

## 1. Introducció

Aquest projecte tracta sobre una versió modificada de Scrabble realitzada a l'IDE NetBeans, on el jugador/s, donades unes fitxes seleccionades aleatòriament, ha de formar una paraula que es valorarà en funció del valor de cada lletra elegida (ja que, com en el joc original, cada lletra consta d'un valor determinat). Si aquesta paraula es troba al diccionari (fitxer ja establert) i es pot formar en funció de les fitxes donades, el jugador obtindrà una certa puntuació. En canvi, si no es compleix una de les condicions anteriors, el jugador no sumarà punts i, a més, obtindrà una penalització de -10 punts a la seva puntuació total.

A més, dins el programa s'han implementat altres opcions que fan que el joc resulti més complet i divertit, com per exemple la visualització d'un historial, una selecció d'idiomes o la opcionalitat de diferents modes de joc. Tot això amb una explicació clara i directa de com funciona el joc i les pases que s'han de seguir en cada moment.

## 2. Disseny

Per dur a terme el MINI-SCRABBLE hem emprat sis classes distintes, la classe Paraula, la classe FitxerParaulesEntrada, la classe FitxerParaulesSortida, la classe FitxerTextEntrada, la classe Fitxa i el main, anomenat Scrabble.

### Main:

Dins el main decidirem què volem fer en tot moment, els altres programes són complementaris. Podrem dur a terme la tria del idioma (Espanyol o Català), triar el mode de joc (1 jugador, PVC o PVP) i certs paràmetres propis de cada mode de joc.

El mode de 1 jugador (o solitari) consisteix en intentar aconseguir la major quantitat de punts durant les rondes que el jugador trii. Per llegir la paraula introduïda per l'usuari feim emprar la classe Paraula amb el mètode llegeixParaula(), que retorna una paraula introduïda per teclat. Aquest mètode s'emprarà sempre que un usuari hagi d'introduir alguna paraula. Per saber quants de punts té una paraula feim emprar retornaPunts(), el qual necessita una paraula per paràmetre per saber la seva puntuació. Aquests mètodes els explicarem més endavant. Un cop finalitzades les rondes, es mostra la puntuació i s'emmagatzema dins el fitxer "historial.txt".

Al mode PVC tenim a una màquina competint contra nosaltres. La manera en la que noltros introduïm les paraules és igual que a 1 jugador, però innovam amb la manera en que la màquina tria la paraula possible. Cervell Superior, amb el mètode triaMillorParaula(), recorr tot el diccionari mentre va mirant quines paraules pot fer, un cop pot fer una, la guarda només si la seva puntuació és major que la de l'anterior, i així fins que recorr el diccionari. El simulador no és com a la pràctica ens recomana fer, sino que depenguent de la dificultat

triada, és capaç de cercar entre un abast més gran de paraules dins el diccionari. En finalitzar, registre el guanyador dins l'història.

El mode PVP ens permet competir entre dos i quatre jugadors, emprant el `llegeixParaula()` que s'empra al solitari, però en aquest cas tenim més variables i paraules en funció del nombre de jugadors elegits. En finalitzar, mostra el guanyador i el guarda dins l'història.

### **Paraula:**

Una "paraula", dins el nostre programa és un array de chars d'una longitud màxima determinada i de una longitud la qual marcarà la longitud real de la paraula

La seva funció principal serà `llegeix()`, la qual retornarà una paraula escrita per teclat i s'emprarà al llarg de tot el programa. Altres funcions importants són `triaMillorParaula()`, emprada per Cervell Superior, i explicada anteriorment, juntament amb `triaSimulador()`, que es la modificació de `triaMillorParaula` per que pugui triar més o menys paraules en funció d'un paràmetre.

### **FitxerTextEntrada:**

Aquesta classe és la que ens permetrà agafar la informació necessària del fitxer "alfabet.alf", que es produirà dins la classe `Fitxa`, concretament en el mètode `creaSacs()`; i el mètode `sorteigFitxes()`; d'aquesta.

### **Fitxa:**

Aquesta classe s'encarrega de preparar les fitxes per a cada ronda, les seves funcions principals són `creaSacs()` i `sorteigFitxes()`.

`CreaSacs()` llegeix l'arxiu "alfabet.alf", i, mitjançant un recorregut, recollecta la informació de dins aquest arxiu, emmagatzemant-les dins els seus atributs.

`SorteigFitxes()` s'encarrega d'agafar onze fitxes aleatòries amb les quals jugarem. Duran la durada de cada ronda les fitxes no es repetiran, però al finalitzar la ronda, totes les fitxes tornen al sac, per dur a terme lo previ, emprarem el mètode `creaNovesFitxes()`, que refresca les fitxes que hem descartat prèviament.

### **FitxerParaulesEntrada:**

Aquesta classe s'encarrega de recórrer els fitxers diccionari tan en l'idioma català com en el castellà per tal de dir si la paraula escrita pel jugador es vàlida o no, concretament es troba als mètodes `estaInclosCatala()` i `estaInclosEspañol()` en el main, de tal manera que una recor el diccionari en català i l'altra fa lo mateix però en el diccionari en español.

Aquests posteriorment s'usen com a condició per determinar si una paraula és vàlida o no.

### **FitxerParaulesSortida:**

Aquesta classe la farem servir només per escriure text a "historial.txt". Per a cada mode de joc hem fet que escrigui un text diferent, depenguent dels punts, nom del jugador, idioma, etc. La informació de l'historial es pot consultar desde el mateix main. També hem implementat que guardi la data en la qual es realitza la partida.

### **3. Conclusions:**

Una vegada finalitzat el treball, els participants d'aquest grup consideren que aquest ha resultat ser un treball molt complet que ha tocat pràcticament totes les coses que s'han vist durant el curs. D'aquesta manera hem pogut acabar d'entendre diverses coses que anteriorment no estaven del tot clares i, a més, hem pogut repasar i practicar per l'examen final de l'assignatura.

Concretament, gràcies a la realització del treball hem aconseguit entendre millor conceptes com la utilització de múltiples classes dins un mateix projecte, el tractament de qualsevol tipus de cadena de text, la utilització de la variable aleatòria o l'implementació i el format d'una data dins un fitxer. Tot això a més de l'especialització en el tractament seqüencial i el disseny descendent d'un programa.

Per una altra banda, considerem que els punts més complexos tractats en el programa han estat, principalment, la creació d'una bona base de joc en solitari (ja que un cop s'ha obtingut aquesta, la resta de programa es mou en funció seva), i alguns mecanismes extra que, més que difícils han resultat tediosos, com per exemple la opció de multijugador amb 4 jugadors, pel simple fet d'haver de tenir en compte a tots aquests dins una mateixa partida tant en català com en castellà i la seva posterior afegida a l'historial.

En conclusió, ha estat una experiència molt agradable ja que, a més de tot el que s'ha après durant el seu desenvolupament, també ha millorat la nostra coordinació en equip i ens ha permès tractar el projecte de forma dinàmica i didàctica.