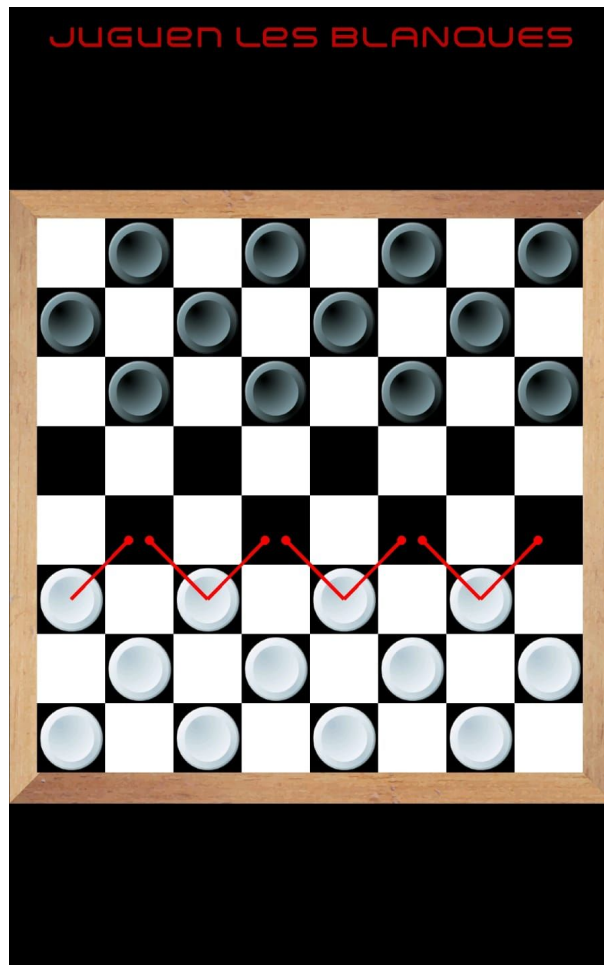


# Intel·ligència Artificial - 21722

## Joc de les dames



### **Alumnes**

López Martínez, Francisco  
Pericàs Serra, Bernat

### **Professor**

Mas Sansó, Ramon

# Classes

Pel que fa a l'entorn proporcionat, només hem realitzat canvis a la classe Cerca i a la classe Game

A la classe *Cerca.java*, tenim:

- La creació de l'objecte *GeneradorDeMoviments*, la declaració i inicialització de les variables *alpha* i *beta*. La declaració de la variable *node* i la de *millor\_puntuació* que inicialment valdrà *MAX\_VALUE*. Finalment un array de 2 posicions que rebrà el nom de *millor\_jugada*.
- El constructor de la classe *Cerca* que reb per paràmetre l'objecte *game*.
- Un mètode *avaluaTauler* que rep per paràmetre un tauler i un torn per saber si han de jugar les blanques o les negres. Aquesta funció retorna un objecte *Moviments* amb el millor moviment trobat.
- La cridada al mètode que s'anomena *generaMovimentsPeces* que rep un tauler i un torn. La seva funció és fer tots els possibles moviments del jugador amb torn al tauler.
- La cridada al mètode *simula\_moviment* que donat un tauler i un moviment, retorna un nou tauler amb el moviment m fet.

- Un mètode *valorPosicio* en el qual es passa per paràmetre els moviments que faran les blanques i les negres a partir del nivell d'on està. Es mira tots els moviments possibles recorrent l'array de moviments. En el cas que faci un salt, se li dóna uns punts, si fa dama, se li dóna uns altres i es van sumant. Quan s'acaba ens quedem amb els punts màxims, és a dir, amb el màxim de les blanques i les negres i retornam un enter (el qual és la resta de la màxima puntuació que fan les blanques i de la màxima posició de les negres, sent aquest nombre positiu).
- Un mètode *minimax* on miram primer si es guanya, si es perd o si s'empata. En cas contrari, obtenim la nova millor nova puntuació, generam un nou moviment i simulam un nou moviment del qual obtenim ja el tauler amb el moviment fet. Després cridam al *minimax* amb aquest tauler amb el moviment fet i un nivell més. I finalment estudiam el contingut de la variable *valor*. Ficam al nivell de millor jugada el millor destí i retornam la millor puntuació.

A la classe *Game.java*, només hem afegit un mètode *canviaTorn* que fa que el torn següent mogui el rival.

## Conclusions

Una vegada superats els problemes inicials d'instal·lació i configuració de l'entorn, no hem tengut cap dificultat destacable a l'hora de realitzar la pràctica, ja que en tot moment sabíem que havíem de fer.

Aquesta pràctica ens ha permès, en primer lloc, aprendre a usar l'*Android Studio* i a obtenir un joc per a android per primer cop en tota la carrera, el qual és un fet que s'agraeix.

Però sobretot ens ha permès posar en pràctica una de les tècniques més usades en les cerques amb adversaris i a arribar a conèixer el funcionament d'aquesta.

És d'agrair que sigui una pràctica curta; és la darrera i els exàmens finals són molt a prop, amb uns objectius clars; ben especificada, i que es proporcioni un entorn amb el qual familiaritzar-se i no haver de començar de zero.