Documentazione progetto M226A

Christian Arzani I2BB

Sommario

[Introduzione al progetto 2](#_Toc158893119)

[Descrizione del gioco: 2](#_Toc158893120)

[Suddivisione del progetto in quattro classi: 2](#_Toc158893121)

[Classi 3](#_Toc158893122)

[Table 3](#_Toc158893123)

[Player 5](#_Toc158893124)

[Sound 5](#_Toc158893125)

[Program 6](#_Toc158893126)

# Introduzione al progetto

## Descrizione del gioco:

Il progetto Prato Fiorito è un'applicazione Java che simula il gioco “Campo Minato". Lo scopo del gioco è evitare di cliccare sui fiori presenti nel campo di gioco, cercando di scoprire tutte le celle vuote senza fiori.

## Suddivisione del progetto in quattro classi:

* Table: Rappresenta la tabella di gioco e gestisce la generazione dei fiori, il controllo delle celle e la visualizzazione della tabella.
* Player: Rappresenta il giocatore, gestisce il punteggio e il nome del giocatore.
* Sound: Gestisce la riproduzione del suono di sottofondo durante il gioco in loop.
* Program: Classe principale che avvia il gioco e gestisce l'interazione con l'utente.

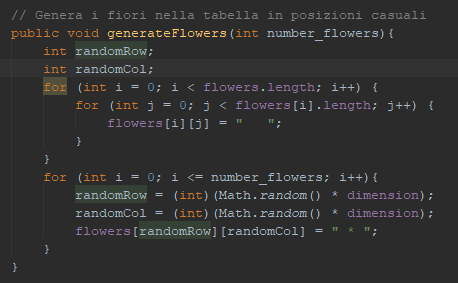
# Classi

## Table

La classe Table rappresenta la tabella di gioco e contiene i metodi per gestirla e per gestire la tabella dei fiori.

Metodi principali:

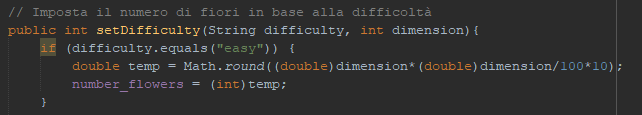
createTable(): Crea la tabella di gioco e la riempie con #.

generateFlowers(int number\_flowers): genera una tabella vuota dove vengono aggiunti dei fiori in posizioni casuali e in basa alla difficoltà si stabilisce il numero di fiori.

checkFlowers(int row, int column): Controlla se c’è un fiore della posizione specificata e in basa a questo ritorna se il gioco finisce o se deve continuare. Inoltre, viene sottratto 1 alla colonna e alla riga inserita per rendere il gioco più user friendly in modo che l’utente non debba inserire 0.

checkScore(int row, int column): Controlla se assegnare il punto al giocatore, se l’utente inserisce un punto della tabella che contiene una spazio vuoto o un asterisco non dal punto, altrimenti si.

setDifficulty(String difficulty, int dimension): Controlla se l’utente inserisce come difficoltà 3 stringhe: easy, medium e hard.   
In basa alla difficoltà viene impostata una percentuale che stabilisce il numero di fiori all’interno della tabella (easy = 10%, medium = 20%, hard = 30%).   
Per ogni difficoltà si crea una variabile di tipo double che contiene il calcolo per il numero di fiori, ovvero: dimension\*dimension/100\*10 (easy).

Per la dimension essendo una variabile di tipo int viene fatto un casting in double per fare il calcolo. Il risultato del calcolo viene approssimato tramite il metodo Math.round. Siccome il risultato della variabile è di tipo double dobbiamo fare un casting in int perché il metodo dovrà restituire un valore di tipo int.

## Player

La classe Player gestisce il giocatore e il suo punteggio.

Metodi principali:

addScore(): Incrementa il punteggio del giocatore.

winGame(int dimension): Verifica se il giocatore ha vinto la partita quando lo score dell’utente corrisponde a (dimension\*dimension) -1.

## Sound

La classe Sound gestisce la riproduzione della musica durante il gioco. Riproduce un file .wav in loop.

## Program

* Richiede il nome del giocatore e la dimensione della tabella.
* Gestisce la selezione della difficoltà del gioco.
* Avvia la creazione della tabella e la generazione dei fiori.
* Consente all'utente di selezionare le celle della tabella e controlla che non inserisca dati non consentiti.
* Controlla se il giocatore ha vinto la partita e stampa il risultato finale.