

# *PROJECTE*

# *BACKTRACKING*

Algorísmia i Estructures de Dades I



**Universitat**  
de les Illes Balears

*Joan Alberola, Harpo*

*Marí Enseñat, Juan José*

## Guia d'ús del programa

El programa *Backtracking* consta de 20 classes, entre les que hi podem destacar 6 (contingudes dins el *Package "View"*) que gestionen la interacció gràfica del programa. És gràcies a aquesta GUI que l'usuari té l'opció de triar la peça amb que es resol el problema i la posició en que aquesta comença.

Un cop iniciada l'execució, venen predeterminades les opcions de peça (*Knight*), fila (0) i columna (0), com s'adjunta a la figura 1. Si es desitja canviar aquests valors, es disposa d'un menú desplegable i uns botons per augmentar/disminuir els valors.

Quan s'hagi configurat així com es vol el joc, s'ha de prémer el botó de "*Compute*" i el programa calcularà la ruta de la peça. Quan es trobi un resultat, es mostraran a cada una de les caselles del tauler els nombres que indiquen l'ordre en què es mou la peça.

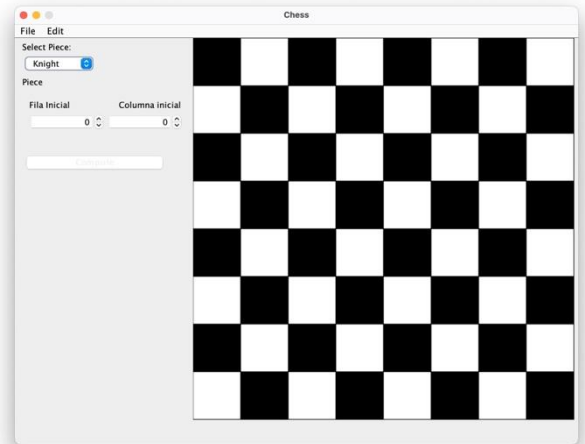


Figura 1. Finestra del programa

Si es vol crear una nova peça, s'haurà de crear una classe nova usant la classe *Piece* com a base, per a poder seleccionar-la s'haurà de implementar el seu nom dins la classe "*Backtracker*" dins la carpeta "*Controler*", a més de crear un nou ítem al combo box de la classe "*MainWindow*" a la línia 73 del codi.

## Mètode iteratiu i recursiu

Per defecte està posat el mètode iteratiu, no és necessari posar l'iterador "*k*" a la cridada del mètode a "*solveTour*" a la classe "*Piece*", però en el cas del recursiu si que s'ha de posar, essent aquest 2.

En el cas recursiu la peça *Knight* és la que més tarda, però si s'executa desde el mig del tauler és més ràpid, la resta és independent d'on es comenci.

El mètode recursiu treballa agafant el primer moviment vàlid, fent així una lectura per inordre, mentre que en l'altre cas agafam desde el darrer moviment possible per tant fem un postordre.

## Funcions de poda

Per a millorar l'eficiència del programa, s'usa una funció de poda que ens verifica quan un moviment d'una peça és vàlid o no. Per a això, dins la classe abstracta "*Piece*" s'inclou el mètode "*isValid*", el qual rep per paràmetre la posició d'una peça i comprova que sigui un moviment vàlid. A continuació s'adjunta el codi:

```
private boolean isValid(int[][] t, int x, int y) {  
    int n = t.length;  
    return x >= 0 && x < n && y >= 0 && y < n && t[x][y] == 0;  
}
```