



# Corsa allo spazio

Autori:

Ivone Danilo, Esposito Daniele, Camardi Martina, Gravina Antonio,  
Siragusa Mattia

Copyright © 2022

# Sommario

---

<b>1 Le tecnologie utilizzate</b>	2
1.1 I linguaggi	2
1.2 Il codice	2
1.3 Il packaging	2
<b>2 Avviare dal codice sorgente</b>	3
<b>3 Documentazione del codice</b>	3
<b>3.1 Funzioni</b>	3
3.1.1 functionLayout.js	3
3.1.2 navMenu.js	3
3.1.3 Stars.js	3
3.1.4 testBuchiBianchi.js	4

# 1 Le tecnologie utilizzate

## 1.1 I linguaggi

Per lo sviluppo di Corsa allo spazio sono stati utilizzati:

- **HTML5** per la realizzazione della struttura delle pagine del sistema
- **CSS3** per la gestione dell'aspetto estetico delle pagine
- **JavaScript** per la gestione dell'interattività del sistema

## 1.2 Il codice

Escludendo i file HTML e CSS per la struttura e l'aspetto grafico dell'applicazione, i file utilizzati sono:

- **functionLayout.js** per consentire l'uscita dall'applicazione
- **navMenu.js** per il funzionamento del sidebar menu e dei sottomenu da esso accessibili
- **Stars.js** per impostare la velocità di movimento delle stelle in background e il loro colore
- **testBuchiBianchi.js** per la gestione dei test sui buchi bianchi
- **testBuchiNeri.js** per la gestione dei test sui buchi neri
- **testFenomeni.js** per la gestione dei test sui fenomeni astronomici
- **testGalassie.js** per la gestione dei test sulle galassie
- **testPianeti.js** per la gestione dei test sui pianeti
- **testSatelliti.js** per la gestione dei test sui satelliti
- **testStelle.js** per la gestione dei test sulle stelle
- **testStoria.js** per la gestione dei test sulla storia dell'astronomia

## 1.3 Il packaging

Il packaging dei file per la realizzazione dell'applicazione è stato effettuato mediante *Electron*, un framework open source. Inoltre, sono stati scaricati *Git* (per l'interfaccia a riga di comando) e *NodeJS*, un runtime system open source per l'esecuzione di codice JavaScript.

## 2 Avviare dal codice sorgente

Per avviare Corsa allo spazio da codice sorgente è necessario accedere alla cartella *src* della repository e cliccare sul file *title.html* per accedere alla schermata iniziale mediante il browser predefinito.

Tuttavia, per questioni di dimensioni, è fortemente consigliato avviare Corsa allo spazio direttamente dall'applicazione installata con l'installer, come descritto nel manuale utente.

## 3 Documentazione del codice

### 3.1 Funzioni

#### 3.1.1 functionLayout.js

##### **function start ()**

Fa partire l'applicazione.

##### **function closeApp 1/2/3()**

Apri la pagina di chiusura dell'applicazione.

Nota per la funzione *closeApp*: quando si prova a chiudere una finestra non aperta da JavaScript si visualizza un messaggio di errore, per cui la funzione non può chiudere la schermata principale dell'applicazione. Per risolvere questo problema, la pagina di chiusura a cui l'utente è indirizzato mostra all'utente il tasto ✕ da premere per uscire dall'applicazione.

#### 3.1.2 navMenu.js

##### **function()**

Se il menu viene aperto si attiva il blur della section. In caso si trovasse sulla pagina di test, disattiva il bottone di conferma.

Analogamente, se il menu viene chiuso il blur della section è disattivato.

Inoltre, nella funzione è inclusa la funzionalità di aprire o chiudere gli ulteriori sottomenu.

#### 3.1.3 Stars.js

##### **for (let i = 0; i < numBalls; i++)**

All'interno di questo ciclo for si inizializzano gli oggetti *Ball* (le stelle in background) che ottengono colore e dimensione casuale entro un range dato. Questi elementi vengono poi mostrati in schermo.

*numBalls* è il numero di stelle da mostrare sullo sfondo, definito all'inizio del file (50).

**balls.forEach((el, i, ra)**

Questa funzione è utilizzata per definire le animazioni degli oggetti *Ball*.

**3.1.4 testBuchiBianchi.js****function loadQuiz()**

Carica il test con le domande disposte in ordine casuale.

**function deselectAnswers()**

Deseleziona le risposte alla generazione del test.

**function getSelected()**

Consente di selezionare la risposta.

**submitbtn.addEventListener('click', () =>**

Conta il punteggio e dà il risultato consentendo la ricarica alla fine del test.

Le funzioni degli altri file di test sono analoghe a quelle elencate sopra.