

# PROGRAMACIÓ AVANÇADA

## PRÀCTICA 5. TARDOR 2015

---

**Objectiu 1: Tècnica del backtracking.** Anàlisi d'un problema

**Objectiu 2:** Resoldre un Joc de taula

**Objectiu 3:** Aplicar la tècnica de l'anàlisi descendent per obtenir un bon disseny de classes

**Durada:** Una sessió

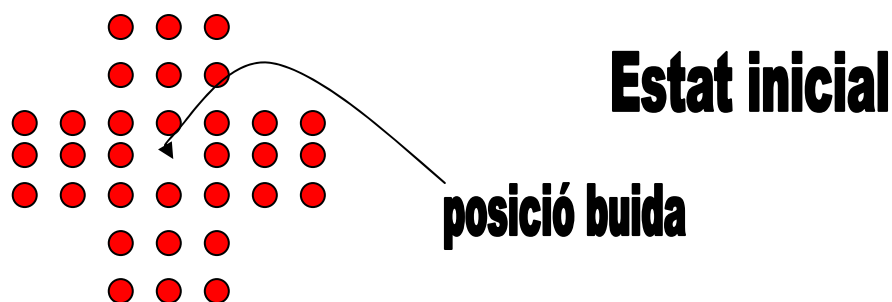
**Lliurament:** Llistat imprès dels fonts i anàlisi del problema i penjar el projecte al Moodle

**Data Lliurament:** Dia abans de l'examen 2 (13/Desembre)

### Enunciat. El Continental

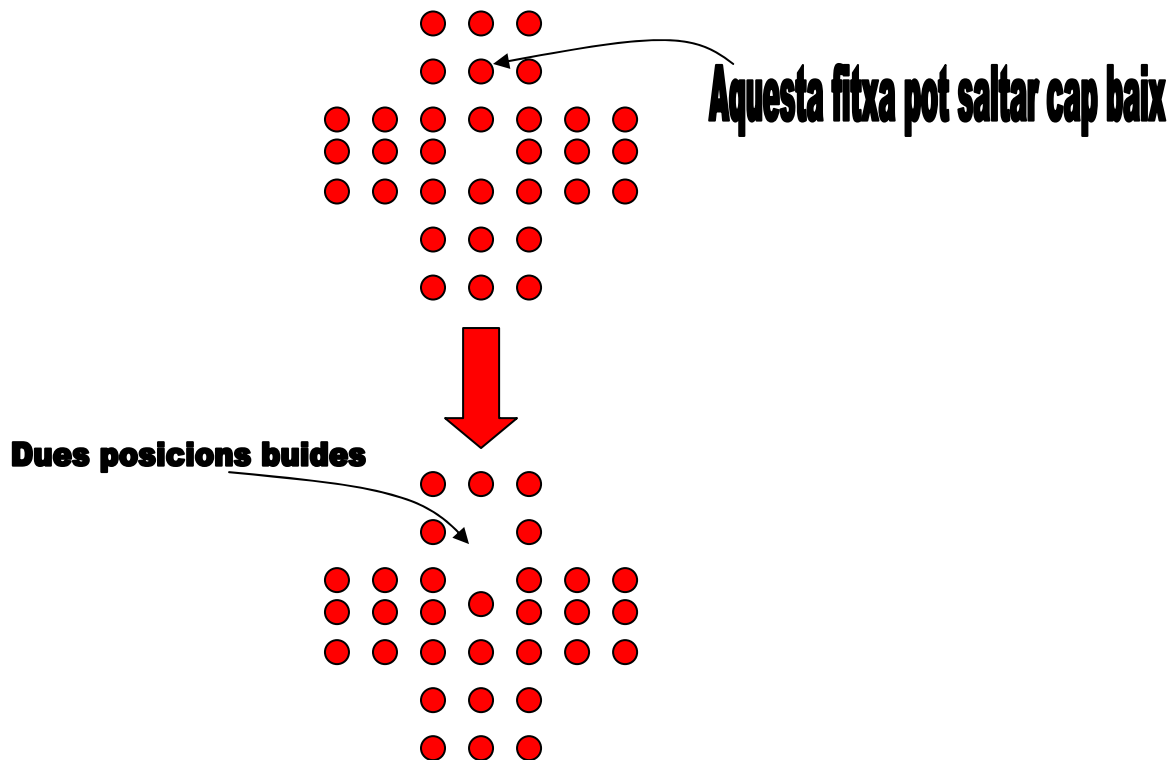
S'ha d'escriure un programa que usant la tècnica del backtracking trobi 2 solucions diferents al joc del Continental.

El **continental** és un joc solitari de taula que consta de 32 peces que es col·loquen en un taulell que té forma de creu i que consta de 33 caselles. Al començar el joc es col·loquen totes les peces sobre el taulell deixant una única casella buida, la central; tal com es mostra en el següent gràfic, les rodones denoten les peces del joc i l'única casella buida està situada a la part central del taulell:



**L'objectiu** és arribar a un **estat final** on el taulell tingui una única peça situada a la **posició central**, la resta de caselles han d'estar buides. Les peces es van matant d'una manera molt similar al joc de dames, en aquest cas els únics moviments possibles són horitzontal dreta, horitzontal esquerra, vertical amunt i vertical baix (4 moviments). Una peça pot fer un d'aquests moviments saltant sobre una de les seves veïnes i anant a parar a una casella buida; tingueu en compte que no es pot sortir del taulell. La peça saltada es treu del taulell.

**Exemple de moviment** possible sobre l'estat inicial:



**Cada cop que es fa un moviment el taulell té una fitxa menys.**

Es tracta d'escriure un algorisme que trobi **dues seqüències de moviments (salts) que, partint de l'estat inicial arribi a l'estat final**. L'estat final ve donat, com s'ha dit anteriorment, quan totes les peces són eliminades del taulell a excepció d'una que queda ubicada a la casella central.

#### Consideracions:

→ 1.- El taulell es representa mitjançant una matriu de dues dimensions, cada posició pot tenir uns dels següents valors:

- lliure (no té cap peça)
- ocupada (per una peça del joc)
- posició NO vàlida (fora del taulell, és una posició que no forma part de la creu)

→ 2.- La solució cal posar-la dins d'una **taula**, cada salt ocuparà una posició de la taula i representarà un moviment del joc. El primer moviment serà emmagatzemant en la posició d'índex 0, el segon en la posició d'índex 1, .... fins el darrer. Per cada moviment es vol emmagatzemar:

- 1<sup>er</sup>.- la posició de la fitxa que va a saltar ( $X_{inici}, Y_{inici}$ )
- 2<sup>on</sup>.- la posició on queda la fitxa després de realitzar el salt ( $X_{fi}, Y_{fi}$ )
- 3<sup>er</sup>.- la posició ( $x, y$ ) de la fitxa menjada.
- Pel salt de l'exemple anterior s'haurà d'emmagatzemar (recordeu que la indexació en una matriu s'inicia per 0):

Objecte classe **Coordenada**

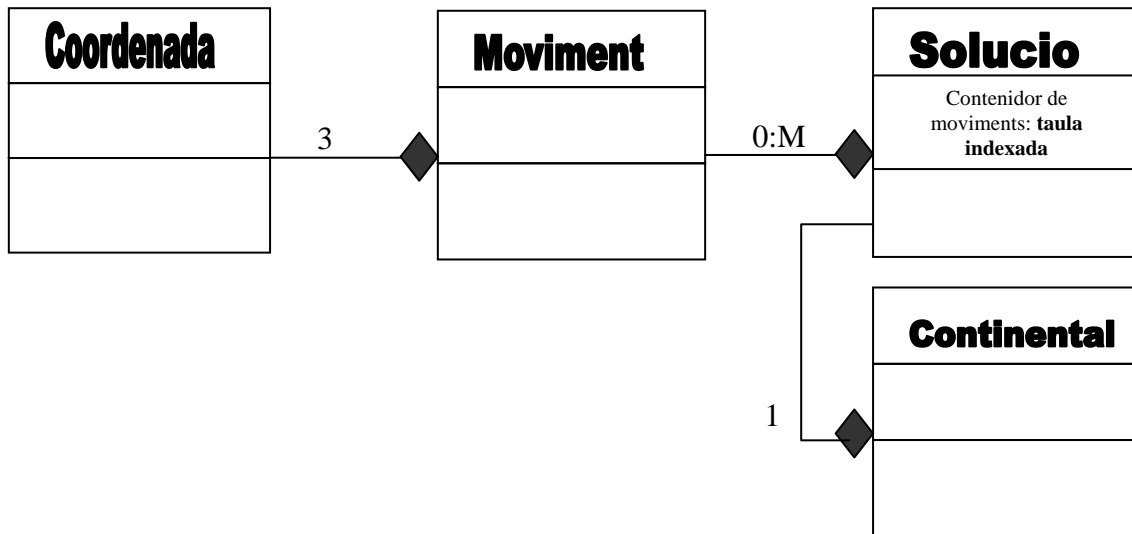
$(X_{inici}, Y_{inici}) = (1, 3)$

$(X_{fi}, Y_{fi}) = (3, 3)$

$(x, y) = (2, 3)$

Objecte classe **Moviment**

Per resoldre aquest problema segons els criteris exposats prèviament cal dissenyar unes classes i relacionar-les de la següent manera. Gràficament:



### Aclariments:

1.- **Classe Coordenada**: objecte per emmagatzemar les coordenades d'una casella del taulell (fila,columna)

2.- **Classe Moviment**: un objecte d'aquesta classe conté 3 objectes de la classe Coordenada, per emmagatzemar les 3 coordenades que involucra un salt (moviment).

3.- **Classe Solucio**: un objecte d'aquesta classe emmagatzemarà la solució del problema, és a dir tindrà un atribut que serà una taula indexada d'objectes Moviment, tal com demana l'enunciat.

4.- **Classe Continental**: un objecte d'aquesta classe tindrà com a mínim dos atributs, un per el taulell de joc i altre l'objecte de la classe solució per emmagatzemar la solució que es demana. Hi podeu afegir tots els atributs i mètodes necessaris per trobar les solucions dels problema. El mètode que implementarà el Backtracking serà ubicat en aquesta classe.

**Es valorarà l'anàlisi descendent, és a dir féu mètodes privats que ajudin al backtracking i facin més llegible i entenedor aquest mètode. I també, l'eficiència de la solució trobada.**

El mètode main, punt d'entrada de l'execució, el podeu ubicar dins d'aquesta mateixa classe Continental o afegir altra classe amb aquesta finalitat.

**Es necessari que un cop es trobin les solucions del problema es visualitzin a pantalla, mitjançant un mètode que l'ensenyi "gràficament". Per tant, redefiniu el mètode toString dins de les classes adients. Trobeu una visualització entenedora, fàcil d'entendre i d'interpretar les solucions trobades.**

**Prèviament a resoldre la pràctica, s'ha de fer un anàlisi del problema.**

**Anàlisi del problema. Qüestions a contestar prèviament a la implementació**

- 1.- Per què l'esquema de backtracking és aplicable per a resoldre aquest enunciat.
- 2.- Quines decisions ha de prendre la tècnica del backtracking en aquest exercici.
- 3.- Quin és el criteri per determinar si una decisió és o no acceptable.
- 4.- Quin és el criteri per determinar si un conjunt de decisions és o no completable.
- 5.- Quin és el criteri per determinar si un conjunt de decisions són o no solució.
- 6.- Dibuixeu l'espai de cerca del problema, és a dir l'arbre que recorrerà la tècnica del backtracking, especificant quina serà l'alçada i l'amplada, indicant si són valors exactes o valors màxims. Usareu marcatge?.
- 7.- Indiqueu quina variant de l'esquema del backtracking cal aplicar, trobar totes les solucions, només una o la millor.
- 8.- En el cas de tenir de trobar la millor solució, és a dir si el vostre enunciat és d'optimització, indiqueu quin és el criteri que seguireu per determinar si una solució és o no millor a una altra ja trobada prèviament.
- 9.- El vostre enunciat pot resoldre's aplicant la **tècnica voraç**?. Justifica la resposta. En cas afirmatiu, quina seria la funció de selecció?. Trobaria la millor solució?.
- 10.- Expliqueu el procediment que seguieu per trobar només dues solucions del problema.

**Qualsevol implementació que NO s'ajusti a l'espai de cerca que indiqueu en la contestació de les preguntes prèvies es considerarà INCORRECTE, malgrat funcioni.**

**Què se us subministra?**

Fitxer amb l'enunciat de la pràctica.

**Què s'ha de lliurar i com?**

S'ha de lliurar la carpeta que conté el projecte Eclipse amb el vostre desenvolupament de la pràctica. La carpeta s'ha de lliurar amb tot el seu contingut i comprimida amb ZIP o RAR.

També s'ha de lliurar el llistat en paper de tot el codi desenvolupat. I en aquesta pràctica també cal un llistat en paper de l'anàlisi del plantejament i resolució del problema on quedin contestades totes les qüestions indicades prèviament.

**On s'ha de lliurar?**

El lliurament del projecte es farà a través de la plataforma Moodle i no s'acceptarà cap altra via. Feu atenció a la data i hora límit.

**El lliurament en paper es farà directament a la professora abans de començar l'examen, el dia 14 de desembre a les 9h.**

**És imprescindible lliurar també, imprès en paper, la contestació a totes les qüestions plantejades i prèvies a la resolució del problema.**

**Quan s'ha de lliurar?**

El lliurament es podrà fer fins el **13 de desembre a les 23:50** (hora del sistema Moodle). Tingueu present que a partir d'aquesta hora el sistema bloquejarà, de manera automàtica, la possibilitat de lliurament.