playStopAudio(audioTagId, buttonRef, stopButtonId)** Permette di avviare, mettere in pausa o stoppare un audio.

**showExitDialog()** Mostra il dialogo di richiesta di conferma di uscita.

**openInBrowser(link)** Apre un link nel browser predefinito.

**openModal(content, [options], [windowIcon])** Apre una finestra modale mostrante il contenuto richiesto.

**openOnKeyboardShortcut(shortcut, content, [openAsModal])** Apre, tramite una shortcut da tastiera, una finestra mostrante il contenuto richiesto.

### 3.1.1 openChildWindow(pageUrl, [windowIcon])

Apre una finestra “figlia” e modale.

**Kind:** global function

Param	Type	Default	Description
pageUrl	String		L’URL della pagina da aprire (assoluto o relativo)
[windowIcon]	String	./assets/icon.ico	L’icona della finestra.

### 3.1.2 createWindow()

Crea la finestra principale.

**Kind:** global function

### 3.1.3 promptModal(parentWindow, [options], callback)

Creazione della finestra di dialogo.

**Kind:** global function

Param	Type	Description
parentWindow	BrowserWindow	La finestra “genitore”
[options]	Object	Le opzioni della nuova finestra
callback	*	La funzione da richiamare alla chiusura della finestra

### 3.1.4 openMobileNavigation()

Apre la navbar in modalità “mobile”. Questa funzione è mantenuta solo perconsentire un eventuale eccessivo ridimensionamento della finestra.

**Kind:** global function

### 3.1.5 drop(name, [defaultLinkClass])

Permette, alla pressione di un bottone, di aprire una sottolistadella navbar.

**Kind:** global function

---

Param	Type	Description
name	String	L'ID della lista che si vuole controllare
[defaultLinkClass]	String	La classe iniziale del bottone

---

### 3.1.6 setLinks(links)

Cambia i link e i nomi dell'argomento precedente e quello successivoa quello attuale

**Kind:** global function

---

Param	Type	Description
links	Object	Le nuove impostazioni e link
links.previous	String	Il nome della pagina precedente
links.previousLink	String	Il link della pagina precedente (il nome del file <i>senza</i> l'estensione)
links.next	String	Il nome della pagina successiva
links.nextLink	String	Il link della pagina successiva (il nome del file <i>senza</i> l'estensione)

---

---

### 3.1.7 initialize(initial, base)

Funzione che, al caricamento della pagina, si occupa di impostare il numero di tag section presenti all'interno della pagina nella memoria locale del browser, di impostare come sezione visibile corrente la prima (sempre all'interno della memoria locale) e di nascondere tutti i tag section successivi al primo.

**Kind:** global function

Param	Type	Default	Description
initial	String		Il primo argomento
base	String	./	La cartella in cui sono situati i file degli argomenti (default: ./)

### initialize~changeSlide(slide)

La funzione, in base al valore assunto da slide (true/false) cambia la sezione corrente in quella precedente (in caso di slide = false) o in quella successiva (in caso di slide = true). Inoltre si occupa di aggiornare il numero della slide corrente nella memoria temporanea del browser. Inoltre, in base al numero di slide, si occupa di rendere visibili (o nascondere) i relativi pulsanti di spostamento (avanti con id next, indietro con id back e quiz con id quiz).

**Kind:** inner method of initialize

Param	Type	Description
slide	boolean	se è <b>true</b> , avanza di una slide, altrimenti indietreggia di una slide.

### 3.1.8 changeTopic(topicName, [base])

Cambia l'argomento correntemente mostrato.

**Kind:** global function

---

Param	Type	Default	Description
topicName	String		Il prossimo argomento
[base]	String	./	La cartella in cui è situato il file dell'argomento

---

### 3.1.9 initializeQuiz()

Inizializza la pagina del quiz.

**Kind:** global function

### 3.1.10 changeQuizSlide(finalSlide)

Cambia la slide del quiz attualmente mostrata.

**Kind:** global function

---

Param	Type	Description
finalSlide	number	La slide da aprire in seguito alla richiesta di variazione della slide. Tale valore deve essere compreso nell'intervallo $[0, n]$ , dove $n$ è il numero di slide presenti nella pagina.

---

### 3.1.11 checkQuiz()

Calcola i risultati del quiz.

**Kind:** global function

### 3.1.12 playStopAudio(audioTagId, buttonRef, stopButtonId)

Permette di avviare, mettere in pausa o stoppare un audio.

**Kind:** global function

Param	Type	Description
audioTagId	<b>String</b>	L'ID dell'elemento <code>&lt;audio&gt;</code> da controllare
buttonRef	<b>HTMLElement</b>	Un riferimento al bottone che richiama questa funzione
stopButtonId	<b>String</b>	L'ID del bottone di Stop.

### 3.1.13 `showExitDialog()`

Mostra il dialogo di richiesta di conferma di uscita.

**Kind:** global function

### 3.1.14 `openInBrowser(link)`

Apri un link nel browser predefinito.

**Kind:** global function

Param	Type	Description
link	<b>String</b>	Il link da aprire

### 3.1.15 `openModal(content, [options], [windowIcon])`

Apri una finestra modale mostrando il contenuto richiesto.

**Kind:** global function

Param	Type	Default	Description
content	<b>String</b>		Il link (assoluto o relativo) da aprire

---

Param	Type	Default	Description
[options]	Object		Le opzioni della nuova finestra
[windowIcon]	String	./assets/icon.ico	L'icona della finestra modale

---

### 3.1.16 openOnKeyboardShortcut(shortcut, content, [openAsModal])

Apre, tramite una shortcut da tastiera, una finestra mostrante il contenuto richiesto.

**Kind:** global function

---

Param	Type	Default	Description
shortcut	String		La shortcut da utilizzare
content	String		Il link (assoluto o relativo) da aprire
[openAsModal]	boolean	false	Se è <code>true</code> , la finestra sarà aperta come modale, altrimenti sarà aperta nella stessa finestra.

---