



Documentazione

WordCraft

Sfida la tua mente con il nostro intrigante
gioco di parole!

5AITI - IIS Cobianchi
Anastasiya Kozemko
Alice Minuzzo
Brend Margarete Protasio

2023

Indice

1. Introduzione	3
2. Features	3
3. Modalità	4
4. Gestione del Progetto	5
4.1 Work Breakdown Structure (WBS)	5
4.2 Diagramma di Gantt	6
5. Tecnologie usate	6
6. Design	7
7. Classi e Metodi	8
7.1 Partita	8
7.2 Dizionario	11
7.3 Utente	13
8. Windows App	13
8.1 Start Page	13
8.2 Menu Page	14
8.3 Normal Mode Page	15
8.4 Hard Mode Page	17
8.5 Special Mode Page	18
8.6 Results Page	19
9. Risorse	20
10. Versioni future	20

1. Introduzione

Benvenuto in WordCraft, un coinvolgente gioco di parole che metterà alla prova le tue abilità linguistiche e la tua velocità di pensiero! In questo entusiasmante passatempo, ti immergerai in un tabellone di lettere dove il tuo obiettivo sarà quello di cercare e creare il maggior numero possibile di parole entro il limite di tempo prestabilito.

Il concetto di base è semplice: ti verrà presentata una griglia di lettere disposte in modo casuale, e la tua sfida sarà quella di individuare e collegare le lettere adiacenti per formare parole valide. Puoi selezionare le lettere in orizzontale, verticale o diagonale, ma ricorda che le parole devono essere composte solo da lettere contigue all'interno della griglia. Più lunghe saranno le parole che troverai, maggiore sarà il tuo punteggio!

Tuttavia, non sarà un compito facile. Il tempo sarà il tuo avversario principale, poiché dovrai cercare di individuare il maggior numero di parole nel minor tempo possibile. Le parole saranno di varie lunghezze e difficoltà, da quelle semplici a quelle più complesse, per mettere alla prova le tue capacità linguistiche a tutti i livelli.

WordCraft ti offre anche diverse modalità di gioco, così da sfidare te stesso in vari limiti di tempo. Inoltre, con la modalità speciale, giocabile solo da persone 18+, potrai sorprenderti!

Preparati a immergerti in un universo di parole e divertimento senza limiti! Affila la tua mente, allena la tua agilità mentale e dimostra di essere il maestro dei vocaboli. Migliora il tuo punteggio personale e scatena la tua creatività linguistica nel coinvolgente gioco di parole "WordCraft"!

2. Features

Modalità di gioco multiple

WordCraft offre tre emozionanti modalità di gioco: Normal, Hard e Special. Ogni modalità presenta una sfida unica e offre un'esperienza diversa ai giocatori.

Griglie di parole stimolanti

I giocatori si trovano di fronte a una griglia di lettere 10x10, da cui devono cercare il maggior numero possibile di parole entro il limite di tempo stabilito. Le griglie di parole sono state create con cura per offrire un equilibrio tra difficoltà e divertimento, garantendo un'esperienza appagante nella risoluzione delle parole.

Gioco multilingue

WordCraft supporta sia l'inglese che l'italiano, consentendo ai giocatori di mettere alla prova le loro abilità lessicali in due lingue popolari. Che tu voglia ampliare il tuo vocabolario inglese o mostrare la tua conoscenza delle parole italiane.

Coinvolgente design del suono

Immergiti nel mondo di WordCraft con il suo design del suono. Dalla musica di sottofondo ai gradevoli effetti sonori, l'audio del gioco migliora l'esperienza complessiva, rendendola rilassante ma anche divertente da giocare.

3. Modalità

Nel progetto sono presenti tre modalità: NormalMode, HardMode, SpecialMode.

Normal Mode

Il giocatore ha a disposizione 3 minuti per trovare più parole possibili all'interno del tabellone proposto e inserirle nell'input dedicato.

Hard Mode

Al giocatore viene concesso solo un minuto per individuare il maggior numero possibile di parole all'interno del tabellone proposto e inserirle nell'apposito campo di input.

Special Mode

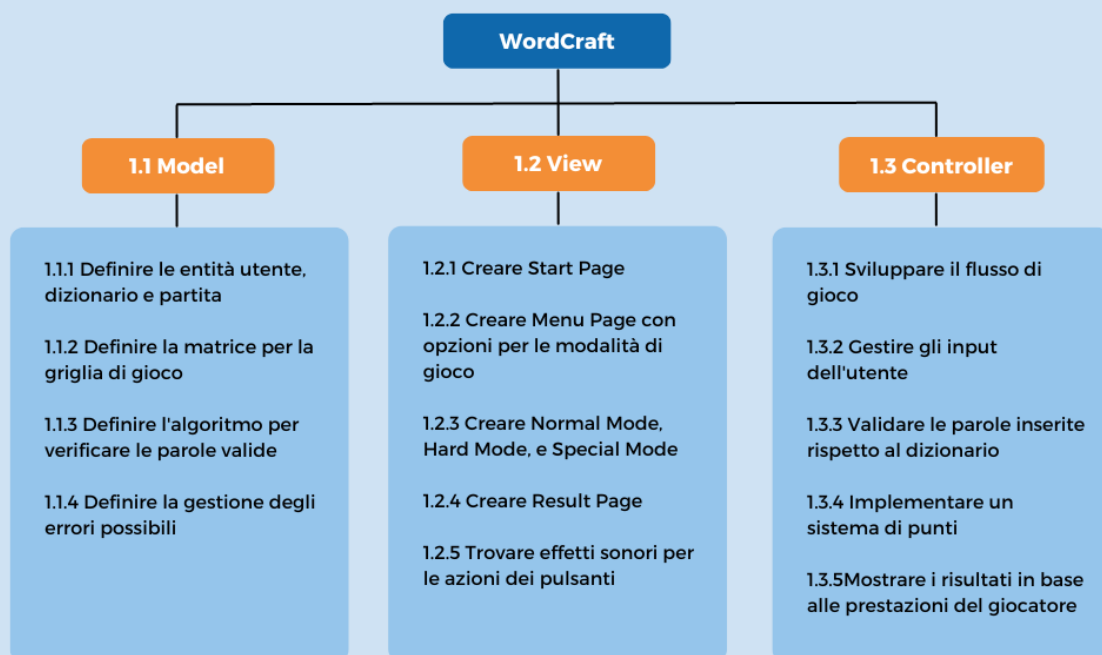
Il giocatore ha a disposizione 3 minuti per trovare più parole possibili che provengono da un dizionario speciale, ed inserirle nell'apposito campo di input.

4. Gestione Del Progetto

Nella gestione del progetto di WordCraft, abbiamo adottato strategie e strumenti per garantire il coordinamento efficace delle attività. Il nostro focus si concentra sulla struttura di scomposizione del lavoro (WBS) e sul diagramma di Gantt. Questi strumenti ci offrono una visione organizzata delle attività e una programmazione temporale, consentendo un migliore controllo e monitoraggio del progetto.

4.1 Work Breakdown Structure (WBS)

Basato su Model View Controller (MVC)

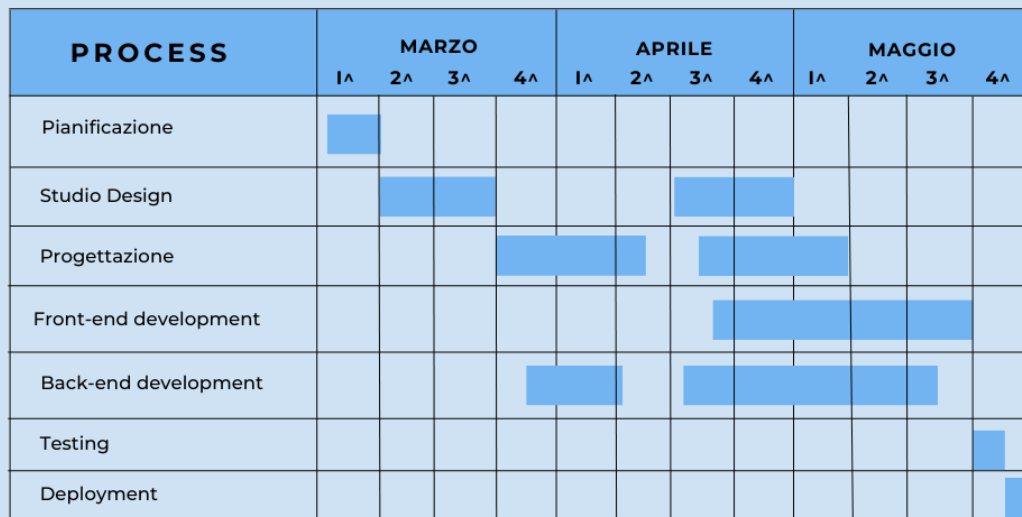


Questo WBS fornisce una suddivisione più chiara dei compiti e delle responsabilità per ciascun componente di WordCraft, garantendo una migliore comprensione del processo di sviluppo.

4.2 Diagramma di Gantt

WordCraft

Gantt Chart



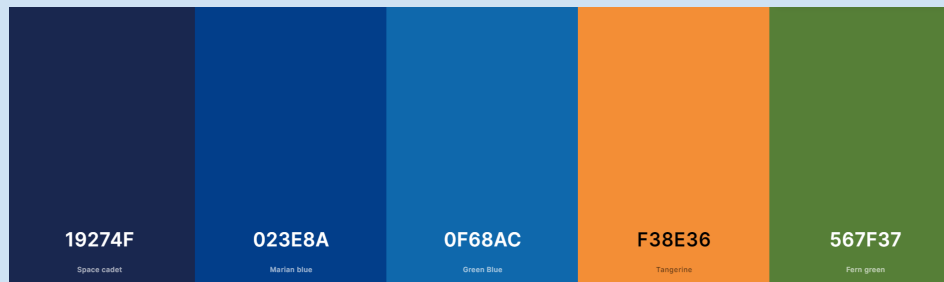
5. Tecnologie Utilizzate

Per la creazione di questo progetto sono state utilizzate le seguenti tecnologie:

- IntelliJ - IDE
- Linguaggio JAVA
- Libreria Java Swing - per la grafica
- Procreate - per la creazione dell'immagine dell'icona
- Notion - per la divisione dei task
- Canva - per la realizzazione del diagramma Gantt
- Git - per tenere traccia delle modifiche apportate ai file nel tempo e coordinare il lavoro con altri membri del team
- GitHub - per la gestione del codice e della collaborazione nel progetto

6. Design

Nella sezione del design, abbiamo curato attentamente ogni dettaglio il gioco. Abbiamo creato una palette di colori appositamente selezionata per trasmettere una sensazione di divertimento, relax e intelligenza, mantenendo al contempo un tocco moderno e un po' eccentrico.



Per quanto riguarda i caratteri tipografici, abbiamo scelto con cura il font "Rubik" per il suo aspetto accattivante e leggibile.

Rubik Bold
abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ
WordCraft

L'interfaccia utente è stata progettata utilizzando Java Swing Designer, consentendo una realizzazione efficiente e precisa del design. Durante il processo di progettazione, abbiamo sempre tenuto a mente l'esperienza dell'utente e l'usabilità (UI/UX), assicurandoci che i giocatori possano facilmente orientarsi e comprendere le funzionalità offerte dal gioco. Il nostro obiettivo principale è stato quello di creare un'interfaccia intuitiva e user-friendly.

7. Classi e Metodi

7.1 Partita

La classe Partita rappresenta un'istanza del gioco.

Attributi

puntiTotali

Viene utilizzato per memorizzare il punteggio totale del giocatore per quella partita. Questo valore viene aggiornato durante il gioco.

utente

Viene utilizzato per tenere traccia dei dati del giocatore che ha inserito il proprio nome dalla pagina iniziale.

parolaMigliore

Viene utilizzato per conservare la parola più lunga trovata dal giocatore durante la partita. Questo valore viene aggiornato quando una parola più lunga viene scoperta.

parolaNum

Viene utilizzato per memorizzare il numero di parole trovate dal giocatore durante la partita. Questo valore viene incrementato ogni volta che una nuova parola viene scoperta.

matrice

Viene utilizzato per rappresentare una griglia 10x10 di lettere casuali. Questa matrice viene creata e visualizzata durante il gioco, fornendo al giocatore il campo di gioco in cui cercare le parole.

paroleArray

Un vettore utilizzato per memorizzare tutte le parole inserite dal giocatore in una determinata istanza di gioco. Questo vettore si popola man mano che il giocatore inserisce nuove parole durante il gioco.

Anche getter e setter implementati per ogni attributo.

Metodi

controllaParolaMigliore()

Controlla se esiste una parola migliore all'interno di un array di parole (paroleArray). Restituisce un valore booleano che indica se la parola migliore è unica rispetto alle altre parole presenti nell'array.

puntiParola(String parola)

Calcola i punti per una parola specifica, in base alla sua lunghezza, e aggiunge la parola all'array delle parole inserite.

contaParola()

Incrementa il contatore delle parole inserite dal giocatore.

doppione(String parolaInserita)

Verifica se una parola inserita è un doppione, ovvero se esiste già nell'array paroleArray e restituisce un valore booleano.

creaMatrice(String[] words)

Genera una matrice di gioco 10x10 con lettere casuali e inserisce 10 parole italiane all'interno in posizioni casuali per assicurarsi che il giocatore trovi una parola esistente piuttosto che inserire semplicemente lettere casuali nella matrice. Rispettando anche le regole specifiche per l'inserimento orizzontale, verticale e diagonale.

```
Java ▾ Copy Caption ***

char[][] matrix = new char[10][10];
Random random = new Random();
String[] wordsInject = new String[10];

// Populate wordsInject array with 10 words of length 10 or less
for (int i = 0; i < wordsInject.length; i++) {
    String randomWord;
    do {
        int randomIndex = random.nextInt(words.length);
        randomWord = words[randomIndex];
    } while (randomWord.length() > 10);
    wordsInject[i] = randomWord.toUpperCase();
    System.out.println("parola added - " + wordsInject[i]);
}

// Insert Italian words horizontally
for (int i = 0; i < wordsInject.length; i++) {
    String word = wordsInject[i];
    int wordLength = word.length();
    int row = random.nextInt(10);
    int col = random.nextInt(10 - wordLength + 1);

    for (int j = 0; j < wordLength; j++) {
        matrix[row][col + j] = word.charAt(j);
    }
}
```

Andando avanti abbiamo messo anche l'algoritmo per mettere parole verticalmente e diagonalmente.

stampaMatrice()

Stampa a schermo la matrice, usato nella fase di testing.

trovaParolaMatrice(String word)

Cerca una parola all'interno della matrice di gioco, verificando se è presente orizzontalmente, verticalmente o diagonalmente.

```
Java ▾ Copy Caption ...

char[][] matrix = this.matrice;
int rows = matrix.length;
int cols = matrix[0].length;
int wordLength = word.length();

// Check horizontally (left to right)
for (int i = 0; i < rows; i++) {
    for (int j = 0; j <= cols - wordLength; j++) {
        boolean found = true;
        for (int k = 0; k < wordLength; k++) {
            if (matrix[i][j + k] != word.charAt(k)) {
                found = false;
                break;
            }
        }
        if (found) {
            return true;
        }
    }
}
```

Andando avanti abbiamo messo anche l'algoritmo per trovare le parole verticalmente e diagonalmente.

checkWord(char[][] matrix, String word, int row, int col, int rowStep, int colStep)

Un metodo ausiliario utilizzato da trovaParolaMatrice() per verificare se una parola è presente in una specifica direzione (riga, colonna, diagonale) all'interno della matrice di gioco.

7.2 Dizionario

Rappresenta un dizionario contenente i metodi per cercare e manipolare parole in diversi file di testo. Utilizza operazioni di I/O sui file e tecniche di manipolazione delle stringhe per svolgere i compiti.

Attributi

words[]

Array di stringhe con una lunghezza di 1068, contiene le parole comuni prelevate dal file “ParoleComuni.txt” utilizzate per inizializzare la matrice iniziale del gioco.

specialwords[]

Array di stringhe con una lunghezza di 419, contiene le parole “speciali” prelevate dal file “fileSpeciale.txt” utilizzate per inizializzare la matrice iniziale del gioco nella modalità “special”.

La classe fornisce metodi getter e setter per gli array words[] e specialwords[] ed un costruttore di default, che non esegue alcuna inizializzazione specifica.

Metodi

trovaParoladiz(String p)

Cerca una parola in un file chiamato "dictionary.txt" situato nella directory "src/file". Legge il file riga per riga e verifica se la parola fornita corrisponde a una riga. Se trovata, stampa un messaggio di controllo e restituisce true; altrimenti, restituisce false. Serve per verificare se le parole inserite da tastiera esistono nel dizionario.

```
Java ▾ Copy Caption ...
public boolean trovaParoladiz(String p){
    boolean b = false;
    try {
        String filePath = "src/file/dictionary.txt";
        BufferedReader reader = new BufferedReader(new FileReader(filePath));
        String line = reader.readLine();
        while (line != null) {
            if(p.equals(line)){
                System.out.println(line+" - trovato :D");
                b = true;
                break;
            }
            line = reader.readLine();
        }
        //System.out.println("FINE");
        reader.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
    return b;
}
```

trovaParolaComuni(String p)

Cerca una parola in un file chiamato "ParoleComuni.txt" situato nella directory "src/file". Segue un approccio simile al metodo precedente.

trovaParolaSpecial(String p)

Cerca una parola in un file chiamato "fileSpeciale.txt" situato nella directory "src/file". Segue un approccio simile ai metodi precedenti.

stampaWords()

Funzione che stampa tutte le parole dell'array "words".

La classe include due metodi per inizializzare l'array "words" dai file:

inizializzaWords()

Legge le parole da un file chiamato "ParoleComuni.txt" situato nella directory "src/file". Suddivide ogni riga in parole singole e le memorizza nell'array "words". Metodo usato per le modalità classiche.

```
public void inicializzaWords() {
    String filePath = "src/file/ParoleComuni.txt";
    File file = new File(filePath);
    try {
        Scanner scanner = new Scanner(file);
        int i = 0;
        while (scanner.hasNextLine()) {
            String line = scanner.nextLine();
            String[] lineWords = line.split(" "); // Split della riga
            for (String word : lineWords) {
                this.words[i] = word; // metti parola in array words
                i++;
            }
        }
        scanner.close();
        // Now you can use the 'words' array to process the words as ne
        // System.out.println("parole totali: " + i);
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    }
}
```

inizializzaWordsSpecial()

Legge le parole da un file chiamato "fileSpeciale.txt" situato nella directory "src/file". Segue un approccio simile al metodo precedente. Metodo usato per la modalità speciale.

7.3 Utente

Rappresenta un giocatore di WordCraft.

Attributi

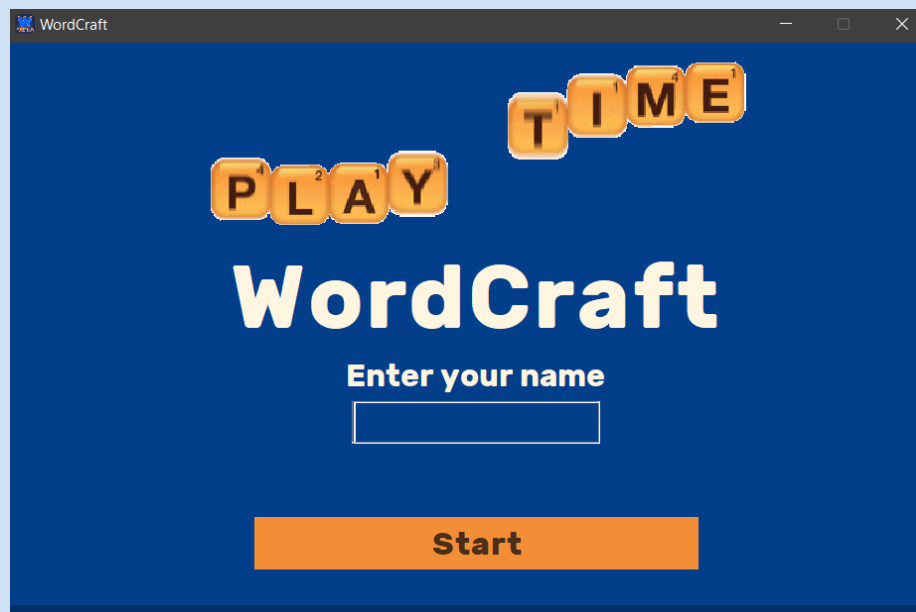
Nickname

Memorizza il nome utente del giocatore. Il nome utente viene passato al costruttore della classe durante la creazione.

Anche getter e setter implementati per ogni attributo.

8. Windows App

8.1 Start Page



La classe `StartPage` fornisce l'interfaccia utente iniziale per il gioco WordCraft, consentendo all'utente di inserire il proprio nome e avviare il gioco tramite il pulsante di avvio.

All'interno del costruttore `StartPage()`, vengono impostate diverse proprietà della finestra dell'interfaccia utente, come l'icona, il titolo, le dimensioni, la posizione e la visibilità. Viene inoltre gestito l'evento del clic sul pulsante di avvio `startButton` e del tasto Invio nel campo di testo `nomeBox`. Quando

viene cliccato il pulsante di avvio, viene riprodotto un suono, viene creato un oggetto **Utente** con il nome inserito nel campo di testo e viene creata e visualizzata una nuova pagina **MenuPage**.

Il metodo **buttonSound()** gestisce la riproduzione di un suono quando viene premuto il pulsante di avvio.

8.1 Menu Page



La classe **MenuPage** fornisce l'interfaccia utente del menu del gioco WordCraft, consentendo all'utente di selezionare diverse modalità di gioco o tornare alla pagina di avvio.

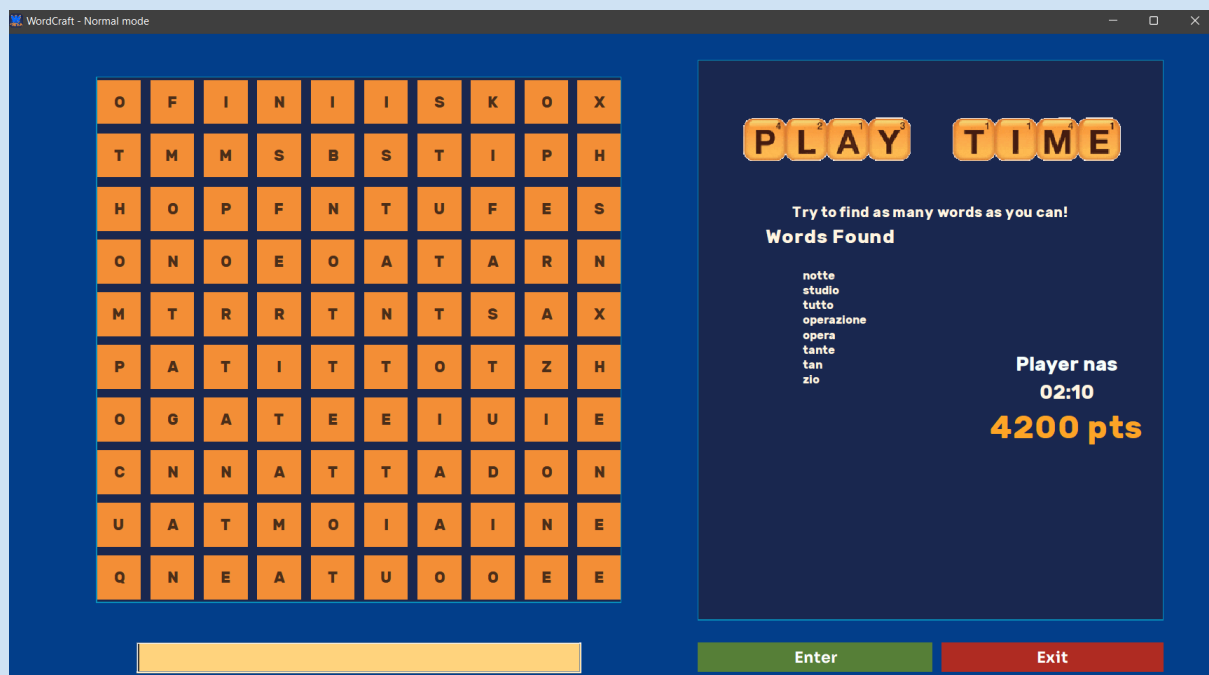
All'interno del costruttore **MenuPage(Utente giocatore)**, vengono impostate diverse proprietà della finestra dell'interfaccia utente, come l'icona, il titolo, le dimensioni, la posizione e la gestione dell'evento di chiusura della finestra. Inoltre, vengono gestiti gli eventi dei pulsanti per le diverse modalità di gioco e il pulsante di ritorno.

Quando viene premuto uno dei pulsanti delle modalità di gioco, viene riprodotto un suono e viene creata e visualizzata una nuova pagina di gioco

EasyMode, HardMode o **SpecialMode**. Quando viene premuto il pulsante di **ritorno**, viene riprodotto un suono e viene creata e visualizzata una nuova pagina di avvio StartPage.

Il metodo **buttonSound()** gestisce la riproduzione di un suono quando viene premuto un pulsante nel menu.

8.1 Normal Mode Page



La classe EasyMode rappresenta l'interfaccia grafica della modalità Normal di WordCraft. Consente al giocatore 3 minuti per trovare quante più parole possibile.

Contiene vari metodi che gestiscono gli eventi generati dagli elementi grafici, come premere un pulsante o inserire una parola nel campo di testo. Quando l'utente preme il pulsante **Enter** dopo aver inserito una parola nel campo di testo, controlla se la parola inserita dall'utente è valida. La cerca nella matrice di gioco e nel **dizionario** delle parole **valide**, e aggiorna il punteggio del giocatore e la lista delle parole trovate di conseguenza.

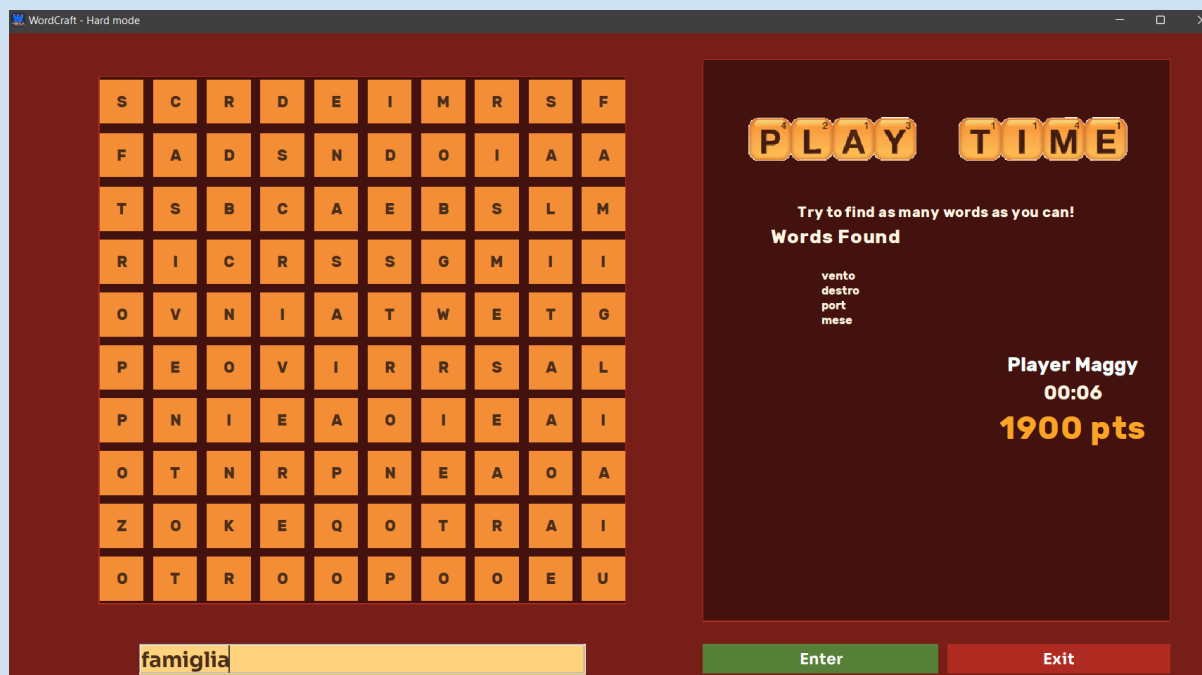
La classe utilizza anche un **timer** della libreria Swing per tenere traccia del tempo rimanente durante il gioco. Il timer viene avviato quando la finestra

di gioco viene visualizzata e viene aggiornata ogni secondo. Quando il tempo scade, il timer viene interrotto e viene visualizzata la pagina dei risultati.

Inoltre, la classe contiene alcuni metodi di supporto che gestiscono la riproduzione dei suoni quando l'utente inserisce una parola **corretta** o **errata**.

```
Java ▾  
@Override  
public void actionPerformed(ActionEvent e) {  
    String parolaInserita = parolaInput.getText();  
  
    //controlla se parola esiste nella matrice  
    String parolaInseritaUpper = parolaInserita.toUpperCase();  
    boolean trovata = p.trovaParolaMatrice(parolaInseritaUpper);  
    if (trovata) {  
        System.out.println("Parola " + parolaInserita + " è nella matrice!");  
  
        //controlla se parola esiste nel dizionario  
        if (diz.trovaParolaDiz(parolaInserita)) {  
            if (p.doppione(parolaInserita) == false) {  
                playCorrectWord();  
                System.out.println("Parola trovata nel dizionario!");  
                p.aggiungiPunti(parolaInserita);  
                System.out.println("Punti della parola: " + p.puntiParola(parolaInserita));  
                //aggiorna punti totali  
                labelPunti.setText(p.getPuntiTotali() + " pts");  
                p.contaParola(); //contatore  
                //mette nella lista di parole  
                String listaParole = "<html>" + labelParoleTrovate.getText();  
                labelParoleTrovate.setText( listaParole + "<br>" + parolaInserita);  
            } else {  
                playWrongWord();  
                System.out.println("Hai già inserito questa parola >:(");  
            }  
        } else {  
            playWrongWord();  
            System.out.println("Questa parola non esiste nel dizionario >:(");  
        }  
    } else {  
        playWrongWord();  
        System.out.println("Parola " + parolaInserita + " non esiste nella matrice :(");  
    }  
    parolaInput.setText(""); //cancella input quando preme invio  
}
```


8.1 Hard Mode Page



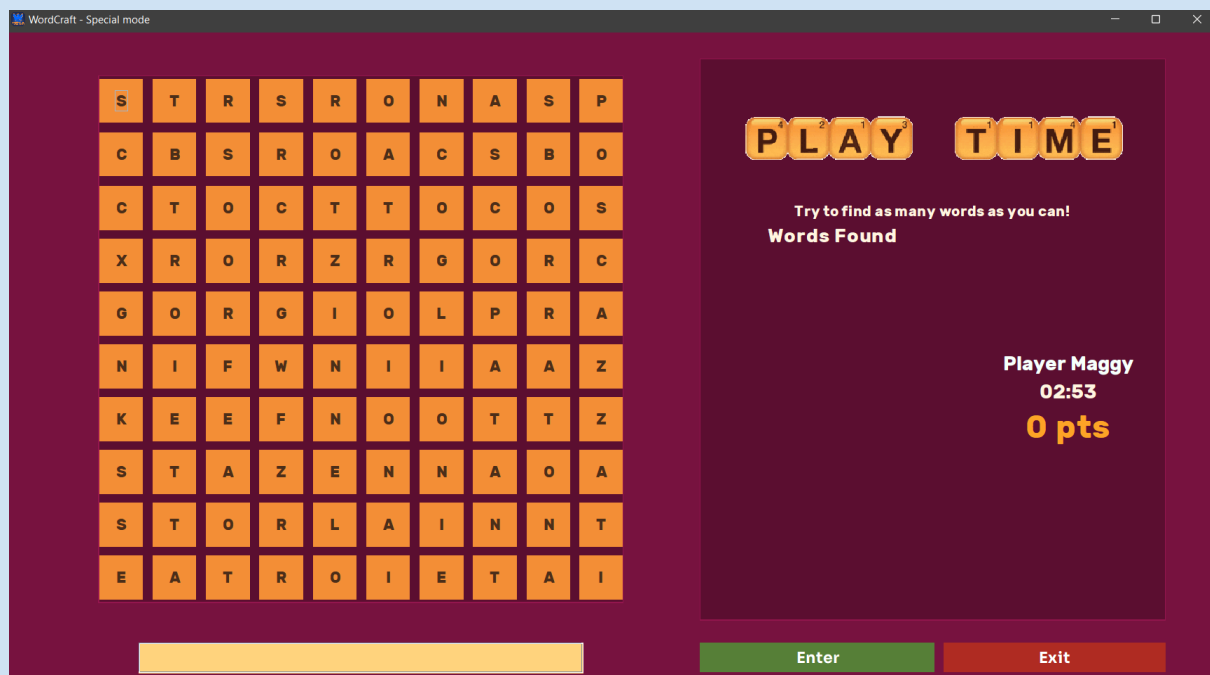
La classe `HardMode` rappresenta l'interfaccia grafica della modalità Hard di WordCraft. Consente al giocatore solo 1 minuto per trovare quante più parole possibile.

Contiene vari metodi che gestiscono gli eventi generati dagli elementi grafici, come premere un pulsante o inserire una parola nel campo di testo. Quando l'utente preme il pulsante **Enter** dopo aver inserito una parola nel campo di testo, controlla se la parola inserita dall'utente è valida. La cerca nella matrice di gioco e nel **dizionario** delle parole **valide**, e aggiorna il punteggio del giocatore e la lista delle parole trovate di conseguenza.

La classe utilizza anche un **timer** della libreria Swing per tenere traccia del tempo rimanente durante il gioco. Il timer viene avviato quando la finestra di gioco viene visualizzata e viene aggiornata ogni secondo. Quando il tempo scade, il timer viene interrotto e viene visualizzata la pagina dei risultati.

Inoltre, la classe contiene alcuni metodi di supporto che gestiscono la riproduzione dei suoni quando l'utente inserisce una parola **corretta** o **errata**.

8.1 Special Mode Page



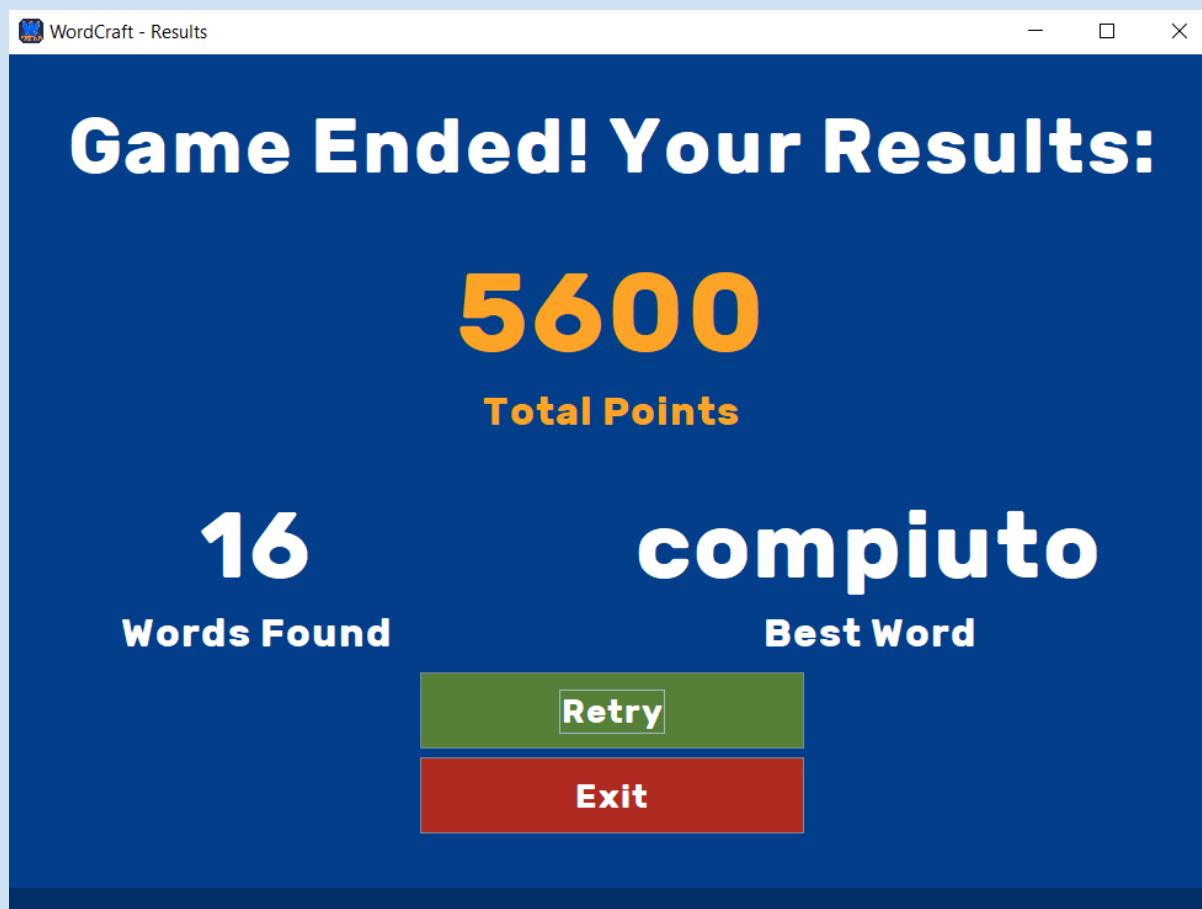
La classe `SpecialMode` rappresenta l'interfaccia grafica della modalità Special di WordCraft giocabile solo da persone 18+. Consente al giocatore 3 minuti per trovare quante più parole possibile.

Contiene vari metodi che gestiscono gli eventi generati dagli elementi grafici, come premere un pulsante o inserire una parola nel campo di testo. Quando l'utente preme il pulsante **Enter** dopo aver inserito una parola nel campo di testo, controlla se la parola inserita dall'utente è valida. La cerca nella matrice di gioco e nel **dizionario normale** e **dizionario speciale** delle parole **valide**, e aggiorna il punteggio del giocatore e la lista delle parole trovate di conseguenza.

La classe utilizza anche un **timer** della libreria Swing per tenere traccia del tempo rimanente durante il gioco. Il timer viene avviato quando la finestra di gioco viene visualizzata e viene aggiornata ogni secondo. Quando il tempo scade, il timer viene interrotto e viene visualizzata la pagina dei risultati.

Inoltre, la classe contiene alcuni metodi di supporto che gestiscono la riproduzione dei suoni quando l'utente inserisce una parola **corretta** o **errata**.

8.1 Results Page



La classe `ResultPage` fornisce l'interfaccia utente dei risultati della partita appena terminata del gioco WordCraft, consentendo all'utente di visualizzare il numero dei punti totalizzati, il numero delle parole trovate e la parola migliore che ha trovato; inoltre può selezionare **Retry** se vuole fare una nuova partita e successivamente selezionare la modalità di gioco, altrimenti **Exit** per tornare alla pagina iniziale del gioco.

Quando finisce la partita, si apre questa pagina accompagnata da un simpatico suono di congratulazioni, gestito dal metodo `playSound()`. All'interno del costruttore `ResultPage(Partita p, Utente g)`, vengono impostate diverse proprietà della finestra dell'interfaccia utente, come l'icona, il titolo, le dimensioni, la posizione e la visibilità. Viene inoltre gestito l'evento del clic sul pulsante `retryButton` per giocare nuovamente e del tasto `exitButton` per tornare alla pagina `MenuPage`. Quando vengono cliccati i pulsanti viene riprodotto un suono, che viene gestito dal metodo `buttonSound()`.

9. Risorse

Le risorse utilizzate per la realizzazione del progetto sono

- ChatGPT - per consigli su come migliorare il codice e implementare il best practice
- Youtube - per l'implementazione di pezzi di codice, come il timer e trovare effetti sonori
- StackOverflow - per consigli su ottimizzazione del codice
- Pixabay - per l'audio principale del gioco

10. Sviluppi Futuri

Abbiamo considerato l'implementazione dei seguenti elementi per le prossime evoluzioni:

- Se si arriva ad inserire un certo numero di parole entro un certo tempo si aggiunge del tempo
- Collegamento ad un database
- Avendo un collegamento ad un database una sezione per le statistiche di gioco.
- La possibilità di selezionare la grandezza del tabellone