# UpWords

## Samuele Pasini-890037

### Appello 23 Febbraio 2018

Per realizzare il gioco UpWords abbiamo usato il linguaggio C++ sulla piattaforma Qt Creator.

Abbiamo utilizzato la programmazione a oggetti perché l’abbiamo ritenuta più adatta a risolvere il problema e per le sue caratteristiche che permetterebbero migliorie e modifiche al gioco in futuro molto facilmente, riutilizzo del codice per altri progetti simili, e divisione del lavoro nel team creato, con annesse scadenze temporali con l’aiuto di MS Project.

Per il testing di ogni parte abbiamo utilizzato un doppio debug, quello offerto da Qt e uno strutturato con cout/cerr a console.

Per lavorare in gruppo ci siamo affidati alla piattaforma GitHub.

Grazie a questo approccio siamo riusciti e suddividere il programma in parti autosufficienti che composte danno vita ad UpWords, procederemo spiegandole una ad una.

## IL SACCHETTO

## IL SUGGERIMENTO

Quella del suggerimento è stata senza dubbio una delle parti più complicate da realizzare e la cui implementazione è soggetta a imperfezioni.

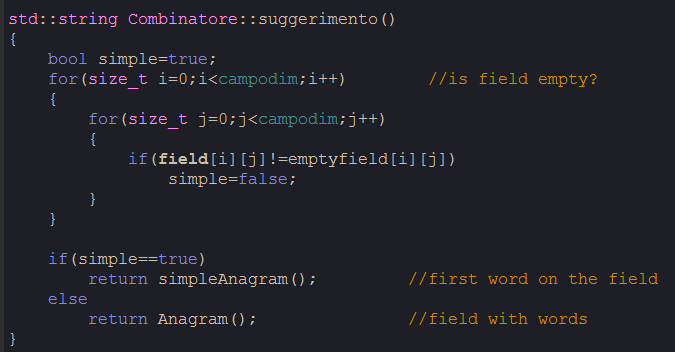
I problemi sono sorti quando agli algoritmi complessi si è aggiunto il fattore “tempo di esecuzione” che ci ha portato a fare delle scelte non sempre ottimali.

La classe “Combinatore”, che si occupa del suggerimento, lavora come un’IA, è capace di mischiare le lettere a disposizione del giocatore e distribuirle su linee e colonne di gioco fino ad ottenere la migliore parola da suggerire all’utente.

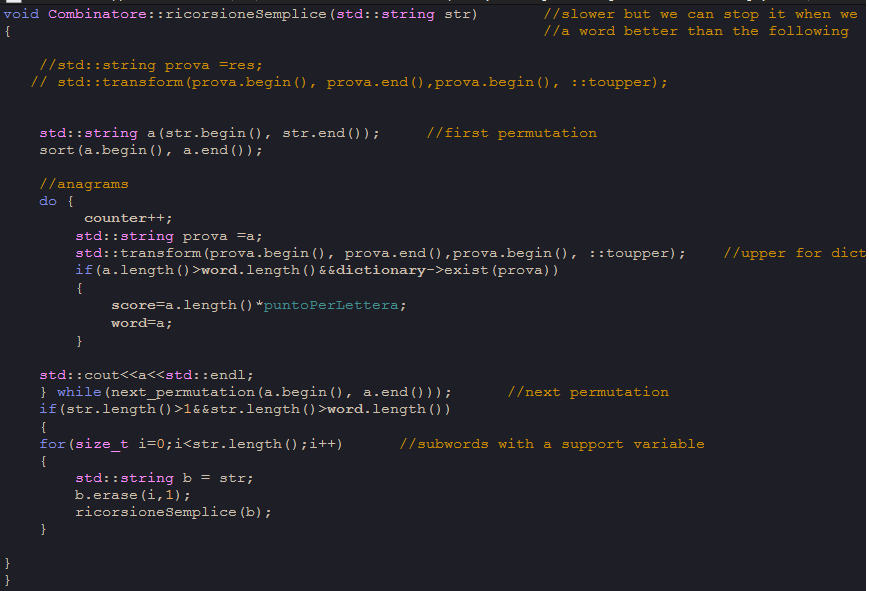
Questa classe potrebbe essere usata come punto di partenza per creare un giocatore automatico contro il quale giocare.

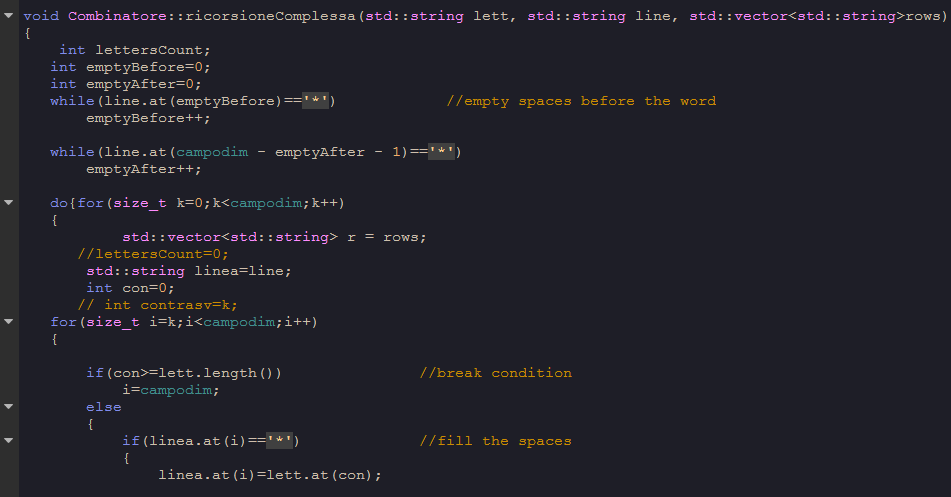
Purtroppo ottenere sempre la parola migliore rendeva il tempo di elaborazione ingestibile, per cui abbiamo omesso le sovrapposizioni e considerato la più lunga parola valida la migliore (lo è nella maggior parte dei casi), e anche con queste semplificazioni a volte si rischia di aspettare parecchio per un suggerimento, a seconda delle condizioni del campo da gioco (tempo variabile tra pochi secondi fino a 3/5 minuti nel caso peggiore).

I metodi della classe agiscono in due modalità differenti a seconda del campo da gioco.



Quando il campo è vuoto il combinatore suggerisce l’anagramma più lungo valido con le tessere del giocatore, mischiandole con un do while, confrontandole con il dizionario e ripetendo l’operazione ricorsivamente su tutte le subwords.



Con il campo già occupato da altre parole si fa più difficile, il combinatore analizza tutte le colonne mischiando le lettere disponibili negli spazi e verificando la validità della parola, e riducendo ricorsivamente le lettere utilizzate

Verificata la validità della parola singola verifica anche quella delle sue trasversali, per poi ripetere l’operazione con le righe

