Documentazione Trova la Parola

1 Introduzione 3

1.1 Informazioni sul progetto 3

1.2 Abstract 3

1.3 Scopo 3

2 Analisi 4

2.1 Analisi del dominio 4

2.2 Analisi e specifica dei requisiti 4

2.2.1 Spiegazione elementi tabella dei requisiti: 6

2.3 Use case 7

2.4 Pianificazione 7

2.5 Analisi dei mezzi 8

2.5.1 Software 8

2.5.2 Hardware 8

3 Progettazione 9

3.1 Design dell’architettura del sistema 9

3.2 Design dei dati e database 9

3.3 Design delle interfacce 9

3.4 Design procedurale 9

4 Implementazione 10

4.1 GUI 10

4.2 Classe Gioco 10

4.2.1 Costruttore 11

4.2.2 Funzioni 11

5 Test 36

5.1 Protocollo di test 36

5.2 Risultati test 41

5.3 Mancanze/limitazioni conosciute 44

6 Consuntivo 44

7 Conclusioni 44

7.1 Sviluppi futuri 45

7.2 Considerazioni personali 45

8 Glossario 45

9 Bibliografia 46

9.1 Bibliografia per articoli di riviste: 46

9.2 Bibliografia per libri 46

9.3 Sitografia 46

10 Allegati 46

# Introduzione

## Informazioni sul progetto

* Allievo del progetto: Marco Conforti
* Docente Responsabile: Geo Petrini
* Scuola: Scuola d’Arti e Mestieri Trevano
* Sezione: Informatica
* Materia: Modulo 306
* Data Inizio: 01.09.2023
* Termine Consegna: 01.12.2023 15:45

## Abstract

The objective of the Cruci Puzzle application is to provide users with an engaging and fun gaming experience through the resolution of a cruci puzzle. The application created thanks to an html page and the style managed thanks to CSS and all the rest of the code is managed by JavaScript. The website was created for a magazine that publishes these games, thanks to this project the magazine will be able to create these words without having to do it and in a much faster way than usual. The site contains a generator with modes, difficulties, fonts, and the possibility of inserting a personalized dictionary.

## Scopo

Lo scopo del progetto è creare un’applicazione ad una rivista che pubblica giochi che permetti di generare il gioco *trova la parola*. L’applicazione necessità di altre personalizzazioni per la creazione della matrice come la modalità, la difficoltà l’inserimento di un dizionario personalizzato e il font.

# Analisi

## Analisi del dominio

Il programma richiesto serve ad una rivista che pubblica giochi di enigmistica, il programma deve generare un gioco chiamato “trova la parola”, dove avendo una lista di parole si dovrà creare una tabella dove saranno presenti tutte le parole della lista, queste parole potranno essere messe in orizzontale in verticale o in senso obliquo, le parole possono essere da sotto verso sopra e viceversa.

## Analisi e specifica dei requisiti

|  |  |
| --- | --- |
| **ID: REQ-001** | |
| **Nome** | Implementazione generatore puzzlecruci |
| **Priorità** | Alta |
| **Versione** | 1.0 |
| **Note** | Dal sito web base si dovrà generare un puzzlecruci |
| **Sotto requisiti** | |
| **001** | Si dovrà avere una funzione per generare parole |
| **002** | Si dovrà avere una funzione per inserire le parole all’interno della griglia in modo corretto (direzione corretta, sovrapposizione no ma incroci si) |
| **003** | Si dovrà avere una funzione per visualizzare la tabella |
| **004** | Si dovrà avere una funzione per visualizzare la lista delle parole |

|  |  |
| --- | --- |
| **ID: REQ-002** | |
| **Nome** | Creazione Export + Possibilità di stampare |
| **Priorità** | Alta |
| **Versione** | 1.0 |
| **Note** | Bisogna creare un export in html e/o txt e la possibilità di stampare il puzzle crucis. |
| **Sotto requisiti** | |
| **001** | Si dovrà avere una funzione per esportare la tabella in html e/o txt |
| **002** | Si dovrà avere una funzione per la stampa |

|  |  |
| --- | --- |
| **ID: REQ-003** | |
| **Nome** | Creazione soluzione |
| **Priorità** | Media |
| **Versione** | 1.0 |
| **Note** | Si dovranno aggiungere la soluzione al puzzlecruci generato, utile anche come test. |
| **Sotto requisiti** | |
| **001** | Si dovrà avere una funzione per rilevare la soluzione. |

|  |  |
| --- | --- |
| **ID: REQ-004** | |
| **Nome** | Creazione sito web di base |
| **Priorità** | Alta |
| **Versione** | 1.0 |
| **Note** | Sito web dove aggiungere funzionalità aggiuntive. |
| **Sotto requisiti** | |
| **001** | Si dovrà aggiungere una funzione per far scegliere all’utente la modalità. |
| **002** | Si dovrà aggiungere una funzione per far scegliere all’utente il font. |
| **003** | Si dovrà aggiungere una funzione per far scegliere all’utente la difficoltà. |
| **004** | Si dovrà aggiungere una funzione per far scegliere all’utente parole nuove tramite un file. |

|  |  |
| --- | --- |
| **ID: REQ-005** | |
| **Nome** | Implementazione generatore puzzlecruci |
| **Priorità** | Media |
| **Versione** | 2.0 |
| **Note** | Implementazione di funzionalità aggiuntive. |
| **Sotto requisiti** | |
| **001** | Si dovrà aggiungere una funzione per cambiare la modalità (bambino, normale, parola nascosta casuale solo per normale, per bambino lettere casuali). |
| **002** | Si dovrà aggiungere una funzione per modificare il font della griglia. |
| **003** | Si dovrà aggiungere una funzione per cambiare la difficoltà della griglia anche in base alla modalità se modalità bambini n. parole se modalità normale grandezza campo. |
| **004** | Si dovrà aggiungere una funzione per far scegliere all’utente parole nuove tramite un file. |

|  |  |
| --- | --- |
| **ID: REQ-006** | |
| **Nome** | Rendere l’applicativo multipiattaforma |
| **Priorità** | Alta |
| **Versione** | 1.0 |
| **Note** | Applicativo funzionante su più piattaforme |
| **Sotto requisiti** | |
| **001** | L’applicativo dovrà funzionare su Windows. |
| **002** | L’applicativo dovrà funzionare su Linux. |
| **003** | L’applicativo dovrà funzionare su Mac OS. |

|  |  |
| --- | --- |
| **ID: REQ-007** | |
| **Nome** | Parole devono essere gestite a regola |
| **Priorità** | Alta |
| **Versione** | 1.0 |
| **Note** | Gestione delle lettere delle parole |
| **Sotto requisiti** | |
| **001** | Le lettere delle parole dovranno essere scritte in maiuscolo |
| **002** | Le lettere delle parole non dovranno avere accenti |
| **003** | Le lettere delle parole non dovranno avere caratteri speciali |

### Spiegazione elementi tabella dei requisiti:

**ID**: identificativo univoco del requisito

**Nome**: breve descrizione del requisito

**Priorità**: indica l’importanza di un requisito nell’insieme del progetto, definita assieme al committente. Ad esempio, poter disporre di report con colonne di colori diversi ha priorità minore rispetto al fatto di avere un database con gli elementi al suo interno. Solitamente si definiscono al massimo di 2-3 livelli di priorità.

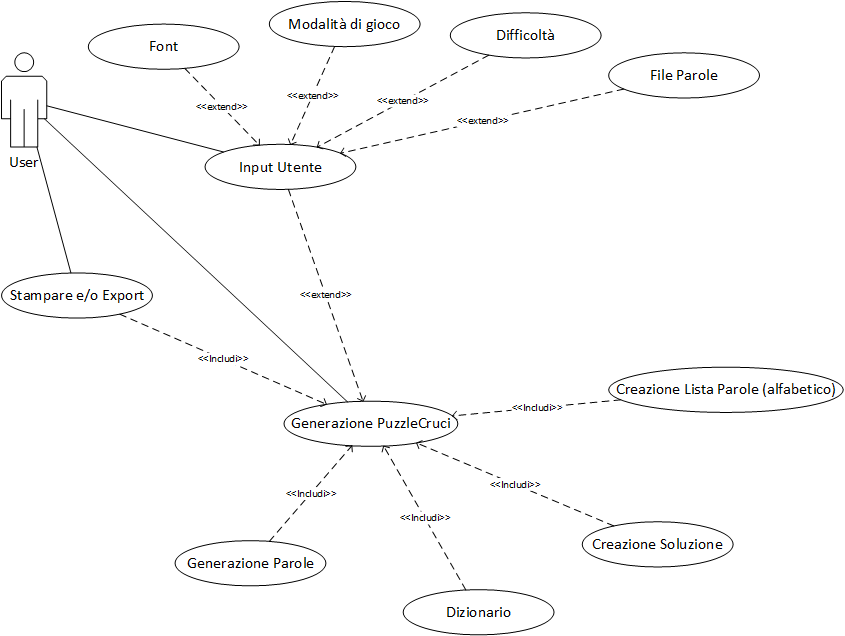
**Versione**: indica la versione del requisito. Ogni modifica del requisito avrà una versione aggiornata.

Sulla documentazione apparirà solamente l’ultima versione, mentre le vecchie dovranno essere inserite nei diari.

**Note**: eventuali osservazioni importanti o riferimenti ad altri requisiti.

**Sotto requisiti**: elementi che compongono il requisito.

## Use case



All’interno del mio applicativo ho esclusivamente un tipo di 1 utente.

* User: L’utente in questione avrà accesso a tre tipi di input, uno dove inserire degli input più specifici per la creazione del proprio Puzzle Cruci, quest’ultimo è un’estensione della generazione del Puzzle Curci. Un altro che appunto genera il Puzzle Cruci includendo anche la lista parole, la generazione delle parole, il dizionario e la soluzione. Infine la possibilità di esportare il Puzzle Cruci generato, anche questa funzionalità è inclusa nella generazione del Puzzle Cruci ma dovrà essere espressamente richiesta dall’utente a contrario della soluzione, la lista, la generazione delle parole e del dizionario.

## Pianificazione

|  |
| --- |
| Figura 1: Esempio di diagramma di Gantt. |

## Analisi dei mezzi

### Software

* **Visual Studio Code 1.78.2**
* **HTML 5**
* **Javascript 1.8.5**
  + **Estensione Live Server 5.7.9**
* **CSS 3**
* **AJAX 3.7**
* **Google Chrome 117.0.5938.150**

### Hardware

* **1 Server Web:** L’applicativo avrà bisogno un server web o dovrà usare l’estensione Live Server di visual studio code per funzionare, poiché richiama delle funzioni Ajax, durante lo sviluppo viene usata l’estensione Live Server.
* **1 Computer:** Il PC usato avrà le seguenti componenti:
  + CPU: i7-9700
  + RAM: 32 GB
  + SSD: 512 GB
  + SCHEDA VIDEO: NVIDIA GeForce RTX 2060

# Progettazione

## Design procedurale

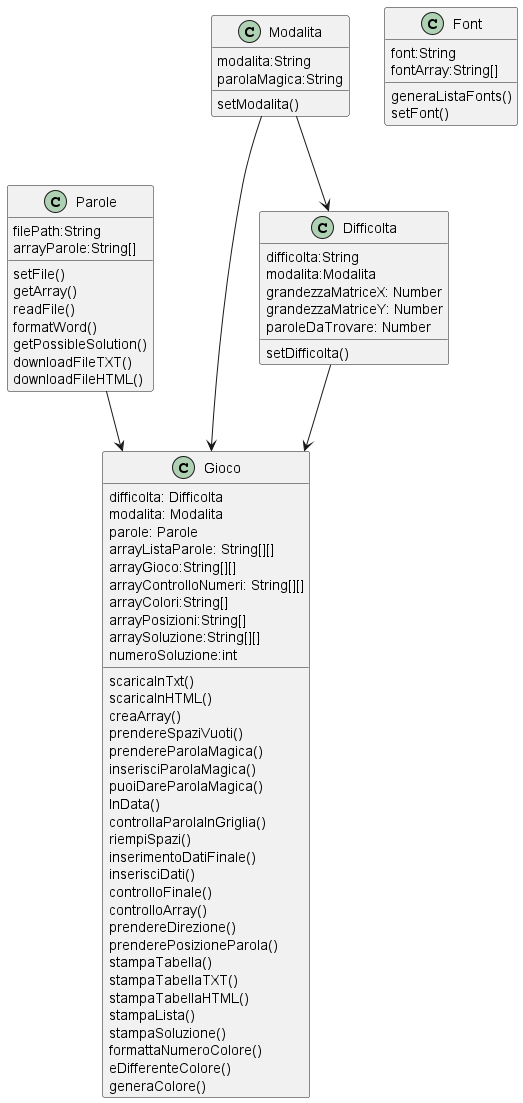


Figura 2 - Class Diagram

Nel seguente Class Diagram si vede come sono le classi all’interno dei file JavaScript.

# Implementazione

## GUI

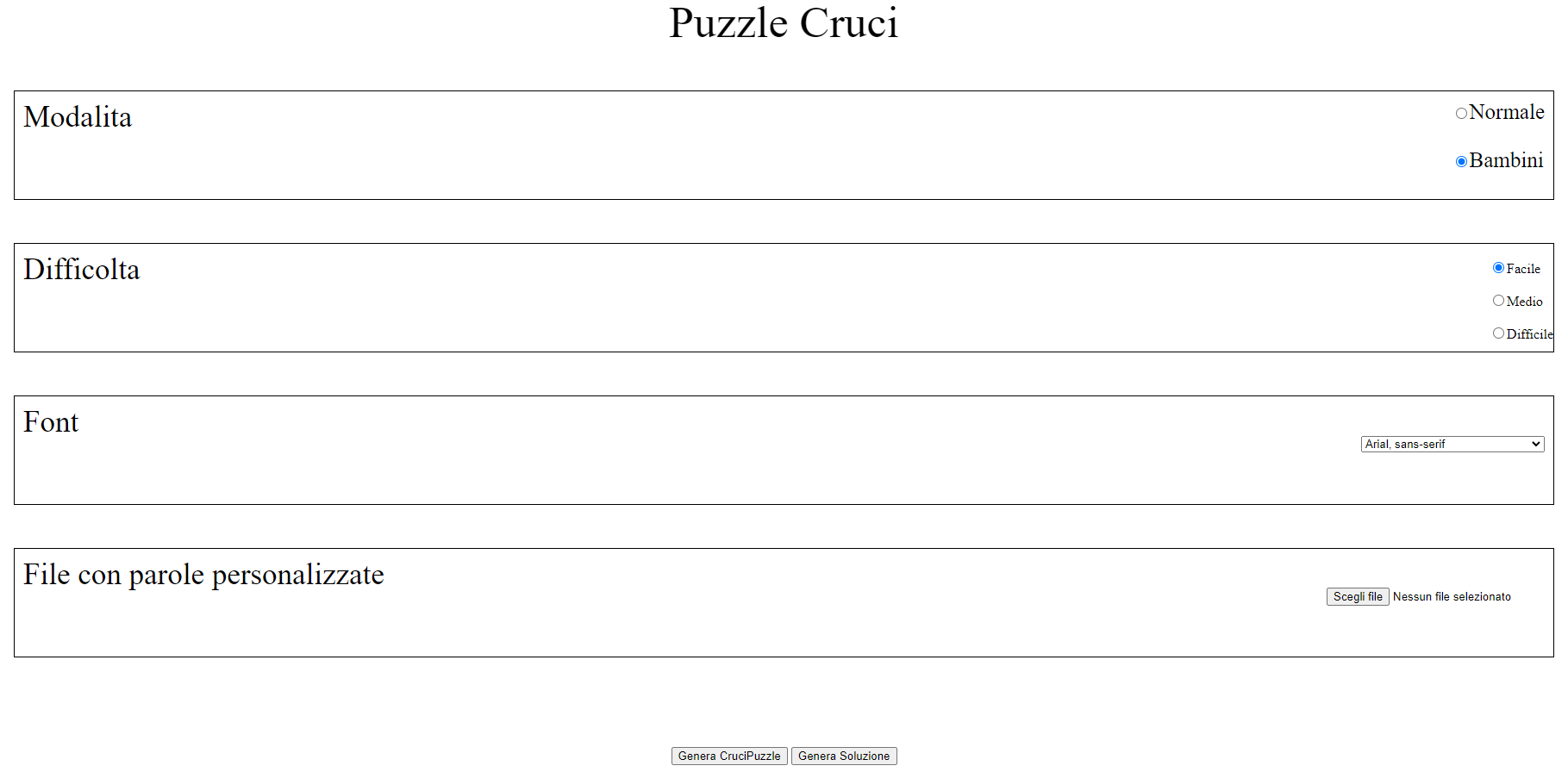
****

Figura 3 – GUI

Nella figura sopra è raffigurata la GUI finale del progetto. Possiamo notare che all’interno della GUI del sito ci sono due modalità una per bambini che è quella di default e quella normale, la differenza tra le due è che la prima non ha la parola nascosta ma lettere casuali al suo interno mentre la modalità normale contiene la parola nascosta che si potrà trovare una volta trovate tutte le parole della lista. Sotto questa funzionalità possiamo trovare la difficoltà che può essere di tre tipi facili, medio o difficili, la difficoltà cambia anche in base alla modalità infatti se abbiamo la modalità bambini e mettiamo facile avremo 15 parole da trovare in una matrice 10x10, con difficoltà media avremo 50 parole da cercare in 18x18, con difficoltà difficile avremo 70 parole da cercare in una matrice 22x22, se abbiamo invece la modalità normale possiamo e difficoltà facile avremo un numero indefinito di parole da cercare in una matrice 15x15, se inseriamo la difficoltà media avremo sempre un numero indefinito di parole da cercare in una matrice 20x20, se inseriamo la difficoltà difficile avremo un numero indefinito di parole in una matrice 25x25. Come penultima funzionalità da inserire troviamo il font fa modificare si possono scegliere 13 font diversi compreso il default. L’ultima funzione consente di importare un file di estensione “.txt”, all’interno del file si dovranno avere delle parole da inserire all’interno della matrice del gioco, maggiori informazioni sono nel capitolo dedicato al file con parole.

## Classe Gioco

La classe gioco si occupa della generazione del cruci puzzle e la sua soluzione aiutato dalle varie classi difficoltà, modalità e Parole. Oltre alla classe Font che funziona in modo autonomo, la classe Gioco è la sola classe che comunica con il file script.js che interagisce con index.html, quindi è la classe centrale del progetto.

### Costruttore

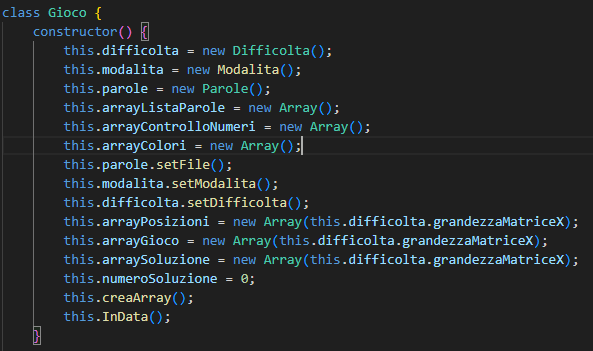


Figura 4 - Costruttore Classe Gioco

Nel seguente costruttore vediamo l’instanziazione della Classe *Difficolta* che serve soprattutto per la grandezza della matrice, inoltre vediamo l’instanziazione della Classe *Modalita* che serve principalmente per appunto sapere la modalità e quindi quale funzione deve fare la Classe, in seguito vediamo l’instanziazione della classe *Parole* che serve principalmente a prendere le parole dai dizionari e metterle dentro un array o anche per recuperare le possibili soluzioni stampabili e per la stampa tramite file. Inoltre verranno inizializzati gli array *ListaParole* che è appunto un array dove all’interno troviamo le parole del dizionario selezionato, l’array *arrayControlloNumeri* che salva al suo interno salva le parole che sono già state usate all’interno della matrice in modo che non ci possono essere doppioni, l’array *arrayColori* che serve appunto a salvare i colori usati nella soluzione in modo che non possono essere più utilizzati per altre soluzioni, l’array *arrayPosizioni* che ha al suo interno la direzione delle parole, questo serve a fare in modo che non ci siano due parole sovrapposte con la stessa direzione, l’array *arrayGioco* che al suo interno ha tutte le lettere del gioco ,l’array *arraySoluzione* che serve a tenere traccia delle parole dove si trovano all’interno della matrice grazie all’aiuto della variabile *numeroSoluzione* che per ogni parola ha un numero diverso e all’interno dell’array citato prima vengono inseriti questi numeri così da sapere dove si trovano le lettere della parola. Inoltre vengono invocati i seguenti metodi *setFile* dalla classe *parole*, *setModalita* dalla classe *Modalita*, *setDifficolta* dalla classe *Difficolta* e i metodi della classe Gioco *creaArray* e *InData*.

### Funzioni

All’interno di questo capitolo si vedranno i metodi della Classe Gioco.

#### Funzione creaArray

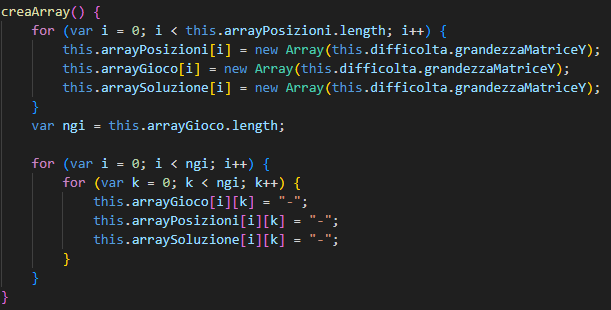


Figura 5 - Funzione creaArray

La Funzione creaArray serve appunto a inizializzare gli array mettendo un ulteriore array all’interno degli array già inizializzati nel costruttore, per ogni cella inserita nel costruttore viene messo un array, così facendo diventa la tabella che poi verrà stampata. Infine, ho deciso di riempire gli array con il carattere *“-”* così da capire quali celle rimanevano vuote.

#### Funzione prendereDirezione

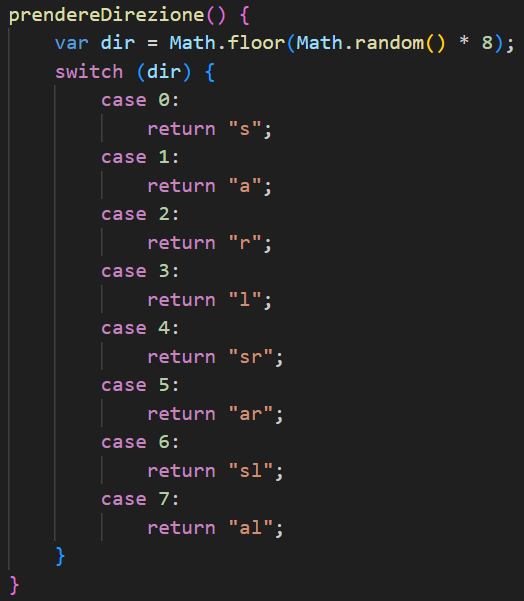


Figura 6 - Funzione getDirezione

La funzione prendereDirezione genera una posizione casuale per come inserire la parola se la funzione ritornerà **“s”** significa che la parola andrà dal lato destro a quello sinistro, mentre se ritornerà **“a”** significa che la parola andrà da destra verso sinistra, mentre se ritornerà **“r”** significa che la parola andrà dall’alto verso il basso, mentre se ritornerà **“l”** significa che la parola andrà da sotto verso sopra, mentre se ritornerà **“sr”** significa che la parola andrà dal alto sinistra verso il basso destra in obliquo, mentre se ritornerà **“ar”** significa che la parola andrà dal basso sinistra verso l’alto destra in obliquo, mentre se ritornerà **“sl”** significa che la parola andrà dal basso destra verso l’alto sinistra in obliquo, mentre se ritornerà **“al”** significa che la parola andrà dal alto destra verso basso sinistra in obliquo.

#### Funzione prenderePosizioneParola

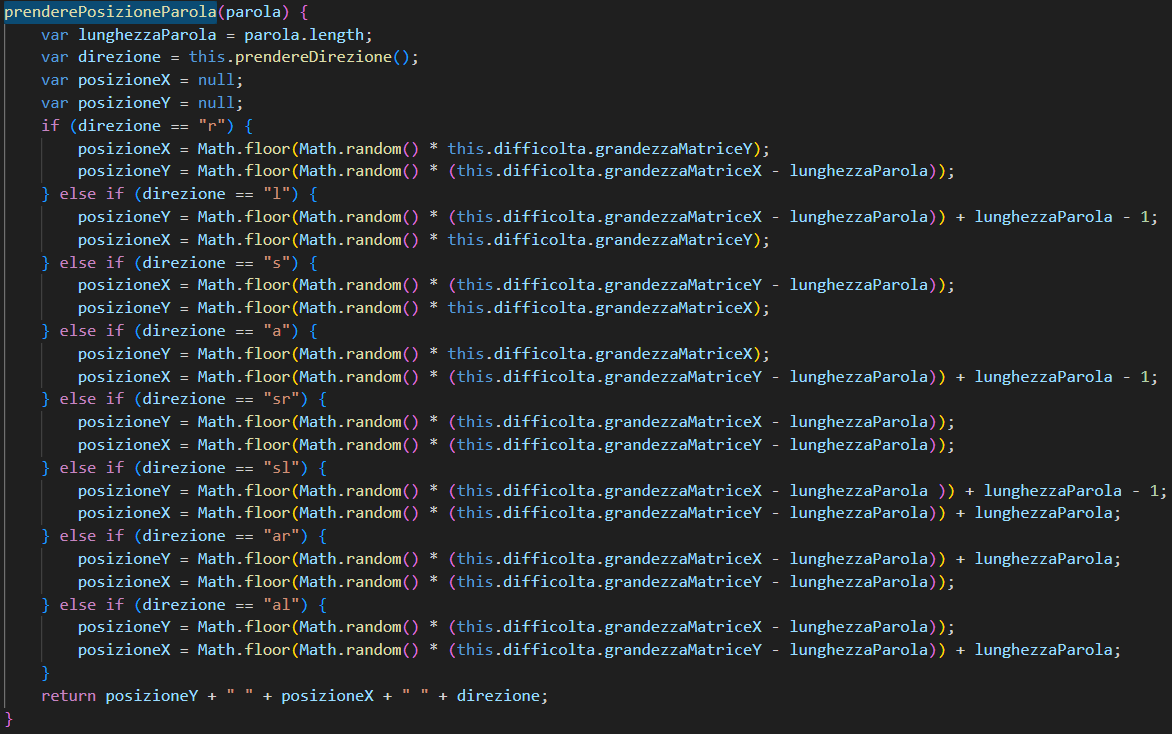


Figura 7 - Funzione getPositionParola

La funzione prenderePosizioneParola serve a dare una coordinata (x;y) dove inizia la parola da inserire. Per ogni direzione la funzione gestisce la generazione dei due punti grazie alla lunghezza della parola che dovrà essere passata come parametro. Se la direzione è dall’altro verso il basso la coordinata y sarà limitata alle posizioni superiori mentre la x è libera stessa cosa contraria se la direzione è dal basso verso l’alto. Il contrario sarà con la direzione da destra verso sinistra, infatti, la y rimarrà casuale mentre la x sarà limitata a dipendenza della direzione. Discorsi simili per le altre quattro posizioni, semplicemente nessuna coordinata sarà libera ma tutte limitate.

#### Funzione controlloFinale

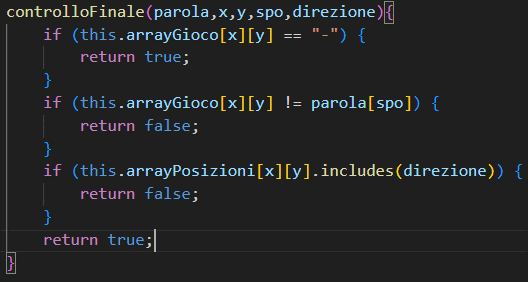


Figura 8 - Funzione controlloFinale

La funzione controlloFinale serve a controllare le lettere specifiche della parola che si sta controllando l’inserimento. Il metodo per funzionare userà i parametri, parola che è la parola che si sta controllando, x che è la coordinata del punto della lettera che si sta controllando, y è l’altra coordinata del punto della lettera che si sta controllando, il numero del carattere che si sta controllando (spo) e la direzione della parola. Se all’interno della cella c’è il carattere “- “significa che la cella è libera, perciò, si può tornare true, se nella cella c’è una lettera diversa da quella della parola si ritornerà false, se la cella contiene la stessa direzione della parola non si potrà inserire perciò tornerà false. Se nessuna delle tre condizioni viene soddisfatta la funzione ritornerà true, poiché non ci sarà nessun problema all’inserimento.

#### Funzione controlloArray

Immagine che contiene testo, schermata, Carattere, software

Descrizione generata automaticamente

Figura 9 - Funzione controlloArray

La funzione controlloArray serve a controllare che all’interno dell’array matrice può essere effettivamente inserita la parola, questa funzione richiede come parametri la parola, la x dove inizia la parola, la y dove inizia la parola e la direzione che deve avere la parola. La funzione controlla facendo un ciclo per ogni lettera della parola controlla che all’interno della matrice la lettera può essere effettivamente inserita, se anche solo una lettera non va bene ritornerà false altrimenti torna true, infine in entrambe le situazioni si aumentare il counter in modo che si controlleranno anche le altre parole. ovviamente per ogni direzione vengono controllate tutte le lettere.

#### Funzione inserisciDatiFinale

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Figura 10 - Funzione inserisciDatiFinale

La funzione inserisciDatiFinale serve ad inserire i dati all’interno degli array, questa funzione è stata creata per sostituire le parti ripetute nella funzione inserisciDati documenta dopo la seguente. I parametri vengono dati dalla funzione appena citata, grazie ai parametri si riescono a inserire i dati corretti nei posti corretti, il parametro parola passa la parola che si sta inserendo, il parametro x è appunto la coordinata all’interno dell’array bidimensionale sull’asse della x, mentre la y è la coordinata all’interno dell’array bidimensionale sull’asse y. Il parametro spo indica numericamente la posizione della lettera all’interno della parola. La funzione inserisce all’interno dell’arrayGioco la lettera della parola, in seguito inserisce la direzione della parola nella stessa posizione della lettera ma in un array dedicato alle posizioni. L’ultima cosa che fa è inserire il numero di soluzione all’interno dell’array dedicato alla soluzione, grazie a questo numeri all’interno dell’array è possibile creare la soluzione della tabella generata.

#### Funzione inserisciDati

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Figura 11 - Funzione InserisciDati

La funzione inserisciDati serve a fare un ciclo dove poi verranno inseriti i dati tramite la funzione vista precedentemente, la variabile *spo* conterrà, come descritto nella descrizione dell’altra funzione, il numero della parola che si sta inserendo in quel momento, ogni volta che viene fatto un ciclo viene aumentata di uno, la variabile sss contiene la coordinata x che dovrà essere convertita in un numero altrimenti invece che sommare parola.lenght si concateneranno (nelle altre direzione potrebbe essere anche o solo la coordinata y rispetto a x non cambia nulla solo che si guarda l’altro asse). Queste variabili servono al corretto inserimento, infatti viene creato un ciclo personalizzato per ogni direzione, all’interno di ogni ciclo vengono inseriti i parametri spiegati nella funzione inserisciDatiFinale, come spiegato nella funzione precedente.

#### Funzione controllaParolaInGriglia

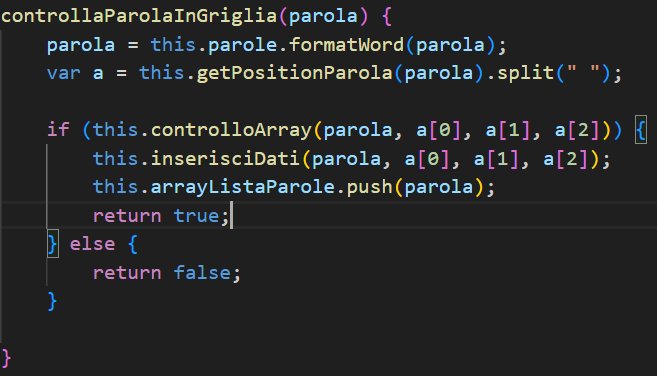


Figura 12- Funzione controllaParolaInGrid

La Funzione controllaParolaInGriglia prende la posizione x e y casuali generati nella funzione getPositionParola e effettua che la parola può essere effettivamente inserita all’interno della matrice, se i controlli vanno a buon fine, la parola viene inserita all’interno della matrice e inserita all’interno dell’array della lista delle parole e ritorna true, mentre se i controlli non vanno a buon fine la funzione ritornerà false.

#### Funzione prendereSpaziVuoti

Immagine che contiene testo, schermata, Carattere, software

Descrizione generata automaticamente

Figura 13 - Funzione prendereSpaziVuoti

La funzione prendereSpaziVuoti serve appunto a prendere gli spazi vuoti all’interno della matrice di gioco. Viene utilizzata esclusivamente per questa funzione, facendo un ciclo sull’array di gioco questa matrice conta quanti “-” trova aumentando per ogni volta che ne vede uno la variabile spaziVuoti, che sarà la variabile finale con tutte gli spazi vuoti che ritornerà la funzione.

#### Funzione prendereParolaMagica

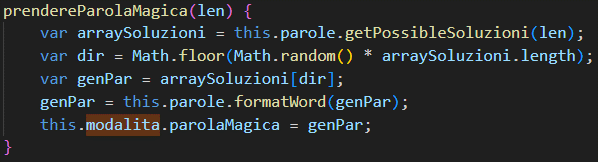


Figura 14 - Funzione prendereParolaMagica

La funzione prendereParolaMagica serve appunto a generare la parola magica (o segreta) gli viene dato come parametro la lunghezza che deve avere la parola e usando la funzione della classe Parole prende tutte le soluzioni possibili, dopodiché genera un numero casuale in base alla grandezza dell’array, e poi prende la parola la formatta a regola e la inserisci come parola magica nella classe Modalita. Non sarà suo compito inserire nell’array o nella stampa.

#### Funzione puoiDareParolaMagica

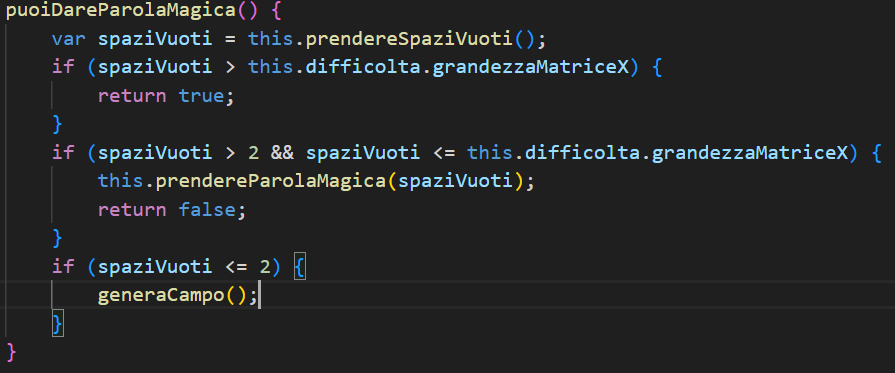


Figura 15 - Funzione puoiDareParolaMagica

La funzione puoiDareParolaMagica serve a vedere se si devono ancora inserire parole all’interno della griglia oppure è possibile inserire la parola magica (o segreta) all’interno della griglia. Il funzionamento è molto semplice, infatti se gli spazi vuoti presi dalla funzione prendereSpaziVuoti sono maggiori della larghezza (uguale all’altezza) dice al ciclo di continuare a inserire parole, nel caso invece che gli spazi vuoti sono maggiori di due e minori della grandezza della matrice verrà generata la parola magica che sarà di lunghezza degli spazi vuoti ovviamente.

#### Funzione stampaTabella

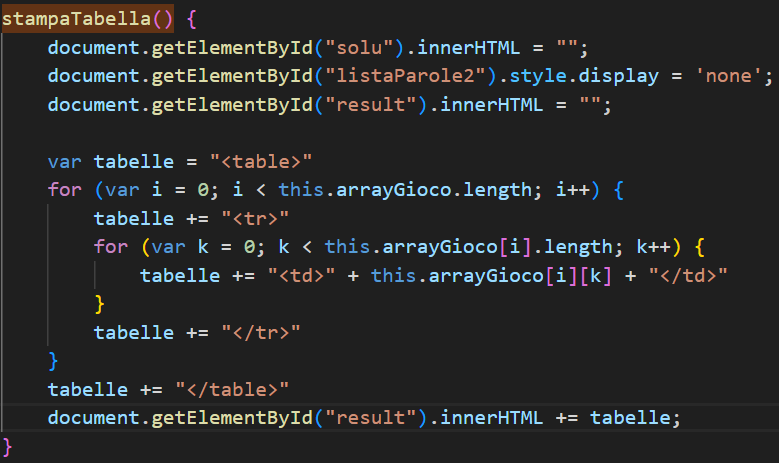


Figura 16 - Funzione stampaTabella

La funzione stampaTabella serve a creare la struttura della tabella e farla visualizzare all’interno della pagina web. Inizialmente svuota la soluzione, in modo che, se precedentemente era stata visualizzata una soluzione verrà rimossa, stessa cosa per la *lista delle parole 2* che sarebbe la lista delle parole delle soluzioni solo che al posto di svuotarla la rendiamo invisibile, questo perché nella funzione dove si inserisce la lista della tabella viene aggiunta anche quella della soluzione quindi non rendendola visibile ma inserendo i dati risolviamo il problema. Dopodiché viene svuotata la tabella visibile, in modo che non vengono concatenate le due tabelle insieme ma quella generata precedente non verrà rimossa. Dopo aver fatto questo la funzione tramite due cicli crea la tabella visualizzabile e la inserisce all’interno della matrice.

#### Funzione stampaLista

Immagine che contiene testo, schermata, Carattere, software

Descrizione generata automaticamente

Figura 17 - Funzione stampaLista

La funzione stampaLista ha il compito di stampare le liste della soluzione e della matrice principale. Inizialmente svuota le due liste in modo cancella la lista della generazione precedente, dopo aver fatto questo ordina in modo alfabetico la lista riempita nella funzione controllaParolaInGriglia come richiesto da requisiti. Dopo aver fatto questo crea una lista di parole all’interno di un div per poi stamparlo all’interno di un div che automaticamente separerà la lista in cinque colonne.

#### Funzione inserisciParolaMagica

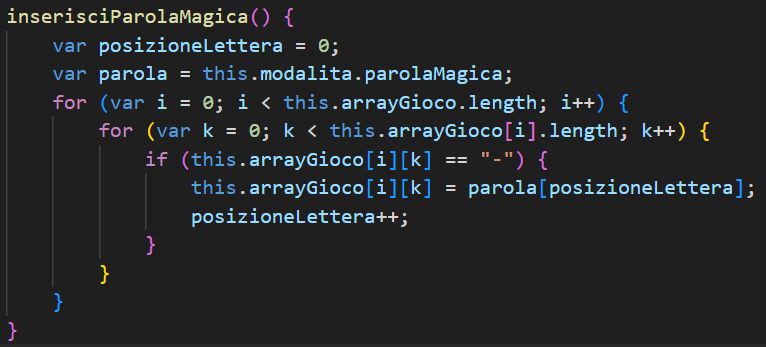


Figura 18 - Funzione inserisciParolaMagica

La funzione inserisciParolaMagica serve ad inserire la parola magica all’interno della matrice. Inizialmente inizializza una variabile che indica la posizione della lettera della parola, in seguito prende appunto la parola magica. Infine, abbiamo un doppio ciclo che fa un ciclo su tutto l’array bidimensionale, ogni posizione controllo che l’array non abbia niente di inserito nel caso specifico che il carattere sia *“- “.* Nel caso che ci sia proprio questo carattere viene inserita la lettera della parola e viene aumentato il counter della posizione delle lettere. Fa questo fino a quando non viene finito di fare il ciclo sull’array, alla fine otteniamo l’array completo con la soluzione.

#### Funzione riempiSpazi

Immagine che contiene testo, schermata, schermo, software

Descrizione generata automaticamente

Figura 19 - Funzione riempiSpazi

La funzione riempiSpazi serve a riempire gli spazi rimasti vuoti nella modalità bambini, inizialmente viene inizializzata la variabile alfabeto dove appunto abbiamo tutto le lettere dell’alfabeto possibili, quest’ultime vengono rese tutte in maiuscolo. In seguito, viene fatto un ciclo sull’array della matrice di gioco principale con il doppio ciclo, per ogni posizione viene fatto il controllo che non sia una cella vuota in questo caso con il carattere “-”, se effettivamente contiene il carattere viene preso un numero random da 0 a 25 e viene presa la lettera a quella posizione e inserita alla posizione corrente.

#### Funzione InData

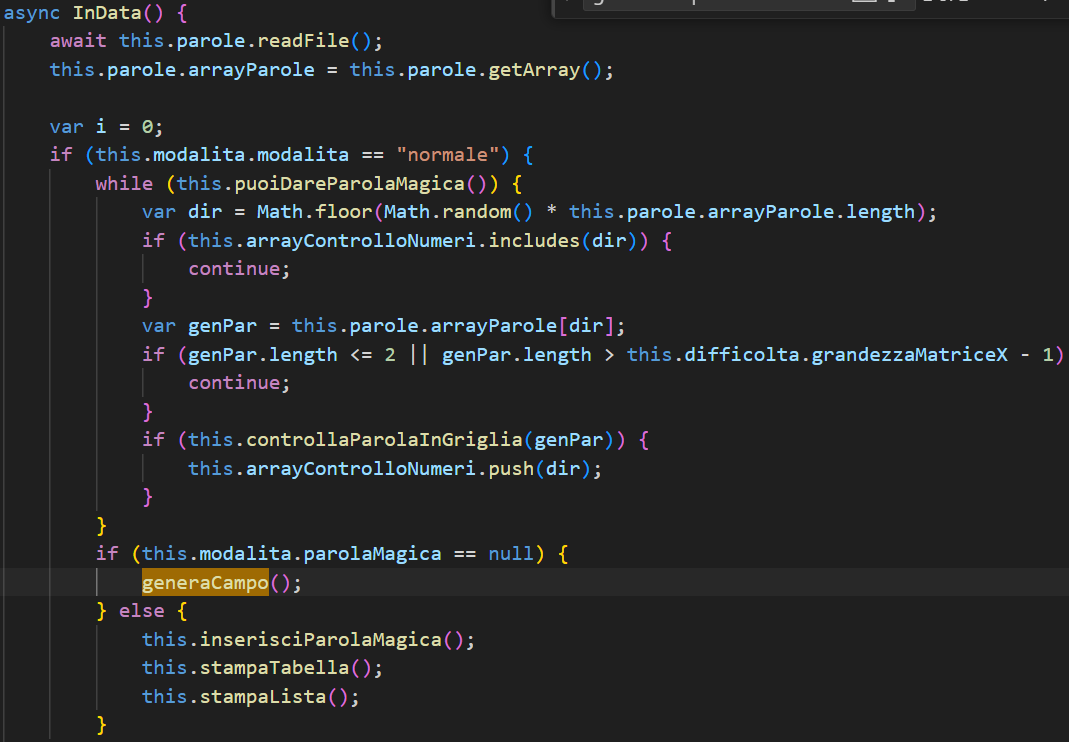


Figura 20 - Funzione InData - Modalità Normale

La Funzione InData serve principalmente a verificare che le parole sono state inserite tutte in base alla modalità inserita dall’utente e dopo che è stato verificato vengono invocati metodi per la buona riuscita del programma. Inizialmente carica il dizionario in un array, inoltre effettua la verifica che la parola possa essere effettivamente inserita all’interno dell’array senza essere troppo grande e alla verifica che la parola non sia effettivamente già stata inserita all’interno della matrice se rispetta queste due condizioni procede verso controlli specifici e se vengono rispettati si inserirà all’interno del array che controlla che non ci siano parole ripetute il numero della parola così non potrà essere ripetuta. All’interno dell’immagine vediamo la modalità normale, quella con la parola nascosta; infatti, vediamo che non c’è un limite di parole all’interno del ciclo ma controlla se la parola magica può essere inserita nel caso che può essere inserita invocherà altri metodi per le varie stampe.

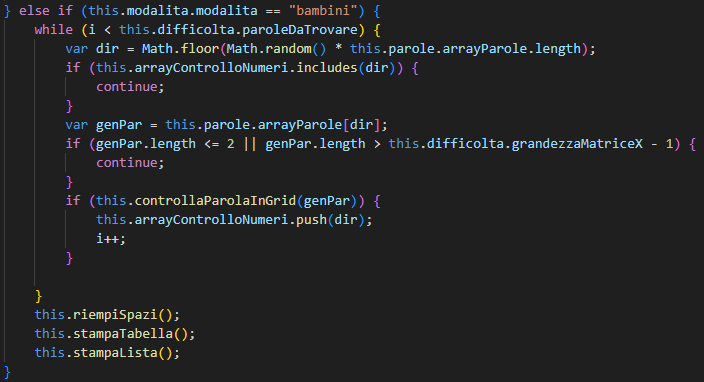


Figura 21 - Funzione InData - Modalità Bambini

Mentre nell’immagine sovrastante si vede la modalità bambini, con lettere casuali al posto delle lettere dalla parola nascosta; infatti, vediamo che nel ciclo viene richiesto il numero di parole da inserire e dopo che sarà stato effettuato questo procederà con riempire gli spazi.

#### Funzione stampaTabellaTXT

Immagine che contiene testo, schermata, software, schermo

Descrizione generata automaticamente

Figura 22 - Funzione stampaTabellaTXT

La funzione stampaTabellaTXT serve ad assegnare al file txt il contenuto che dovrà avere. Inizialmente esegue un ciclo sulla tabella e la mette nella variabile e in seguito si inserisce nella variabile la lista delle parole, a riguardo di questo, ho dovuto effettuare una modifica; infatti, se il programma percepisce che ci sono ancora due argomenti li inserisce uno dopo l’altro, mentre se invece ce n’è uno inserirà solo quello.

Se la modalità sarà normale inserirà anche uno spazio dove mettere al parola magica.

#### Funzione stampaTabellaHTML



Figura 23 - Funzione stampaTabellaHTML

La funzione stampaTabellaHTML serve ad assegnare il contenuto della tabella html al file appunto html. Inizialmente esegue un ciclo sulla tabella e la mette nella variabile e in seguito si inserisce nella variabile la lista delle parole, a riguardo di questo, ho dovuto effettuare una modifica; infatti, se il programma percepisce che ci sono ancora due argomenti li inserisce uno dopo l’altro, mentre se invece ce n’è uno inserirà solo quello. Se la modalità sarà normale inserirà anche uno spazio dove mettere al parola magica.

#### Funzioni scaricaInTXT e scaricaInHTML

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Figura 24 - Funzione scaricaInTXT e Funzione scaricaInHTML

Le funzioni scaricaInTXT e scaricaInHTML servono a scaricare i file nelle rispettive estensioni, le funzioni in sé vengo prese dalla classe Parole ma vengono ancora inizializzati in questa classe in modo che al file script si dovrà inserire esclusivamente la classe Gioco senza usarne un'altra inoltre è molto utile per passare le tabelle e molto più semplice.

#### Funzione eDifferenteColore

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Figura 25 - Funzione eDifferenteColore

La funzione eDifferenteColore serve a vedere se il colore che viene passato come parametri r che indica il colore rosso, g che indica il colore verde, b che indica il colore blu. Se il colore è già presente nell’array viene ritornato false, mentre se non è presente torna true ed inserisce nell’array di colori il colore in modo che non può essere usato.

#### Funzione formattaNumeroColore

Immagine che contiene testo, Carattere, schermata, numero

Descrizione generata automaticamente

Figura 26 - Funzione formattaNumeroColore

La funzione formattaNumeroColore serve a formattare il colore, mettendo davanti al numero del colore. Ho dovuto fare questa funzione altrimenti mi dava alcuni errori poiché la formazione richiesta è rgb(rrr,ggg,bbb), e ovviamente se uscita un numero, per esempio 5 per rosso 10 per blu e 30 per verde il risultato era rgb(510,300,000) ed ovviamente non usciva quello richiesto anche perché i valori massimi sono 255, per risolvere questo problema i numeri precedenti fanno diventare il risultato nel seguente modo rgb(005,010.030) che è effettivamente come richiesto dal programma.

#### Funzione generaColore

Immagine che contiene testo, schermata, schermo, software

Descrizione generata automaticamente

Figura 27 - Funzione generaColore

La funzione generaColore serve a generare un colore casuale, la funzione genera un colore casuale, evitando gli estremi in modo che il colore non è né troppo chiaro né troppo scuro. Dopo aver generato formatta a regola come descritto nella funzione sopra e in seguito controlla se esiste già il colore se non esiste lo inserisce in un array grazie alla funzione eDifferenteColore, e continua con il ciclo, altrimenti genera un altro colore. La funzione genererà già tutto i colori per ogni soluzione grazie al ciclo inziale.

#### Funzione stampaSoluzione

Immagine che contiene testo, schermata, software

Descrizione generata automaticamente

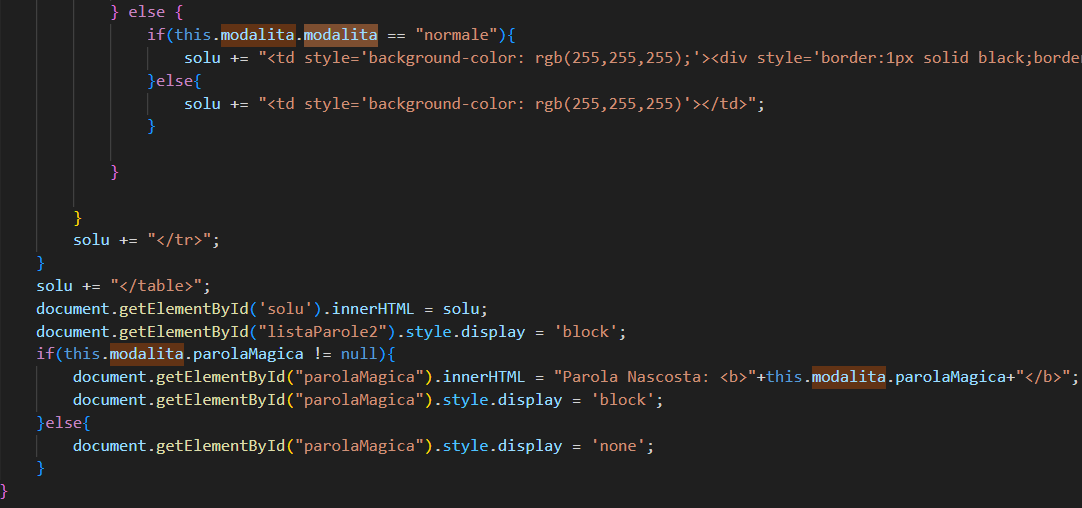


Figura 28 - Funzione stampaSoluzione

La funzione stampaSoluzione serve appunto a stampare la soluzione all’interno della pagina web, il funzionamento è molto semplice, inizialmente genera tutti i colori con la funzione generaColore, dopo fa un ciclo di tutto l’array delle soluzioni e per ogni cella guarda quali numeri ci sono dopo aver fatto questo li somma alla variabile mediaColor per ogni colore quindi se in una cella ci sono due numeri significa che ci sono due colori, quindi somma entrambi i valori rgb nelle variabili specifiche, alla fine delle somma viene fatta una media dei colori in modo che non spicca un colore in particolare se una cella viene usata da più parole ma c’è un colore simile per tutte le parole. Infine, crea la tabella con il colore specifico e in base alla modalità inserisce o meno la parola magica. Conclude rendendo visibili le cose corrette.

## Classe Modalita

La classe Modalita gestisce le modalità del gioco, le modalità sono esclusivamente due, la prima chiamata modalità bambini è quella dove si devono esclusivamente trovare parole e non c’è nessuna parola finale da trovare. L’altra modalità invece si chiama modalità normale e come nell’altra si devono trovare le parole nelle otto direzioni possibili, ma in questa modalità una volta trovate tutte le parole rileva una parola finale.

### Costruttore

Immagine che contiene testo, Carattere, schermata

Descrizione generata automaticamente

Figura 29 - Costruttore Classe Modalita

Il costruttore di questa classe è molto semplice, infatti, contiene esclusivamente due variabili che sono la modalità e la parola magica. La variabile modalità indica esclusivamente la modalità che l’utente sta usando di default viene messa la modalità bambini. La variabile parolaMagica invece serve a memorizzare la parolaMagica.

### Funzioni

#### Funzione setModalita

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Figura 30 - Funzione setModalita

La funzione setModalita serve a impostare la modalità, questa funzione prende l’elemento checkbox con id modNormale e guarda se è stato selezionato se non è stato selezionato controlla se è selezionato il checkbox con id modBambini e in base a quale è selezionato viene inserita la modalità. Il programma è fatto apposta che l’utente deve obbligatoriamente selezionare

## Classe Difficolta

La classe Difficolta gestisce tutto quello che riguarda appunto la difficolta, questa classe dipende molto dalla classe Modalita; infatti, la difficolta cambia molto grazie alla modalita, se abbiamo la modalita da bambini ci saranno un numero definito di parole da generare in una certa tabella dove la grandezza della matrice di x e y sono uguali, infatti, viene generato un quadrato. Se la modalità è normale vengono generate un numero indefinito di parole in una matrice dove la grandezza di x e y sono uguali, le parole sono indefinite poiché ne inserirà affinché non potrà aggiungerne e si potrà inserire la parola nascota. In conclusione, questa classe gestisce tutte le grandezze della matrice, ed eventualmente il numero di parole da cercare.

### Costruttore

Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente

Figura 31 - Costruttore classe Difficolta

Il costruttore di questa classe è molto semplice, troviamo come prima cosa l’inizializzazione della variabile modalita che, come spiegato nella descrizione della classe, serve a capire cosa selezionare e modificare. Troviamo di seguito la difficolta che viene indicata “facile” di default, le altre due modalità possibili sono “medio” e “difficile” tra difficoltà cambia solamente la grandezza della matrice ed eventualmente le parole da trovare. La variabile che troviamo dinanzi è la variabile delle parole da trovare che, come spiegato precedentemente, viene usata e modificata solamente se la modalita indicata è la modalita bambini, questa variabile serve appunto ad indicare il numero di parole che l’utente dovrà cercare. Le ultime due variabili indicano la grandezza della matrice x e y che attualmente sono in qualsiasi difficolta uguali ma per uno sviluppo futuro ho voluto già mettere due grandezze diverse, queste due appunto sono la grandezza della matrice indicativamente x e y.

### Funzioni

#### Funzione setDifficolta

Immagine che contiene testo, schermata

Descrizione generata automaticamente

Figura 32 - Funzione setDifficolta

La funzione setDifficolta serve a impostare la difficolta e le sue funzioni principali ovvero la grandezza della matrice ed eventualmente le parole da cercare. Innanzitutto, la funzione imposta la modalita, di seguito controlla la modalità e controlla quale checkbox è stato selezionato ed in base a quello selezionato riesce inserisce la grandezza della matrice, la difficolta e, se la modalita è bambino, le parole da che l’utente dovrà trovare.

## Classe Parole

La classe Parole serve a gestire tutto quello che usa le parole, come dizionari, la lettura di file, la formattazione delle parole, gestire le possibili soluzioni ed infine tutto quello che riguarda il download dei file.

Questa classe non dipende da nessun’altra classe bensì è indispensabile alla classe centrale Gioco.

### Costruttore

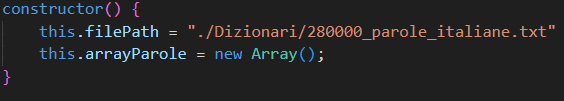


Figura 33 - Costruttore classe Parole

Il costruttore della classe parole è molto semplice, infatti, contiene esclusivamente queste due variabili la prima contenente il percorso del file delle parole, mentre la seconda è l’array dove verranno inserite tutte le parole che sono state prese dal file.

### Funzioni

#### Funzione setFile

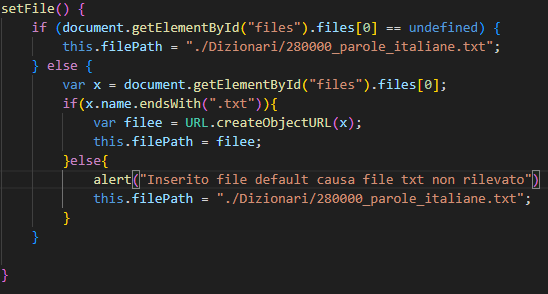


Figura 34 - Funzione setFile

La funzione setFile serve appunto ad impostare il file usato come dizionario nella variabile filePath. Inizialmente controlla che nel file chooser ci sia un file selezionato se non c’è alcun file selezionato inserisce quello di default inserito anche nel costruttore. Altrimenti prende il file e gli crea un URL fittizio con *URL.createObjectURL()* infatti questa funzione serve a creare un URL sostituito a quello reale poiché il browser per privacy non permette la visualizzazione del URL del file inserito dall’utente, perciò, l’unico modo è usare questo modo. E di seguito viene assegnata alla variabile filePath.

#### Funzione readFile

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Figura 35 - Funzione readFile

La funzione readFile serve a leggere il file. Come prima cosa crea una promise ovvero un’azione asincrona questo serve per fare in modo che prima che venga fatta qualsiasi altra operazione viene terminata questa, questo viene effettuato dal metodo resolve(), infatti questo metodo serve a dire che l’operazione è terminata con successo e si può andare avanti. Dentro la promise viene letto il file grazie ad una richiesta GET con XMLHttpRequest, dopodiché ogni parola presa dal file viene inserita all’interno dell’array e questo array viene inserito in un div invisibile dove si può trovare questo array, ho usato questa soluzione poiché se provavo ad utilizzare l’array inizializzato nel costruttore dava un errore e non funzionava; perciò, ho utilizzato questa soluzione anche se ce n’è sono di migliori.

#### Funzione getArray

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

La funzione getArray è molto legata alla funzione readFile, infatti, prende le parole nel div invisibile e le inserisce in nell’array delle parole, inoltre è necessario levare l’ultima parola poiché è un carattere vuoto, e lo faccio usando il metodo pop. Ed infine ritorno l’array delle parole con tutte le parole inserite.

#### Funzione formatWord

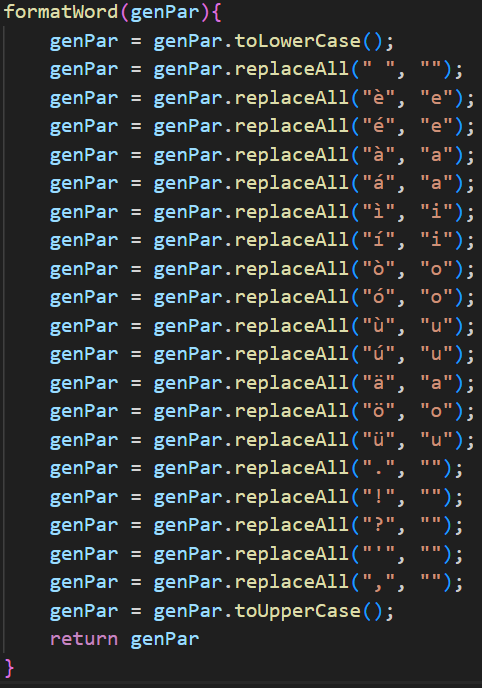


Figura 36 - Funzione formatWord

La funzione formatWord serve a formattare, come richiesto dai requisiti, infatti, non devono essere presenti accenti e altri caratteri. Inizialmente mette tutte le parole minuscole e controlla che non ci siano accenti altrimenti li sostituisce con parole senza accento. Ed infine come richiesto la parola viene messa in maiuscolo e viene ritornata.

#### Funzione getPossibleSolution

Immagine che contiene testo, schermata, Carattere, software

Descrizione generata automaticamente

Figura 37 - Funzione getPossibleSolution

La funzione getPossibleSolution serve a prendere le soluzioni possibili dal array di parole. Inizialmente carica inserisce le parole all’interno dell’array, di seguito crea un array dove si inseriranno le possibili soluzioni. Grazie al parametro dove si indica la lunghezza che deve avere la parola magica e facendo un ciclo dell’array delle parole riusciremo a trovare tutte le parole con quella certa lunghezza che, se rispettano questa condizione, verranno inseriti all’interno dell’array.

#### Funzione downloadFileTXT

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Figura 38 - Funzione downloadFileTXT

La funzione donwloadFileTXT serve a far scaricare all’utente la tabella nel formato txt. Inizialmente viene creato un blob (Oggetto di dati) dove viene inserito il contenuto passato come parametro del metodo e gli viene assegnato il tipo testo, viene anche in questo caso creato un URL fittizio per nascondere il path, in seguito viene creato l’elemento gli viene assegnato questo URL e gli si mette l’attributo che al click dovrà partire il download del file con quel contenuto.

#### Funzione donwloadFileHTML

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Figura 39 - Funzione downloadFileHTML

La funzione donwloadFileHTML serve a far scaricare all’utente la tabella nel formato html. Inizialmente viene creato un blob (Oggetto di dati) dove viene inserito il contenuto passato come parametro del metodo e gli viene assegnato il tipo testo, viene anche in questo caso creato un URL fittizio per nascondere il path, in seguito viene creato l’elemento gli viene assegnato questo URL e gli si mette l’attributo che al click dovrà partire il download del file con quel contenuto.

## Classe Font

La classe font serve a gestire tutto quello che riguarda il font della tabella. Questa classe, oltre a quella di gioco, è l’unica che comunica direttamente con l’html. E inoltre è una classe che non ha bisogno altre classi per essere gestita o che serve ad altre classi per funzionare.

### Costruttore

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Figura 40 - Costruttore classe Font

Il costruttore di questa classe contiene esclusivamente due variabili una che è un array mentre l’altra una semplice stringa. L’arrayFont è appunto l’array dove sono contenuti tutti i font (12 tipi di font). Mentre la variabile indica il font che si sta usando attualmente di defaul è stato inserito il font Courier New, monospace.

### Funzioni

#### Funzione generaListaFonts

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Figura 41 - Funzione generaListaFonts

La funzione generaListaFonts serve a generare all’interno della pagina html il dropdown dove scegliere il font, questo verrà visualizzato con tutti i font inseriti con il loro nome e l’anteprima di come appaiono. Inizialmente il metodo crea il dropdown dopo facendo un ciclo dell’array di font visto nel costruttore e per ogni font viene aggiunto il nome e gli viene dato il proprio stile di font. Conclude inserendo nello spazio apposito il dropdown.

#### Funzione setFont

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Figura 42 - Funzione setFont

La funzione setFont serve a rendere effettive le modifiche del font selezionato. Il font viene modificato ogni volta che ne viene selezionato uno diverso. La prima cosa che fa la funzione è prendere il font selezionarlo e inserilo nella variabile dedicata al font. Successivamente la funzione rende la tabella di gioco, di soluzione e le liste e la parola magica con il font selezionato.

## Script

Il file script.js è colui con dentro le funzioni da far fare alla pagina web.

### Funzioni

#### Funzione generaCampo

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Figura 43 - Funzione generaCampo

La funzione generaCampo serve a istanziare la classe Gioco quindi viene fatto tutto quello che è stato spiegato in quella classe e in seguito rende visibili i pulsanti per la stampa e il download dei file txt e html.

#### Funzione onStart

Immagine che contiene testo, Carattere, schermata, numero

Descrizione generata automaticamente

Figura 44 - Funzione onStart

La funzione onStart serve a generare la lista di font all’inizio dell’esecuzione della pagina web.

#### Funzione txt

Immagine che contiene Carattere, testo, schermata, Elementi grafici

Descrizione generata automaticamente

Figura 45 - Funzione txt

La funzone txt serve a far scaricare all’utente il file in estensione txt.

#### Funzione html

Immagine che contiene Carattere, testo, schermata, Elementi grafici

Descrizione generata automaticamente

Figura 46 - Funzione html

La funzone html serve a far scaricare all’utente il file in estensione html.

#### Funzione stampa

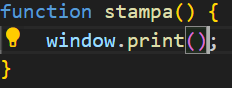


Figura 47 - Funzione stampa

La seguente funzione servirà a far stampare la tabella all’utente.

#### Funzione stampaSoluzione

Immagine che contiene testo, Carattere, schermata, Elementi grafici

Descrizione generata automaticamente

Figura 48 - Funzione stampaSoluzione

La funzione stampaSoluzione serve a far vedere la soluzione all’utente.

#### Funzione cambiaFont

Immagine che contiene testo, Carattere, schermata, Elementi grafici

Descrizione generata automaticamente

Figura 49 - Funzione cambiaFont

La funzione cambiaFont serve a far cambiare font all’utente.

# Test

## Protocollo di test

Definire in modo accurato tutti i test che devono essere realizzati per garantire l’adempimento delle richieste formulate nei requisiti. I test fungono da garanzia di qualità del prodotto. Ogni test deve essere ripetibile alle stesse condizioni.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-001  REQ-001 | **Nome:** | Controllo stampa della tabella |
| **Descrizione:** | Si dovrà verificare che la tabella viene generata correttamente. | | |
| **Prerequisiti:** | Aver avviato l’estensione *live* di Visual Studio o aver il sito web in un Web Server | | |
| **Procedura:** | 1. Aprire il sito web all’interno del proprio browser oppure andare su <https://samtinfo.ch/i21MarcoConforti/Progetto/> 2. Non modificare font, modalita, difficolta e file. 3. Schiacciare il pulsante *Genera CruciPuzzle.* | | |
| **Risultati attesi:** | La tabella è visibile all’interno della pagina | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-002  REQ-001 | **Nome:** | Controllo stampa della lista |
| **Descrizione:** | Si dovrà verificare che la lista viene generata correttamente. | | |
| **Prerequisiti:** | Aver avviato l’estensione *live* di Visual Studio o aver il sito web in un Web Server | | |
| **Procedura:** | 1. Aprire il sito web all’interno del proprio browser oppure andare su <https://samtinfo.ch/i21MarcoConforti/Progetto/> 2. Non modificare font, modalita, difficolta e file. 3. Schiacciare il pulsante *Genera CruciPuzzle.* | | |
| **Risultati attesi:** | Sotto la tabella è ben visible la lista con 15 parole totali nelle colonne (4 colonne) | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-003  REQ-001 | **Nome:** | Controllo parole |
| **Descrizione:** | Si dovrà verificare che la tabella contiene tutte le parole. | | |
| **Prerequisiti:** | Aver avviato l’estensione *live* di Visual Studio o aver il sito web in un Web Server | | |
| **Procedura:** | 1. Aprire il sito web all’interno del proprio browser oppure andare su <https://samtinfo.ch/i21MarcoConforti/Progetto/> 2. Non modificare font, modalita, difficolta e file. 3. Schiacciare il pulsante *Genera CruciPuzzle.* 4. Cercare all’interno della tabella tutte le parole (se necessario usare soluzione) | | |
| **Risultati attesi:** | Tutte le parole vengono trovate | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-004  REQ-001 | **Nome:** | Controllo Incroci |
| **Descrizione:** | Si dovrà verificare che la tabella viene generata correttamente. | | |
| **Prerequisiti:** | Aver avviato l’estensione *live* di Visual Studio o aver il sito web in un Web Server | | |
| **Procedura:** | 1. Aprire il sito web all’interno del proprio browser oppure andare su <https://samtinfo.ch/i21MarcoConforti/Progetto/> 2. Non modificare font, modalita, difficolta e file. 3. Schiacciare il pulsante *Genera CruciPuzzle.* 4. Verificare che gli incroci funzioni correttamente | | |
| **Risultati attesi:** | Gli incroci non sono della stessa direzione e non separano una parola. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-005  REQ-004  REQ-005 | **Nome:** | Controllo Funzionamento modalita Bambino |
| **Descrizione:** | Controllare che modalità Bambino funziona correttamente | | |
| **Prerequisiti:** | Aver avviato l’estensione *live* di Visual Studio o aver il sito web in un Web Server | | |
| **Procedura:** | 1. Aprire il sito web all’interno del proprio browser oppure andare su <https://samtinfo.ch/i21MarcoConforti/Progetto/> 2. Non modificare font, difficolta e file. 3. Inserire la modalità bambini (inserita già di default) 4. Schiacciare il pulsante *Genera CruciPuzzle.* | | |
| **Risultati attesi:** | La tabella stampata dovrà avere tutte le regole dei test precedenti, non dovrà avere la parola magica ma ci dovranno essere delle lettere casuali al posto di questa. Per verificare è possibile effettuare tutta la correzione a mano altrimenti è possibile aiutarsi con la soluzione. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-006  REQ-004  REQ-005 | **Nome:** | Controllo Funzionamento modalita Normale |
| **Descrizione:** | Controllare che modalità Normale funziona correttamente | | |
| **Prerequisiti:** | Aver avviato l’estensione *live* di Visual Studio o aver il sito web in un Web Server | | |
| **Procedura:** | 1. Aprire il sito web all’interno del proprio browser oppure andare su <https://samtinfo.ch/i21MarcoConforti/Progetto/> 2. Non modificare font, difficolta e file. 3. Inserire la modalità normale 4. Schiacciare il pulsante *Genera CruciPuzzle.* | | |
| **Risultati attesi:** | La tabella stampata dovrà avere tutte le regole dei test precedenti, dovrà avere la parola magica per verificare è possibile effettuare tutta la correzione a mano altrimenti è possibile aiutarsi con la soluzione. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-007  REQ-004  REQ-005 | **Nome:** | Controllo Funzionamento modalità Bambino con difficoltà facile |
| **Descrizione:** | Controllare che modalità Bambino con difficoltà facile funziona correttamente | | |
| **Prerequisiti:** | Aver avviato l’estensione *live* di Visual Studio o aver il sito web in un Web Server | | |
| **Procedura:** | 1. Aprire il sito web all’interno del proprio browser oppure andare su <https://samtinfo.ch/i21MarcoConforti/Progetto/> 2. Non modificare font e file. 3. Inserire la modalità bambino 4. Inserire la difficolta facile 5. Schiacciare il pulsante *Genera CruciPuzzle.* | | |
| **Risultati attesi:** | La tabella stampata dovrà avere tutte le regole dei test precedenti, dovrà essere una tabella 10x10, dovranno esserci all’interno della matrice 15 parole per verificare è possibile leggere il numero di parole all’interno della lista. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-008  REQ-004  REQ-005 | **Nome:** | Controllo Funzionamento modalità Bambino con difficoltà media |
| **Descrizione:** | Controllare che modalità Bambino con difficoltà media funziona correttamente | | |
| **Prerequisiti:** | Aver avviato l’estensione *live* di Visual Studio o aver il sito web in un Web Server | | |
| **Procedura:** | 1. Aprire il sito web all’interno del proprio browser oppure andare su <https://samtinfo.ch/i21MarcoConforti/Progetto/> 2. Non modificare font e file. 3. Inserire la modalità bambino 4. Inserire la difficolta media 5. Schiacciare il pulsante *Genera CruciPuzzle.* | | |
| **Risultati attesi:** | La tabella stampata dovrà avere tutte le regole dei test precedenti, dovrà essere una tabella 18x18, dovranno esserci all’interno della matrice 50 parole per verificare è possibile leggere il numero di parole all’interno della lista. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-009  REQ-004  REQ-005 | **Nome:** | Controllo Funzionamento modalita Bambino con difficolta difficile |
| **Descrizione:** | Controllare che modalità Bambino con difficoltà difficile funziona correttamente | | |
| **Prerequisiti:** | Aver avviato l’estensione *live* di Visual Studio o aver il sito web in un Web Server | | |
| **Procedura:** | 1. Aprire il sito web all’interno del proprio browser oppure andare su <https://samtinfo.ch/i21MarcoConforti/Progetto/> 2. Non modificare font e file. 3. Inserire la modalità bambino 4. Inserire la difficolta difficile 5. Schiacciare il pulsante *Genera CruciPuzzle.* | | |
| **Risultati attesi:** | La tabella stampata dovrà avere tutte le regole dei test precedenti, dovrà essere una tabella 22x22, dovranno esserci all’interno della matrice 70 parole per verificare è possibile leggere il numero di parole all’interno della lista. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-010  REQ-004  REQ-005 | **Nome:** | Controllo Funzionamento modalita Normale con difficolta facile |
| **Descrizione:** | Controllare che modalità Bambino con difficoltà facile funziona correttamente | | |
| **Prerequisiti:** | Aver avviato l’estensione *live* di Visual Studio o aver il sito web in un Web Server | | |
| **Procedura:** | 1. Aprire il sito web all’interno del proprio browser oppure andare su <https://samtinfo.ch/i21MarcoConforti/Progetto/> 2. Non modificare font e file. 3. Inserire la modalità normale 4. Inserire la difficolta difficile 5. Schiacciare il pulsante *Genera CruciPuzzle.* | | |
| **Risultati attesi:** | La tabella stampata dovrà avere tutte le regole dei test precedenti, dovrà essere una tabella 12x12, all’interno della tabella ci saranno un numero indefinito di parole. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-011  REQ-004  REQ-005 | **Nome:** | Controllo Funzionamento modalita Normale con difficolta media |
| **Descrizione:** | Controllare che modalità Bambino con difficoltà facile funziona correttamente | | |
| **Prerequisiti:** | Aver avviato l’estensione *live* di Visual Studio o aver il sito web in un Web Server | | |
| **Procedura:** | 1. Aprire il sito web all’interno del proprio browser oppure andare su <https://samtinfo.ch/i21MarcoConforti/Progetto/> 2. Non modificare font e file. 3. Inserire la modalità normale 4. Inserire la difficolta media 5. Schiacciare il pulsante *Genera CruciPuzzle.* | | |
| **Risultati attesi:** | La tabella stampata dovrà avere tutte le regole dei test precedenti, dovrà essere una tabella 16x16, all’interno della tabella ci saranno un numero indefinito di parole. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-012  REQ-004  REQ-005 | **Nome:** | Controllo Funzionamento modalita Normale con difficolta difficile |
| **Descrizione:** | Controllare che modalità Bambino con difficoltà facile funziona correttamente | | |
| **Prerequisiti:** | Aver avviato l’estensione *live* di Visual Studio o aver il sito web in un Web Server | | |
| **Procedura:** | 1. Aprire il sito web all’interno del proprio browser oppure andare su <https://samtinfo.ch/i21MarcoConforti/Progetto/> 2. Non modificare font e file. 3. Inserire la modalità normale 4. Inserire la difficolta difficile 5. Schiacciare il pulsante *Genera CruciPuzzle.* | | |
| **Risultati attesi:** | La tabella stampata dovrà avere tutte le regole dei test precedenti, dovrà essere una tabella 20x20, all’interno della tabella ci saranno un numero indefinito di parole. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-013  REQ-004  REQ-005 | **Nome:** | Controllo Font |
| **Descrizione:** | Controllare che il programma cambia Font in modo corretto | | |
| **Prerequisiti:** | Aver avviato l’estensione *live* di Visual Studio o aver il sito web in un Web Server | | |
| **Procedura:** | 1. Aprire il sito web all’interno del proprio browser oppure andare su <https://samtinfo.ch/i21MarcoConforti/Progetto/> 2. Non modificare font, file modalita e difficolta. 3. Schiacciare il pulsante *Genera CruciPuzzle.* 4. Schiacciare il pulsante *Genera Soluzione.* 5. Andare nella sezione font e scegliere un font differente da quello selezionato. | | |
| **Risultati attesi:** | La tabella, la lista, la soluzione e la lista della soluzione dovranno essere modificate con il font selezionato. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-014  REQ-004  REQ-005 | **Nome:** | Controllo Inserire un file |
| **Descrizione:** | Controllare che il file viene inserito correttamente | | |
| **Prerequisiti:** | Aver avviato l’estensione *live* di Visual Studio o aver il sito web in un Web Server | | |
| **Procedura:** | 1. Aprire il sito web all’interno del proprio browser oppure andare su <https://samtinfo.ch/i21MarcoConforti/Progetto/> 2. Non modificare font, modalità e difficoltà. 3. Premere su *Scegli file* 4. Scegliere un file con estensione txt 5. Schiacciare il pulsante *Genera CruciPuzzle.* | | |
| **Risultati attesi:** | La tabella stampata dovrà contenere esclusivamente le parole nel file selezionato. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-015  REQ-004  REQ-005 | **Nome:** | Controllo Gesione file |
| **Descrizione:** | Controllare che il file viene gestito correttamente | | |
| **Prerequisiti:** | Aver avviato l’estensione *live* di Visual Studio o aver il sito web in un Web Server | | |
| **Procedura:** | 1. Aprire il sito web all’interno del proprio browser oppure andare su <https://samtinfo.ch/i21MarcoConforti/Progetto/> 2. Non modificare font, modalità e difficoltà. 3. Premere su *Scegli file* 4. Scegliere un file con estensione che non è txt 5. Schiacciare il pulsante *Genera CruciPuzzle.* | | |
| **Risultati attesi:** | Dovrà uscire un alert dove esce un errore e la tabella dovrà essere stampata con le parole di default. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-016  REQ-002 | **Nome:** | Controllo Export stampando la tabella |
| **Descrizione:** | Si dovrà verificare che il file di viene stampato correttamente. | | |
| **Prerequisiti:** | Aver avviato l’estensione *live* di Visual Studio o aver il sito web in un Web Server | | |
| **Procedura:** | 1. Aprire il sito web all’interno del proprio browser oppure andare su <https://samtinfo.ch/i21MarcoConforti/Progetto/> 2. Non modificare font, modalita, difficolta e file. 3. Schiacciare il pulsante *Genera CruciPuzzle.* 4. Schiacciare il pulsante in basso *Stampa.* 5. Stampare tramite il print browser che viene aperto. | | |
| **Risultati attesi:** | Controllare all’interno della stampa si vede correttamente la griglia e la lista delle parole. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-017  REQ-002 | **Nome:** | Controllo Export come file txt |
| **Descrizione:** | Si dovrà verificare che il file di estensione *txt* viene scaricato correttamente e che il suo contenuto sia corretto. | | |
| **Prerequisiti:** | Aver avviato l’estensione *live* di Visual Studio o aver il sito web in un Web Server | | |
| **Procedura:** | 1. Aprire il sito web all’interno del proprio browser oppure andare su <https://samtinfo.ch/i21MarcoConforti/Progetto/> 2. Non modificare font, modalita, difficolta e file. 3. Schiacciare il pulsante *Genera CruciPuzzle.* 4. Schiacciare il pulsante in basso *ScaricaTXT.* | | |
| **Risultati attesi:** | All’interno del browser dovrebbe partire un download di un file chiamato *CruciPuzzleByMarco.txt* verificare aprendolo che ci sia la tabella e la lista di parole. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-018  REQ-002 | **Nome:** | Controllo Export come file html |
| **Descrizione:** | Si dovrà verificare che il file di estensione *html* viene scaricato correttamente e che il suo contenuto sia corretto. | | |
| **Prerequisiti:** | Aver avviato l’estensione *live* di Visual Studio o aver il sito web in un Web Server | | |
| **Procedura:** | 1. Aprire il sito web all’interno del proprio browser oppure andare su <https://samtinfo.ch/i21MarcoConforti/Progetto/> 2. Non modificare font, modalita, difficolta e file. 3. Schiacciare il pulsante *Genera CruciPuzzle.* 4. Schiacciare il pulsante in basso *ScaricaHTML.* | | |
| **Risultati attesi:** | All’interno del browser dovrebbe partire un download di un file chiamato *CruciPuzzleByMarco.html* verificare aprendolo che ci sia la tabella e la lista di parole. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-019  REQ-003 | **Nome:** | Controllo Soluzione |
| **Descrizione:** | Si dovrà verificare che la soluzione è corretta e funziona correttamente | | |
| **Prerequisiti:** | Aver avviato l’estensione *live* di Visual Studio o aver il sito web in un Web Server | | |
| **Procedura:** | 1. Aprire il sito web all’interno del proprio browser oppure andare su <https://samtinfo.ch/i21MarcoConforti/Progetto/> 2. Non modificare font, modalita, difficolta e file. 3. Schiacciare il pulsante *Genera CruciPuzzle.* 4. Schiacciare il pulsante *Genera Soluzione.* | | |
| **Risultati attesi:** | All’interno della pagina sotto la matrice di gioco dovrebbe esserci un’altra matrice colorata, verificare che ci sono tutte le parole e che i colori della matrice sono diversi e se c’è un’intersezione di più parole il colore è circa la media dei due colori. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-020  REQ-003 | **Nome:** | Controllo Parole senza accenti e senza caratteri non consentiti |
| **Descrizione:** | Si dovrà verificare che la matrice non conterrà accenti o caratteri non consentiti | | |
| **Prerequisiti:** | Aver avviato l’estensione *live* di Visual Studio o aver il sito web in un Web Server | | |
| **Procedura:** | 1. Aprire il sito web all’interno del proprio browser oppure andare su <https://samtinfo.ch/i21MarcoConforti/Progetto/> 2. Non modificare font, modalita, difficolta. 3. Selezionare un dizionario con parole con accenti 4. Schiacciare il pulsante *Genera CruciPuzzle.* 5. Schiacciare il pulsante *Genera Soluzione.* | | |
| **Risultati attesi:** | Controllare che all’interno delle due matrici non ci sono caratteri non consentite e accenti. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-021  REQ-006 | **Nome:** | Controllo Multipiattaforma |
| **Descrizione:** | Si dovrà verificare che il sito web si possa vedere su tutti i sistemi operativi richiesti. | | |
| **Prerequisiti:** | Aver avviato l’estensione *live* di Visual Studio o aver il sito web in un Web Server  Avere macchina virtuale MAC e LINUX. | | |
| **Procedura:** | 1. Aprire le macchina virtuale 2. Aprire il browser sulla macchina virtuale, aprire il sito web tramite l’estensione di Visual Studio oppure tramite web server oppure andare su <https://samtinfo.ch/i21MarcoConforti/Progetto/> 3. Controllare il funzionamento completo del sito. | | |
| **Risultati attesi:** | Il sito dovrebbe funzionare correttamente su tutti i sistemi operativi. | | |

## Risultati test

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Esito | Commenti | Data |
| TC-001 | Passato | La tabella viene generata correttamente ed è visibile all’interno della pagina. | 01.12.2023 |
| TC-002 | Passato | La lista sotto la tabella contiene tutte e 15 le parole nella 4 colonne | 01.12.2023 |
| TC-003 | Passato | La matrice dopo una ricerca prolungata contiene tutte le parole. In tutte le direzioni possibili. | 01.12.2023 |
| TC-004 | Passato | Tutti gli incroci della tabella vengono inseriti correttamente, non ci sono parole “separate” dagli incroci e non ci sono parole con la stessa direzione che hanno in comune uno o più intersezioni. | 01.12.2023 |
| TC-005 | Passato | Dopo aver inserito la modalità bambino ho notato che tutte e 15 le parole presenti erano inserite all’interno della matrice, i posti dove non ci sono le parole sono correttamente riempite con delle lettere casuali. | 01.12.2023 |
| TC-006 | Passato | Dopo aver inserito la modalità normale ho notato che tutte le parole presenti erano inserite all’interno della matrice, i posti dove non ci sono le parole sono correttamente riempite con le lettere della parola nascosta. | 01.12.2023 |
| TC-007 | Passato | Dopo aver inserito la modalità bambino e la difficoltà facile ho notato correttamente che la matrice è grande 10x10, all’interno della matrice e della lista ci sono correttamente 15 parole ed il resto sono lettere casuali. | 01.12.2023 |
| TC-008 | Passato | Dopo aver inserito la modalità bambino e la difficoltà media ho notato correttamente che la matrice è grande 18x18, all’interno della matrice e della lista ci sono correttamente 50 parole ed il resto sono lettere casuali. | 01.12.2023 |
| TC-009 | Passato | Dopo aver inserito la modalità bambino e la difficoltà difficile ho notato correttamente che la matrice è grande 22x22, all’interno della matrice e della lista ci sono correttamente 70 parole ed il resto sono lettere casuali. | 01.12.2023 |
| TC-010 | Passato | Dopo aver inserito la modalità normale e la difficoltà facile ho notato correttamente che la matrice è grande 12x12, all’interno della matrice e della lista ci sono correttamente tutte le parole possibili ed il resto sono lettere della parola nascosta. | 01.12.2023 |
| TC-011 | Passato | Dopo aver inserito la modalità normale e la difficoltà media ho notato correttamente che la matrice è grande 16x16, all’interno della matrice e della lista ci sono correttamente tutte le parole possibili ed il resto sono lettere della parola nascosta. | 01.12.2023 |
| TC-012 | Passato | Dopo aver inserito la modalità normale e la difficoltà facile ho notato correttamente che la matrice è grande 20x20, all’interno della matrice e della lista ci sono correttamente tutte le parole possibili ed il resto sono lettere della parola nascosta. | 01.12.2023 |
| TC-013 | Passato | Dopo aver inserito un font diverso è cambiato immediatamente il font sia della tabella, della soluzione e delle due liste. | 01.12.2023 |
| TC-014 | Parzialmente  Passato | Dopo aver provato ad inserire un file txt con tante parole al suo interno funziona correttamente. Mente se inserisco un testo txt con poche parole non funziona e il programma va in crash, questo sarà da modificare. | 01.12.2023 |
| TC-015 | Passato | Dopo aver provato ad inserire un file che non ha come estensione txt il sito web reagisce in modo corretto infatti dà il messaggio di errore e usa il dizionario di default. | 01.12.2023 |
| TC-016 | Passato | All’interno del print browser preview si vede correttamente la tabella e lista. La stampa si esegue correttamente. | 01.12.2023 |
| TC-017 | Passato | Installazione e visualizzazione file txt corretta. | 01.12.2023 |
| TC-018 | Passato | Installazione e visualizzazione file html corretta. | 01.12.2023 |
| TC-019 | Passato | La Soluzione è visualizzata correttamente | 01.12.2023 |
| TC-020 | Passato | All’interno delle matrici non sono inseriti né caratteri non accettati né accenti. | 01.12.2023 |
| TC-021 | Passato | In tutti e tre i sistemi non cambia, al massimo cambia il browser perciò non è necessario testare ma se si fa il test si vede che in tutti i browser funziona correttamente. | 01.12.2023 |

# Consuntivo

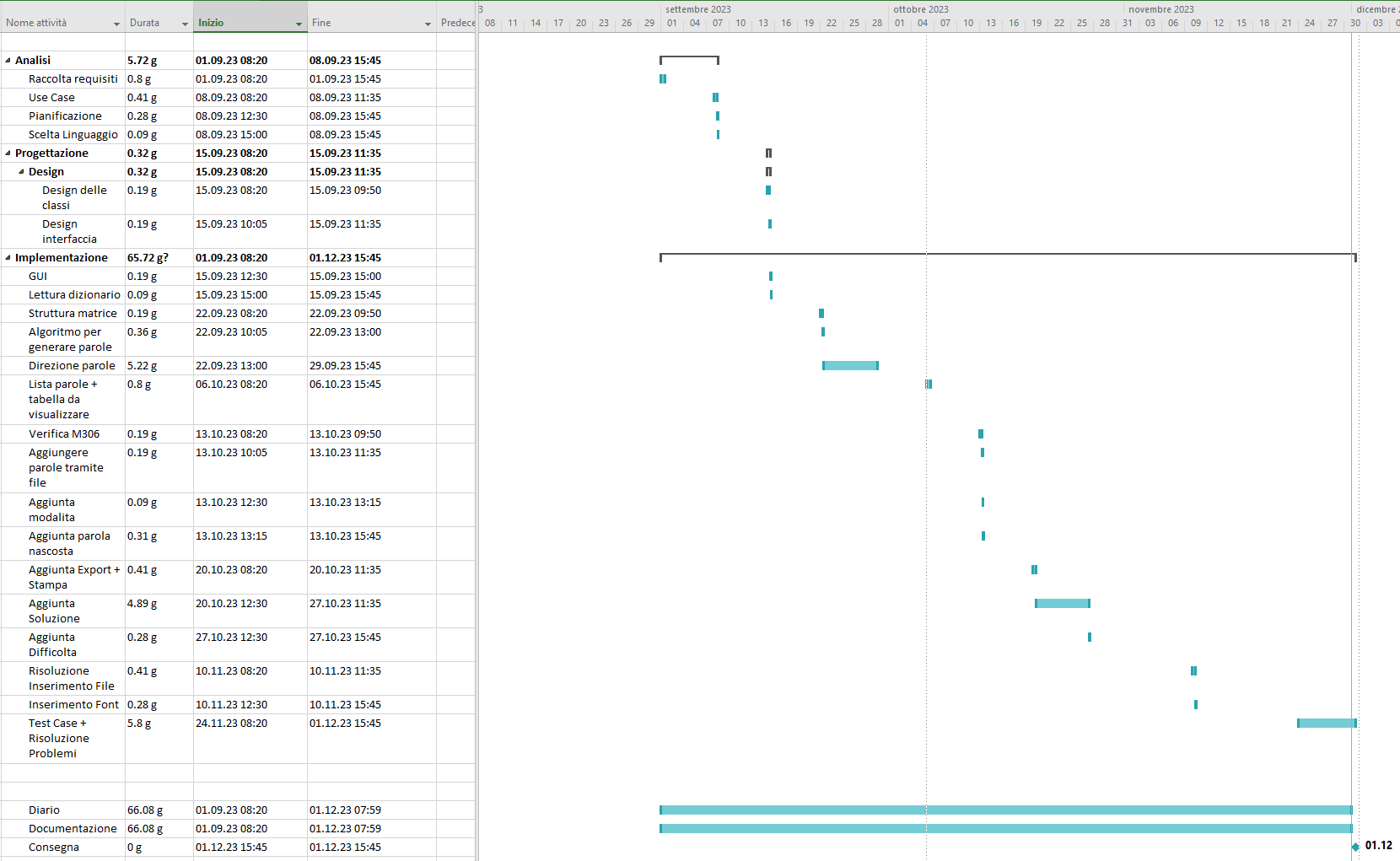


Figura 50 - Gantt Consuntivo

Vediamo che rispetto al gantt di progettazione abbiamo delle aggiunte, la verifica del modulo 305 che non era prevedibile che fosse in quel giorno, risoluzione inserimento file, questo è stato inserito per i vari problemi riscontrati nell’aggiungere un dizionario. Del resto il gantt è molto simile, cambia semplicemente l’ordine, spesso ho cambiato ordine perché ritenevo più importante una cosa rispetto ad un’altra perciò gli davo priorità oppure semplicemente capivo che per effettuare una cosa necessitavo prima della cosa precedente perciò ero costretto a cambiare l’ordine.

# Conclusioni

La soluzione non ha un impatto considerevole nel mondo e non penso lo avrà mai, poiché non aiuta a gestire delle situazioni difficili del mondo. Può comunque tornare utile a chi vuole giocare al cruci puzzle, a dei maestri che vogliono proporre un’attività didattica in classe o semplicemente per un momento di svago. In questo caso una rivista che pubblica questi giochi ha chiesto di fare un’applicazione che permette di generare questo gioco.

## Sviluppi futuri

In futuro sarà possibile aggiungere delle modalità personalizzate in più rispetto ad adesso, per esempio facendo inserire all’utente la grandezza della matrice o il numero di parole che devono contenersi all’interno della matrice. Sicuramente si dovrà modificare il funzionamento dell’inserimento del dizionario personalizzato poiché ci sono delle cose che non funzionano correttamente e che avrei voluto sistemare, ovvero se un file ha troppo poche parole le prende dal file di default, purtroppo non ho avuto il tempo di implementare una funzione che risolve questo problema quindi in futuro. Inoltre mi piacerebbe rendere più veloce la generazione della griglia per la modalità normale.

## Considerazioni personali

Grazie a questo progetto ho imparato l’importanza di avere una progettazione precisa e di rispettarla e di spenderci più tempo all’inizio per non perderne dopo per riprogettare tutto. Ho imparato a programmare orientato ad oggetti in JavaScript, poiché negli anni precedenti non abbiamo programmato orientato a oggetti in JavaScript ma con altri linguaggi di programmazione. Ho imparato a pulire meglio il codice rispetto agli anni precedenti anche se penso che devo ancora migliorare. Ho imparato cos’è un Use Case, come usare Git come fare una gantt cos’è un Activity Diagram o una Swimlane.

# Glossario

|  |  |
| --- | --- |
| **Termine** | **Descrizione** |
| AJAX | **Asynchronous JavaScript And XML**: una tecnica che permette di eseguire richieste ed ottenere dati da una pagina web in modo asincrono. |
| CSS | **Cascading Style Sheets**: linguaggio che permette di definire il layout e la grafica di una pagina web. |
| HTML | **Hyper Text Markup Language:** è il linguaggio di marcatura più usato per i documenti web |
| BLOB | **Binary Large Object:** Un oggetto binario di spesso di grandi dimensioni è una tipologia di dati che funziona solo con informazioni binarie. Si tratta spesso di oggetti di grandi dimensioni che richiedono pertanto uno spazio di archiviazione significativo. |

# Bibliografia

## Sitografia

* https://www.w3schools.com/, Learn to Code (JS, AJAX, CSS), 08-09-23
* https://stackoverflow.com/questions/14964035/how-to-export-javascript-array-info-to-csv-on-client-side, How to export JavaScript array info to csv (on client side), 20-10-2023
* https://www.html.it/pag/18966/bordi-arrotondati-con-i-css/, Bordi arrotondati con i CSS, 10-11-2023

# Allegati

Elenco degli allegati, esempio:

* Diari di lavoro
* Codici sorgente/documentazione macchine virtuali
* Istruzioni di installazione del prodotto (con credenziali di accesso) e/o di eventuali prodotti terzi
* Documentazione di prodotti di terzi
* Eventuali guide utente / Manuali di utilizzo
* Mandato e/o QdC
* Prodotto
* Gantt (Progettazione)
* Mockup
* Use Case
* Diagrammi di flusso