

# Tebes: Estudi de tècniques aplicades a un joc abstracte

Treball de Fi de Grau

**Sofia Sales Salvatierra**

Grau d'Enginyeria Informàtica

**Director**

Vicenç Torra Reventós

3 de juny del 2015



Aquesta obra està subjecta a una llicència de  
[Reconeixement-NoComercial-SenseObraDerivada 3.0  
Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

## Fitxa del treball de fi de grau

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                 |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------|
| <b>Títol del treball:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | <i>Tebes : Estudi de tècniques aplicades a un joc abstracte</i> |
| <b>Nom de l'autor:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | <i>Sofia Sales Salvatierra</i>                                  |
| <b>Nom del consultor:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | <i>Vicenç Torra Reventós</i>                                    |
| <b>Data de lliurament (mm/aaaa):</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | <i>06/2015</i>                                                  |
| <b>Àrea del Treball Final:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | <i>05.629 - TFG Intel.ligència artificial</i>                   |
| <b>Titulació:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | <i>Grau en Enginyeria Informàtica</i>                           |
| <b>Resum del Treball:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                                 |
| <p>La disciplina de la intel·ligència artificial té un abast molt ampli, com ara, aplicacions en la robòtica, en els videojocs, mineria de dades o processament del llenguatge natural entre d'altres. En aquest projecte, ens centrarem en aquelles tècniques que ens poden ajudar a solucionar un joc de taula abstracte com el Tebes, que posteriorment explicarem.</p> <p>En aquest estudi es parteix d'una solució inicial i s'evoluciona cap a diverses variants per a poder analitzar-les i comparar-les unes amb altres. L'objectiu ha estat, d'una banda construir un adversari per al jugador humà que pugui resultar un repte, i d'altra comparar diverses tècniques per trobar la millor alternativa pel programa, tenint en compte el temps i recursos dels que disposem.</p> |                                                                 |

**Abstract:**

Artificial intelligence has a very wide scope including applications in robotics, video games, and natural language processing. In this Bachelor's thesis we will focus on those techniques that can help solve an abstract board game as Tebes, which will be later explained.

This study is based on an initial solution which evolves into different playing options that are analyzed and compared. The first objective was to build an adversary for a human player which could be a challenge, and the second to compare different techniques to find the best alternative for the program, given user's time and resource constraints.

**Paraules clau:**

*Intel·ligència artificial, Minimax, Monte-Carlo Tree Search, Heurística, Joc abstracte, Tebes.*

## Índex de continguts

|                                                               |           |
|---------------------------------------------------------------|-----------|
| <b>Fitxa del treball de fi de grau .....</b>                  | <b>3</b>  |
| <b>Índex de continguts.....</b>                               | <b>5</b>  |
| <b>Índex de figures.....</b>                                  | <b>7</b>  |
| <b>Índex de taules .....</b>                                  | <b>8</b>  |
| <b>CAPÍTOL 1 : INTRODUCCIÓ .....</b>                          | <b>9</b>  |
| 1.1 Justificació.....                                         | 9         |
| 1.2 Objectius .....                                           | 9         |
| 1.3 Planificació del projecte .....                           | 10        |
| 1.4 Productes obtinguts .....                                 | 11        |
| 1.5 Descripció dels capítols de la memòria.....               | 12        |
| <b>CAPÍTOL 2 : EL JOC TEBES .....</b>                         | <b>13</b> |
| 2.1 El joc tebes .....                                        | 13        |
| 2.2 Regles del joc .....                                      | 14        |
| 2.3 Classificació del tipus de joc.....                       | 17        |
| 2.4 Avaluació de la complexitat.....                          | 18        |
| 2.5 Representació del joc .....                               | 19        |
| <b>CAPÍTOL 3 : SOLUCIÓ INICIAL .....</b>                      | <b>20</b> |
| 3.1 Algorisme de cerca ( minimax ).....                       | 20        |
| 3.2 Poda Alfa-Beta .....                                      | 22        |
| 3.3 Funció d'avaluació .....                                  | 25        |
| 3.4 Mapa de pesos .....                                       | 27        |
| 3.5 Millores i adaptacions .....                              | 29        |
| <b>CAPÍTOL 4 : ALTRES ESTRATÈGIES CONSIDERADES.....</b>       | <b>32</b> |
| 4.1 MCTS.....                                                 | 32        |
| 4.2 MCTS amb heurística.....                                  | 35        |
| 4.3 Segona versió de l'heurística.....                        | 37        |
| <b>CAPÍTOL 5 : ANÀLISIS DELS RESULTATS .....</b>              | <b>38</b> |
| 5.1 Resultats diferents profunditats del Minimax .....        | 38        |
| 5.2 Resultats de les diferents heurístiques.....              | 41        |
| 5.3 Resultats del MCTS + heurística amb diferents temps ..... | 43        |
| 5.4 Resultats dels diferents algorismes .....                 | 44        |
| 5.5 Conclusions de l'estudi comparatiu .....                  | 45        |

|                                               |           |
|-----------------------------------------------|-----------|
| <b>CAPÍTOL 6 : IMPLEMENTACIÓ .....</b>        | <b>46</b> |
| 6.1 Eines utilitzades .....                   | 46        |
| 6.2 Procediment seguit .....                  | 46        |
| 6.3 Diagrama de classes .....                 | 47        |
| <b>CAPÍTOL 7 : CONCLUSIONS GENERALS .....</b> | <b>48</b> |
| <b>GLOSSARI .....</b>                         | <b>49</b> |
| <b>BIBLIOGRAFIA .....</b>                     | <b>50</b> |
| <b>ANNEX .....</b>                            | <b>52</b> |
| Manual de l'aplicació .....                   | 52        |

## Índex de figures

|                                                               |    |
|---------------------------------------------------------------|----|
| Figura 2.1 Caselles del Tebes .....                           | 11 |
| Figura 2.2 Tauler del Tebes .....                             | 11 |
| Figura 2.3 Fitxes del Tebes .....                             | 11 |
| Figura 2.4 Tauler estàndard .....                             | 12 |
| Figura 2.5 Tauler estàndard amb fitxes .....                  | 14 |
| Figura 2.6 Tauler aleatori 1 .....                            | 12 |
| Figura 2.7 Tauler aleatori 2 .....                            | 12 |
| Figura 2.8 Moviments possibles .....                          | 13 |
| Figura 2.9 Moviment incorrecte.....                           | 15 |
| Figura 2.10 Matar a dues posicions.....                       | 16 |
| Figura 2.11 Matar a 1 posicio.....                            | 16 |
| Figura 2.12 Representació tauler de 4x4.....                  | 17 |
| Figura 2.13 Representació tauler de 6x6.....                  | 17 |
| Figura 3.1 Arbre Minimax .....                                | 20 |
| Figura 3.2 Pseudocodi Minimax .....                           | 21 |
| Figura 3.3 Poda Alfa-Beta.....                                | 22 |
| Figura 3.4 Pseudocodi Minimax amb poda Alfa-Beta .....        | 23 |
| Figura 3.5 Pseudocodi funció avaluació.....                   | 26 |
| Figura 3.6 Matriu de bonificació.....                         | 25 |
| Figura 3.7 Matriu de bonificació amb cantonada esquerra ..... | 27 |
| Figura 3.8 Matriu de penalització .....                       | 28 |
| Figura 3.9 Matriu de bonificació tauler aleatori.....         | 26 |
| Figura 3.10 Matriu penalització tauler aleatori .....         | 26 |
| Figura 4.1 Fases del MCTS.....                                | 32 |
| Figura 4.2 Pseudocodi del MCTS .....                          | 33 |
| Figura 4.3 Pseudocodi funció UCTValor.....                    | 36 |
| Figura 5.2 MCTS + Heurística diferents temps .....            | 43 |
| Figura 5.3 Número de victòries de tots els algorismes.....    | 44 |
| Figura 6.1 Model Vista Controlador .....                      | 46 |
| Figura 6.2 Diagrama de Classes.....                           | 47 |

## Índex de taules

|                                                |    |
|------------------------------------------------|----|
| Taula 1.1 Planificació.....                    | 10 |
| Taula 2.1 Tipus de joc.....                    | 17 |
| Taula 3.1 Resultats poda alfa-beta .....       | 24 |
| Taula 5.1 Tauler 4x4 Minimax.....              | 38 |
| Taula 5.2 Tauler 6x6 Minimax.....              | 38 |
| Taula 5.3 Tipus de finals.....                 | 39 |
| Taula 5.4 Mitjana de torns minimax .....       | 40 |
| Taula 5.5 Comparació entre heurístiques.....   | 41 |
| Taula 5.6 Mitjana de torns heurístiques .....  | 42 |
| Taula 5.7 MCTS+Heurística diferents temps..... | 43 |
| Taula 5.9 Comparació entre algorismes.....     | 44 |



## CAPÍTOL 1 : INTRODUCCIÓ

### 1.1 Justificació

Quan parlem de jocs de taula en trobem de molts tipus, de més complexos i de més simples. Aquest projecte es centrarà en el joc “Tebeș”, (que és un tipus de joc abstracte amb una certa complexitat), com a pretext per a fer una comparació entre diversos algorismes i heurístiques.

Així doncs, es crearà una aplicació amb diverses opcions, que ens facilitaran la tasca a realitzar en relació a l’anàlisi de les diferents solucions.

### 1.2 Objectius

L’objectiu d’aquest projecte és fer un estudi comparatiu de diverses tècniques dins del món de la intel·ligència artificial per tal d’aconseguir que una màquina pugui resoldre els moviments correctes que la duguin a guanyar el joc del Tebeș i que pugui resultar un repte per a un jugador humà.

L’estratègia ha estat començar per una solució inicial, que s’ha anat millorant per aconseguir una el més correcte possible, tenint en compte els recursos de què disposem.

Recaptarem informació tot fent un estudi comparatiu entre les diferents solucions. Per a assolir aquest objectiu farem jugar la màquina contra ella mateixa amb diferents algorismes, ja que així podrem fer les estadístiques de partides guanyades o perdudes, fet que ens donarà una idea de la qualitat de la solució i ens ajudarà a comparar les diferents tècniques d’una manera més precisa.

Per a fer això, implementarem una aplicació que oferirà diferents possibilitats, per exemple, la de poder jugar contra un contrincant humà, i la de poder fer simulacions de partides amb diferents configuracions per a poder extraure’n informació rellevant.

### 1.3 Planificació del projecte

|  | <b>Fita</b>                                | <b>Inici</b> | <b>Fi</b>  |
|--|--------------------------------------------|--------------|------------|
|  | Pla de treball (PAC1)                      | 26-02-2015   | 11-03-2015 |
|  | Memòria                                    | 26-02-2015   | 03-06-2015 |
|  | Introducció                                |              |            |
|  | Justificació                               | 11-03-2015   | 13-03-2015 |
|  | Objectius                                  | 11-03-2015   | 13-03-2015 |
|  | Planificació del projecte                  | 13-03-2015   | 15-03-2015 |
|  | Descripció dels productes obtinguts        | 30-05-2015   | 03-06-2015 |
|  | Descripció dels capítols                   | 30-05-2015   | 03-06-2015 |
|  | El joc Tebes                               |              |            |
|  | Descripció de les regles del joc           | 15-03-2015   | 16-03-2015 |
|  | Avaluació de la complexitat                | 17-03-2015   | 19-03-2015 |
|  | Avaluació del tauler                       | 20-03-2015   | 26-03-2015 |
|  | Implementació                              |              |            |
|  | Anàlisi (diagrama de classes)              | 26-03-2015   | 28-03-2015 |
|  | Implementació de l'estructura (MVC)        | 28-03-2015   | 04-04-2015 |
|  | Implementació del funcionament bàsic       | 28-03-2015   | 08-04-2015 |
|  | Millora de la implementació arbre de cerca | 22-04-2015   | 30-04-2015 |
|  | Avaluació Heurística                       | 22-04-2015   | 06-05-2015 |
|  | Implementació del funcionament (part 2)    | 22-04-2015   | 06-05-2015 |
|  | Implementació d'altres estratègies         | 01-05-2015   | 25-05-2015 |
|  | Estudi comparatiu                          | 06-05-2015   | 20-05-2015 |
|  | Presentació                                |              |            |
|  | Creació de diapositives                    | 20-05-2015   | 03-06-2015 |
|  | Enregistrament d'un vídeo ( presentació )  | 20-05-2015   | 03-06-2015 |

Taula 1.1 Planificació

## 1.4 Productes obtinguts

Els productes obtinguts són, la memòria amb l'estudi comparatiu de les diferents tècniques, l'aplicació i un vídeo de presentació.

**Memòria:** La memòria consta de diferents apartats on s'introdueixen les característiques de les tècniques emprades, els resultats de les estadístiques i les conclusions pertinents.

**Aplicació:** L'aplicació està dividida en dues parts, d'una banda es pot jugar contra la màquina i de l'altre es pot simular partides màquina contra màquina. Ofereix quatre opcions diferents de taulers, en relació amb la mida, 4x4 o 6x6, i en relació amb el tipus, aleatori o clàssic.

**Vídeo de presentació:** On s'expliquen i es mostren les diapositives que resumiran el treball dut a terme.

## 1.5 Descripció dels capítols de la memòria

Aquest treball final de grau consta d'un cos de 7 capítols, que van des d'una introducció del projecte i els detalls d'aquest, fins a unes conclusions, passant per el tema que ens interessa que són les diferents tècniques emprades i les seves característiques, aplicades a la resolució del joc Tebes.

### Capítol 1: Introducció

En aquest capítol fem una breu introducció de què serà aquest projecte, tot descrivint tant la temporització d'aquest com els productes i resultats obtinguts.

### Capítol 2: El joc Tebes

El capítol 2 es centra en la descripció del joc escollit, en primer lloc, des d'un punt de vista genèric i, en segon lloc, des d'un enfocament més tècnic gràcies a l'anàlisi de la tipologia i la complexitat d'aquest.

### Capítol 3: Solució Inicial

En el capítol 3, descriurem la primera estratègia a tenir en compte que serà, l'algorisme de cerca minimax amb poda alfa beta, l'heurística que el complementa i una breu explicació de com ha estat implementada.

### Capítol 4: Altres estratègies considerades

En el capítol 4 ens ocupem de la descripció d'altres estratègies utilitzades per a comparar amb la solució inicial, com són el MCTS i el MCTS amb heurística. També introduïm la segona versió de l'heurística.

### Capítol 5: Anàlisis dels resultats

En el capítol 5 es desplega un estudi comparatiu de les diferents solucions, tot mostrant-ne els resultats i les conclusions d'aquest.

### Capítol 6: Implementació

Aquest capítol està reservat a una breu explicació de les eines i els mètodes per a la implementació de la nostra aplicació.

### Capítol 7: Conclusions

En aquest últim capítol explicarem les conclusions finals a què ens ha conduït el nostre projecte.

## CAPÍTOL 2 : EL JOC TEBES

### 2.1 El joc tebes

El joc Tebes és un joc de taula català editat el 2008 i dissenyat per Jordi Rossich, que es classifica dins la tipologia de jocs que s'anomenen jocs abstractes. Quan ens referim a jocs abstractes tractem sobre aquells jocs que no presenten un tema o una ambientació, és a dir, les fitxes i caselles no tenen cap similitud amb alguna cosa real o imaginària. Tot i això, la idea del joc està inspirada en la llegenda grega dels Set contra Tebes.

El joc és per a dos jugadors, amb un marge d'edat de 8 en endavant i la durada aproximada de la partida és de 30 minuts. Es pot jugar en un tauler de 4x4 si el jugador que hi juga és iniciat o bé, en un de 6x6 si és avançat. En el nostre projecte ens basarem fonamentalment en el tauler 6x6, tot i que també implementarem el tauler 4x4, ja que ens ajudarà a entendre millor la mecànica del joc gràcies. No obstant això, ambdues versions del joc són idèntiques en regles i només canvia la complexitat de cadascuna.

El joc consta de diversos components:

36 peces quadrades per a construir el tauler: 18 blanques i 18 negres.

36 fitxes rodones per a col·locar sobre del tauler: 18 blanques i 18 negres.



*Figura 2.1 Caselles del Tebes*



*Figura 2.2 Tauler del Tebes*



*Figura 2.3 Fitxes del Tebes*

El tauler es crea amb les peces quadrades, pot ser un tauler estàndard com en les dames o els escacs, o bé un tauler qualsevol, aleatori, com el que es mostra en la imatge. Els taulers poden ser aleatoris, però han de tenir sempre el mateix nombre de caselles blanques i negres.

## 2.2 Regles del joc

Per a explicar les regles ens centrarem en el tauler estàndard però aquestes seran igualment vàlides per altres configuracions de taulers.

Un cop disposades les llosetes de tauler, es posen les fitxes d'un color a les tres files superiors i les del altre color a les tres files restants.

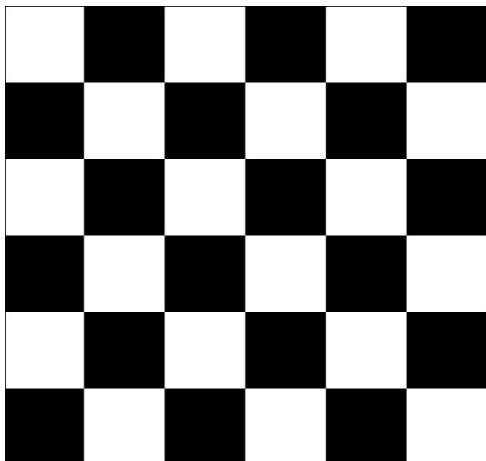


Figura 2.4 Tauler estàndard

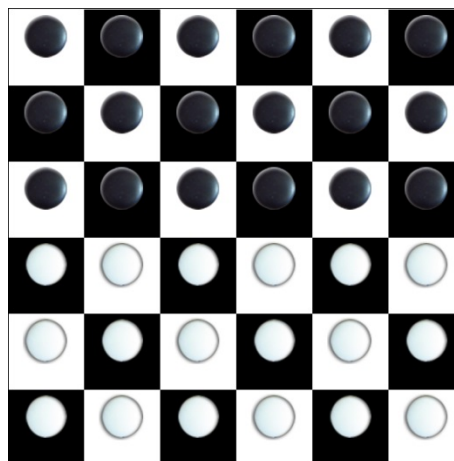


Figura 2.5 Tauler estàndard amb fitxes

L'objectiu del joc és arribar amb alguna de les fitxes pròpies a les dues **cantonades** de la última fila del contrincant.

**Important!** Quan un jugador arriba a una de les dues cantonades, guanya un torn i disposa d'un moviment addicional! Aquest torn extra, però, només es pot obtenir una vegada per jugador.

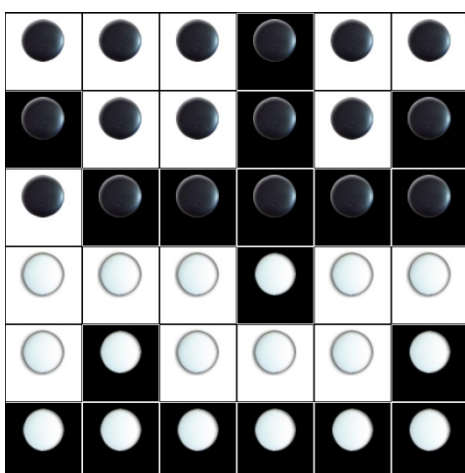


Figura 2.6 Tauler aleatori 1

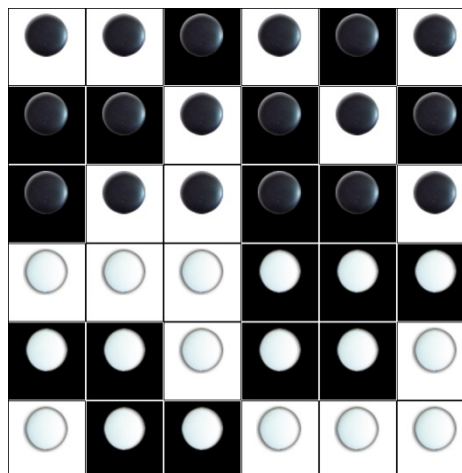


Figura 2.7 Tauler aleatori 2

Com es pot veure en les imatges, els taulers poden ser generats de manera aleatòria cosa que fa que el joc canvi depenent de la col·locació dels colors.

Per tal d'arribar a les cantonades cada jugador farà un moviment per torn. Els moviments han de complir totes aquestes regles:

- Quan es mou una fitxa es pot avançar una o dues caselles.
- Les fitxes només es poden moure cap endavant, cap als costats i en diagonal, mai cap enrere.
- Si es mouen dues caselles, les fitxes no poden saltar per sobre d'una fitxa del contrincant, només poden saltar una fitxa del color propi, o bé per sobre d'una casella buida.
- Una fitxa mai pot acabar el moviment en una casella on ja es troba instal·lada una fitxa del propi color, només pot col·locar-se en una casella buida o amb fitxes del color contrari, sempre que compleixi les normes per matar la fitxa.

[no es pot col·locar a la casella de la fitxa  
blanca però sí la pot saltar]

[No pot matar la fitxa negra i tampoc la pot saltar]

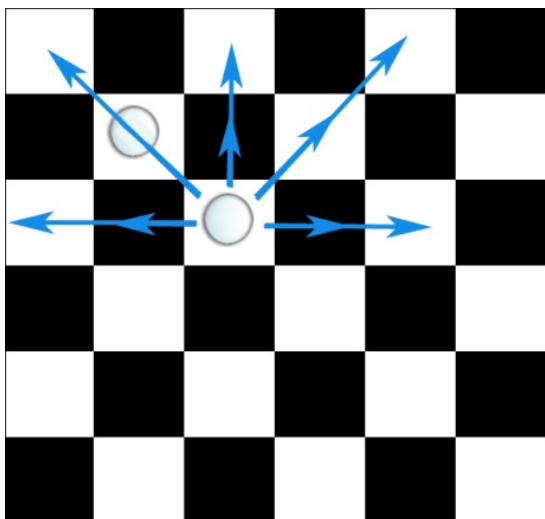


Figura 2.8 Moviments possibles

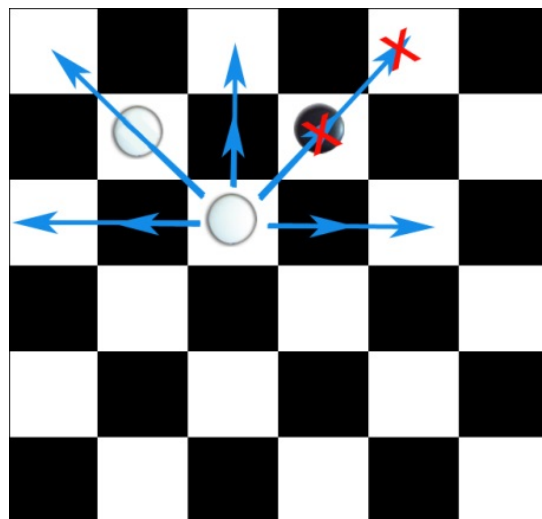


Figura 2.9 Moviment incorrecte

- Per a matar fitxes del adversari hem de tenir en compte el color de la casella on està originalment la fitxa que mourem i que farem servir per matar la fitxa del rival:
  - Si la fitxa es troba en una casella del **nostre color**: podrem matar fitxes que estiguin situades exactament a **2 posicions** de distància.

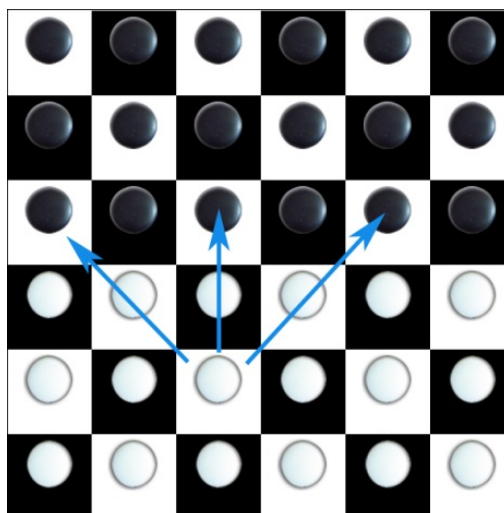


Figura 2.10 Matar a dues posicions

- Si la fitxa es troba a una casella del **color de l'adversari**; podrem matar fitxes que estiguin exactament a **1 posició** de distància.

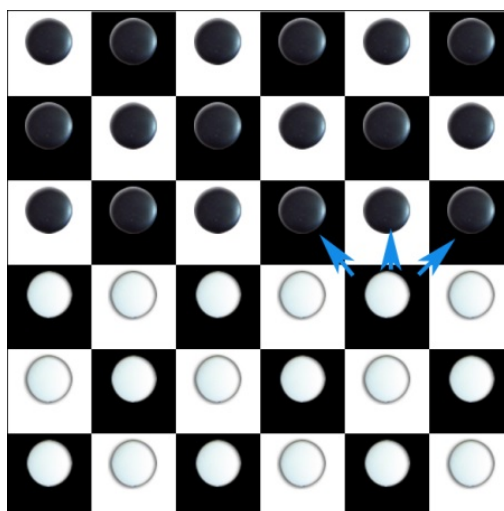


Figura 2.11 Matar a una posició

La partida finalitza quan un jugador té dues de les seves fitxes a les dues cantonades del rival. També es considera el final del joc quan un dels dos jugadors té menys de dues fitxes o no es pot moure, és a dir, no té més moviments possibles.



## 2.3 Classificació del tipus de joc

Els diferents tipus de jocs els podríem classificar amb la següent taula:

|                       | Determinístics              | Estocàstics                          |
|-----------------------|-----------------------------|--------------------------------------|
| Informació Perfecta   | Escacs, dames, go , othello | Backgammon, monopoly                 |
| Informació Imperfecta | Battleship Kriegspiel       | Bridge, poker, scrabble, nuclear war |

*Taula 2.1 Tipus de joc*

Un joc amb **informació perfecta** seria quan els jugadors tenen coneixement de tot el que passa i de tots els components que ho componen des del principi fins al final.

El joc Tebes, tot i gaudir de la opció de generar un tauler aleatori, és un joc de informació perfecta, ja que coneixem el tauler i la situació de les fitxes abans d'iniciar el joc.

Pel que fa a l'**atzar**, el nostre joc no depèn massa d'ell, en el tauler estàndard.

Els jocs sense cap mena d'element aleatori s'anomenen **deterministes**. En el nostre joc l'atzar es mínim, pel que fa als taulers aleatoris, aquests sí que gaudeixen d'un cert atzar perquè els moviments possibles depenen de com estan col·locades les caselles de diferents colors que formen el tauler. Aquest fet, però, no afecta a la classificació del nostre joc com a determinista, ja que si hi juguem amb un tauler aleatori, aquest ja està generat i és conegut un cop s'ha iniciat la partida, per tant, no intervé l'atzar.

Així doncs, podríem dir que el nostre joc formaria part de la categoria **determinista** amb **informació perfecta**, cosa que té certs avantatges a l'hora d'implementar-lo.

A més, existeixen altres factors que cal tenir en compte a l'hora d'implementar l'aplicació del joc Tebes. Un dels més importants és la complexitat del joc, que en avaluar-la ens trobem amb un arbre de cerca bastant gran. Posteriorment, analitzarem aquesta qüestió amb més detall, així com les tècniques per a reduir la necessitat d'avaluar tots els nodes, amb el que anomenem poda alfa-beta.

## 2.4 Avaluació de la complexitat

Quan ens referim a complexitat, parlem de dos factors, el factor de ramificació de l'arbre que es tracta del nombre de fills de cada node i la profunditat de l'arbre del joc, és a dir, la profunditat de la cerca màxima.

Al factor de ramificació l'anomenarem  $b$  i a la profunditat de l'arbre  $d$ .

Així doncs passem a calcular el factor de ramificació que com que no es constant haurem de fer la mitjana de tots els possibles moviments per torn de diferents torns.

El valor del factor de ramificació obtingut per al joc Tebes és de 35.

Pel que fa a la profunditat mitjana, fem la mitjana del nombre de torns de diverses partides i ens surt unes 30 jugades per partida, aquesta seria la nostra profunditat de cerca màxima.

Així doncs obtenim la complexitat de l'arbre del nostre joc, que es:

$O(b^d)$  ---> On  $b$  es = 35 i  $d$  = 30.

Per tant, ens queda una complexitat de  $O(35^{30})$  que ens dona:

$$O(35^{30}) = 2.099139643 * 10^{46}$$

## 2.5 Representació del joc

Ara passem a definir el model del joc. Tenint en compte la dimensió del tauler, les matrius que defineixen el nostre espai poden ser de  $4 \times 4$  o bé de  $6 \times 6$ , aquest espai estarà representat amb les coordenades  $x$  i  $y$  que es referiran als dos índex de les matrius.

$D$  la dimensió del tauler de Tebes (pot ser 4 o 6)

$\mathbb{C} = \{b, n\}$  els dos colors de Tebes (blanc i negre)

Definim el model  $\mathcal{M}$  (que representa l'estat d'un joc de Tebes) com la tupla:

$\mathcal{M} := \{C, F, t, d\}$  on :

$C := (c_{i,j})_{D \times D}$  amb  $c_{i,j} \in \mathbb{C}$ , és la matriu dels colors de les caselles

$F := (f_{i,j})_{D \times D}$  amb  $f_{i,j} \in \mathbb{C} \cup \{\lambda\}$ , és la matriu de fitxes.  $\lambda$  representa una posició sense fitxa

$t \in \mathbb{C}$  indica quin jugador (blanc o negre) té el torn

$d$  és un predicat  $\forall j \in \mathbb{C}: d(j) \Leftrightarrow$  el jugador  $j$  disposa del doble torn

Per a una més fàcil definició dels moviments de manera teòrica i per a la representació de partides, aquests estaran definits per un origen i un destí, sent aquests dos enters representats tenint en compte la següent numeració.

|    |    |    |    |
|----|----|----|----|
| 0  | 1  | 2  | 3  |
| 4  | 5  | 6  | 7  |
| 8  | 9  | 10 | 11 |
| 12 | 13 | 14 | 15 |

Figura 2.12 Representació tauler de  $4 \times 4$

|    |    |    |    |    |    |
|----|----|----|----|----|----|
| 0  | 1  | 2  | 3  | 4  | 5  |
| 6  | 7  | 8  | 9  | 10 | 11 |
| 12 | 13 | 14 | 15 | 16 | 17 |
| 18 | 19 | 20 | 21 | 22 | 23 |
| 24 | 25 | 26 | 27 | 28 | 29 |
| 30 | 31 | 32 | 33 | 34 | 35 |

Figura 2.13 Representació tauler de  $6 \times 6$

Així doncs una partida estaria definida com en l'exemple següent:

**B:** 19 – 14 , **N:** 9 – 21 , **B:** 28 – 21, **N:** 11 - 21

## CAPÍTOL 3 : SOLUCIÓ INICIAL

### 3.1 Algorisme de cerca ( minimax )

El minimax és un algorisme que ens permet trobar el moviment òptim suposant que l'adversari sempre juga el millor moviment possible. En aquest algorisme es representen els torns dels dos jugadors: un anomenat MAX, que es tractarà del propi jugador que està decidint el moviment i que maximitzarà les possibilitats de victòria; i un anomenat MIN que representarà el jugador oponent i minimitzarà les possibilitats de vèncer.

L'algorisme minimax es basa en desplegar l'arbre de jocs possibles (la qual cosa es fa mitjançant un recorregut en profunditat) fins a trobar nodes terminals, a partir d'aquí s'avalua la solució i s'indica si és victòria, derrota o empat. Un cop avaluada la solució es propaga cap al node inicial la millor opció.

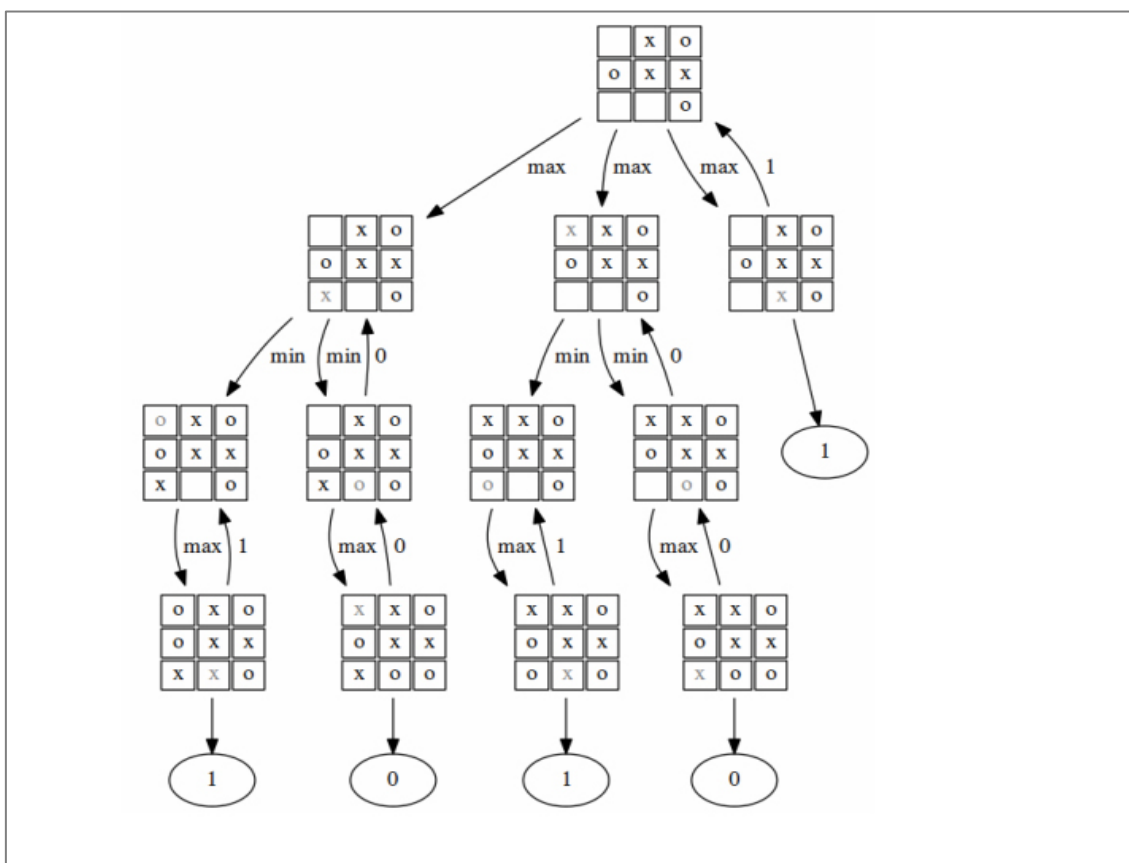


Figura 3.1 Arbre Minimax

El diagrama mostra un exemple de fragment d'arbre generat per minimax pel joc "tres en ratlla". Per aquest joc, arribar fins als nodes terminals de l'arbre és possible, perquè la seva complexitat és petita, però per a jocs amb més complexitat, com el tractat en aquest projecte, això és inviable, ja que com mencionem a la secció 2.4 la complexitat de l'arbre es de  $O(35^{30})$ .

És per aquesta raó que una funció heurística és imprescindible per a avaluar els nodes, perquè a pesar de la millora de la poda alfa-beta la complexitat de l'arbre és encara massa gran com per a poder-lo desplegar tot.

## Pseudocodi minimax

```
funció minimax(jugador,Tauler):  
  
    Si (node es terminal)  
        retornar guanyador  
    //si el node és fulla o bé hem trobat solució tornem el valor  
  
    nodes_fills=CalcularMoviments(jugador,Tauler);  
    //calculem tots els moviments possibles del jugador actual  
  
    Si (jugador==MAX)  
        //si és el jugador MAX agafem el valor més alt  
  
        Per cada node de nodes_fills:  
            valor=Maxim(valor,minimax(MIN,Tauler));  
            fiper  
  
        retorna valor;  
    fisi  
  
    Si(jugador==MIN)  
        //Si és el jugador MIN agafem el valor més petit  
  
        Per cada node de nodes_fills:  
            valor=Minim(valor,minimax(MAX,Tauler));  
            fiper  
  
        retorna valor;  
    fisi  
fifuncio
```

*Figura 3.2 Pseudocodi Minimax*

### 3.2 Poda Alfa-Beta

La poda alfa-beta ens permet millorar l'eficiència de l'algorisme minimax. La idea bàsica és no examinar els nodes de l'arbre que sabem que no ens aportaran informació que modifiqui la solució final.

Com es pot observar en la imatge inferior, en la última línia que pertany al jugador MAX fem una poda de la branca del node 5, ja que  $\alpha \geq \beta$ , és a dir que 5 és més gran que 4 i com que el jugador MIN tria el mínim valor no té sentit examinar nodes més grans que el de 4.

Amb la branca del node 5 passa el mateix,  $\alpha \geq \beta$  per tant es fa poda, com que el jugador MAX (node arrel) tria el valor més gran, al tenir un 5 en el tercer node del jugador MIN es faria poda, degut a que al triar el jugador min el valor més petit no té sentit seguir examinant ja que qualsevol numero inferior a 5 serà inferior a 6, per tant no serà triat per el node MAX (node arrel).

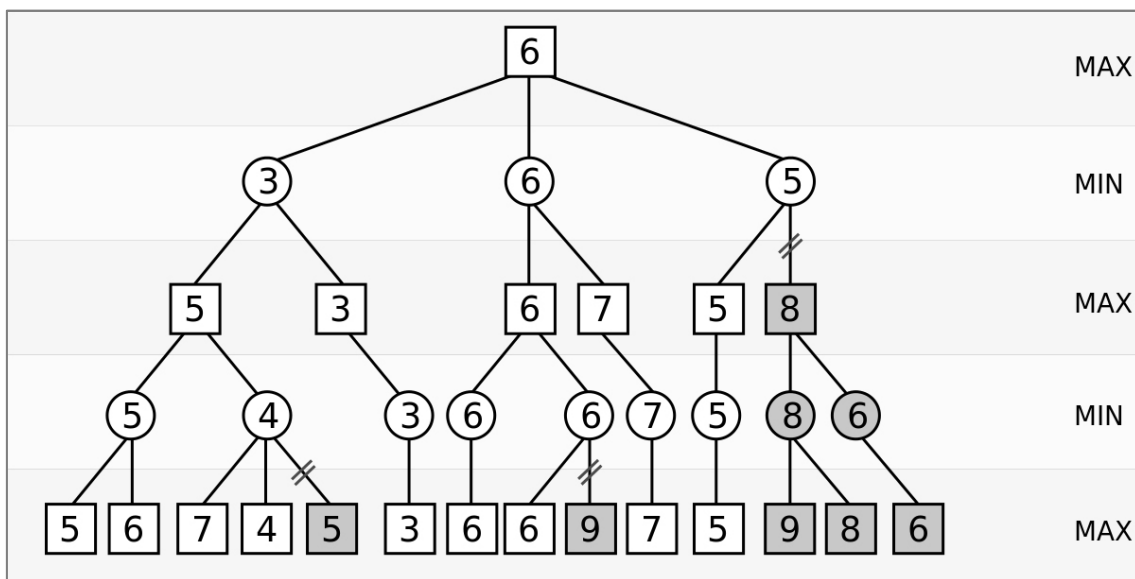


Figura 3.3 Poda Alfa-Beta

Per a entendre millor aquesta idea, tot seguit afegim l'algorisme en pseudocodi del minimax amb poda alfa-beta.

## Pseudocodi minimax amb poda alfa-beta

```

funció minimaxAlphaBeta(jugador,Tauler,alfa,beta,profunditat):

    Si (profunditat==0 || esSolucio(Tauler)):
        valor=EvaluacioTauler(Tauler)
        retornar valor
    //si hem arribat a la profunditat màxima o bé
    // hem trobat solució tornem el valor
    Fisi

    nodes_fills=CalcularMoviments(jugador,Tauler);
    //calculem tots els moviments possibles

    Si (jugador==MAX)
    //si és el jugador MAX agafem el valor més alt

        Per cada node fill:
            valor = minimax(MIN,Tauler,alfa,beta,profunditat-1);

            si(valor > alfa)
                alfa=valor;
                //li assignem el valor més alt a alfa
            fisi

            si(alfa >= beta)
                retorna alfa;// fem poda
            fisi
        fiper
        retorna alfa;
    fisi

    Si(jugador==MIN)
    //Si és el jugador MIN agafem el valor més petit

        Per cada node fill:
            valor = minimax(MAX,Tauler,alfa,beta,profunditat-1);

            si(valor < beta)
                beta=valor;
                //li assignem el valor més petit a beta
            fisi

            si(alfa >= beta)
                retorna beta; // fem poda
            fisi
        fiper
        retorna beta;
    fisi

    fifuncio

```

Figura 3.4 Pseudocodi Minimax amb poda Alfa-Beta

Així doncs la poda alfa-beta suposa una gran millora ja que recordant la complexitat del joc que expliquem en l'apartat 2.4, amb la poda s'aconsegueix una gran reducció.

Es pot afirmar que la quantitat de nodes visitats de l'arbre seria:

$$O(35^{30/2}) = 1.448840793 * 10^{23}$$

És a dir, és el mateix que tenir un factor de ramificació de  $\sqrt{b}$ .

La millora mitjana de la poda alfa-beta està estimada en un 30%.

Podem observar un anàlisi de la reducció de nodes avaluats gràcies a aquesta optimització en la següent taula.

### 3.2.1 Avaluació dels resultats de l'algorisme minimax amb poda alfa-beta

| Profunditat | Nodes avaluats –<br>Alfa/Beta | Temps    | Nodes teòrics sense<br>poda |
|-------------|-------------------------------|----------|-----------------------------|
| 2           | 629                           | ≈0.0015s | $35^2 = 1225$               |
| 4           | 66.023                        | ≈0.15s   | $35^4 = 1.500.625$          |
| 6           | 10.300.195                    | ≈19s     | $35^6 = 1.838.265.625$      |

Taula 3.1 Resultats poda alfa-beta

La reducció  $O(b^{d/2})$  que mencionem però, es la mitjana teòrica que s'espera. En el nostre joc en concret, tenint en compte l'heurística i degut a que només considerem 3 casos de profunditat, no podem extreure regles generals.



### 3.3 Funció d'avaluació

Pel que fa a la funció heurística, aquesta avaluarà l'estat del joc depenent per a quin dels dos jugadors és calculada, seguidament resumim les característiques d'aquesta funció.

$H(m,j)$  és una funció heurística sobre un estat de la partida  $m$  i un jugador  $j$ .

Si  $H(m,j) = 0$  ambdós jugadors tenen les mateixes possibilitats de guanyar.

$$-MAXVALOR \leq H(m,j) \leq MAXVALOR$$

Si  $H(m,j) = -MAXVALOR$  el jugador  $j$  ha perdut la partida.

Si  $H(m,j) = MAXVALOR$  el jugador  $j$  ha guanyat la partida.

L'heurística utilitzada es basarà en dues idees bàsiques:

- S'avaluarà, en primer lloc, com de bona és es una configuració del tauler per al jugador Max, ajudant-nos amb les matrius de pesos que posteriorment explicarem; i, en segon lloc, com de dolenta és la configuració del tauler per al mateix jugador tenint en compte les fitxes del adversari. Un cop fet això, a la puntuació positiva del nostre jugador li restarem la puntuació negativa.
- La segona idea bàsica de la nostra heurística és que la quantitat de peces que es tenen donen una bonificació o penalització a l'hora d'avaluar el tauler, per aquesta raó sumem un valor, en el nostre cas 10 per cada fitxa, si la fitxa és del nostre jugador serà positiu, si és de l'adversari serà negatiu.

En **resum**, els dos factors que indiquen les possibilitats de victòria d'un jugador són el nombre de fitxes (com més millor) i com estan situades aquestes en el tauler.

Així doncs, per a implementar la primera idea base de la nostra funció d'avaluació farem ús d'unes matrius de pesos que varien segons la disposició del tauler.

Aquestes matrius seran dues, la mBonificacio que seran els pesos que sumarem a la nostra heurística i la mPenalitzacio que seran els pesos que restaran a la nostra heurística ja que és l'avantatge de l'adversari.

Podem descriure la nostra funció heurística com a :

$$\sum_{i=1..D} \sum_{j=1..D} ((mBonificacio[i,j] + 10 * fitxaNegra?[i,j]) - (mPenalitzacio[i,j] + 10 * fitxaBlanca?[i,j]))$$

Així doncs, la funció d'avaluació farà el següent:

- Sumar els valors positius per a les fitxes de la màquina (negres) amb la matriu de Bonificació.
- Restar valors de les fitxes de l'oponent (matriu de penalitzacions).
- Tenir en compte el valor de la diferència de peces.
- Si s'arriba a solució (és a dir, té les dues cantonades) li assignem el valor màxim, per a que aquesta solució sigui la triada.
- Si el contrincant arriba a solució li assignem el valor mínim.

### Pseudocodi de la funció d'avaluació

Per a entendre millor l'anterior explicació adjunto el pseudocodi que correspondria a la funció d'avaluació, on tenim una classe mapa que és la que conté les matrius (de bonificació i de penalització).

```

funció evaluar(Tauler):

    valor = valornegatiu = 0
    per (x=0; x < mida_tauler ; x++)
        per (y=0; y < mida_tauler ; y++)

            si (Tauler[x][y].getFitxa()==Negra)
                //si la fitxa és negra
                valor = valor+mapa.getVal(x,y,mBonificacio)+10
                //agafem valor matriu 1 (bonificació)
                //agregem +10 perquè hi ha fitxa
            sino si (Tauler[x][y].getFitxa()==Blanca)
                //si la fitxa és blanca
                valornegatiu = valornegatiu
                    + mapa.getVal(x,y,mPenalitzacio)+10
                //agafem valor matriu 2 ( penalització )
                //incrementem quantitat fitxes blanques
            fisi
        fiper
    fiper

    retorna valor-valornegatiu //Restem penalitzacions al valor
fifunció

```

*Figura 3.5 Pseudocodi funció avaluació*

Posteriorment explicarem com són i com es generen els pesos de la matriu de pesos.

### 3.4 Mapa de pesos

Una vegada descrita la funció d'avaluació, explicarem com es generen els pesos de la matriu de pesos.

#### Matriu de bonificació:

|    |    |    |    |    |    |
|----|----|----|----|----|----|
| B5 |    |    |    |    | B5 |
|    |    |    |    |    |    |
|    |    | B6 |    | B6 |    |
| B6 |    | B3 | B2 | B6 | B4 |
|    | B2 |    | B6 |    | B2 |
| B1 | B3 | B3 | B2 | B2 | B1 |

Figura 3.6 Matriu de bonificació

|    |  |    |    |    |    |
|----|--|----|----|----|----|
| B5 |  |    |    |    | B5 |
|    |  |    |    |    |    |
|    |  | B6 |    | B6 |    |
|    |  |    | B2 | B6 | B4 |
|    |  |    | B6 |    | B2 |
| B1 |  |    | B2 | B2 | B1 |

Figura 3.7 Matriu de bonificació amb cantonada esquerra

Els pesos de la matriu de pesos de bonificació aniran ordenats jeràrquicament com s'indica a continuació:

$$\text{MAXVALOR} > B1 > B2 > B3 > B4 > B5 > B6 > 0$$

Per començar, tindrem en compte per a l'exemple que la màquina és les negres en la posició superior, òbviament donarem el pes màxim (B1) a les cantonades del contrincant, com es pot observar en la imatge. Tenint en compte el color de la màquina, ara valorem amb el segon pes (B2) les caselles que donen accés a la cantonada. Com juguem amb les negres, tindrem accés des de les caselles de diferent color, ja que aquestes poden matar a un requadre de distància. Per tant, donem el valor (B2) a les caselles blanques adjacents a les cantonades. Assignem valor (B2), també, a les caselles negres a dos requadres de distància de les cantonades, es a dir, als accessos a aquestes.

Per últim, valorem amb el pes (B3) les caselles que tenen accés a les caselles que tenen valor (B2), és a dir, accés a les caselles que tenen accés a les cantonades.

Aquesta matriu pot variar a mesura que avança el joc, per exemple, quan la màquina arriba a una cantonada, les caselles adjacents a aquesta es posen sense pes, ja que no interessa que es tingui en compte aquesta cantonada.

**Matriu de penalització:**

|    |    |    |    |    |    |
|----|----|----|----|----|----|
| P1 | P3 | P3 | P2 | P2 | P1 |
|    | P3 |    |    |    | P3 |
|    |    |    | P3 | P3 |    |
|    |    |    |    |    |    |
|    |    |    |    |    |    |
|    |    |    |    |    |    |

Figura 3.8 Matriu de penalització

Els pesos de la matriu de pesos de penalització aniran ordenats jeràrquicament com s’indica a continuació:

$$\text{MAXVALOR} > P1 > P2 > P3 > 0$$

La idea de la matriu de penalització és semblant a l’anterior matriu, però a l’inrevés, es posen pesos a les cantonades pròpies i als accessos a aquestes, però, a diferència del mapa anterior, aquests pesos es resten, és a dir:

Tenim la variable valor que la modifiquem d’aquesta manera:

Recorrent totes les caselles amb fitxes negres, sumem a la variable valor els pesos de la matriu de bonificacions, només on hi ha fitxes negres.

Posteriorment, recorrem les caselles amb fitxes blanques ( l’oponent ) i restem els valors dels pesos de la matriu de penalització, és a dir, les solucions on hi hagi fitxes blanques a les posicions que tenen P1 o P2 restaran aquest numero a la variable valor, per tant seran solucions menys òptimes.

Així doncs, un exemple de matriu de pesos per a un tauler aleatori es el següent:

|    |    |    |    |    |    |
|----|----|----|----|----|----|
| B5 |    |    |    |    | B5 |
|    |    |    |    |    |    |
|    |    | B6 |    |    |    |
| B6 |    | B2 | B3 | B6 | B6 |
| B2 | B2 | B6 |    |    |    |
| B1 | B3 | B2 | B3 | B2 | B1 |

Figura 3.9 Matriu de bonificació (tauler aleatori)

|    |    |    |    |    |    |
|----|----|----|----|----|----|
| P1 | P3 | P2 | P3 | P2 | P1 |
| P2 |    |    |    | P2 |    |
|    | P2 | P2 | P2 | P2 |    |
|    |    |    |    |    |    |
|    |    |    |    |    |    |
|    |    |    |    |    |    |

Figura 3.10 Matriu de penalització( tauler aleatori)

## 3.5 Millors i adaptacions

### 3.5.1 Doble torn

El doble torn és un detall a tenir en compte a l'hora de fer simulacions de partides o recórrer l'arbre de cerca del minimax.

Per aquesta raó, el nostre model emmagatzema i té constància de quan ha succeït un doble torn, és a dir, quan el jugador actual accedeix a una cantonada per primera vegada, i, per tant, té un torn addicional.

Aquest fet es soluciona executant dues vegades seguides el minimax amb el jugador MAX o MIN, depenent de qui sigui el jugador que hagi aconseguit el doble torn.

Tenint en compte aquest fet, observem que aleshores en l'arbre, comparem solucions normals, és a dir, de max-min-max-min, amb solucions amb doble torn, això ens pot dificultar el fet d'aconseguir el millor moviment possible.

Per aquesta raó, adaptem el minimax a aquesta situació, no decremant el nivell de la pròxima crida del mínimax quan te lloc el doble torn. Aquest fet ens crea un arbre asimètric en quant a profunditat, però ens dona una millor aproximació al millor moviment possible.

### 3.5.2 Nivells de l'algorisme minimax i aleatorietat

Tenint en compte els nivells del minimax, s'observa que, en el cas que dues victòries estiguin en diferents profunditats de l'arbre aquestes aconsegueixen el mateix valor. Per conseqüència, si ens trobem a un moviment de la victòria però també hi ha una victòria possible en tres moviments, podia succeir que l'algorisme escollís la segona i no decidís guanyar en el següent torn a causa de la coincidència de valor heurístic d'ambdues solucions.

Aquest fet, queda solucionat tenint en compte el nivell de profunditat d'una solució, així doncs, en el valor de l'heurística d'un moviment se li afegeix la informació del nivell de profunditat on és aconseguit. Posteriorment, quan una solució té el mateix valor que una altra (és victòria) prioritzem aquella que és immediata.

D'altra banda, pot donar-se el cas que diversos moviments aconsegueixin tots el mateix valor heurístic i el mateix nivell, així doncs, en una primera solució s'escollia el primer, però aquest fet tenia com a conseqüència la no aleatorietat de les partides quan es feien simulacions d'aquestes (eren totes iguals al enfrontar diversos minimaxs).

Per a solucionar aquesta situació, afegim un número aleatori entre 0 i 1 (no inclosos), aquest número serà massa petit com per suposar una diferència entre valors diferents de les heurístiques de diverses solucions, però ens servirà per desempatar en el cas d'heurístiques amb el mateix valor.

En afegir un nombre aleatori entre les solucions amb el mateix valor heurístic, s'aconsegueix que el minimax gaudeixi d'aleatorietat i, per tant, que les partides siguin diferents, fet que ajuda a dificultar la predictibilitat dels moviments de l'adversari.

### 3.5.3 Evitar cicles

Una situació que cal evitar en la simulació de partides i en la creació de l'arbre de cerca del minimax, són els cicles.

Els cicles es produeixen quan es torna a un estat del joc ja visitat, és a dir, quan els dos jugadors executen els mateixos moviments repetits una i altra vegada.

Aquest fet ens podia conduir a bucles infinits a l'hora de simular una partida o a moviments totalment innecessaris.

Per a evitar aquest fet, hem de comprovar que no s'ha passat per un estat del joc ja visitat, mantenint un registre dels estats passats. Per evitar tenir registre dels estats complets de tota la partida farem dues millores:

- 1- Identificar els taulers de forma única amb un valor numèric.
- 2- Enregistrar només els estats passats que són susceptibles de ser repetits des de l'estat actual.

Per a identificar l'estat del joc, durant una mateixa partida, només ens caldrà tenir en compte la informació que canvia durant la partida (informació dinàmica). Aquesta informació inclou:

- La disposició de les fitxes
- Quin jugador té el torn
- Quins jugadors disposen de doble torn

Identificarem l'estat del tauler amb la següent fórmula:

$$id(M) = 8 \left( \sum_{i=1..D} \sum_{j=1..D} 3^{D*(i-1)+(j-1)} v f_{i,j} \right) + 4vt + 2vdt(n) + vdt(b)$$

$D$  es la dimensió del tauler ( $D = 4$  o  $6$ )

$v f_{i,j} = 0 \Leftrightarrow$  en la posició  $(i, j)$  no hi ha fitxa

$v f_{i,j} = 1 \Leftrightarrow$  en la posició  $(i, j)$  hi ha una fitxa blanca

$v f_{i,j} = 2 \Leftrightarrow$  en la posició  $(i, j)$  hi ha una fitxa negra

$vt = 0 \Leftrightarrow$  és el torn de les blanques

$vt = 1 \Leftrightarrow$  és el torn de les negres

$vdt(j) = 0 \Leftrightarrow$  el jugador  $j$  té el doble torn

$vdt(j) = 1 \Leftrightarrow$  el jugador  $j$  ha exhaurit el doble torn

Tenint en compte les dimensions possibles del joc, el valor de l'identificador és inferior a  $2^{63}$ , així doncs, podem representar aquest valor en un enter de 64 bits.

Una vegada tenim l'identificador de l'estat del joc, és fàcil calcular el id(M) incrementalment a partir d'un id anterior.

D'aquesta manera, ara només cal emmagatzemar l'id de l'estat del joc per aquells estats que creiem susceptibles de ser repetits.

Un moviment que avança, és a dir, que mou una peça cap endavant mai podrà produir un cicle, ja que segons les característiques del joc, les peces no poden tirar cap endarrere. Si matem una fitxa o be consumim el doble torn tampoc és possible entrar en cicle.

Així doncs, els moviments que poden produir cicles són aquells que no maten cap peça, són horitzontals i no consumeixen doble torn.

El procediment que cal seguir doncs, ha estat emmagatzemar l'estat del joc quan es produeix un d'aquests moviments i comprovar que no es repeteixi.

Aquesta comprovació és du a terme en la classe model, la mateixa classe que ens retorna els moviments possibles, així doncs, per a la simulació de partides, s'ha decidit que la mateixa classe model ens elimini els moviments que repeteixen un estat ja visitat, és a dir, si quan simulem una partida hi han moviments que produeixen cicles, aquests no es tenen en compte en retornar els moviments possibles.

El fet d'eliminar els moviments que produeixen cicles de la llista de moviments possibles només és du a terme en les simulacions ja que el fet de impossibilitar els cicles en el model real seria contradir les normes del joc ( ja que el joc permet que es produeixin cicles) .

Així doncs, si el jugador no té més remei que moure un moviment i entrar en cicle, perquè d'una altra manera perdria i l'adversari d'aquest també ha d'entrar en cicle per evitar perdre, es podria produir un cicle que repetit moltes vegades es considerarien tables, de totes maneres, la possibilitat, a causa de la mecànica del joc es bastant mínima.

Quant al jugador humà, aquest sí que té permès du a terme moviments en cicle, de manera que, en una partida contra un humà és possible entrar en cicle si aquesta és la seva decisió.

## CAPÍTOL 4 : ALTRES ESTRATÈGIES CONSIDERADES

### 4.1 MCTS

El Monte-Carlo Tree Search (MCTS) és un mètode de cerca basat en exploracions aleatòries en l'espai de cerca, utilitzades per a predir el moviment més prometedor.

Gràcies a l'estructura d'arbre i a un elevat nombre de simulacions aleatòries, el MCTS pot fer una estimació del valor de cada moviment possible.

A diferència de la cerca abans tractada, el MCTS no necessita una heurística (coneixement sobre el joc). Algunes de les condicions necessàries per a aplicar-lo són que la puntuació del joc sigui acotada, que les normes siguin conegudes i que les simulacions finalitzin en un període de temps curt, és a dir, que la longitud del joc estigui limitada.

En un principi, les exploracions dels nodes són aleatòries, posteriorment, gràcies als resultats de les exploracions prèvies, el MCTS és capaç de seleccionar els nodes més prometedors.

La selecció dels nodes és una qüestió a tenir en compte, ja que s'ha de trobar un equilibri entre l'*exploitation* (explotació) dels nodes amb alts nombres de victòries i l'*exploration* (exploració) dels nodes amb poques simulacions, és a dir, els més desconeguts. Per a aquesta tasca utilitzarem la fórmula UCT (Upper Confidence Bound applied to trees) que es descriu en els apartats següents.

#### 4.1.1 L'algorisme MCTS

Com es pot veure en la següent imatge el MCTS està dividit en quatre fases repetides tantes vegades com el temps de què disposem.

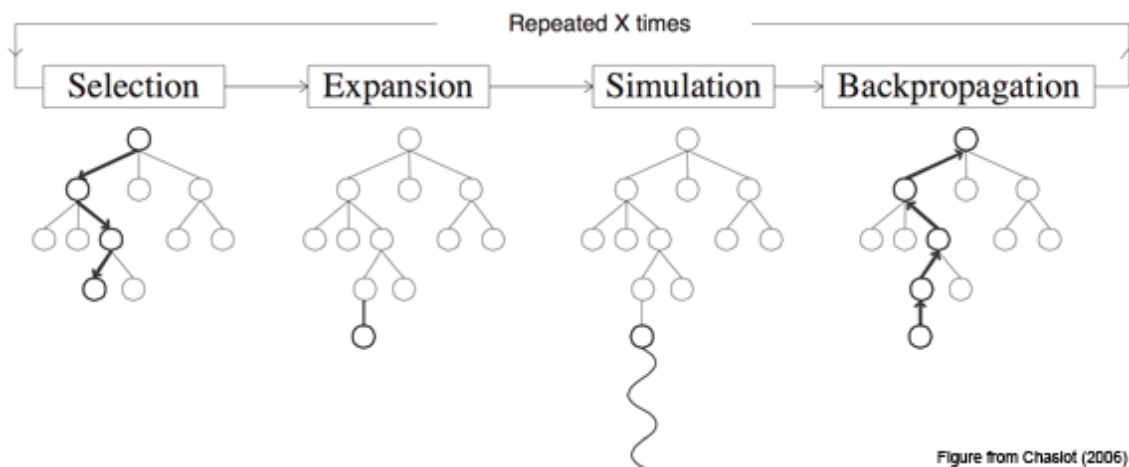


Figura 4.1 Fases del MCTS



Posteriorment, passem a descriure les fases de l'algorisme:

- **La Selecció:** Avancem des de l'arrel del arbre de cerca fins a trobar un node fulla i seleccionar-lo. Per a aquesta tasca utilitzarem l'estratègia UCT per la seva simplicitat i eficiència.
- **L'Expansió:** Aquesta fase afegeix nodes al nostre arbre MCTS
- **La Simulació:** A partir del node seleccionat, realitzem una simulació de la partida, realitzant moviments aleatoris per a tots dos jugadors.
- **La Retropropagació:** S'actualitzen els valors resultants de la simulació als nodes, des del node fulla fins al node arrel del nostre arbre.

## Pseudocodi del MCTS

```

funció MCTS (estat):
    Mentre(Hi_Ha_Temps) fer
        nodeActual=Arrel

        //si el node no és fulla, va seleccionant fins a
        arribar a un node fulla
        Mentre (!nodeActual.esFulla()) fer
            nodeActual=Seleccionar(nodeActual)

        fimentre
            //expandim el node seleccionat
            nodeActual=Expandir(nodeActual)

            // calculem el valor de la simulació (1, 0)
            Valor=Simulacio(nodeActual)

            // Retropropagem el valor cap al node arrel
            Mentre (nodeActual ∈ Arbre ) fer
                Retropropagacio(nodeActual,Valor)
                nodeActual=nodeActual.pare
            fimentre

        fimentre
            // retornem el millor moviment segons el valor màxim
            retorna MillorFill(Arrel)

    fifunció
  
```

*Figura 4.2 Pseudocodi del MCTS*

### 4.1.2 UCT

L'estratègia UCT (Upper Confidence Bound applied to trees) es utilitzada per a la fase de selecció, amb la finalitat de mantenir un equilibri entre l'explotació i l'exploració, aquesta estratègia es basa en la fórmula següent:

$$UCT = \frac{W_i}{n_i} + C \cdot \sqrt{\frac{\ln n}{n_i}}$$

On:

$W_i$  seria el valor del nombre de victòries acumulades.

$n_i$  el nombre de simulacions realitzades des del node  $i$ .

$C$  constant, a la practica s'usa sovint  $\sqrt{2}$ .

$n$  nombre de simulacions totals.

### 4.1.3 Selecció del moviment resultant

Després de fer les diferents simulacions, hem d'escollir el millor moviment a realitzar, que serà un dels fills del node arrel, per a aquesta tasca hi ha diferents estratègies com ara:

- Valor màxim: El fill que més valor tingui.
- Més robust: El fill que tingui més visites.
- Robust-valor màxim: El fill que gaudeixi del màxim tant de valor com de visites. En el cas de que no hi hagi cap fill amb el robust-valor màxim es continua simulant fins que en trobem un.

En aquest projecte utilitzarem el valor màxim per a triar el millor moviment.

### 4.1.4 Condició final MCTS

En la implementació del algorisme MCTS, s'ha triat com a condició per a l'acabament de les simulacions, i per tant per al final de l'execució de l'algorisme, un període de temps, aquest està configurat per a que sigui de quatre segons.

## 4.2 MCTS amb heurística

L'algorisme MCTS abans esmentat es basa en l'aleatorietat i en una bona estratègia de selecció de nodes més prometedors, però li manca el que podríem anomenar "coneixement".

És per aquesta raó que en aquesta part del projecte unirem l'eficàcia de l'algorisme MCTS amb el coneixement sobre els diferents estats del joc que ens proporciona una heurística.

Millorarem la precisió de la fase de selecció del nostre algorisme MCTS afegint a la nostra estratègia UCT el valor de la nostra heurística.

Per a aquesta tasca caldrà transformar els valors de la nostra heurística (de MAXVALOR a -MAXVALOR) a un rang de valors amb els quals pugui treballar el MCTS, que són de 0 a 1 (1 per a les victòries del jugador MAX).

Així doncs, aconseguirem el nostre valor heurístic per a un estat del tauler de la següent forma:

$$H(\text{estat}) = \frac{\text{valor}(\text{estat}) + \text{MAXVALOR}}{2 \cdot \text{MAXVALOR}}$$

On  $\text{valor}(\text{estat})$  és el valor heurístic de l'estat del joc (un valor entre  $-\text{MAXVALOR}$  i  $\text{MAXVALOR}$ )

$H(\text{estat})$  és un valor entre 0 i 1 (probabilitat), indicant 1 la victòria del jugador MAX i 0 la derrota del jugador MAX.

MAXVALOR és el màxim valor que té la nostra heurística.

Així doncs, en la fase d'expansió, avaluarem l'estat del joc un cop dut a terme el moviment del node en qüestió i guardarem aquest valor en el node per a utilitzar-lo posteriorment en la fase de selecció.

És important trobar un equilibri entre els beneficis que ens proporciona l'heurística a l'hora de seleccionar el millor node a expandir i els beneficis que ens proporciona l'UCT en conjunt amb l'exploració de l'arbre de cerca.

Per aquesta raó afegim una variable  $\beta$  que ens reduirà el valor de l'heurística a mesura que es vagi sobrepasant un nombre "k" de simulacions.

$$\beta = \frac{k}{k + n_i}$$

On  $n_i$  es el nombre de simulacions.

D'aquesta manera el nostre algorisme de selecció per al MCTS ens retornarà els valors segons:

$$UCT_{heurística} = \beta \cdot H(\text{estat}) + (1-\beta) \cdot \frac{w_i}{n_i} + C \cdot \sqrt{\frac{\ln(n)}{n_i}}$$

Com es pot observar al llarg de l'execució del nostre algorisme MCTS i les seves respectives simulacions, la nostra heurística comença tenint molt pes (per ajudar a la tria ja que la informació de què disposa el MCTS a l'inici es mínima sense heurística) i, posteriorment va disminuint la seva importància ja que passades unes quantes simulacions el MCTS ja té prou informació per a seleccionar els nodes més prometedors.

```

funció UCTValor (node):
    uctValor=0
    beta=k/k+node.numVisites
    si (node.numVisites==0)
        uctValor = node.valorHeuristic
    //si el node no té visites només ens centrem en l'heurística
    sino
        uctValor = beta* node.valorHeuristic + (1-beta)
            * victòries/node.numVisites +constant *
            Math.sqrt(Math.log(node.pare.numVisites)/node.numVisites)
    // si el node ja ha estat visitat apliquem la fórmula
    fisi
    uctValor += numpetit * aleatori

    // Sumem un nombre molt petit i aleatori per a desempatar en el cas de
    // nodes amb el mateix valor, això fa que l'algorisme gaudeixi de certa
    // aleatorietat

fifunció

```

Figura 4.3 Pseudocodi funció UCTValor

### 4.3 Segona versió de l'heurística

Així com la primera versió de l'heurística era bastant agressiva, ja que fomentava l'avanç cap a les cantonades, tot donant valors elevats a les posicions de l'àrea de l'adversari, la segona versió es podria dir que és més de desgast, per aquest fet, com veurem en les estadístiques, la mitjana de torns d'una partida és superior a la mitjana de torns d'una partida amb la primera heurística.

Per a la segona versió tindrem en compte les amenaces, és a dir, calcularem la diferència entre la quantitat de fitxes a les quals podem matar i la quantitat de fitxes nostres a les quals l'adversari pot matar. D'aquesta manera l'heurística li donarà més valor a posicions en què podem matar fitxes.

Posteriorment, canviem la fórmula amb què calculàvem la diferència de peces, per l'explicada posteriorment, que ens donarà valors més alts al fet de tenir més peces que l'adversari, quan estiguem al final de la partida.

Aquests dos factors ens potenciarà el fet de tenir posicions on amenacem peces i el fet de matar-les, és per aquesta raó que serà una heurística més de desgast, ja que la posició respecte el tauler i l'avanç estarà en un segon lloc.

Definirem, doncs, la nova heurística com una combinació lineal de diversos paràmetres, tot seguit, explicarem cadascun d'ells.

$$\text{Combinació lineal de paràmetres} = C1 * \text{parametre1} + C2 * \text{parametre2} + C3 * \text{parametre3}$$

Els paràmetres que tindrem en compte per aquesta heurística seran:

- Parametre1= diff(posicional)
- Parametre2= diff(num\_peces)
- Parametre3= diff (amenaces)

On la **diferència posicional** serà el resultat de la resta dels valors estimats de les peces segons el mapa de pesos del jugador MAX, menys els de l'adversari (diferència ja utilitzada en l'anterior heurística).

La **diferència de peces** la modificarem respecte a la nostra heurística anterior, en aquesta heurística farem servir la fórmula:

$$\frac{\text{fitxes}_1 - \text{fitxes}_2}{\text{fitxes}_1 + \text{fitxes}_2} * \text{Constant}$$

D'aquesta manera si es té una fitxa més d'avantatge a l'inici de la partida no serà tant rellevant com tenir-ne una més al final de la partida, on l'avantatge tindrà més valor.

Pel que fa a la **diferència entre amenaces** ens referirem a la suma de totes les possibilitats de matar a les peces de l'adversari, menys les possibilitats que té ell de matar les nostres.

## CAPÍTOL 5 : ANÀLISIS DELS RESULTATS

### 5.1 Resultats diferents profunditats del Minimax

Per a generar les estadístiques s'han fet paquets de 100 partides amb les diferents característiques indicades a les taules.

#### Tauler estàndard

##### Tauler 4x4

| Profunditat Minimax     | Negres  |         |         |         | Victòries Blanques | 55% |
|-------------------------|---------|---------|---------|---------|--------------------|-----|
| Blanques                | 2       | 3       | 4       | 5       |                    |     |
| 2                       | 60 / 40 | 43 / 57 | 4 / 96  | 23 / 77 | 32.5               |     |
| 3                       | 72 / 28 | 46 / 54 | 18 / 82 | 21 / 79 | 39.25              |     |
| 4                       | 89 / 11 | 82 / 18 | 46 / 54 | 41 / 59 | 64.5               |     |
| 5                       | 94 / 6  | 88 / 12 | 85 / 15 | 68 / 32 | 83.75              |     |
| <b>Victòries Negres</b> | 21.25   | 35.25   | 61.75   | 61.75   |                    |     |
| <b>45%</b>              |         |         |         |         |                    |     |

Taula 5.1 Tauler 4x4 Minimax

##### Tauler 6x6 (heurística1)

| Profunditat Minimax     | Negres  |         |         |         | Victòries Blanques | 59% |
|-------------------------|---------|---------|---------|---------|--------------------|-----|
| Blanques                | 2       | 3       | 4       | 5       |                    |     |
| 2                       | 83 / 17 | 29 / 71 | 10 / 90 | 4 / 96  | 31.5               |     |
| 3                       | 96 / 4  | 78 / 22 | 17 / 83 | 10 / 90 | 50.25              |     |
| 4                       | 100 / 0 | 98 / 2  | 62 / 38 | 16 / 84 | 69                 |     |
| 5                       | 100 / 0 | 98 / 2  | 87 / 13 | 57 / 43 | 85.5               |     |
| <b>Victòries Negres</b> | 5.25    | 24.25   | 56      | 78.25   |                    |     |
| <b>41%</b>              |         |         |         |         |                    |     |

Taula 5.2 Tauler 6x6 Minimax

En principi, quan la profunditat del minimax és poca hi ha una clara superioritat de les blanques, a mesura que s'incrementa el nivell s'igualen els resultats, donant unes xifres bastant similars en enfrontaments amb el mateix nivell. Les blanques continuen tenint un cert percentatge superior de victòries.

Podríem dir també que com més profunditat gaudeix el minimax, més possibilitats hi ha de victòria.

## Tipus Finals 4x4 – 6x6

| Característiques<br>Profunditat<br>Minimax |          | Tipus de final       |              |                        |          |                      |             |
|--------------------------------------------|----------|----------------------|--------------|------------------------|----------|----------------------|-------------|
| Negres                                     | Blanques | Arribar a cantonades |              | No hi ha mes moviments |          | No hi ha prou fitxes |             |
|                                            |          | 4x4                  | 6x6          | 4x4                    | 6x6      | 4x4                  | 6x6         |
| 2                                          | 2        | 93                   | 100          | 0                      | 0        | 7                    | 0           |
| 2                                          | 3        | 92                   | 100          | 0                      | 0        | 8                    | 0           |
| 2                                          | 4        | 91                   | 99           | 0                      | 0        | 9                    | 1           |
| 2                                          | 5        | 93                   | 100          | 0                      | 0        | 7                    | 0           |
| 3                                          | 2        | 97                   | 100          | 1                      | 0        | 2                    | 0           |
| 3                                          | 3        | 90                   | 100          | 0                      | 0        | 10                   | 0           |
| 3                                          | 4        | 76                   | 100          | 0                      | 0        | 24                   | 0           |
| 3                                          | 5        | 61                   | 100          | 4                      | 0        | 35                   | 0           |
| 4                                          | 2        | 89                   | 100          | 0                      | 0        | 11                   | 0           |
| 4                                          | 3        | 87                   | 100          | 0                      | 0        | 13                   | 0           |
| 4                                          | 4        | 85                   | 94           | 2                      | 0        | 13                   | 6           |
| 4                                          | 5        | 67                   | 99           | 1                      | 0        | 32                   | 1           |
| 5                                          | 2        | 82                   | 100          | 0                      | 0        | 18                   | 0           |
| 5                                          | 3        | 88                   | 100          | 0                      | 0        | 12                   | 0           |
| 5                                          | 4        | 73                   | 97           | 3                      | 0        | 24                   | 3           |
| 5                                          | 5        | 83                   | 100          | 0                      | 0        | 17                   | 0           |
| <b>Mitjana</b>                             |          | <b>84.18</b>         | <b>99.31</b> | <b>0.68</b>            | <b>0</b> | <b>14.68</b>         | <b>0.68</b> |

Taula 5.3 Tipus de finals

Podem observar com en la gran majoria de partides acaben amb l'objectiu principal, que es arribar a les cantonades, la causa d'un 14% aproximat de finals és la situació on un dels jugadors es queda sense fitxes i una part mínima ( 0.68% ) es degut a que un jugador no té moviments possibles.

Si es juga amb el tauler 4x4 es produeixen més victòries per quedar-se sense fitxes que si es juga amb el tauler 6x6, el motiu d'aquest fet és en part a la dimensió reduïda i el nombre inferior de fitxes en comparació amb les emprades en el tauler més gran. Si es juga en el tauler 6x6, gairebé el 100% de victòries s'esdevenen quan arriba a les cantonades del camp contrari.

### Mitjana de torns tauler 6x6 (heurística1)

| Característiques    |          | Mitjana dels torns |
|---------------------|----------|--------------------|
| Profunditat Minimax |          |                    |
| Negres              | Blanques |                    |
| 2                   | 2        | 36                 |
| 2                   | 3        | 30                 |
| 2                   | 4        | 32                 |
| 2                   | 5        | 30                 |
| 3                   | 2        | 34                 |
| 3                   | 3        | 33                 |
| 3                   | 4        | 35                 |
| 3                   | 5        | 32                 |
| 4                   | 2        | 37                 |
| 4                   | 3        | 37                 |
| 4                   | 4        | 43                 |
| 4                   | 5        | 38                 |
| 5                   | 2        | 35                 |
| 5                   | 3        | 34                 |
| 5                   | 4        | 38                 |
| 5                   | 5        | 38                 |
| <b>Mitjana</b>      |          | <b>35.125</b>      |

Taula 5.4 Mitjana de torns minimax

La mitjana de torns per partida és de 35.125, es pot observar com les partides amb més profunditat (5-4) en els jugadors generen un nombre major de torns, a diferència de les de menor profunditat.



## 5.2 Resultats de les diferents heurístiques

Per comparar les diferents heurístiques farem simulacions de partides amb el minimax amb 5 de profunditat i el MCTS + Heurística amb 4 segons de límit de temps. A continuació es mostren els resultats obtinguts.

Comparació entre heurístiques 6x6

| Algorisme                               | Negres                 |                        |                       |                       | Mitjana victòries<br>Blanques |
|-----------------------------------------|------------------------|------------------------|-----------------------|-----------------------|-------------------------------|
|                                         | Minimax<br>heurística1 | Minimax<br>heurística2 | MCTS +<br>heurística1 | MCTS +<br>heurística2 |                               |
| <b>Blanques</b>                         |                        |                        |                       |                       |                               |
| Minimax<br>heurística1                  | 57 / 43                | 24 / 76                | 67 / 33               | 38 / 62               | 46.5                          |
| Minimax<br>heurística2                  | 87 / 13                | 36 / 64                | 94 / 6                | 62 / 38               | 69.75                         |
| MCTS +<br>heurística1                   | 47 / 53                | 10 / 90                | 45 / 55               | 39 / 61               | 35.25                         |
| MCTS +<br>heurística2                   | 61 / 39                | 44 / 56                | 84 / 16               | 66 / 34               | 63.75                         |
| <b>Mitjana<br/>victòries<br/>negres</b> | 37                     | 71.5                   | 27.5                  | 48.75                 |                               |

Taula 5.5 Comparació entre heurístiques

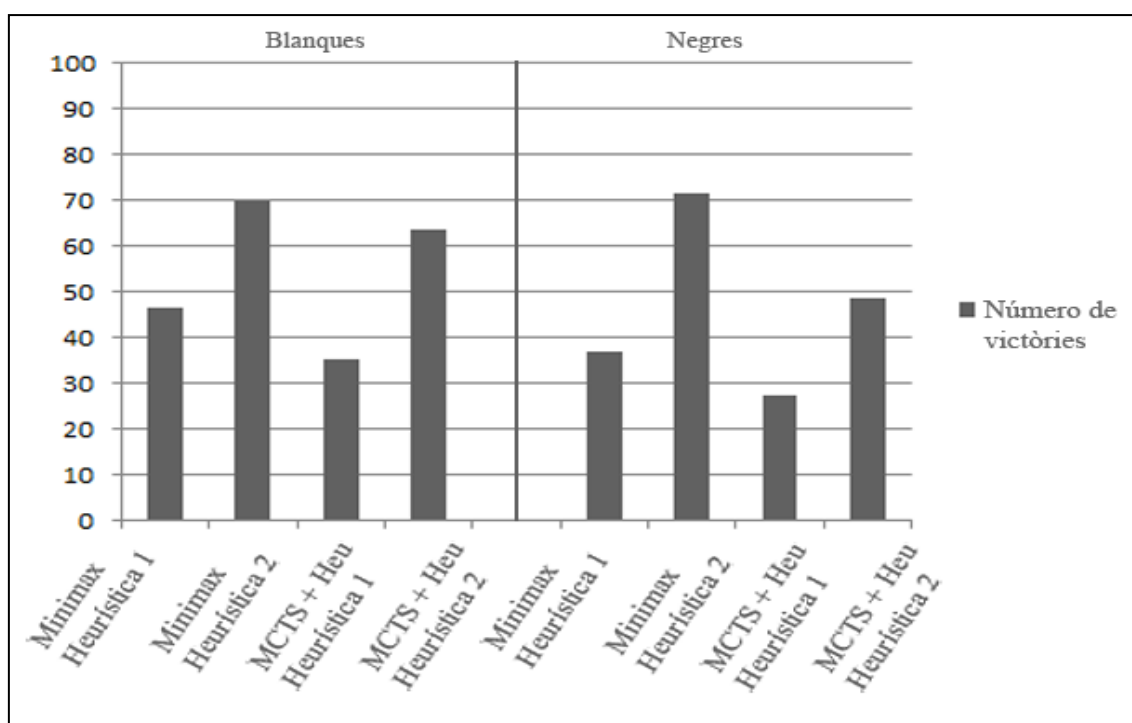


Figura 5.1 Número de victòries heurístiques

Es pot observar com els resultats de l'heurística 2 són millors que els de l'heurística 1, ja que presenta un major nombre de victòries i sempre són superiors als de l'heurística 1 quan s'enfronten l'una amb l'altra. El motiu d'això és en part el fet que l'heurística 2 contempla més paràmetres, i aquests són adients, per tant, fa una avaluació més acurada.

| Algorismes utilitzats         |                               | Mitjana de torns |
|-------------------------------|-------------------------------|------------------|
| Negres                        | Blanques                      |                  |
| <b>Minimax</b><br>heurística1 | <b>Minimax</b><br>heurística1 | 38               |
| <b>Minimax</b><br>heurística1 | <b>Minimax</b><br>heurística2 | 36               |
| <b>Minimax</b><br>heurística1 | <b>MCTS +</b><br>heurística1  | 43               |
| <b>Minimax</b><br>heurística1 | <b>MCTS +</b><br>heurística2  | 47               |
| <b>Minimax</b><br>heurística2 | <b>Minimax</b><br>heurística1 | 42               |
| <b>Minimax</b><br>heurística2 | <b>Minimax</b><br>heurística2 | 38               |
| <b>Minimax</b><br>heurística2 | <b>MCTS +</b><br>heurística1  | 39               |
| <b>Minimax</b><br>heurística2 | <b>MCTS +</b><br>heurística2  | 45               |
| <b>MCTS +</b><br>heurística1  | <b>Minimax</b><br>heurística1 | 43               |
| <b>MCTS +</b><br>heurística1  | <b>Minimax</b><br>heurística2 | 39               |
| <b>MCTS +</b><br>heurística1  | <b>MCTS +</b><br>heurística1  | 52               |
| <b>MCTS +</b><br>heurística1  | <b>MCTS +</b><br>heurística2  | 51               |
| <b>MCTS +</b><br>heurística2  | <b>Minimax</b><br>heurística1 | 42               |
| <b>MCTS +</b><br>heurística2  | <b>Minimax</b><br>heurística2 | 42               |
| <b>MCTS +</b><br>heurística2  | <b>MCTS +</b><br>heurística1  | 50               |
| <b>MCTS +</b><br>heurística2  | <b>MCTS +</b><br>heurística2  | 57               |
| <b>Mitjana</b>                |                               | <b>44</b>        |

Taula 5.6 Mitjana de torns heurístiques

Com podem veure, la mitjana de torns ha augmentat respecte els 35 que obteníem en les diferents simulacions amb la combinació de minimax amb l'heurística 1.

El MCTS+heurística te una mitjana de torns superior al minimax, succeix el mateix amb l'heurística 2, pel fet de ser una heurística més aviat de desgast i no tant agressiva com la 1, també te una mitjana de torns superior. En conseqüència, les partides triguen més a acabar ja que no va tant directe al objectiu final.

### 5.3 Resultats del MCTS + heurística amb diferents temps

#### Comparació MCTS + Heurística amb diferents temps (heurística final)

| Algorisme                               | Negres                  |                         |                                       |
|-----------------------------------------|-------------------------|-------------------------|---------------------------------------|
| <b>Blanques</b>                         | MCTS+ HEU<br>(4 segons) | MCTS+ HEU<br>(6 segons) | <b>Mitjana victòries<br/>Blanques</b> |
| MCTS+ HEU<br>(4 segons)                 | 66 / 34                 | 41 / 59                 | 53.5                                  |
| MCTS+ HEU<br>(6 segons)                 | 72 / 28                 | 62 / 38                 | 67                                    |
| <b>Mitjana<br/>victòries<br/>negres</b> | 31                      | 48.5                    |                                       |

Taula 5.7 MCTS+Heurística diferents temps

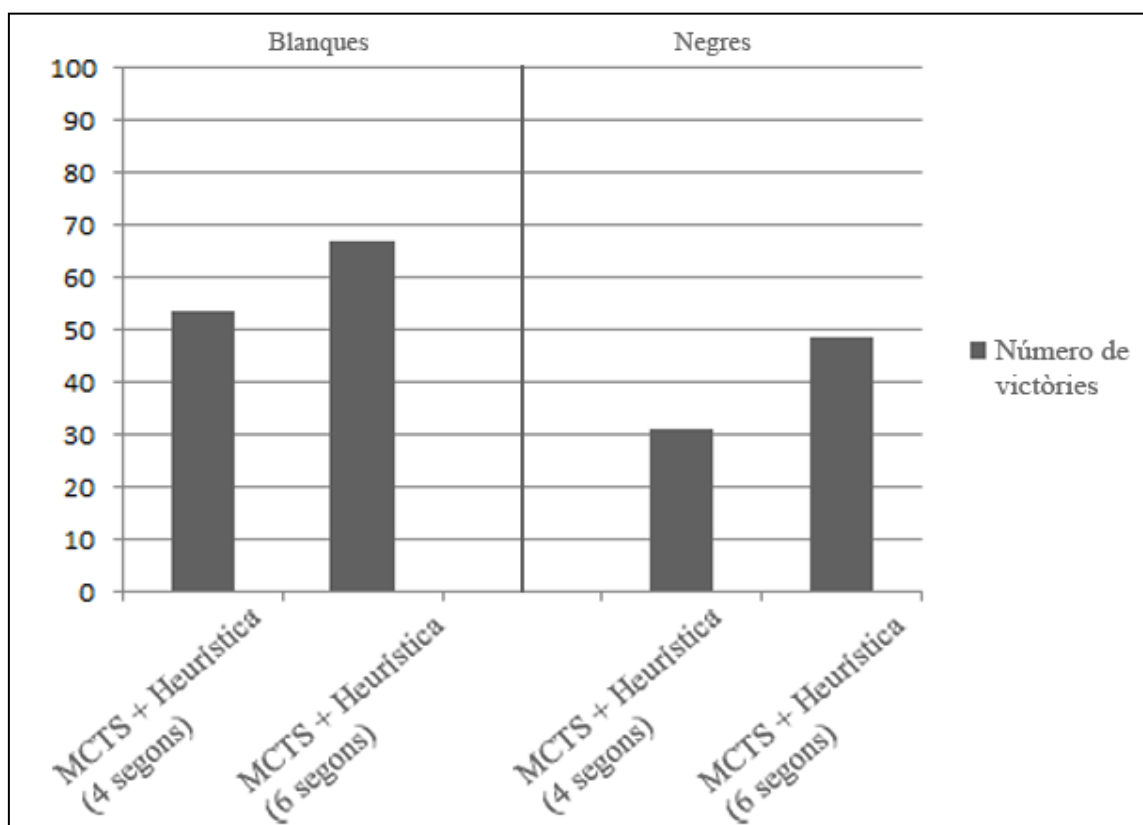


Figura 5.2 MCTS + Heurística diferents temps

Podem observar quan augmentem el temps límit d'execució del MCTS+ Heurística, que l'eficàcia d'aquest algorisme millora i obtenim un percentatge de victòries superior. Aquests resultats ens permeten suposar que l'algorisme està implementat de forma correcta i que l'aleatoreïtat va disminuint per passar a una tria més acurada del moviment correcte gràcies a les simulacions que l'algorisme du a terme.

## 5.4 Resultats dels diferents algorismes

### Comparació entre algorismes 6x6 (heurística final)

| Profunditat<br>Minimax         | Negres   |                     |                    | Mitjana<br>Victòries<br>Blanques | 52% |
|--------------------------------|----------|---------------------|--------------------|----------------------------------|-----|
|                                | Blanques | Minimax<br>(prof.5) | MCTS<br>(4 segons) |                                  |     |
| Minimax<br>(prof.5)            | 36 / 64  | 100 / 0             | 62 / 38            | 66                               |     |
| MCTS<br>(4 segons)             | 1 / 99   | 49 / 51             | 10 / 90            | 20                               |     |
| MCTS+ HEU<br>(4 segons)        | 44 / 56  | 97 / 3              | 66 / 34            | 69                               |     |
| Mitjana<br>Victòries<br>Negres | 73       | 18                  | 54                 |                                  |     |
| 48%                            |          |                     |                    |                                  |     |

Taula 5.9 Comparació entre algorismes

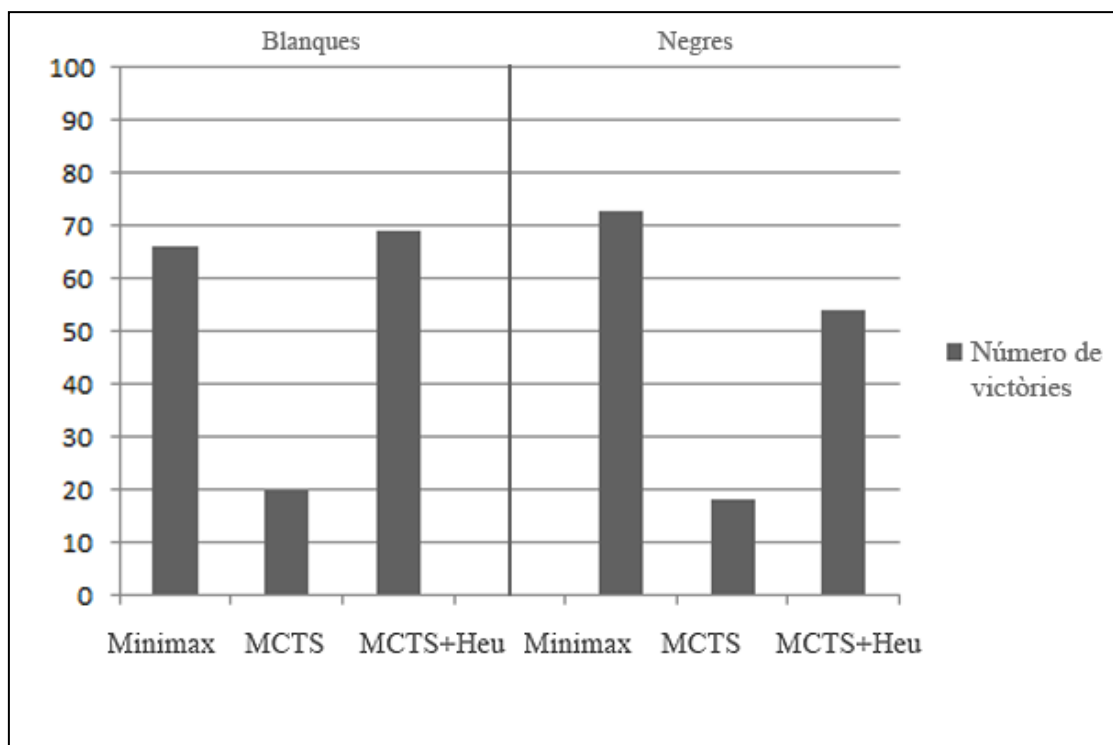


Figura 5.3 Número de victòries de tots els algorismes

Tot i que el minimax guanya al MCTS+Heurística quan s'enfronten un amb l'altre, pel que fa al nombre de victòries si es juga amb les blanques estan bastant igualats i en canvi, jugant amb les negres el minimax presenta una lleugera superioritat respecte al montecarlo amb heurística.

S'observa també que el pes de l'heurística és bastant important ja que els algorismes que la usen obtenen més nombre de victòries que per exemple, el MCTS simple que no usa heurística (coneixement).

## 5.5 Conclusions de l'estudi comparatiu

Podem resumir les conclusions a les que arribem amb els resultats de simular paquets de 100 partides i comparar-los entre les diferents heurístiques i algorismes, en diferents punts:

- El minimax amb major profunditat té un major nombre de victòries, el que ens dona a entendre que l'algorisme està ben implementat.
- El tipus de finals del joc en taulers 4x4 acostumen a ser més diversos que en el tauler 6x6 a causa del seu tamany, en el de 6x6 gairebé un 100 % dels finals resultaven d'arribar a les cantonades.
- El minimax augmenta la seva mitjana de torns quan més gran és la profunditat. D'altra banda, el MCTS+Heurística i l'heurística 2 encara fan partides més llargues arribant a obtenir una mitjana de torns de fins i tot 57.
- L'heurística 2 és bastant millor que la inicial, ja que contempla diferents paràmetres i fa una avaluació més acurada de l'estat del joc.
- A l'hora de fer partides amb diferents temps per al MCTS+Heurística, observem que quan augmentem el límit de temps, (fet que fa que el nombre de simulacions sigui major), les victòries també augmenten. Aquest algorisme podria arribar a millorar la seva eficàcia amb límits més elevats de temps d'execució.
- Pel que fa als resultats finals dels diferents algorismes, observem com el minimax i el MCTS+Heurística estan bastant igualats en victòries, tot i que si enfrontem l'un amb l'altre el minimax és lleugerament superior. En relació amb el MCTS sense heurística, està bastant allunyat pel que fa a número de victòries dels seus companys, ja que el fet de no tenir una heurística és bastant rellevant.

Cal dir, que si es gaudís d'un computador més potent, tant el minimax com el MCTS o el MCTS+ Heurística millorarien la seva eficàcia perquè podríem augmentar el nombre de profunditat en el minimax i el nombre de simulacions per segon en el montecarlo.

## CAPÍTOL 6 : IMPLEMENTACIÓ

### 6.1 Eines utilitzades

Per a crear l'aplicació del joc Tebes, he triat el llenguatge java així com la llibreria swing per a la interfície gràfica.

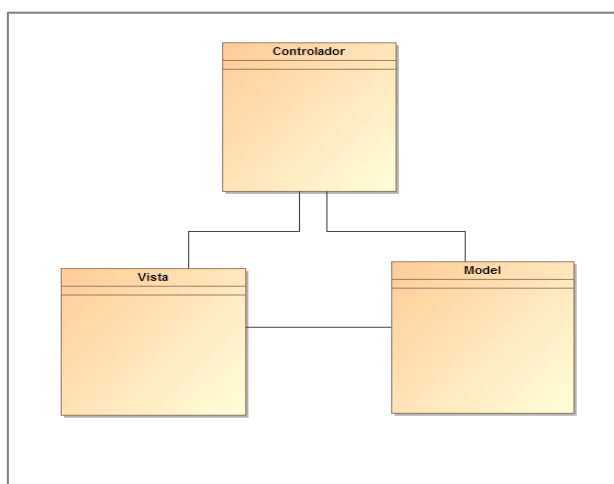
L'IDE utilitzat ha estat Eclipse, concretament la versió Kepler.

Per a generar l'arxiu executable .jar a partir d'un projecte, utilitzem el plug-in Fat Jar, que ha resultat ser molt senzill i eficaç per a aquesta tasca.

### 6.2 Procediment seguit

A l'hora d'iniciar el procés d'implementació de l'aplicació del joc Tebes, partim d'una base creada conforme el patró MVC.

El patró Model-Vista-Controlador, construeix diferents components, per un costat definirem el que serà la representació de la informació, i per l'altre, la interacció amb l'usuari.



*Figura 6.1 Model Vista Controlador*

Una vegada creada l'estructura bàsica del nostre projecte, passem a definir el diagrama de classes, com es mostra en el següent apartat.

En el procés de creació de l'aplicació hem partim d'una solució inicial, en el nostre cas, l'algorisme minimax amb una heurística bàsica en la qual només es té en compte la diferència posicional respecte a un mapa de pesos. Posteriorment, hem creat unes versions alternatives amb l'algorisme MCTS sense heurística i el MCTS amb heurística, per a poder fer un estudi comparatiu.

Finalment, hem creat una segona versió de l'heurística que contempla més paràmetres que la primera, per tant, com s'ha pogut veure en les estadístiques, la millora notablement.

### 6.3 Diagrama de classes

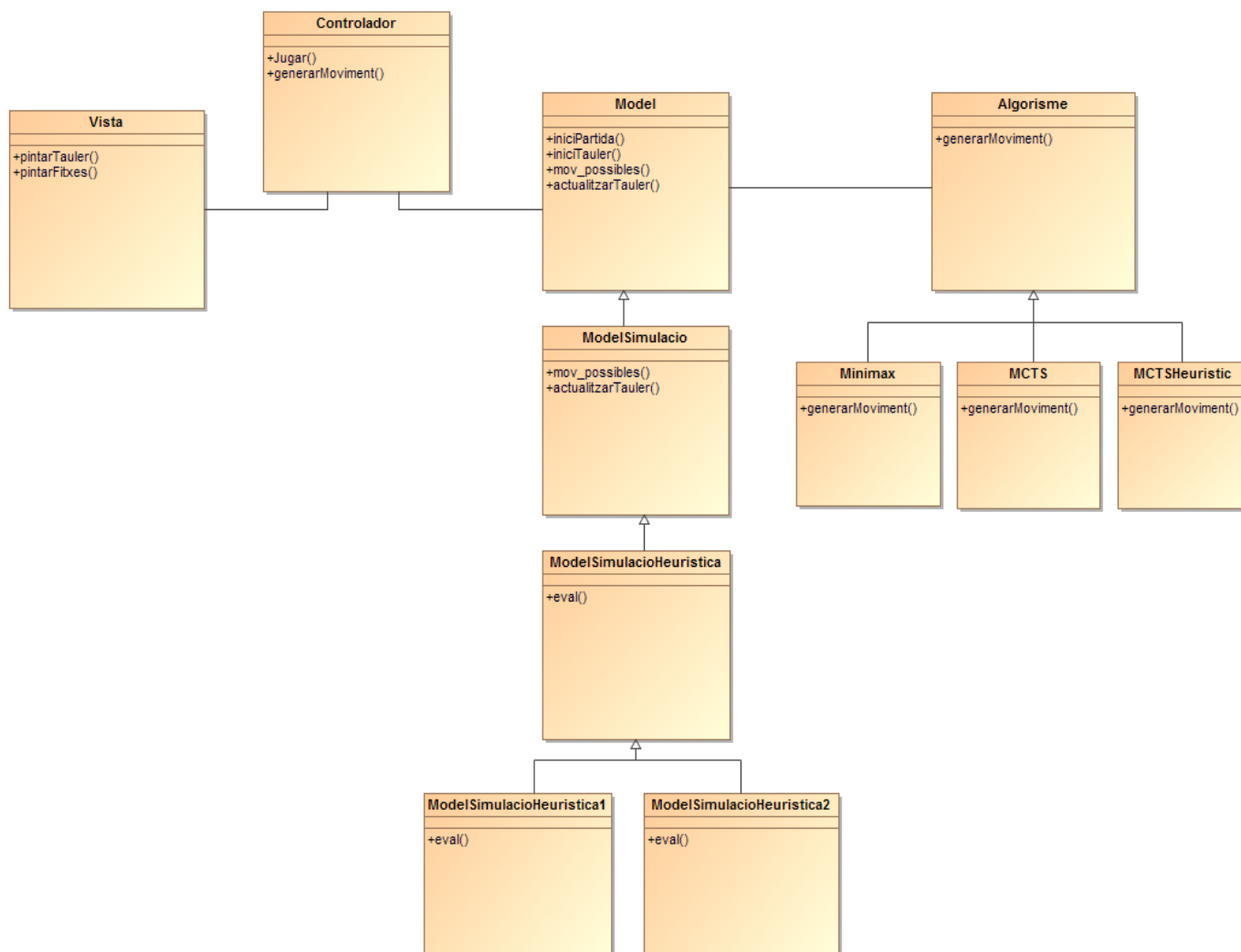


Figura 6.2 Diagrama de Classes

Com es pot observar en l'imatge apliquem el patró MVC.

Pel que fa a les heurístiques i als algorismes utilitzats aniran en relació amb la classe model, ja que es pot considerar que són representació de l'informació.

Així doncs, de la classe Model es deriva la classe ModelSimulació, que reimplementa la funció `mov_possibles`, amb la opció d'evitar cicles, i la funció `actualitzarTauler`.

La classe ModelSimulacióHeurística es una classe abstracta que ens defineix la funció `eval()` que implementaran les classes ModelSimulacioHeurística1 i ModelSimulacioHeurística2, és a dir, és la funció que ens retornarà el valor heurístic d'un model en un estat en concret.

Per a implementar els diferents algorismes, hem creat una classe Algorisme de la que heretaran la classe Minimax, MCTS i MCTSheuristic i que defineix la funció `generarMoviment()`, ja que els algorismes ens han de retornar el moviment a executar.

## CAPÍTOL 7 : CONCLUSIONS GENERALS

Clourem aquest projecte explicant algunes impressions i lliçons apreses durant el procés de creació d'aquest.

El fet de començar per una solució inicial i anar evolucionant cap a les diferents tècniques posteriorment analitzades ha facilitat el procés d'aprenentatge, ja que la dificultat ha anat augmentant progressivament.

L'estudi comparatiu de les diferents tècniques ens ha dut a pensar en les avantatges i inconvenients de cadascuna d'elles, fent cada tècnica útil per a diferents problemàtiques depenent de les seves característiques.

Així doncs, per exemple, el MCTS seria més adequat quan no es té un coneixement del joc ni una heurística definida, a diferència d'altres algorismes analitzats.

Hem pogut veure que la utilitat de tenir una heurística ben definida és força important, ja que les solucions que gaudeixen d'una bona heurística milloren considerablement els resultats, també veiem, que encara que les diferències entre dues heurístiques semblin trivials, poden suposar una diferència considerable en relació al nombre de victòries que aconsegueixen.

Una de les problemàtiques trobades durant la realització del projecte ha estat la capacitat de còmput de la maquinària utilitzada, ja que per exemple, no hem pogut analitzar profunditats de més de 6 per al minimax i els resultats obtinguts no són tant bons com els que obtindríem si disposéssim d'una maquinària més potent.

Fixant-nos en les estadístiques, intuïm que si gaudíssim de més capacitat de còmput les nostres solucions millorarien.

Per últim, una possible via d'ampliació d'aquest projecte, que no hem pogut abastar per manca de temps, consistiria en implementar un MCTS amb memòria, és a dir, implementar una solució que anés guardant l'arbre creat en l'anterior jugada per a no tenir que tornar-lo a generar i partir del sub-arbre en què ens trobem tot descartant el restant, d'aquesta manera l'arbre del qual partiríem ja tindria alguns nodes explorats a partir dels quals podríem partir per a generar més simulacions.

Com a objectiu aconseguit s'ha creat una aplicació capaç de jugar contra un humà, arribant a una dificultat mitjanament acceptable, depenent de l'algorisme i heurística triats. A més, l'aplicació creada ens ha servit per a comparar i avaluar les diferents tècniques a través del seu anàlisi.



## GLOSSARI

**Joc estocàstic:** Joc dinàmic amb transicions probabilístiques.

**Joc determinista:** Joc predictable, és l'oposat al joc estocàstic.

**Pseudocodi:** Descripció d'alt nivell d'un algorisme.

**MCTS:** Monte-Carlo Tree Search.

**UCT:** Upper Confidence Bound applied to trees.

**Explotació:** Selecció dels nodes més prometedors.

**Exploració:** Selecció dels nodes menys visitats.

**Heurística:** Tècnica dissenyada per resoldre problemes.

**Cicle:** Quan en el joc s'arriba a un estat ja visitat.

**Moviment:** Transició d'un estat a un altre del joc.

**BIBLIOGRAFIA**

**Baier, H., Winands, M. H. M.,** (2014) *Monte-Carlo Tree Search and Minimax Hybrids with Heuristic Evaluation Functions*, Proc. CGW 2014, CCIS 504 45-63.

**Bouzy, B.** (2008) *Old-fashioned Computer Go vs Monte-Carlo Go*, Proc. SYNASC 2008.

**Chaslot, G. M. J.-B., Winands, M.H.M., van den Herik, J.H., Uiterwijk, J.W.H.M., Bouzy, B.** (2008) *Progressive strategies for monte-carlo tree search*, New mathematics and natural computation 4 343-357.

**Chaslot, G., Bakkes, S., Szita, I., Spronck, P.** (2008) *Monte-Carlo Tree search: A New Framework for Game AI*, Proc. AIIDE 2008.

**Drake, P., Uurtamo, S.** (2007) *Heuristics in Monte Carlo Go*. Proc. IC-AI 2007 171-175.

**Garcia Serrano, A.** (2012) *Inteligencia Artificial. Fundamentos, practica y aplicaciones*.

**Gelly, S., Silver, D.** (2007) *Combining Online and Offline Knowledge in UCT*, Proc. ICML 2007 273-280.

**Gelly, S., Silver, D.** (2011) *Monte-Carlo Tree Search and Rapid Action Value Estimation in Computer Go*, Artificial Intelligence 175 1856-1875.

**Gelly, S., Wang, Y., Munos, R., Teytaud, O.** (2006) *Modification of UCT with Patterns in Monte-Carlo Go*, HAL [archives-ouvertes.fr](http://archives-ouvertes.fr) <http://hal.univ-lille3.fr/inria-00117266v3/document>

**Kocsis, L., Szepesvári, C.** (2006) *Bandit based Monte-Carlo Planning*, Proc. ECML 2006 282-293.

**Korf, R. E., Chickering, D. M.** (1996) *Best-first minimax search*, Artificial Intelligence 84 299-337.

**Lanctot, M., Winands, M. H. M., Pepels, T., Sturtevant, N. R.** (2014) *Monte Carlo Tree Search with Heuristic Evaluations using Implicit Minimax Backups*, Proc. CIG 2014.

**Mandziuk, J. Kusiak, M. Waledzik, K.** (2007) *Evolutionary-based heuristic generators for checkers and give-away checkers*, Expert Systems 24 189-211.

**Mathias Ganter; Jonas Klink** (2015) *A New Experience: O-Thell-Us – An AI Project*, <http://courses.cs.washington.edu/courses/cse573/04au/Project/mini1/O-Thell-Us/Othellus.pdf>

**Miles, C., Quiroz, J., Leigh, R., Louis, S. J.** (2007) *Co-Evolving Influence Map Tree Based Strategy Game Players*, Proc. CIG 2007.

**Plaat, A., Schaeffer, J., Pijls, W., de Bruin, A.** (1996) *Best-first fixed-depth minimax algorithms*, Artificial Intelligence 87 255-293.

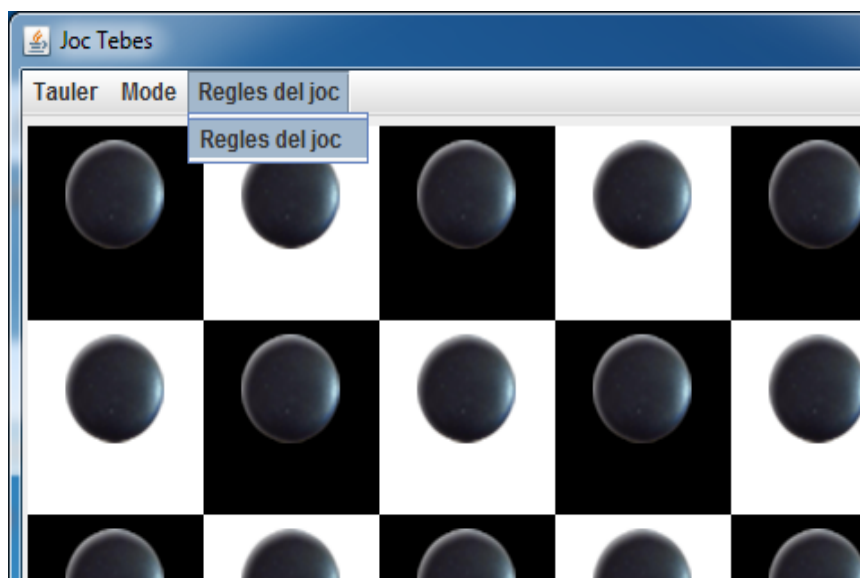
**Schaeffer, J., Björnsson, Y., Burch, N., Kishimoto, A., Müller, M., Lake, R., Lu, P., Sutphen, S.** (2007) *Solving Checkers*, Science 317 1518-1522.

**Schaeffer, J., van den Herik, H. J.** (2002) *Games, computers, and artificial intelligence*, Artificial Intelligence 134 1-7.

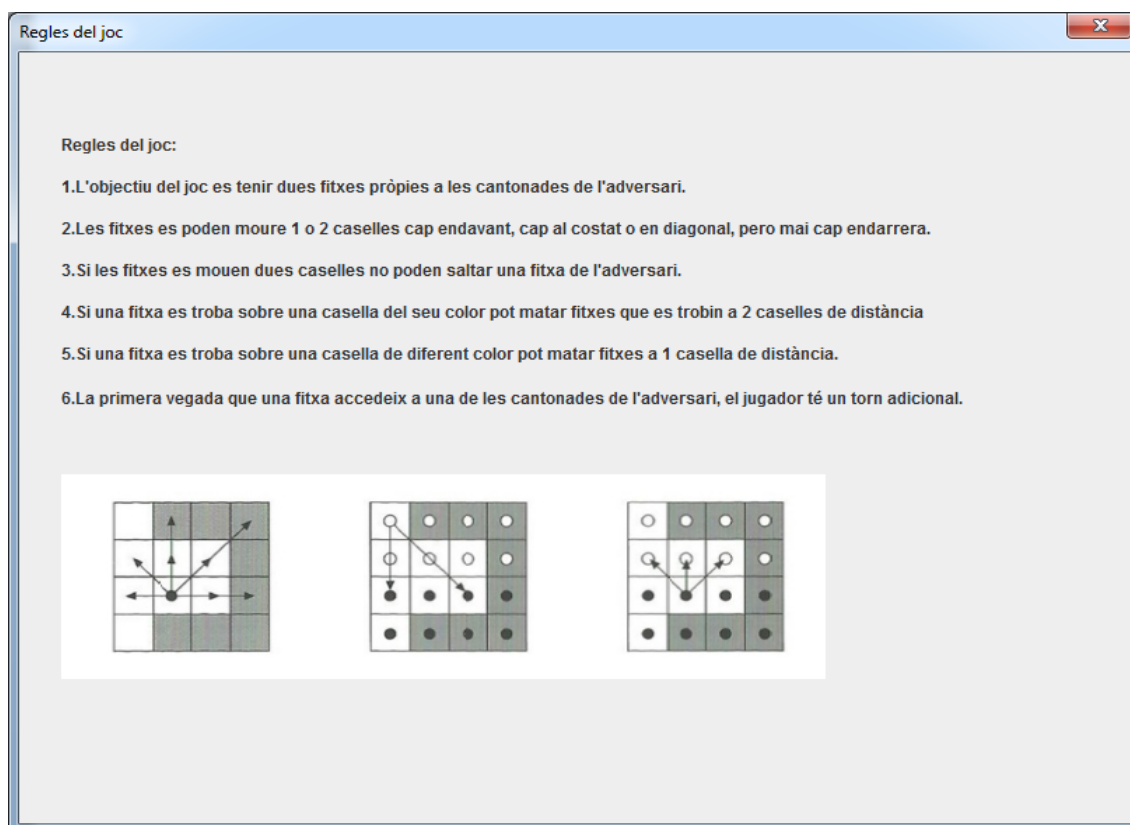
**Van den Herik, H. J., Uiterwijk, J. W. H. M., van Rijswijk, J.** (2002) *Games solved: Now and in the future*, Artificial intelligence 134 277-311.

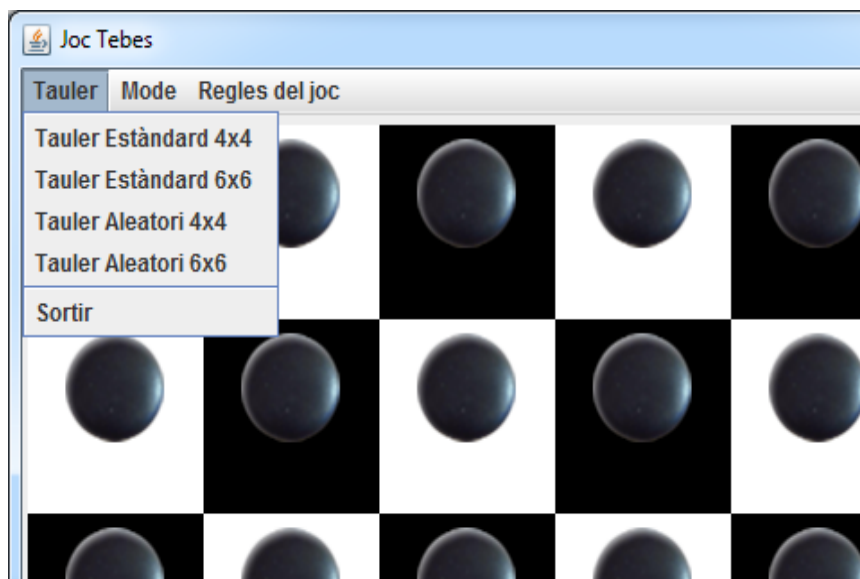
## ANNEX

## Manual de l'aplicació

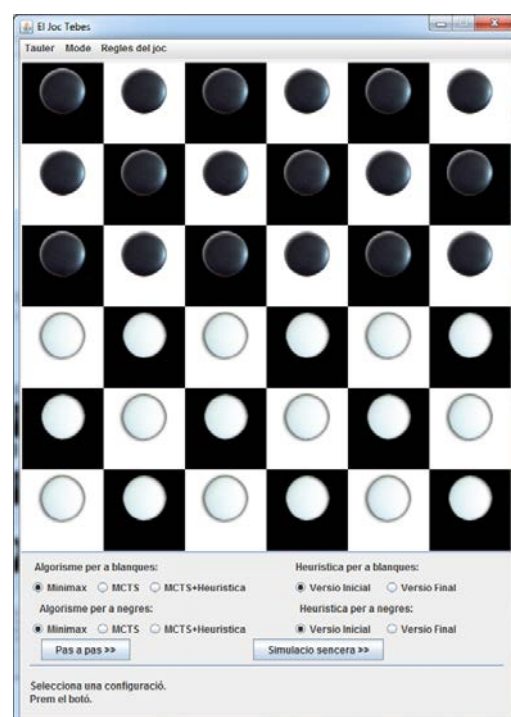
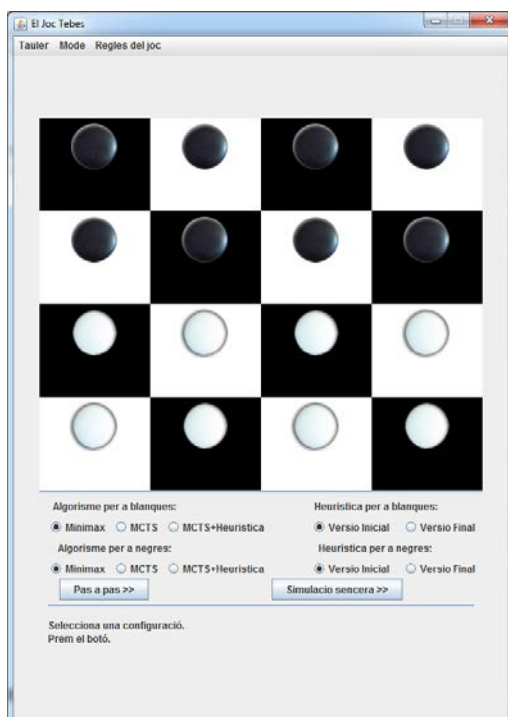


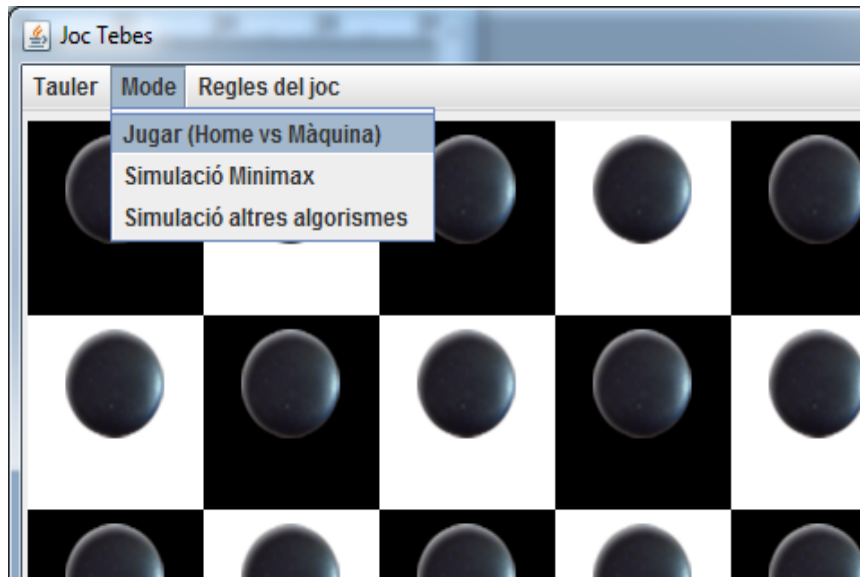
Per començar, en el menú podem trobar les regles del joc per així fer-nos una idea de com funciona, si premem aquesta opció del menú trobarem la següent pantalla.





En el menú podem veure altres opcions, la primera és la de configuració del Tauler, on les opcions són varies, el tauler estàndard amb les seves mides 4x4 i 6x6 i el tauler aleatori també amb les seves mides 4x4 i 6x6.

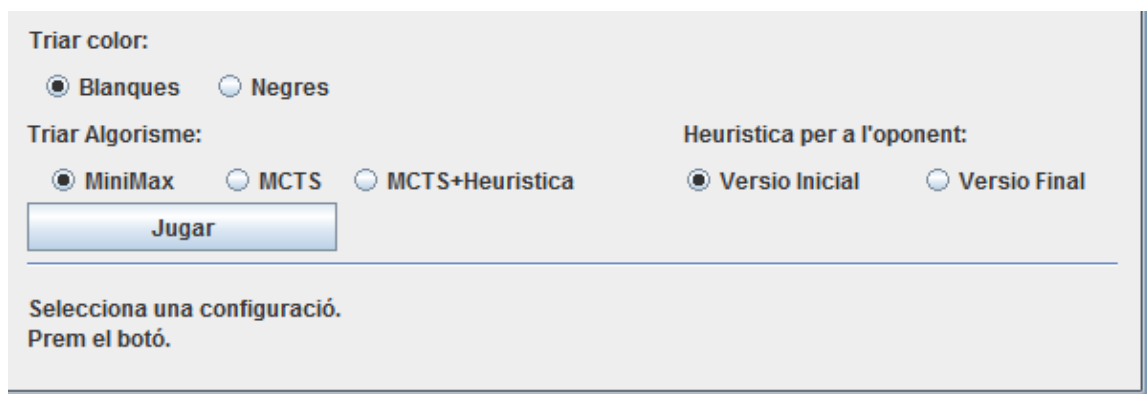




Un cop seleccionat el tauler, en l'opció de menú Mode, podem trobar les diferents maneres d'utilitzar l'aplicació, que són:

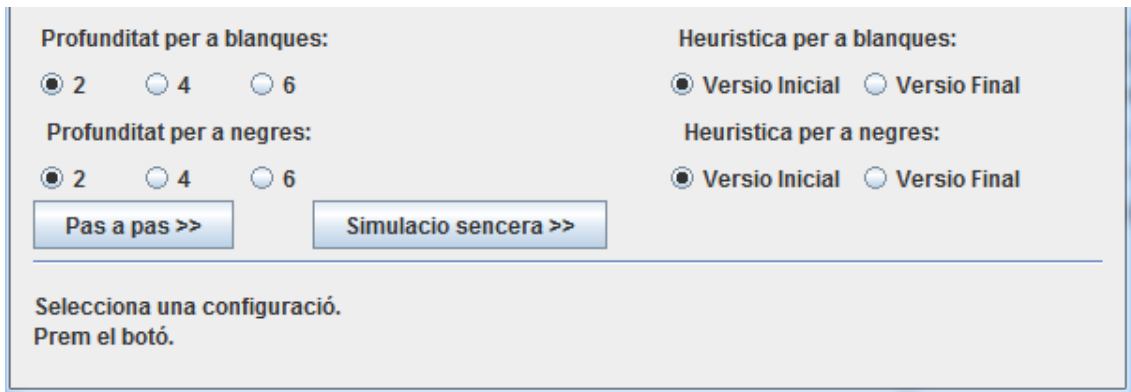
- La primera, és Jugar, on pots jugar contra la màquina amb els diferents algorismes i heurístiques.
- La segona opció, amb la qual pots fer simulacions de l'algorisme Minimax amb diferents configuracions.
- La tercera opció, amb la qual pots fer simulacions de partides amb els diferents algorismes i heurístiques implementades.

### Opció de Jugar



En aquest mode es pot triar el color amb el que es jugarà ( blanques o negres ), l'algorisme de l'adversari, així com també l'heurística.

### Opció de simulació del Minimax



Profunditat per a blanques:  
 2    4    6

Profunditat per a negres:  
 2    4    6

Heurística per a blanques:  
 Versio Inicial    Versio Final

Heurística per a negres:  
 Versio Inicial    Versio Final

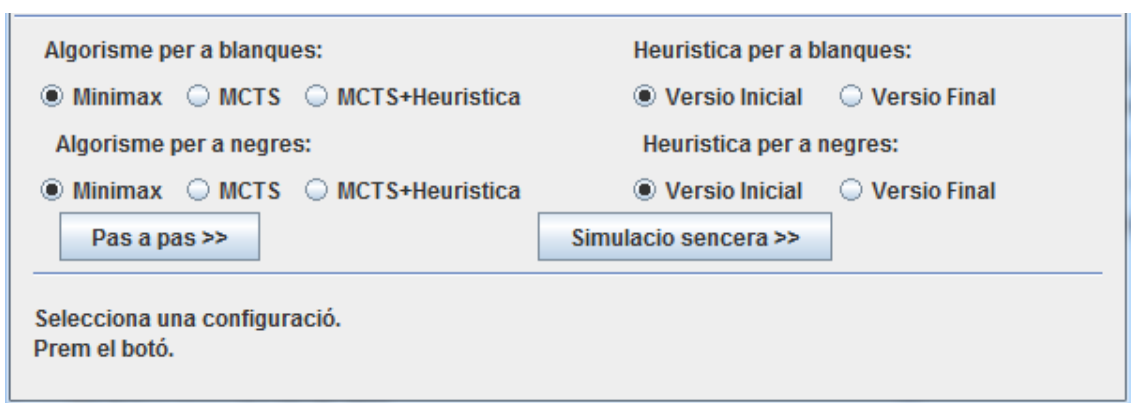
---

Selecciona una configuració.  
Prem el botó.

En aquest mode, podem enfrontar dos algorismes minimax, podem triar les diferents característiques per a les blanques i per a les negres, com per exemple:

- La profunditat 2,4, o 6. ( s'ha de tenir en compte que la profunditat 6 va una mica més lenta a causa de les dimensions de l'arbre generat) .
- L'heurística, tant per les negres com per les blanques
- El tipus de simulació, el mode pas a pas ens simula la partida torn per torn, amb el que has de prémer el botó a cada pas, i la simulació sencera, on la simulació no s'atura fins a arribar al final de la partida.

### Opció de simulació d'altres algorismes



Algorisme per a blanques:  
 Minimax    MCTS    MCTS+Heurística

Algorisme per a negres:  
 Minimax    MCTS    MCTS+Heurística

Heurística per a blanques:  
 Versio Inicial    Versio Final

Heurística per a negres:  
 Versio Inicial    Versio Final

---

Selecciona una configuració.  
Prem el botó.

Per últim ens trobem el mode simulació d'altres algorismes, on es pot enfrontar els diferents algorismes creats amb les diferents heurístiques també creades.

Els botons, com en el mode anterior, ens donen l'opció de simular la partida pas a pas, o bé, fer una simulació sencera fins al final de la partida.