PRESENTAZIONE LAVORO

Il lavoro in compresenza con i professori di matematica e coding dell'anno scolastico 2021/2022 darà come prodotto: un codice risolutore che genera tutte le possibili combinazioni di lettere che formulano solo parole di senso compiuto nella lingua italiana. Lo scopo principale è quello di fondere l'argomento matematico: lo studio del calcolo combinatorio, con l'argomento di coding, ovvero la stesura di un codice Python (linguaggio di programmazione).

Sullo sfondo di ciò che è stato detto in precedenza, l'intenzione è quella di creare un gioco basato sul codice trascritto, interagendo tra utente-computer; Il gioco è ispirato all'omonima app Ruzzle: un gioco con alla base l'idea di stimolare e allenare la mente, attraverso la ricerca di parole prodotte da lettere che vengono generate casualmente. Il suo fondamento si basa quindi sugli anagrammi, i quali sono basati sul concetto di permutazione, un concetto approfondito in ambito matematico attraverso il calcolo combinatorio. La realizzazione di questo gioco è stata possibile grazie ad alcune librerie in ambiente Python. La nostra versione vuole essere un gioco sia offline che online. Nel primo caso la sfida è contro se stessi, data una griglia 4x4 composta da lettere casuali, l'obiettivo è quello di trovare quante più parole possibili in un tempo massimo di 2 minuti, allo scadere del tempo il computer restituirà tutte le parole possibili confrontandole con quelle trovate, definendo dunque il punteggio. Nella seconda modalità invece è possibile giocare 'online', l'utente deve immettere il codice-partita per poter partecipare alla stessa sessione di un'altro giocatore, da tener conto che la partita si svolgerà sempre contro se stessi, in quanto il confronto avviene con punteggi finali, ma ogni giocatore giocherà la propria partita, ma con le stesse regole e stesse lettere casuali, in modo tale che i punteggi possono essere confrontati e stabilire il vincitore.

FUNZIONALITÀ

Nello specifico il gioco inizierà nel momento in cui viene schiacciata nuova partita, il timer partirà e la griglia delle lettere verrà riempita ogni volta con lettere generate a caso.

Nella prima delle 2 modalità la sfida è contro il computer ma in realtà contro se stessi dato che la CPU vincerà sempre, il gioco consiste quindi nel cercare più parole possibili digitando nel box sottostante alla griglia cercando di trovare tutte le parole possibili con quelle lettere, ogni volta che si trova una parola verrà aggiornato il punteggio nella griglia affianco alla scritta del giocatore(G1 ad esempio), nel tempo massimo di 8 minuti finiti i quali il computer restituirà tutte decretando la fine del gioco.

Nella modalità a 2 giocatori si può giocare sullo stesso computer con un amico, durante questa modalità le parole andranno scritte nel campo di testo a turni poiché una volta verranno assegnate al giocatore uno e una volta al giocatore due e così via, allo scadere del tempo massimo di 5 minuti il computer restituirà sempre nella colonna più a sinistra tutte la parole possibili con quelle lettere e il gioco terminerà decretando come vincitore il giocatore che avrà più punti ovvero colui che avrà trovato più parole

REGOLE

Il gioco termina non appena finisce il tempo, nelle due modalità le regole sono simili, in entrambe l'unico obiettivo è trovare e scrivere nel box di testo le parole, utilizzando le lettere presenti nella griglia, generate casualmente. Invece nella modalità a due giocatori, questi ultimi hanno come unica accortezza quella di scrivere le parole alternativamente, prima uno e poi l'altro.

Il codice del gioco è stato scritto seguendo una programmazione a oggetti.

Programmazione a oggetti

La programmazione ad oggetti, meglio conosciuta come Object Oriented Programming (OOP),è l'uso di classi e oggetti, è una delle tecniche di programmazione più diffuse e consolidate nel modo dello sviluppo software. I principi fondamentali sono adatti sia ad applicativi WEB, che a software più complessi, come giochi 3d.

Classi e Oggetti

La classe è un'astrazione che descrive le caratteristiche comuni ad un insieme di strutture dati, descrive quindi una famiglia di variabili (chiamate Oggetti) con caratteristiche e comportamenti comuni. L'oggetto viene definito anche come istanza della classe, perché a differenza della classe che è una descrizione astratta, l'oggetto rappresenta la descrizione di un elemento in particolare.

Attributi e Metodi

I metodi sono tutte quelle funzioni che si possono applicare ad una stringa, mentre gli attributi sono elementi destinati a contenere dati relativi al singolo oggetto.

Calcolo combinatorio

In definitiva il calcolo combinatorio fornisce quegli strumenti di calcolo per determinare il numero di raggruppamenti che si possono formare con un numero k di oggetti presi da un insieme contenente n oggetti (n ≥ k) secondo le modalità seguenti:

- i k oggetti possono formare gruppi ordinati (disposizioni);
- se k = n otterremo dei gruppi ordinati (permutazioni);
- i k oggetti possono formare gruppi non ordinati (combinazioni).

Esaminiamo in dettaglio questi raggruppamenti.

Disposizioni semplici

Si considera un insieme formato da n elementi distinti ed un numero k ≤ n. Si chiamano disposizioni semplici degli n elementi presi a k a k (o disposizioni della classe k) un gruppo ordinato formato da k degli n elementi dell'insieme dato in modo che valgano le seguenti proprietà:

- in ciascun raggruppamento figurano k oggetti senza ripetizione;
- due di tali disposizioni si ritengono diverse quando differiscono per almeno un elemento oppure per l'ordine con cui gli stessi elementi si presentano.

Il numero delle disposizioni semplici di n elementi distinti della classe k, si indica con il simbolo Dn, k il cui valore è uguale al prodotto di k numeri interi consecutivi decrescenti dei quali il primo è n.

Si ha cioè:

$$D_{(n,k)} = \frac{n!}{(n-k)!} = n \cdot (n-1) \cdot \ldots \cdot (n-k+1)$$

Il simbolo n! si legge n fattoriale e non è altro che il prodotto di n numeri interi decrescenti a partire da n e per definizione si pone 0! = 1. Il suo calcolo è eseguito in uno dei metodi della classe calcComb().

Disposizioni con ripetizione

Si considera un insieme costituito da n elementi distinti ed un numero naturale k senza alcuna limitazione superiore. Il problema che si pone è quello di costruire tutti i possibili raggruppamenti distinti prendendo k oggetti in modo che:

- in ciascun raggruppamento figurano k oggetti ed uno stesso oggetto può figurare, ripetuto, fino ad un massimo di k volte;
- due qualsiasi raggruppamenti sono distinti se uno di essi contiene almeno un oggetto che non figura nell'altro, oppure gli oggetti sono diversamente ordinati, oppure gli oggetti che figurano in uno figurano anche nell'altro ma sono ripetuti un numero diverso di volte.

Il numero delle disposizioni con ripetizione si indica con il simbolo D'n, k e si dimostra che tale numero è dato da:

$$D^r_{n,k} = n^k$$

Permutazioni semplici

Le permutazioni semplici altro non sono che le disposizioni di n oggetti presi ad n ad n. Ossia, dato un insieme di n oggetti, si dicono permutazioni di tali n oggetti tutti i gruppi che si possono formare con gli n oggetti dati prendendoli tutti. Se ne deduce allora che le permutazioni semplici differiscono soltanto per l'ordine con cui sono disposti gli n oggetti distinti contenuti nei vari raggruppamenti. Dalla definizione segue quindi che le permutazioni coincidono con le disposizioni semplici di classe n, quindi il calcolo delle permutazioni è uguale al calcolo del numero delle disposizioni semplici di n elementi di classe n; quindi il numero delle permutazioni di n elementi distinti è uguale al prodotto dei primi n numeri naturali (escluso lo zero) ed è dato dal fattoriale del numero n, ossia:

$$P_n = n!$$

Gli anagrammi altro non sono che le permutazioni che si ottengono da una parola variando solo il posto delle lettere.

Permutazioni con ripetizione

Le permutazioni con ripetizione di n elementi, di cui h, k, ... ripetuti, sono tutti i gruppi formati dagli n elementi, che differiscono per l'ordine in cui si presentano gli elementi distinti e la posizione che occupano gli elementi ripetuti:

$$P_n^r = \frac{n!}{h!\,k!\ldots}$$

Combinazioni semplici

Dato un insieme di n elementi, si dicono combinazioni semplici degli n elementi presi a k a k (o di classe k) con $k \leq n$ tutti i gruppi di k elementi, scelti fra gli n dell'insieme dato, in modo che ciascun gruppo differisca dai restanti almeno per uno degli elementi in esso contenuti (senza considerare, quindi, l'ordine degli elementi). Da notare la differenza fra disposizioni e combinazioni (semplici): mentre nelle disposizioni si tiene conto dell'ordine, nelle combinazioni semplici, invece, si considerano distinti solo quando due i raggruppamenti differiscono almeno per un elemento. Per determinare il numero delle combinazioni semplici di n elementi di classe k, e che indichiamo con il simbolo Cn, k, ci serviamo della formula:

$$C_{n,k} = \frac{D_{n,k}}{P_k}.$$

Il numero di combinazioni viene indicato anche con il simbolo , che si chiama coefficiente binomiale e si legge “n su k”. Il coefficiente binomiale di due numeri n e k, con 0 ≤ k ≤ n, è il numero:

Combinazioni con ripetizione

Considerando un insieme formato da n elementi e fissando un numero k (senza alcuna limitazione superiore), si costruiscono i possibili raggruppamenti distinti prendendo k elementi dell’insieme dato in modo che:

- ogni elemento può essere ripetuto al massimo fino a k volte;
- non interessa l’ordine con cui gli elementi si presentano;
- è diverso il numero di volte col quale un elemento compare.

La formula che dà il numero delle combinazioni con ripetizione di n elementi di classe k è:

$$C'_{n,k} = \binom{k+n-1}{k} = \frac{(n+k-1)!}{k!\,(n-1)!}$$

Probabilità

Si definisce probabilità di un evento il rapporto fra il numero dei casi favorevoli ed il numero dei casi possibili, supposti tutti ugualmente possibili.

$$P(E) = \frac{(\text{numero dei casi favorevoli})}{(\text{numero dei casi possibili})}$$

SPIEGAZIONE DIAGRAMMA DI FLUSSO

All’interno del diagramma di flusso compaiono numerosi blocchi i quali verranno spiegati di seguito.

- **Inizializzazione**: all’interno di questo blocco sono contenute le istruzioni che utilizzando tkinter inizializzano la schermata visualizzata dall’utente una volta avviato il programma contenendo il box di testo nel quale l’utente può scrivere il suo nome, inoltre compare un’altro blocco di testo nel quale qualora si volesse giocare contro un’altro giocatore sarà possibile inserire un codice, questo corrisponde al codice che identifica la specifica partita del giocatore così da poter giocare entrambi con le stesse lettere.
- **Inizializzazione 1:** dopo l’inizializzazione dell’interfaccia e la scrittura da parte dell’utente del proprio nome e, nel caso si volesse giocare contro un altro giocatore, del codice partita. Dopo ciò viene mostrata a schermo un’ulteriore interfaccia all’interno della quale nel caso si abbia scelto di giocare da solo sono presenti 3 bottoni con sopra scritto facile normale e difficile, questi corrispondono alla difficoltà della partita la quale è calcolata utilizzando la probabilità.Viene infine avviata la partita una volta cliccato il bottone, dando avvio alla variabile che tiene il tempo.
- **Inizializzazione 2:** dopo l’inizializzazione dell’interfaccia e la scrittura da parte dell’utente del proprio nome e, nel caso si volesse giocare contro un altro giocatore, del codice partita. Dopo ciò viene mostrata a schermo un’ulteriore interfaccia all’interno della quale nel caso si abbia scelto di giocare con un’altro giocatore e dunque inserito il codice di una partita già esistente verranno presi dal database le parole e la combinazione di lettere della partita già esistente. Viene infine avviata la partita, dando avvio alla variabile che tiene il tempo e fornendo in un angolo della schermata il codice della partita.
- **Check parola:** qui una funzione chiamata confUtil verifica se la combinazione selezionata è corretta confrontandola con una wordlist contenente tutte le parole della lingua italiana restituisce un valore boleano che può essere appunto vero(SI) o falso(NO)
- **Aggiorna punteggio:** mostra all’utente se la parola è corretta o meno e tiene conto del punteggio salvandolo in memoria
- **Calcola punteggio:** prende la variabile che tiene conto del punteggio e la salva sul database inserendolo nella classifica.

- **Mostra schermata finale:** mostra un’ulteriore interfaccia in cui si visualizza la classifica con il proprio punteggio e si chiede eventualmente di rigiocare cliccando un pulsante.