Dattilo App

Sommario

[1 Introduzione 3](#_Toc122697800)

[1.1 Informazioni sul progetto 3](#_Toc122697801)

[1.2 Abstract 3](#_Toc122697802)

[1.3 Scopo 3](#_Toc122697803)

[2 Analisi 4](#_Toc122697804)

[2.1 Analisi del dominio 4](#_Toc122697805)

[2.2 Analisi e specifica dei requisiti 4](#_Toc122697806)

[2.3 Use case 7](#_Toc122697807)

[2.4 Pianificazione 8](#_Toc122697808)

[2.5 Analisi dei mezzi 9](#_Toc122697809)

[2.5.1 Software 9](#_Toc122697810)

[2.5.2 Hardware 9](#_Toc122697811)

[3 Progettazione 10](#_Toc122697812)

[3.1 Design delle interfacce 10](#_Toc122697813)

[3.2 Design procedurale 11](#_Toc122697814)

[3.2.1 Classi 11](#_Toc122697815)

[4 Implementazione 14](#_Toc122697816)

[4.1 Membri e Proprietà della Classe Apprendimento 14](#_Toc122697817)

[4.2 variabili aiuto Classe Apprendimento 15](#_Toc122697818)

[4.3 costruttore della classe Apprendimento 16](#_Toc122697819)

[4.4 Metodo GeneraLivello Classe Apprendimento 16](#_Toc122697820)

[4.5 Metodo ConfrontaChar Classe Apprendimento 17](#_Toc122697821)

[4.6 metodo AggiornaCronometro classe Apprendimento 17](#_Toc122697822)

[4.7 Metodo ConvertToSM Classe Apprendimento 18](#_Toc122697823)

[4.8 classe MainViewModel 18](#_Toc122697824)

[5 Consuntivo 19](#_Toc122697825)

[6 Conclusioni 20](#_Toc122697826)

[6.1 Sviluppi futuri 20](#_Toc122697827)

[6.2 Considerazioni personali 20](#_Toc122697828)

[7 Allegati 20](#_Toc122697829)

# Introduzione

## Informazioni sul progetto

Autore: Alex Ierardi

Scuola: Arti e Mestieri Trevano

Classe: I3BB

Anno scolastico: 2022/23

Sezione: Informatica

Materia: Progetti individuali

Docente responsabile: Geo Petrini

Data d’inizio: 8.09.2022

Data di consegna: 23.12.2022

## Abstract

Scrivere con 10 dita è una cosa che prima o poi tutti dobbiamo saper fare, ad oggi i computer sono parte integrante della nostra vita ed è molto importante che una persona impari, soprattutto le persone che lavorano tutti i giorni con i computer.

Il problema di non saper scrivere in modo adeguato porta la persona a dover guardare continuamente la posizione dei tasti sulla tastiera e di conseguenza perdere molto tempo, questo influisce molto nel nostro lavoro siccome non si può perdere tempo continuando a guardare dove si stanno mettendo le dita e magari sbagliare anche a scrivere.

Questo progetto vuole aiutare le persone ad imparare a scrivere con 10 dita attraverso dei livelli incrementali che hanno lo scopo di far capire all’utente come posizionare le dita sulla tastiera per poi, una volta imparato, non doverla più guardare.

## Scopo

Lo scopo di questo progetto e far imparare all’utente a scrivere con 10 dita, ottimizzando il tempo di scrittura e facilitando le persone a commettere meno errori.

# Analisi

## Analisi del dominio

Questo applicativo verrà principalmente usato da utenti inesperti che non sanno ancora scrivere con dieci dita. Di applicativi che insegnano dattilografia ce ne sono molti, questo in particolare oltre ad avere la solita modalità di apprendimento classica con i livelli in qui più vai avanti e più sono i caratteri che l’utente dovrà premere ha anche la possibilità di inserire dei testi che il programma andrà a dettare.

## Analisi e specifica dei requisiti

|  |  |
| --- | --- |
| **ID: REQ-01** | |
| **Nome** | Scelta del metodo di insegnamento |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | scegliere tra due metodi di insegnamento |
| **Sotto requisiti** | |
| **001** | Copiare il testo visto in base all’immagine con la posizione delle dita |
| **002** | Ascoltare il dettato e scrivere senza vedere il testo |
| **ID: REQ-01** | |

|  |  |
| --- | --- |
| **ID: REQ-02** | |
| **Nome** | blocco dei tasti non utilizzabili da parte dell’utente |
| **Priorità** | 3 |
| **Versione** | 1.0 |
| **Note** | Blocco del tasto per cancellare e di invio |
| **ID: REQ-02** | |

|  |  |
| --- | --- |
| **ID: REQ-03** | |
| **Nome** | La dettatura ha una velocità regolabile |
| **Priorità** | 3 |
| **Versione** | 1.0 |
| **Note** | L’utente potrà scegliere la velocità della dettatura in base alla sua esperienza |
| **ID: REQ-03** | |

|  |  |
| --- | --- |
| **ID: REQ-04** | |
| **Nome** | Raccogliere statistiche dell’utente |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Sotto requisiti** | |
| **001** | Quanti tasti al minuto sono stati premuti al minuto |
| **002** | Conta dei tasti sbagliati |
| **003** | Percentuale dei tasti sbagliati |
| **ID: REQ-04** | |

|  |  |
| --- | --- |
| **ID: REQ-05** | |
| **Nome** | L’utente può aggiungere testi |
| **Priorità** | 3 |
| **Versione** | 1.0 |
| **Note** | I testi aggiunti potranno essere dettati |
| **ID: REQ-05** | |

|  |  |
| --- | --- |
| **ID: REQ-06** | |
| **Nome** | L’utente scrive il testo all’interno del programma |
| **Priorità** | 3 |
| **Versione** | 1.0 |
| **Note** | Il testo che l’utente scrive sarà all’interno del programma e non in un altro file |
| **ID: REQ-06** | |

|  |  |
| --- | --- |
| **ID: REQ-07** | |
| **Nome** | Font ridimensionabile in base all’esigenza dell’utente |
| **Priorità** | 3 |
| **Versione** | 1.0 |
| **ID: REQ-07** | |

|  |  |
| --- | --- |
| **ID: REQ-08** | |
| **Nome** | Possibilità di mettere in pausa l’esercizio |
| **Priorità** | 3 |
| **Versione** | 1.0 |
| **Note** | L’utente può mettere in pausa l’esercizio durante lo svolgimento |
| **Sotto requisiti** | |
| **001** | Oscura testo durante la modalità di copiature del testo |
| **002** | Ferma dettato |
| **ID: REQ-08** | |

|  |  |
| --- | --- |
| **ID: REQ-08** | |
| **Nome** | Evidenziamento delle lettere sbagliate |
| **Priorità** | 3 |
| **Versione** | 1.0 |
| **Note** | Quando l’utente sbaglia a premere un tasto la lettera diventa di un altro colore |
| **ID: REQ-08** | |

## Use case

## Pianificazione

Figura Gannt Preventivo

## Analisi dei mezzi

### Software

Visual Studio 2022

Microsoft® Project

Microsoft® Word

### Hardware

Per svolgere il progetto è stato usato un unico PC, ovvero il seguente:

HP EliteDesk 800 G5 TWR

* Intel® Core™ i7-9700 CPU 3 GHz
* 32 GB DDR4
* NVIDIA® GeForce RTX 2060

# Progettazione

## Design delle interfacce

L’utente una volta aperto l’applicativo dovrà scegliere quelle delle due modalità fare attraverso il click dei pulsanti cambierà la view.

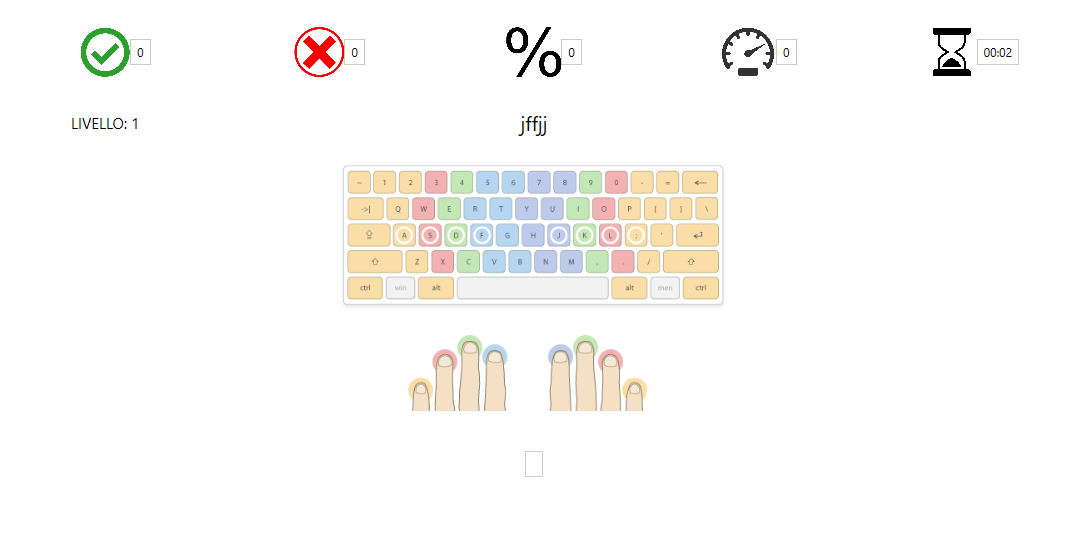
Questa è la view della modalità di apprendimento classica dove l’utente dovrà ricopiare i caratteri che vede stampati all’interno del TextBox situato sotto all’immagine di come posizionare le dite sulla tastiera. Sopra la stampa dei caratteri da copiare sono situati i conteggi dei caratteri giusti che l’utente andrà a scrivere, dei caratteri sbagliati, la percentuale di caratteri giusti, la velocità di battitura e il tempo.

Figura view della modalità di apprendimento

Il progetto prevedeva anche un’altra parte che purtroppo manca, ovvero la parte di dettatura di un testo inserito dall’utente dettata dal programma questa parte avrebbe dovuto contenere un campo dove caricare il testo da dettare e un TextBox dove poter scrivere, in più tutti i conteggi presenti nella modalità di apprendimento.

## Design procedurale

### Classi

#### Classe Apprendimento

|  |  |
| --- | --- |
|  |  |

Figura Classe Apprendimento

La classe Apprendimento è usata per poter calcolare e i dati mostrati attraverso la view e per confrontare i caratteri che il programma genera con il testo inserito dall’utente.

#### Classe ApprendimentoVIewModel

|  |  |
| --- | --- |
|  |  |

Figura Classe ApprendimentoViewModel

La classe ApprendimentoViewModel server a rappresentare i dati nella view e a gestire la visibilità dei pulsanti.

#### Classe MainViewModel

Figura Classe MainViewModel

La classe MainViewModel serve a cambiare le view che mostra l’applicativo

# Implementazione

## Membri e Proprietà della Classe Apprendimento

NCaratteri contiene il numero di caratteri che verranno stampati e che l’utente sceglie all’interno della view, il numero minimo di caratteri è 50.

TestoUtente contiene il testo che l’utente inserisce.

CarLivello viene riempita con i caratteri generati casualmente nel metodo GeneraLivello.

NLivello contiene il numero del livello attuale, in base a quello si generano livelli con caratteri diversi.

CorrectCar aumenta ogni volta che la lettera inserita dall’utente è corretta.

WrongCar aumenta ogni volta che la lettera inserita dall’utente è errrata.

PercCar viene calcolata con il metodo CalcolaPercenutale ed è la percentuale di caratteri giusti inseriti dall’utente.

Velocita viene calcolata con il metodo CalcolaVelocita ed è la velocita di battitura dell’utente.

Cronometro aumenta ogni secondo tramite il metdo AggiornaCronometro e server per poter calcolare le ore, minuti e secondi tramite il metodo ConverToSM.

TextBoxAttivo determina se il TextBox dove l’utente può scrivere è attivo oppure no.

Mentre le variabili mancanti determinano la visibilità dei pulsanti.

## variabili aiuto Classe Apprendimento

rnd è un random che uso per prendere delle lettere casuali dall’array di lettere da stampare.

Livelli contiene tutte le lettere da stampare per ogni livello.

posCar è la posizione del carattere che controllo nel metodo ConfrontaChar.

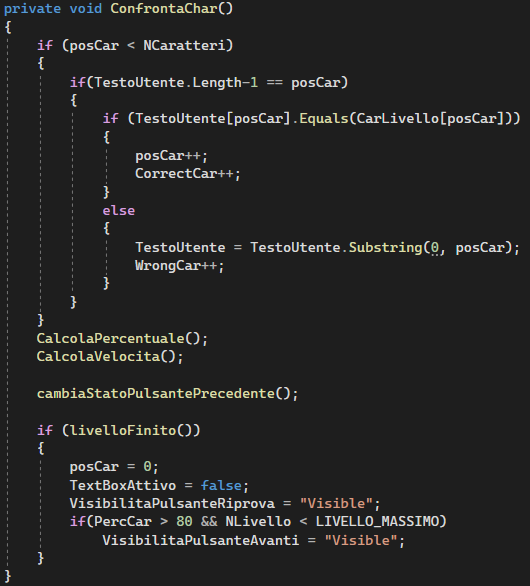
Thread lo uso per incrementare la variabile Cronometro.

## costruttore della classe Apprendimento

Figura costruttore classe Apprendimento

Nel costruttore inizializzo tutte le variabili.

## Metodo GeneraLivello Classe Apprendimento

All’inizio del metodo richiamo il metodo cambiaStatoPulsantePrecedente che serve a far visualizzare il pulsante precedente se il livello è più del primo, azzero CarLivello e in un for con lunghezza di NCaratteri genero un carattere casuale dall’array livelli e se è uno spazio ed è in prima posizione o in ultima posizione oppure se è uno spazio ed il carattere prima è anche esso uno spazio allora rifaccio il for. Infine se nessuna di questa condizione è vera aggiungo il carattere generato a CarLivello.

## Metodo ConfrontaChar Classe Apprendimento

Questo metodo se il la posizione che controllo è minore del numero di caratteri ed è uguale alla lunghezza del testo dell’utente allora e i caratteri sono uguali allora aumento la posizione dal carattere confrontato e aumento la variabile di caratteri corretti altrimenti il TestoUtente rimane uguale a quello di prima.

Quando il livello è finito disattivo il TextBox, faccio apparire il pulsante per riprovare il livello e se la percentuale è maggiore di 80 allora faccio apparire anche il pulsante per andare avanti di livello.

## metodo AggiornaCronometro classe Apprendimento

metodo che aggiorna la variabile cronometro ogni 1000 millisecondi.

## Metodo ConvertToSM Classe Apprendimento

Metodo che calcola minuti e secondi in base al cronometro e lo formatta bene con uno 0 prima del carattere se è minore di 10.

## classe MainViewModel

Questa classe riceve come parametro quale view deve mostrare e la mette all’interno della variabile SelectedViewModel

# Consuntivo

Figura Gannt Consuntivo

# Conclusioni

## Sviluppi futuri

Gli sviluppi futuri del progetto saranno sicuramente implementare la parte di dettatura mancante che a causa di un problema dovuto con il cambio delle view non ho potuto finire perché mi ha fatto perdere molto tempo. Inoltre la velocità di battitura nella modalità di apprendimento è da migliorare siccome mostra sempre 0. In più aggiungere la possibilità di cambiare i livelli attraverso una lista di pulsanti.

## Considerazioni personali

Durante lo svolgimento di questo progetto ho dedotto che il pattern mvvm per sviluppare applicazioni in C# è terrificamene lento per il fatto che tutte le variabili create all’interno della classe model bisogna ricrearle nel viewmodel per poi poterle rappresentare nella view.

# Allegati

Elenco degli allegati, esempio:

* Diari di lavoro
* Codici sorgente
* QdC
* Prodotto
* Gannt Preventivo
* Gannt Consuntivo