

REALIZZAZIONE
IL TEAM F.S.C. PRESENTA:

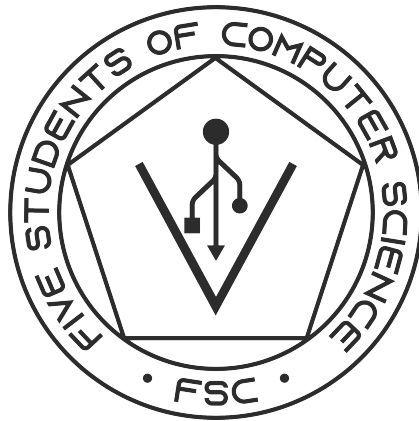


STRUMENTALMENTE

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

1	Le tecnologie utilizzate	9
1.1	I linguaggi	9
1.2	Il codice	9
2	Avviare dal codice sorgente	11
2.1	Il Makefile	11
3	Documentazione del codice	13
3.0.1	Classes	13
3.1	Funzioni	13
3.1.1	Main	14
3.1.2	accordo	14
3.1.3	accordo	15
3.1.4	openMobileNavigation()	16
3.1.5	setLinks(firstSlideNumber, links)	16
3.1.6	initialize(initial, base, totalNumberOfSlides)	17
3.1.7	changeTopic(topicName, [base])	18
3.1.8	initializeQuiz()	18
3.1.9	changeQuizSlide(finalSlide)	18
3.1.10	playStopAudio(audioTagId, buttonRef, stopButtonId)	19
3.1.11	warnIfIncomplete(previousQuizId, previousQuizName, topicToOpenName, callback)	19
3.1.12	showExitDialog()	19
3.1.13	showExitFromQuizDialog(toOpen)	20
3.1.14	showQuizDialog(nomeQuiz, score, total, return_link)	20
3.1.15	openInBrowser(link)	20
3.1.16	openModal(content, [options], [windowIcon])	20
3.1.17	getUsername()	21
3.1.18	setUsername(newUsername)	21
3.1.19	getQuiz(id)	21
3.1.20	openOnKeyboardShortcut(shortcut, content, [openAsModal])	21
3.1.21	script_load()	22
3.1.22	verify_and_store()	22
3.1.23	correct_chord()	22
3.1.24	script_load()	22
3.1.25	verify_and_store()	23
3.1.26	correct_chord()	23
4	Bug e problemi noti	25
4.1	La velocità	25

Elenco delle tabelle

Elenco delle figure

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** (con la presenza di pochi file scritti in *TypeScript*, un dialetto del 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.ts 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

Esegue i suoi compiti utilizzando la classe `Main` definita nel file `StrumentalMente.ts`.

StrumentalMente.ts Contiene la definizione della classe principale che gestisce gli eventi del *main process* di Electron.

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.

quiz.js

accordo.js

2 Avviare dal codice sorgente

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

```
yarn install
```

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

```
yarn 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.

`yarn-install` Installa yarn.

`install` Installa le dipendenze del progetto (con l'opzione `--force` che forza l'eventuale reinstallazione).

`start` Avvia StrumentalMente.

`deploy` Compila StrumentalMente in un eseguibile per la piattaforma su cui il comando è stato eseguito.

`docs` Ricompila tutta la documentazione partendo dai sorgenti \LaTeX .

`jsdoc` Ricompila la documentazione del codice in un file \LaTeX .

`bib2html` Converte la bibliografia in formato $\text{BIB}\text{\TeX}$ in HTML e la importa automaticamente nell'apposita sezione della pagina "about".

`docx-to-html` Converte i file DOCX in file HTML.

3 Documentazione del codice

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

3.0.1 Classes

Main

accordo Classe accordo.

accordo Classe accordo.

3.1 Funzioni

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

setLinks(firstSlideNumber, links) Cambia i link e i nomi dell'argomento precedente e quello successivo a quello attuale

initialize(initial, base, totalNumberOfSlides) 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.

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

warnIfIncomplete(previousQuizId, previousQuizName, topicToOpenName, callback) Mostra un messaggio all'utente se il quiz propedeutico all'argomento scelto non è stato completato. Se l'utente conferma di voler proseguire, viene effettuata l'azione richiesta, altrimenti non si attua alcuna azione.

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

showExitFromQuizDialog(toOpen) Mostra il dialogo di richiesta di conferma di uscita dal quiz.

showQuizDialog(nomeQuiz, score, total, return_link) Mostra il dialogo con il punteggio del quiz.

openInBrowser(link) Apre un link nel browser predefinito.

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

getUsername() Ritorna l'username collegato a StrumentalMente.

setUsername(newUsername) Imposta l'username dell'utente.

getQuiz(id) Ottiene il risultato del quiz scelto

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

script_load() Seleziona un numero casuale compreso tra 1 e 7 e ne imposta l'accordo da richiedere all'utente.

verify_and_store() Verifica che le selezioni effettuate dall'utente siano corrette in base all'accordo presentatogli e memorizza: se la selezione è corretta (1) o non corretta (0), le checkbox selezionate (e non) e l'accordo che l'utente doveva riprodurre.

correct_chord() In base al numero di accordo che l'utente doveva riprodurre, ripristina la sequenza di selezioni corretta nello schema.

script_load() Seleziona un numero casuale compreso tra 1 e 7 e ne imposta l'accordo da richiedere all'utente.

verify_and_store() Verifica che le selezioni effettuate dall'utente siano corrette in base all'accordo presentatogli e memorizza: se la selezione è corretta (1) o non corretta (0), le checkbox selezionate (e non) e l'accordo che l'utente doveva riprodurre.

correct_chord() In base al numero di accordo che l'utente doveva riprodurre, ripristina la sequenza di selezioni corretta nello schema.

3.1.1 Main

Kind: global class

new Main()

La classe contenente la logica principale dell'applicazione.

3.1.2 accordo

Classe accordo.

Kind: global class

- accordo
 - new accordo(nome, dita, tasto_iniziale)
 - new accordo(nome, dita, tasto_iniziale)

new accordo(nome, dita, tasto_iniziale)

Param	Type	Description
nome	String	Stringa che indica il nome dell'accordo

Param	Type	Description
dita	Boolean	Sequenza di valori logici che indicano se la checkbox corrispondente è stata selezionata o meno
tasto_iniziale	number	Indica il numero del capotasto iniziale dell'accordo

new accordo(nome, dita, tasto_iniziale)

Param	Type	Description
nome	String	Stringa che indica il nome dell'accordo
dita	Boolean	Sequenza di valori logici che indicano se la checkbox corrispondente è stata selezionata o meno
tasto_iniziale	number	Indica il numero del capotasto iniziale dell'accordo

3.1.3 accordo

Classe accordo.

Kind: global class

- accordo
 - new accordo(nome, dita, tasto_iniziale)
 - new accordo(nome, dita, tasto_iniziale)

new accordo(nome, dita, tasto_iniziale)

Param	Type	Description
nome	String	Stringa che indica il nome dell'accordo

Param	Type	Description
dita	Boolean	Sequenza di valori logici che indicano se la checkbox corrispondente è stata selezionata o meno
tasto_iniziale	number	Indica il numero del capotasto iniziale dell'accordo

new accordo(nome, dita, tasto_iniziale)

Param	Type	Description
nome	String	Stringa che indica il nome dell'accordo
dita	Boolean	Sequenza di valori logici che indicano se la checkbox corrispondente è stata selezionata o meno
tasto_iniziale	number	Indica il numero del capotasto iniziale dell'accordo

3.1.4 openMobileNavigation()

Apri la navbar in modalità "mobile". Questa funzione è mantenuta solo per consentire un eventuale eccessivo ridimensionamento della finestra.

Kind: global function

3.1.5 setLinks(firstSlideNumber, links)

Cambia i link e i nomi dell'argomento precedente e quello successivo a quello attuale

Kind: global function

Param	Type	Description
firstSlideNumber	number	Il numero della prima slide dell'argomento corrente

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)

3.1.6 initialize(initial, base, totalNumberOfSlides)

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: ./)
totalNumberOfSlides	number		Il numero totale di pagine per la sezione.

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	number	Il numero della slide da aprire.

3.1.7 `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

3.1.8 `initializeQuiz()`

Inizializza la pagina del quiz.

Kind: global function

3.1.9 `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.

3.1.10 playStopAudio(audioTagId, buttonRef, stopButtonId)

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

Kind: global function

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

3.1.11 warnIfIncomplete(previousQuizId, previousQuizName, topicToOpenName, callback)

Mostra un messaggio all'utente se il quiz propedeutico all'argomento scelto non è stato completato. Se l'utente conferma di voler proseguire, viene effettuata l'azione richiesta, altrimenti non si attua alcuna azione.

Kind: global function

Param	Type	Description
previousQuizId	String	L'id del quiz propedeutico
previousQuizName	String	Il nome del quiz (da comunicare all'utente)
topicToOpenName	String	Il nome dell'argomento che si vuole aprire
callback	*	La funzione da eseguire se l'utente accetta di proseguire.

3.1.12 showExitDialog()

Mostra il dialogo di richiesta di conferma di uscita.

Kind: global function

3.1.13 showExitFromQuizDialog(toOpen)

Mostra il dialogo di richiesta di conferma di uscita dal quiz.

Kind: global function

Param	Type	Description
toOpen	String	Il file da aprire se è cliccato il tasto 'Sì'. Il percorso è relativo rispetto alla cartella principale.

3.1.14 showQuizDialog(nomeQuiz, score, total, return_link)

Mostra il dialogo con il punteggio dei quiz.

Kind: global function

Param	Type	Description
nomeQuiz	String	Il nome del quiz.
score	number	Il punteggio ottenuto.
total	number	Il punteggio totale possibile.
return_link	String	Il file da aprire se è cliccato il tasto 'Ok'. Il percorso è relativo rispetto alla cartella principale.

3.1.15 openInBrowser(link)

Apri un link nel browser predefinito.

Kind: global function

Param	Type	Description
link	String	Il link da aprire

3.1.16 openModal(content, [options], [windowIcon])

Apri una finestra modale mostrando il contenuto richiesto.

Kind: global function

Param	Type	Default	Description
content	String		Il link (assoluto o relativo) da aprire
[options]	Object		Le opzioni della nuova finestra
[windowIcon]	String	./assets/icon.ico	L'icona della finestra modale

3.1.17 getUsername()

Ritorna l'username collegato a StrumentalMente.

Kind: global function

3.1.18 setUsername(newUsername)

Imposta l'username dell'utente.

Kind: global function

Param	Type	Description
newUsername	String	Il nuovo username.

3.1.19 getQuiz(id)

Ottiene il risultato del quiz scelto

Kind: global function

Param	Type	Description
id	String	L'id del quiz di cui interessa il risultato.

3.1.20 openOnKeyboardShortcut(shortcut, content, [openAsModal])

Apri, 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 true, la finestra sarà aperta come modale, altrimenti sarà aperta nella stessa finestra.

3.1.21 `script.load()`

Seleziona un numero casuale compreso tra 1 e 7 e ne imposta l'accordo da richiedere all'utente.

Kind: global function

3.1.22 `verify_and_store()`

Verifica che le selezioni effettuate dall'utente siano corrette in base all'accordo presentatogli e memorizza: se la selezione è corretta (1) o non corretta (0), le checkbox selezionate (e non) e l'accordo che l'utente doveva riprodurre.

Kind: global function

3.1.23 `correct_chord()`

In base al numero di accordo che l'utente doveva riprodurre, ripristina la sequenza di selezione corretta nello schema.

Kind: global function

3.1.24 `script.load()`

Seleziona un numero casuale compreso tra 1 e 7 e ne imposta l'accordo da richiedere all'utente.

Kind: global function

3.1.25 verify_and_store()

Verifica che le selezioni effettuate dall'utente siano corrette in base all'accordo presentato e memorizza: se la selezione è corretta (1) o non corretta (0), le checkbox selezionate (e non) e l'accordo che l'utente doveva riprodurre.

Kind: global function

3.1.26 correct_chord()

In base al numero di accordo che l'utente doveva riprodurre, ripristina la sequenza di selezioni corretta nello schema.

Kind: global function

4 Bug e problemi noti

4.1 La velocità

L'applicazione non è scritta come applicazione nativa, quindi la sua velocità e responsività non è lontanamente paragonabile a quella di un'applicazione nativa.

Questa limitazione è dovuta principalmente a tre fattori: il linguaggio di programmazione utilizzato (JavaScript), il *framework* Electron (conosciuto per la sua lentezza) e il motore di *rendering* su cui quest'ultimo si basa (Chromium).

Nonostante gli sforzi compiuti dal team di sviluppo per limitare questo problema, per esempio sacrificando la leggibilità del codice a favore di tipologie di istruzione più rapide (si pensi al `forEach` in confronto al classico ciclo `for`: il primo è mediamente più lento del secondo ma è più leggibile), l'applicazione resta poco rapida.