



Joc de proves del Joc d'Escacs Genèrics

PROJECTE DE PROGRAMACIÓ

GRAU EN ENGINYERIA INFORMÀTICA

*David Pérez Sánchez, u1959299
Laura Galera Alfaro, u1959173*

1 de juny de 2020

Índex

1	Introducció	3
2	Proves en la càrrega de dades	3
2.1	Fitxer de regles	3
2.1.1	Format del fitxer	3
2.1.2	Dimensions del tauler	4
2.1.3	Límits per acabar en taules	5
2.1.4	Definició de peces	5
2.1.5	Rei	5
2.1.6	Moviments de les peces	6
2.2	Fitxer de desenvolupament	7
2.2.1	Format del fitxer	7
2.2.2	Fitxer de regles vàlid	7
2.2.3	Coherència del torn	7
2.2.4	Posicions inicials	7
2.2.5	Correctesa de les jugades	10
2.2.6	Continuar en el mateix punt	11
3	Proves de la dinàmica de joc	11
3.1	Dinàmica de torns	11
3.2	Desfer i refer jugades	12
3.3	Tirades especials	14
3.4	Taules involuntàries	14
4	Proves de les interfícies	14
4.1	Text	15
4.2	Gràfica	17
5	Proves de desar una partida	19
5.0.1	Iniciar i ajornar	19

5.0.2	Carregar i finalitzar	20
5.0.3	Carregar i ajornar	20
5.0.4	Iniciar i finalitzar	20
6	Proves en la logística del Tauler	21
6.1	Mobilitat de les peces	21
6.1.1	Destins dels reis	21
6.1.2	Destins inviables	23
6.1.3	Destins en situació d'escac	24
6.1.4	Peces que protegeixen al rei	25
6.2	Enrocs entre les peces	26
6.3	Promocions	27
6.4	Detecció d'escac	28
6.5	Detecció d'escac i mat	30
6.6	Detecció de rei ofegat	31
7	Proves en el jugador CPU	33
7.1	Selecció de la millor jugada	33
7.2	Selecció de taules	34
7.3	Selecció de promoció	36
7.4	Jugador humà vs Jugador CPU	37
7.5	Jugador CPU vs Jugador CPU	38
7.5.1	Temps de resposta	38
A	Fitxer de regles	39
B	Fitxer de desenvolupament d'una partida ajornada	44
C	Partida entre dos jugadors CPUs	48

1 Introducció

2 Proves en la càrrega de dades

Una part imprescindible del programa és la càrrega de les dades a la partida. Si aquesta no funciona com s'espera, la resta de parts de l'aplicació és difícil que es comportin com haurien de fer-ho. Per tant, és important comprovar bé que es llegeixen correctament les dades dels fitxers JSON i que s'introdueixen a la partida de forma correcta. Així doncs, s'han realitzat diferents proves de format i de correctesa de les dades per verificar el bon funcionament de la càrrega.

2.1 Fitxer de regles

2.1.1 Format del fitxer

El format del fitxer es comprova per totes les parts del JSON, per tant, si hi ha un error tipogràfic en algun dels noms dels camps, si en falta algun, si n'hi ha de més... es notifica a l'usuari que hi ha un error de format. A continuació es mostren diversos exemples de proves realitzades a partir d'un fitxer de regles correcte, però modificant algunes parts per provar si es comprova bé el format a totes elles. En totes les proves el resultat obtingut és:

Error: format del fitxer incorrecte

Exemples:

Un moviment definit amb tres elements en comptes de 4.

```
[  
    1,  
    1,  
    1  
]
```

Una dada que hauria de ser un nombre enter s'ha escrit com una cadena de caràcters.

```
"nFiles": 8,  
"nCols": "8",  
"peces": [
```

Un error tipogràfic al nom d'un camp ("asdf" en comptes de "nFiles").

```
"asdf": 8,  
"nCols": 8,  
"peces": [
```

Una peça sense moviments inicials.

```
"nom": "ALFIL",
"simbol": "A",
"imatgeBlanca": "alfilB.png",
"imatgeNegra": "alfilN.png",
"valor": 3,
"moviments": [
  [
    "a",
    "a",
    1,
    0
  ],
  [
    "a",
    "-a",
    1,
    0
  ]
],
"promocio": false,
"invulnerabilitat": false
```

2.1.2 Dimensions del tauler

Pel que fa a les dimensions del tauler, s'ha provat d'iniciar una partida amb un fitxer de regles que tenia el nombre de files i columnes per sobre i per sota dels límits permesos (de 4 a 16). A continuació es mostren uns exemples amb el resultat obtingut:

"nFiles": 50,	"nFiles": 1,
"nCols": 50,	"nCols": 1,

Resultat obtingut a ambdues proves:

Error en les mides del tauler

Cal destacar que, quan hi ha més peces assignades a les posicions inicials que caselles disponibles al tauler es mostra el següent missatge:

Error: No hi poden haver més peces que caselles.

2.1.3 Límits per acabar en taules

Només s'accepten límits d'escacs i de taules per inacció més grans que 1, ja que, si fossin menors, no es podria jugar. Aleshores, quan un fitxer no compleix aquesta especificació, es mostra per pantalla:

```
Error: els límits han de ser nombres enters més grans que 1
```

2.1.4 Definició de peces

Les peces poden ser definides en el fitxer de regles. S'hi pot personalitzar el nom, símbol, imatges, valor, moviments, capacitat per promocionar i invulnerabilitat. En general, l'usuari pot crear les peces que vulgui, però hi ha uns límits. Per exemple, només es poden definir 25 peces (com a màxim) i, per tant, només se'n carregaran aquesta quantitat, ignorant la resta (sense prioritat ni ordre).

Un aspecte important és que no hi poden haver peces amb nom repetit i, en cas d'entrar-ne al fitxer, es mostra:

```
Error: no hi poden haver peces amb el mateix nom
```

Tampoc hi poden haver peces referenciades a la posició inicial que no hagin estat definides i, en cas contrari, es mostra:

```
Error: a les posicions inicials només poden haver-hi peces definides
```

2.1.5 Rei

El Rei és la peça més especial del tauler, pel que es tracten a part les proves efectuades amb aquesta peça. És imprescindible que a tot joc d'escacs hi hagi un rei definit i que a cada bàndol se n'hagi assignat un. Per tant, si no hi ha cap definició d'una peça que es digui "REI", es mostra l'error:

```
Error: hi ha d'haver obligatòriament una peça "REI" definida
```

Si hi ha un rei definit però a les posicions inicials de les peces no es referencia cap Rei o n'hi ha més d'un, es mostra el missatge:

```
Error: cada bàndol ha de tenir un "REI"
```

A més, perquè es pugui jugar correctament, el Rei ha de tenir el valor més alt. Altrament es mostra:

Error: la peça "REI" ha de tenir el valor més alt

Tampoc ha de poder-se promocionar, sinó un jugador es quedarà sense Rei, cosa que no pot succeir, ja que la partida quedaria en un estat il·legal. La invulnerabilitat tampoc pot ser una característica del rei. En cas de no complir aquestes condicions, es mostra, respectivament:

Error: El REI no ha de poder promocionar

Error: El REI no pot ser invulnerable

2.1.6 Moviments de les peces

A l'hora de definir els moviments d'una peça, és fàcil que s'escriguin moviments o moviments inicials que continguin un mateix destí. Com per exemple $[a, 1, 1, 1]$ i $[n, n, 2, 1]$. En aquest cas només es considerarà un d'ells, mostrant per terminal quin s'ignora amb un missatge com el següent:

Error: el moviment x està solapat, s'ignora

On "x"és la definició del moviment ignorat.

A més, si és un moviment combinat la peça ha de poder saltar sense capturar, pel que si s'intenta carregar un moviment no vàlid segons aquest criteri com:

$[2, 1, 1, 0]$

Es mostra el missatge següent:

El moviment x no és vàlid.

On "x"és la definició del moviment erroni.

El missatge anterior també es mostra si alguna component del moviment no coincideix amb els valors permesos per cadascuna. Per exemple, el moviment $[z, -a, 1, 3]$ donaria error.

Una altra situació possible és que una peça no tingui cap moviment, pel que si s'introdueix una llista de moviments buida es mostra:

Error: Totes les peces han de tenir un moviment

2.2 Fitxer de desenvolupament

2.2.1 Format del fitxer

De la mateixa forma que succeeix amb el fitxer de regles, en el de desenvolupament també es comproven tots els camps seguint el mateix esquema, pel que les proves donen els mateixos resultats que a l'apartat [2.1](#).

2.2.2 Fitxer de regles vàlid

El fitxer de regles de la partida està referenciat al de desenvolupament. Així l'usuari pot introduir només el nom d'un fitxer i tota la informació es carrega directament. Llavors, és imprescindible que la ruta sigui correcta. En cas contrari es mostrerà el missatge:

Interfície gràfica

nomDelFitxerRegles.json
Error obrint el fitxer

Interfície text

Error: no s'ha pogut obrir el
fitxer nomDelFitxerRegles.json

2.2.3 Coherència del torn

Per tal de poder carregar la partida i continuar-la des del punt on es va ajornar, cal que s'especifiqui quin és el jugador que tira en primer lloc. Però, si la primera tirada no coincideix amb el jugador, podria significar que s'ha perdut informació de les primeres tirades i, per tant, la partida no seria vàlida. Així doncs, si el camp `properTorn` és `BLANQUES` i la primera tirada és de `NEGRES`, es mostra l'error:

Error: Els torns no són coherents

2.2.4 Posicions inicials

Pot succeir que l'usuari vulgui definir les posicions inicials de les peces per a cada bàndol. Per fer-ho, haurà d'indicar-ho al fitxer de desenvolupament. Això pot ser un problema si accidentalment se situen dues peces a la mateixa posició. Per exemple:

```
{  
  "pos": "a2",  
  "tipus": "PEO",  
  "moguda": false  
}  
  
{  
  "pos": "a2",  
  "tipus": "TORRE",  
  "moguda": true  
}
```

Aquestes dues peces tenen una definició de la posició inicial correcta, però entre elles ocasionen un conflicte: s'han posicionat a la mateix casella. Per tant, es mostra per pantalla:

Error: No hi pot haver més d'una peça a la mateixa casella

Com és d'esperar, si es vol col·locar una peça fora del tauler, es mostrerà:

La posició x no existeix al tauler

On **x** és la posició errònia.

També cal tenir en compte a l'hora de definir les posicions que les peces referenciades han d'estar descrites al fitxer de regles, altrament es mostrerà:

Error: a les posicions inicials només hi poden haver peces descrites al fitxer de regles

Amb aquestes proves també s'ha comprovat que quan es vol iniciar una partida amb les posicions inicials personalitzades es carreguen les del fitxer de desenvolupament a la partida i no pas les que s'especifiquen al fitxer de regles. Per verificar-ho, s'ha mostrat el contingut de les posicions inicials utilitzant els següents fitxers com a prova:

Fitxer de regles

```
{  
  "nFiles": 8,  
  "nCols": 8,  
  "peces": [  
    {  
      "nom": "REI",  
      "simbol": "R",  
      "imatgeBlanca": "reiB.png",  
      "imatgeNegra": "reiN.png",  
      "valor": 100,  
      "moviments": [  
        [  
          1,  
          0,  
          1,  
          0  
        ]  
      ],  
      "movimentsIniciais": [],  
      "promocio": false,  
      "invulnerabilitat": false  
    }  
  ],  
  "posInicial": [  
    "REI"  
  ],  
  "limitEscacsSeguits": 3,  
  "limitTornsInaccio": 40,  
  "enrocs": []  
}
```

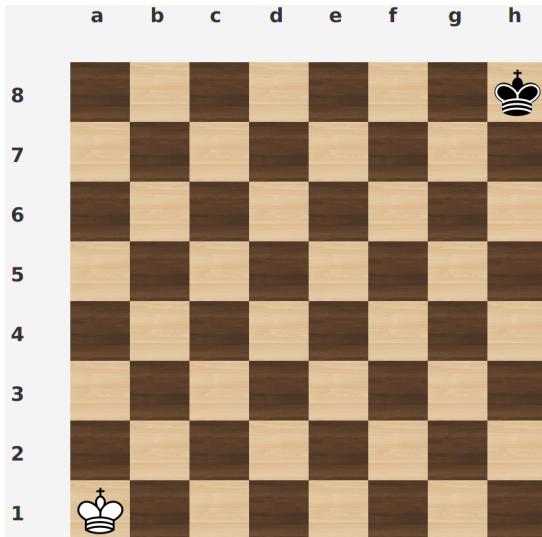
Fitxer de desenvolupament

```
{  
  "fitxerRegles": "regles.json",  
  "posIniBlanques": [  
    {  
      "pos": "a2",  
      "tipus": "REI",  
      "moguda": false  
    }  
  ],  
  "posIniNegres": [  
    {  
      "pos": "a7",  
      "tipus": "REI",  
      "moguda": false  
    }  
  ],  
  "proper_torn": "BLANQUES",  
  "tirades": [  
  ],  
  "resultat_final": ""  
}
```

Els resultats obtinguts són:

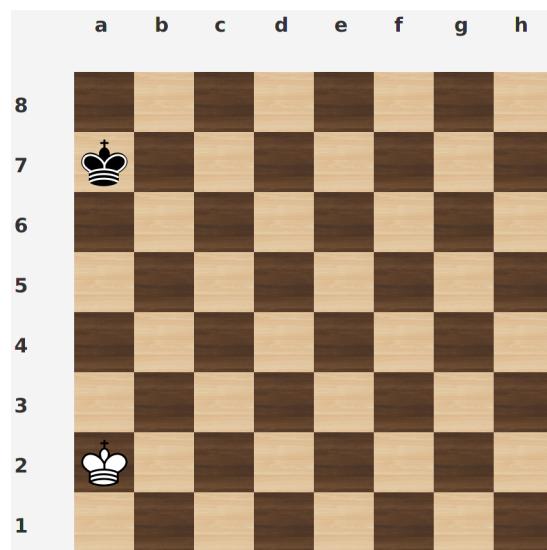
Iniciar:

{a1=REI, h8=REI}



Carregar:

{a2=REI, a7=REI}



Tal com era d'esperar, quan s'inicia la partida es llegeixen les posicions inicials del fitxer de regles i, quan es carrega, les del fitxer de desenvolupament.

2.2.5 Correctesa de les jugades

Quan es carrega una partida ajornada, es tornen a efectuar les tirades de nou. Llavors, es verifica que siguin possibles. En cas que hi hagi una tirada no vàlida es mostrerà un missatge d'error juntament amb el missatge que es mostraria si s'estigués jugant la partida. Si es canvia una tirada ordinària i es posa un destí no vàlid, pot aparèixer el següent missatge d'error:

Error en la càrrega del fitxer:

Error: no es pot fer la jugada ordinària

A més, cal destacar que es poden carregar partides ajornades, però no partides ja finalitzades. En cas de carregar una partida acabada, es mostra:

Error en la càrrega del fitxer:

Error: no es pot fer la jugada ordinària

2.2.6 Continuar en el mateix punt

Per corroborar que aquesta funcionalitat s'executa correctament, es prova de carregar la partida amb el fitxer de desenvolupament de l'annex B. El resultat obtingut és que la partida continua en el mateix punt on es va deixar, havent-se realitzat totes les jugades de nou i amb el torn del jugador que va ajornar la partida. Per tant, no s'ha alterat en cap aspecte el seu curs.

Cal destacar que si es prova de carregar un fitxer de desenvolupament d'una partida que ha finalitzat es mostra un error, ja que no és possible jugar. El missatge és el següent:

Error: la partida ja està acabada

3 Proves de la dinàmica de joc

Aquest apartat té la intenció de posar a prova el correcte funcionament d'una partida. Els punts a tractar són la dinàmica de torns, les capacitats de desfer i refer jugades, la realització de les tirades especials i les taules involuntàries.

3.1 Dinàmica de torns

Una partida es troba condicionada pel torn del jugador. El torn permet saber quines jugades són acceptades, doncs quan és el torn del jugador de blanques les peces negres no es poden moure i viceversa. També serveix per descriure el desenvolupament i el desenllaç del joc.

De fet, el torn es va alternant al llarg d'una partida, excepte quan es decideix desfer o refer jugades. Aquestes dues accions van deslligades del torn, és a dir, poden ser seleccionades per qualsevol dels dos jugadors en algun moment de la partida.

Així doncs, si l'última jugada ha estat feta pel jugador negre, el torn en aquest moment correspon al jugador de blanques. Ara bé, si algú decideix desfer, el torn serà recuperat pel de negres perquè la partida haurà retrocedit fins a aquell punt:

Torn: BLANQUES → Desfer 1 → es desfa 1 jugada → Torn: NEGRES

En canvi, si es decideixen desfer dues jugades, es desfarà primer la del jugador negre i després la del jugador blanc. Per tant, el torn continuarà essent del jugador blanc perquè just en aquell moment, quan les dues jugades no existien, ell era qui tenia el torn:

Torn: BLANQUES → Desfer 2 → es desfan 2 jugades → Torn: BLANQUES

La mateixa metodologia és usada en relació amb refer tirades i el torn dels jugadors.

3.2 Desfer i refer jugades

Una funcionalitat important de la partida és l'opció de desfer i refer jugades. Com bé indica el nom, aquestes accions permeten desfer n jugades i, per tant, retrocedir a un tauler anterior, o bé refer n jugades i recuperar el que existia abans de desfer. Llavors, és evident que les jugades desfetes no poden aparèixer al fitxer de desenvolupament.

Tanmateix, hi ha un límit per desfer i refer, doncs no és possible desfer jugades en un tauler on mai s'ha efectuat cap jugada. De la mateixa manera, no és possible refer jugades si mai se n'ha desfet cap. Per aquesta raó, quan el jugador entra un valor i aquest és superior al límit, es desfan o refan totes les jugades possibles i s'acaba mostrant un missatge informatiu juntament amb el tauler resultant:

NEGRES:

desfer 100

No hi ha res per desfer

Tauler després de desfer el màxim

BLANQUES:

refer 100

No hi ha res per refer

Tauler després de refer el màxim

Un punt clau que no es pot oblidar és l'opció de desfer i refer quan es carrega una partida ajornada. Si un jugador ha decidit carregar una partida que havia ajornat fa 1 hora, ha de poder desfer i refer les jugades com ell desitgi.

A la següent imatge es pot veure què passa si es carrega el fitxer ajornat de l'annex B i s'opta per desfer a l'instant:

ESCAC AL REI DE BLANQUES!

8		a		r		c		a	
	+-----+		+-----+		+-----+		+-----+		+-----+
7		c		p		p		p	
	+-----+		+-----+		+-----+		+-----+		+-----+
6									
	+-----+		+-----+		+-----+		+-----+		+-----+
5									
	+-----+		+-----+		+-----+		+-----+		+-----+
4									
	+-----+		+-----+		+-----+		+-----+		+-----+
3									
	+-----+		+-----+		+-----+		+-----+		+-----+
2		D				P		P	
	+-----+		+-----+		+-----+		+-----+		+-----+
1				C		R		A	
	+-----+		+-----+		+-----+		+-----+		+-----+
	a	b	c	d	e	f	g	h	

BLANQUES:

desfer 1

8		d		a		r		c	
	+-----+		+-----+		+-----+		+-----+		+-----+
7		c		p		p		p	
	+-----+		+-----+		+-----+		+-----+		+-----+
6									
	+-----+		+-----+		+-----+		+-----+		+-----+
5									
	+-----+		+-----+		+-----+		+-----+		+-----+
4									
	+-----+		+-----+		+-----+		+-----+		+-----+
3									
	+-----+		+-----+		+-----+		+-----+		+-----+
2		D				P		P	
	+-----+		+-----+		+-----+		+-----+		+-----+
1				C		R		A	
	+-----+		+-----+		+-----+		+-----+		+-----+
	a	b	c	d	e	f	g	h	

NEGRES:

S'ha des fet la jugada de la dama negra que anava de *a8* fins *h1*.

3.3 Tirades especials

És important comentar les tirades especials, ja que són gestionades directament per la partida. Aquestes tirades inclouen taules sol·licitades, taules acceptades, taules denegades, rendició i ajornament.

L’ajornament i la rendició indiquen a la partida que aquesta ha de ser guardada perquè no es seguirà amb el joc. En canvi, quan es realitza una sol·licitud de taules el jugador obligatòriament ha de respondre-les amb *aceptar* o *denegar*. Si s’accepten, la partida finalitza i es guarda el desenvolupament, però si les denega es continua amb la partida i la sol·licitud no apareixerà al fitxer de guardat quan s’acabi o s’ajorni la partida.

3.4 Taules involuntàries

A part de les taules sol·licitades, existeixen dos tipus més d’empat però involuntaris: taules per escac continu i taules per inacció.

Les taules per escac continu es donen si un dels jugadors sobre passa el límit d’escacs continus establerts. És important comprovar que si una jugada comporta un escac però aquesta es desfà, llavors el comptador d’escacs continus del jugador també canvia. El mateix amb refer.

Les taules per inacció es donen quan els dos jugadors porten masses jugades sense capturar cap peça de l’enemic i se sobrepassa el límit permès. Igual que amb l’anterior, cal tenir present els efectes de desfer i refer jugades que no capturen cap peça.

4 Proves de les interfícies

Amb les proves de les interfícies el que es pretén és demostrar si la interacció entre els usuaris i l’aplicació és còmode i a la vegada correcte. Per tant, es vol comprovar si els missatges enviats i rebuts per cada banda es mostren i s’interpretan com cal.

A més, també és important trobar les diferències i les semblances entre una interfície de text i una interfície gràfica. El fet de treballar amb una interfície gràfica, que es mou per esdeveniments, ha permès afegir certes facilitats que en una interfície de text no és possible. Ara bé, totes dues han de garantir uns requisits i tenir trets en comú.

És recomanable haver llegit el manual d’usuari abans de seguir amb les proves.

4.1 Text

En primer lloc és indispensable per a la interfície que aquesta demani les dades per a continuar o iniciar una partida, amb els jugadors desitjats:

Vols iniciar o carregar una partida? [i/c] :

. . .

Jugador de blanques CPU? [s/n] :

Un cop fet això, un altre punt clau, és mostrar el tauler al principi de la partida i després d'efectuar cada jugada, amb el torn corresponen:

8		t			d		a	c	t					
7		p		p		r		A		p		p		
6					p									
5							C							
4						c		P						
3				C										
2		P		P		P			P		P		P	
1		T		A		a			T		R			
	a	b	c	d	e	f	g	h						

BLANQUES:

També ha de comunicar a l'usuari si la jugada que intenta fer no és possible i llavors esperar que n'entri una de vàlida:

Error: no es pot fer la jugada ordinària

Error: no es pot fer la jugada enroc

Com bé s'ha comentat, els jugadors han de poder realitzar tirades especials com són ajornar, rendir-se o demanar taules. A més, quan es demanen taules, el següent jugador es veu obligat a respondre-les:

NEGRES:

El jugador de BLANQUES t'ha demanat taules. Acceptes? [acceptar/denegar]: meow

Has d'acceptar o denegar les taules abans de continuar

NEGRES:

El jugador de BLANQUES t'ha demanat taules. Acceptes? [acceptar/denegar]:

En cas que el valor entrat pel jugador no correspongui a cap acció especificada cal mostrar un missatge informatiu:

BLANQUES:

miau

Format de la jugada incorrecte

També, quan una jugada comporta una promoció es mostra un missatge amb les peces disponibles per a promocionar. El nom d'una peça que no existeix és considerat un error i cal tornar a entrar un valor:

La peça "CAVALL" que es troba a la posició "b8" pot promocionar.

[ALFIL, CAVALL, PEO, TORRE, DAMA]

Introduceix el nom d'una peça (mateix nom per cancel·lar).

DOFÍ

Cal escollir una de les peces mostrades.

Durant el transcurs de la partida, com també quan es carrega, es mostren els missatges d'escac si n'hi ha. A més, sempre que finalitza una partida, primer es comunica perquè ha finalitzat (escac i mat, taules per inacció, etc.), en segon lloc es mostra el tauler final, a continuació el resultat d'acabament i per últim es demana el nom del fitxer on es guardarà el desenvolupament:

ESCAN I MAT AL REI DE NEGRES!

Estat final del tauler:

8	t			d		a	c	t
7	p	p	p		r	A	p	p
6				p				
5				C	C			
4				c	P			
3								
2	P	P	P	P		P	P	P
1	T		A	a		T	R	
	a	b	c	d	e	f	g	h

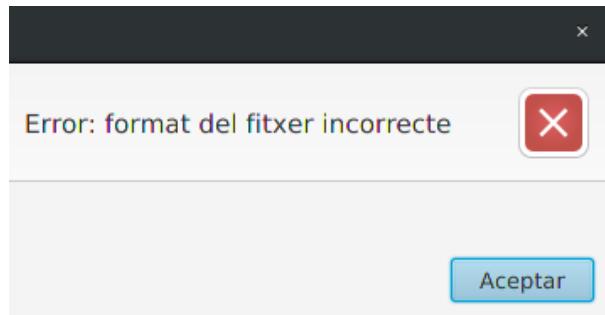
BLANQUES GUANYEN

Entra el nom del fitxer on es guardarà la partida:

Per últim, quan es tracta de desfer, refer, ajornar, taules o rendir-se, en el cas que juguin dos jugadors CPU no es poden demanar. Això és així perquè la interfície no espera rebre cap valor de l'usuari mentre es desenvolupa la partida. Això és diferent quan es tracta de la interfície gràfica.

4.2 Gràfica

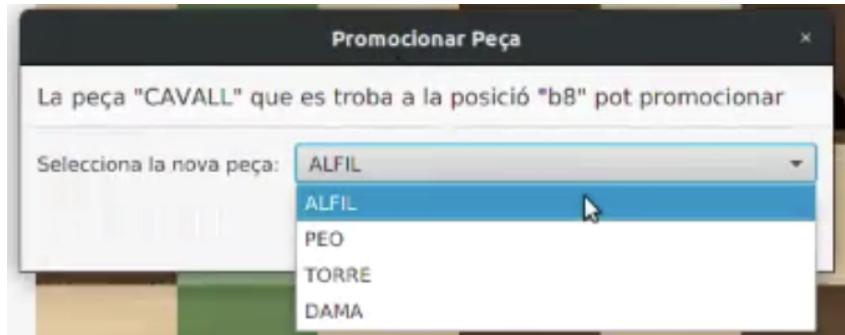
Respecte a la interfície gràfica, les dades de la partida es demanen per mitjà d'una finestra de benvinguda. Si es detecta algun tipus d'anomalia amb el fitxer entrat, es mostra un missatge d'error com el següent:



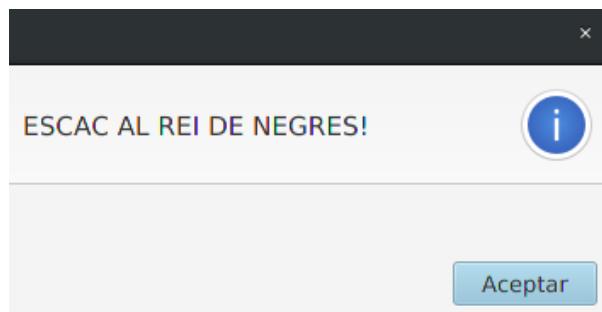
Tot seguit apareix la finestra de joc amb el tauler, els torns i els botons. Els botons de la barra lateral existeixen en funció dels tipus de jugadors -humà o CPU- que disputen

la partida. A partir d'aquí es poden començar a efectuar les jugades. Si alguna jugada no és possible, el moviment s'avorta.

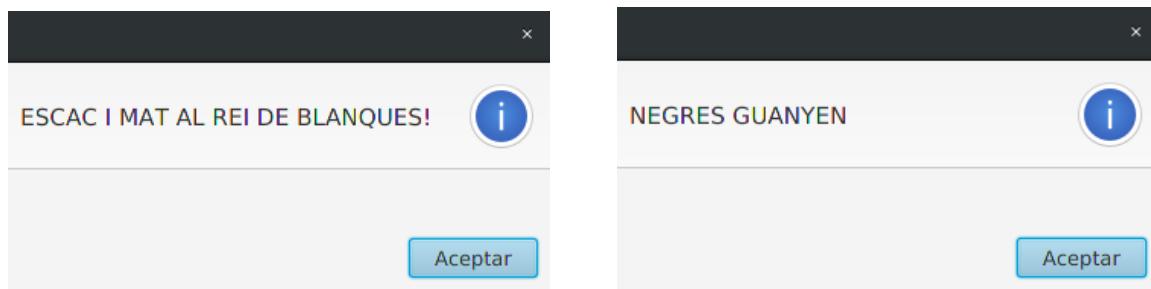
A més, després d'efectuar una jugada ordinària existeix la possibilitat que la peça pugui promocionar. En aquest cas, es mostra una finestra que conté un desplegable on només es poden seleccionar les peces a què es pot promocionar:



A mesura que es realitzen les jugades es pot donar el cas que una d'elles posi en situació d'escac al rei contrari. Cal destacar que aquesta situació també pot donar-se tan bon punt es carrega la partida. Llavors, es mostra un emergent com el següent:



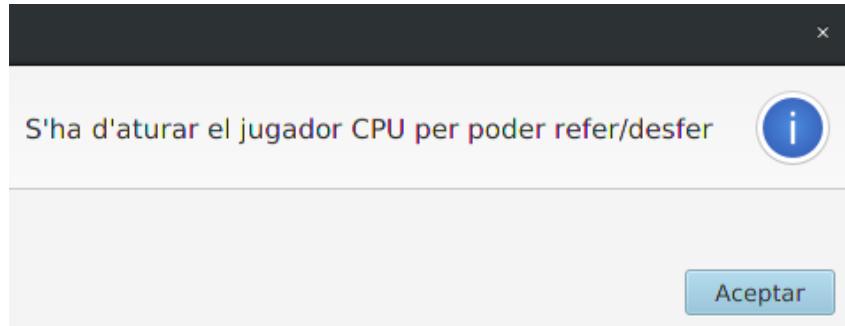
Quan una jugada comporta la finalització de la partida, s'informa el jugador de la raó que ha posat fi al joc per mitjà d'una finestra. Tot seguit, quan es tanca, també es mostra el resultat final de la partida i es procedeix a guardar el fitxer:



Quan es realitza una operació de desfer o refer, es modifica l'estat de la partida, incloent-hi el tauler. Si algun dels jugadors és CPU, el càcul de la seva tirada s'executa en segon pla en un altre procés. Ens vam trobar amb un problema relacionat amb el paral·lelisme:

l'estat del tauler es modificava mentre la CPU decidia la tirada. Aleshores, no es podia assegurar que la tirada calculada per la CPU fos correcta, ja que el tauler agafat per realitzar els càlculs no coincidia amb el de la partida.

Per aquesta raó, vam haver d'implementar una mena de semàfors que permeten a l'usuari refer i desfer sense cap problema. Això ho vam fer afegint uns botons que aturen el càlcul de la CPU un cop s'aplica la jugada pendent i un altre botó per reprendre l'activitat de la CPU. A continuació hi ha el missatge que es mostra quan s'intenta desfer mentre la CPU està calculant:



Un cop implementada aquesta solució les accions de desfer i refer s'executen tal com és d'esperar.

5 Proves de desar una partida

Si la partida finalitza o s'ajorna, l'usuari pot guardar-ne el desenvolupament. És important que les dades del fitxer siguin correctes i s'hi posin en ordre, ja que s'hi ha de poder veure el transcurs de la partida. A més, si s'havia ajornat, s'ha de poder tornar a carregar i continuar en el mateix punt on es va desar el progrés. Aleshores, s'han fet diverses proves per assegurar-ne el bon funcionament utilitzant el fitxer de regles de l'annex A.

5.0.1 Iniciar i ajornar

Un cop iniciada la partida, es prova a finalitzar-la després d'efectuar unes quantes jugades. El resultat és el que es mostra a l'annex B, on es pot veure que s'escriuen correctament les tirades de tots els tipus (també les promocions) i que la causa de finalització de la partida és:

```
"resultat_final": "PARTIDA AJORNADA"
```

5.0.2 Carregar i finalitzar

Amb aquesta prova es veu com s'ignora la tirada d'ajornament i com s'acumulen les noves tirades a les que ja hi havia al fitxer. També apareix la causa de finalització de la partida correctament.

5.0.3 Carregar i ajornar

Un cop carregada la partida, es prova de finalitzar-la sense efectuar cap jugada. Això permet veure si la tirada d'ajornament s'ignora i si les tirades definides tornen a escriure's correctament. El fitxer de sortida, un cop s'ha ajornat novament, és idèntic al de l'annex.

5.0.4 Iniciar i finalitzar

Aquí s'ha provat d'iniciar una partida i jugar-la fins al final. Per agilitzar el procés i per tenir un nombre de tirades significatiu, s'ha fet jugar CPU vs CPU. El resultat es troba a l'annex C i mostra com hi ha constància de totes les tirades, de la causa de finalització de la partida, de les posicions inicials, etc. El resultat de la partida s'anota correctament:

```
"resultat_final": "BLANQUES GUANYEN"
```

6 Proves en la logística del Tauler

En aquesta secció es desitja mostrar les proves que afecten la logística del tauler. En el nostre cas, el tauler no es resumeix en un simple espai de joc, sinó que té sota la seva responsabilitat tasques importants per al desenvolupament d'una partida. Ell és qui controla les capacitats de les peces que conté segons la seva disposició i, a més, analitza les situacions d'escac, escac i mat i rei ofegat per a comunicar-ho a les classes gestores. Llavors, és obvi que ha de dur a terme comprovacions que requereixen estructures i algoritmes de suport. A continuació es pretén posar-los a prova.

6.1 Mobilitat de les peces

Una de les principals responsabilitats del tauler és conèixer, segons els moviments de les peces, a quines posicions es poden moure cadascuna d'elles. Això és el que permet acceptar o denegar les jugades que els usuaris sol·liciten al llarg de la partida. Com és evident, el simple fet de conèixer els moviments d'una peça no és suficient. Apareixen múltiples factors com són la distribució d'altres peces, les propietats d'invulnerabilitat d'algunes, les dimensions del tauler, etc. que afecten la seva realització.

Cada vegada que s'experimenta un canvi al tauler se cerquen tots els possibles destins gràcies a un seguit d'algoritmes que ens permeten, per cada peça i moviment, els punts següents:

1. Trobar els destins segons l'origen i el moviment.
2. Descartar els destins que, efectuant el moviment, deixen al rei del bàndol en escac.
3. Per cada origen, guardar els seus destins accompanyats dels moviments.

Així doncs aquest apartat desitja presentar algunes de les proves que s'han efectuat sobre la mobilitat de les peces del tauler.

6.1.1 Destins dels reis

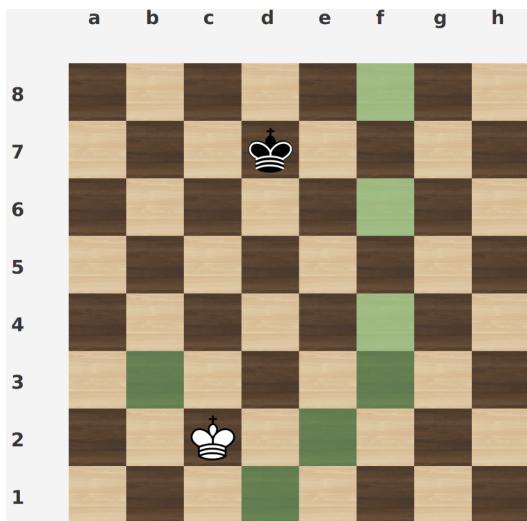
Per començar, el millor és treballar amb poques peces, que continguin moviments diversos, i que ens ajudin a veure ràpidament si els destins es calculen tal com s'espera. Per posar-se en context: es disposa d'un tauler inicial 8 x 8 amb dos reis situats a les caselles *c2*, el blanc, i *d7*, el negre. Aquests reis, a més, disposen de moviments especials que s'adjunten a continuació:

```

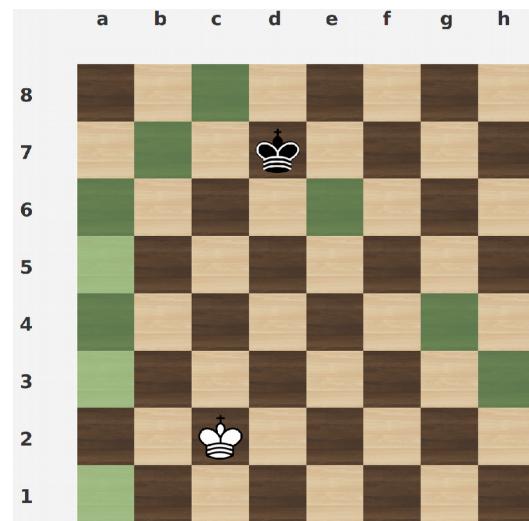
    "moviments": [
      [
        "a",
        "-a",
        0,
        2
      ],
      [
        "m",
        3,
        1,
        1
      ],
      [
        0,
        2,
        0,
        1
      ],
      [
        0,
        -2,
        1,
        2
      ]
    ],
  ]

```

Dit això, comprovem quins són els possibles destins de cada rei:



Destins rei blanc



Destins rei negre

Si ens centrem en el rei blanc, es pot veure que no pot accedir a les caselles $a2$, $f5$ i $f7$. L'explicació és perquè aquests destins els comparteix amb el rei negre i el rei negre té la capacitat de capturar en ells, sigui perquè capture a destí o perquè salta capturant. És important recordar que en els escacs està prohibit moure el rei a una casella on correria perill. També cal remarcar la correctesa dels destins; per exemple, el moviment $["a", -a", 0, 2]$ proporciona les posicions $b3$ i $d1$. Mentre que totes les caselles verdes que apareixen a la columna f corresponen al moviment $["m", 3, 1, 1]$.

Respecte al rei negre, succeeix un fet semblant. Sí que pot accedir a la casella $a4$ perquè el rei blanc no pot capturar a destí amb el moviment $["a", -a", 0, 2]$. Tanmateix, per exemple, no es pot moure fins a la $a2$ perquè és perillosa segons el moviment $[0, -2, 1, 2]$ de la peça blanca. També és interessant de veure com els moviments s'inverteixen correctament quan es tracta de peces negres. A més, en els dos casos, no hi ha hagut problema per trobar els destins dels moviments que involucren literals enters, com seria $["a", -a", 0, 2]$.

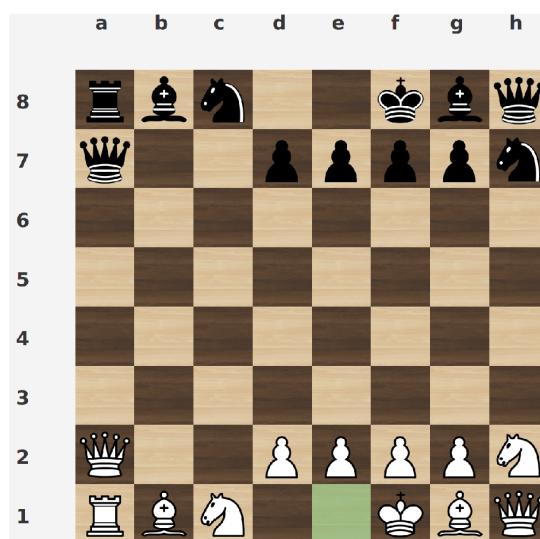
6.1.2 Destins inviables

Un cop vist com funcionen els moviments és hora de passar al següent nivell. Quan es treballa amb els destins cal comprovar un seguit de regles que afecten la captura i mobilitat de les peces.

Per comprovar aquests casos s'usarà el fitxer d'inici de partida que apareix a l'annex A:

1. No permetre els destins on es troba una peça del mateix bàndol

Com es mostra a la fotografia, encara que el rei blanc tingui la capacitat de capturar una casella endavant o a la dreta, aquests destins no són considerats perquè en les dites posicions hi ha peces del mateix color. També podem comprovar que l'enroc entre l'alfil i el rei no és possible perquè el rei acabaria a una posició de perill -el cavall $c8$ amenaça la casella de $d1$ -.



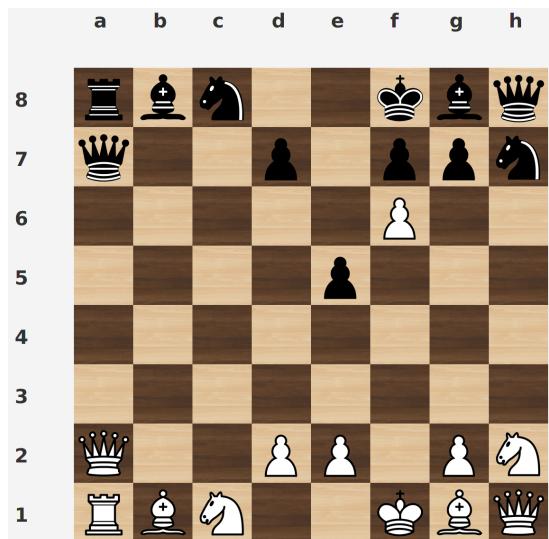
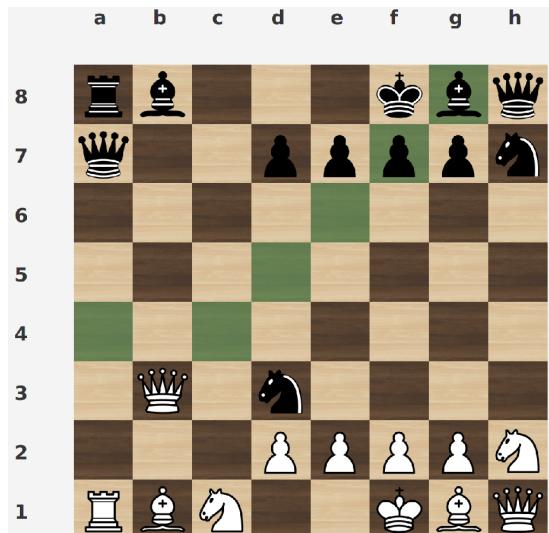
2. No permetre els destins on hi ha una peça contraria invulnerable

En el següent cas el cavall és invulnerable. Llavors, encara que la dama blanca tingui el moviment [0, "m", 2, 1], mai podrà capturar al cavall que hi ha a la posició g3. Per aquesta raó, la casella no està pintada de verd quan el ratolí es posiciona sobre la dama, ja que no és un destí possible.

A més, aquest exemple també ens serveix per comprovar que si el moviment obliga a capturar a destí però els possibles destins no contenen peces, llavors, es descarten.

3. No permetre els destins on hi ha una peça i el moviment no capture a destí

El cas més simple és el del peó. Aquesta peça compta amb el moviment [2, 0, 0, 2] però només es pot efectuar quan no hi ha una peça enemiga a la casella indicada. Així, doncs, quan es posiciona el ratolí sobre el peó de f6 el moviment anterior no es pot efectuar perquè la casella f8 està ocupada.



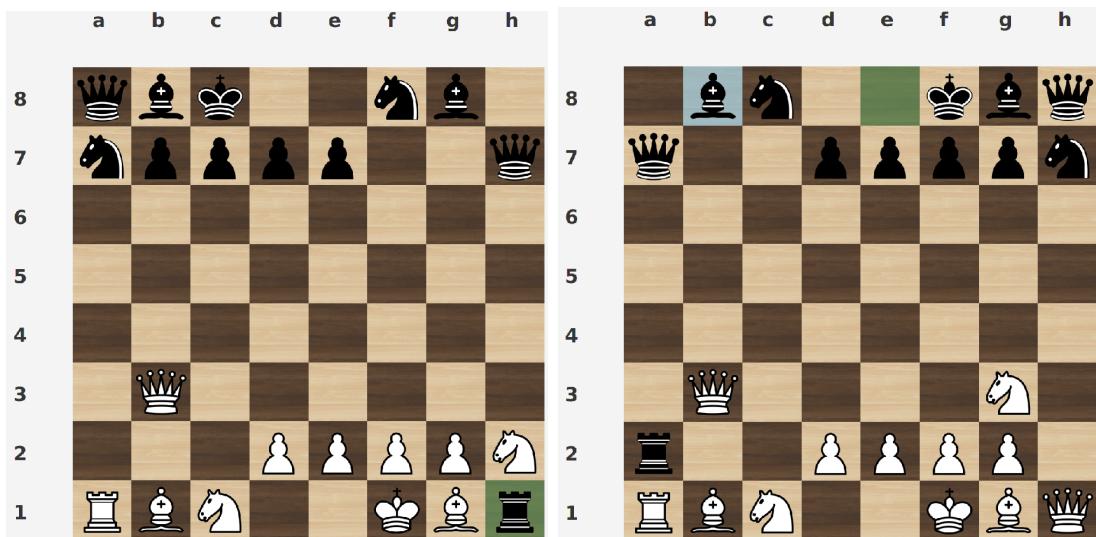
6.1.3 Destins en situació d'escac

Un requisit important és controlar els destins quan el rei del mateix bàndol es troba en escac. Com a regla dels escacs, quan el rei està sota amenaça, només es permeten els moviments que el treuen d'aquesta.

Perquè el rei que es troba en situació d'escac pugui fugir del perill, cal efectuar una d'aquestes tres jugades:

1. Que una peça mati a l'enemiga que amenaça al rei.
2. Que el rei pugui fugir a una posició segura -considerant també els enrocs-.
3. Que una altra peça s'interposi entre l'amenaça si l'enemiga no pot saltar -considerant també els enrocs-.

En les següents imatges es mostren algunes d'aquestes circumstàncies:



S'han usat les peces del fitxer A

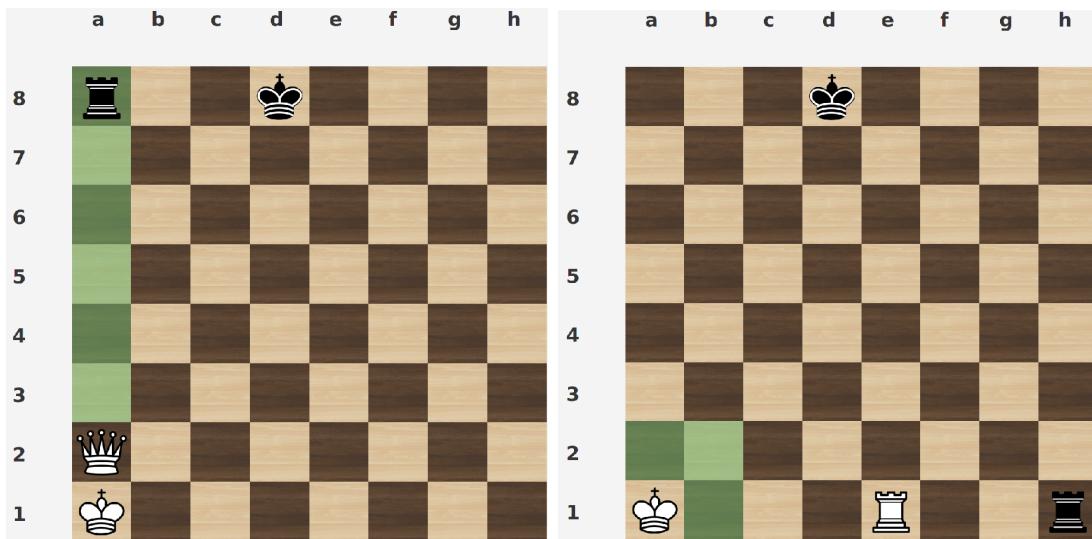
En la primera imatge el rei blanc està amenaçat per la torre negra. Només es pot salvar si la torre blanca mata a la negra.

En la segona imatge el rei negre està amenaçat pel cavall blanc de la casella $g3$. Es pot salvar fent un enroc amb l'alfil o movent-se a la posició $e8$.

6.1.4 Peces que protegeixen al rei

Una mesura presa en el desenvolupament del projecte ha estat ometre els destins de les peces que deixen al mateix rei en escac. De fet, la normativa d'escacs dicta que els moviments que deixen al rei del bàndol en escac han d'estar prohibits. Així, doncs, no s'ha permès moure a una casella una peça que, a conseqüència del moviment, deixaria el seu rei en estat d'escac.

Tot seguit es mostren dues situacions d'aquest tipus:



S'han usat les peces dels escacs convencionals

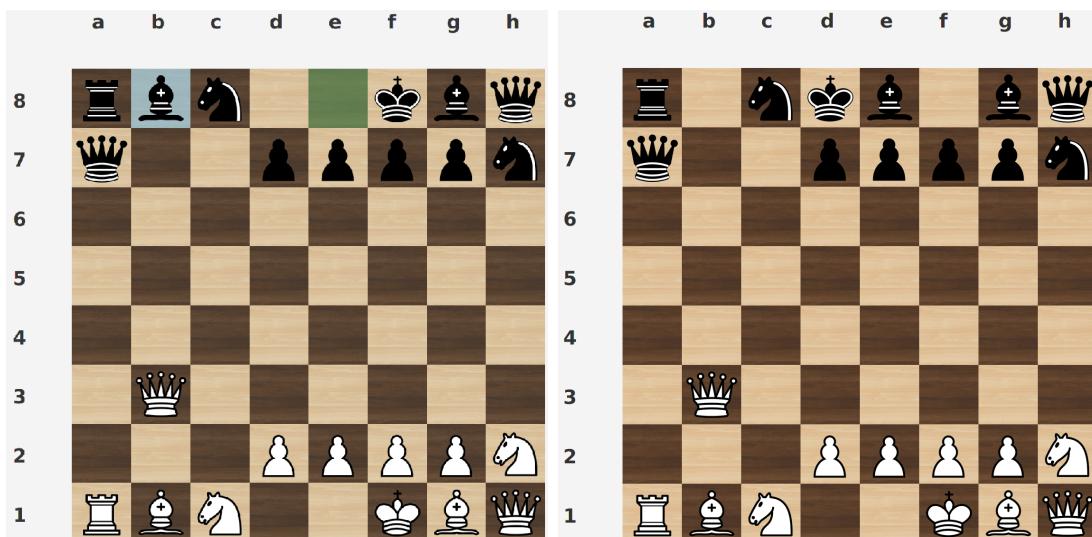
En el primer cas, es pot veure com l'únic moviment permès per a la reina és el de moure's al llarg de la columna, ja que qualsevol altre tipus de moviment deixaria el rei blanc en escac per part de la torre negra.

En el segon cas, el que succeeix és que el rei blanc no pot fer l'enroc amb la torre perquè acabaria a la posició *c1*, on seria amenaçat per part de la torre negra perquè la blanca es trobaria a *b1*.

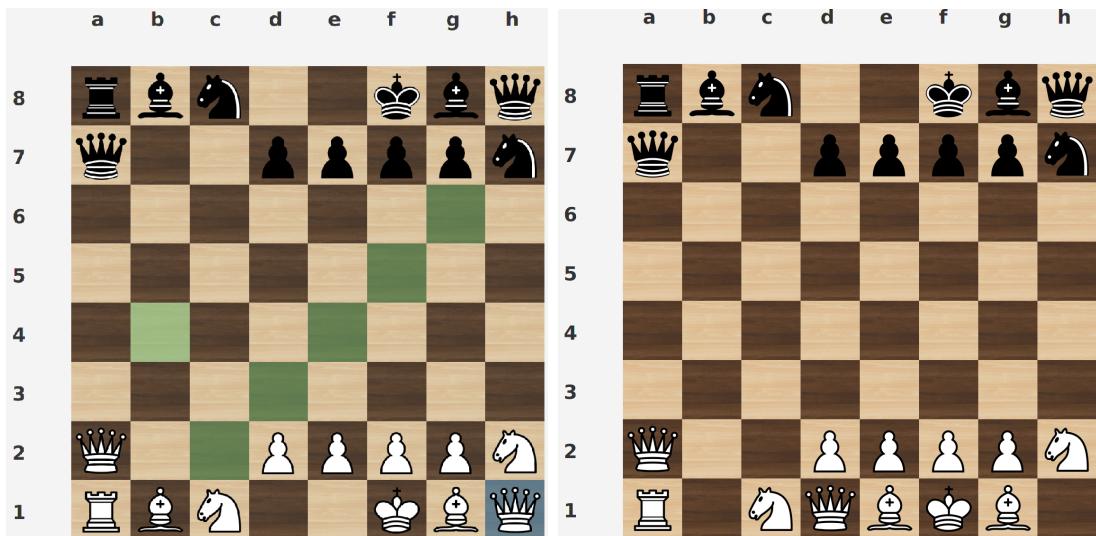
6.2 Enrocs entre les peces

Un dels moviments més emblemàtics dels escacs és l'enroc. En el nostre cas l'enroc té l'especialitat que permet fer-se entre tota mena de peces (incloses dues d' idèntiques), també permet acceptar que aquestes s'hagin mogut i que tinguin entre les caselles intermèdies altres peces.

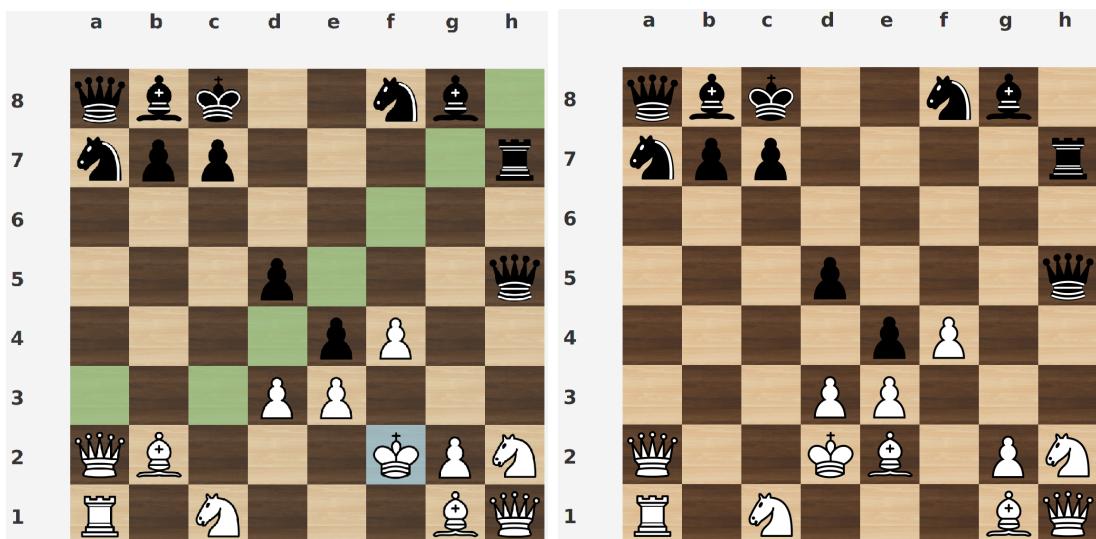
Usant el fitxer de regles que hi ha a l'annex A, s'han pogut comprovar els següents casos d'enroc:



Resultat abans i després de fer l'enroc entre l'alfil b8 i el rei f8



Resultat abans i després de fer l'enroc entre l'alfil b1 i la dama h1



Resultat abans i després de fer l'enroc entre l'alfil b2 i el rei f2, quan aquests s'han mogut

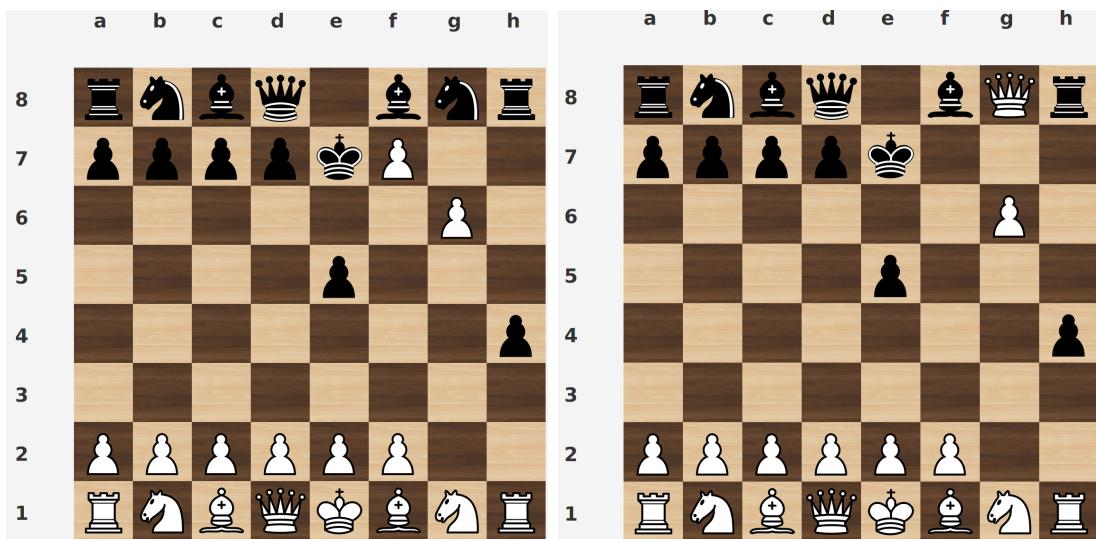
En canvi, no és possible realitzar l'enroc entre la torre i el rei quan es comença la partida, ja que no poden existir peces entre la parella de l'enroc. De la mateixa manera, una vegada que s'hagi mogut un alfil o una dama no es podrà realitzar cap més enroc entre ells.

6.3 Promocións

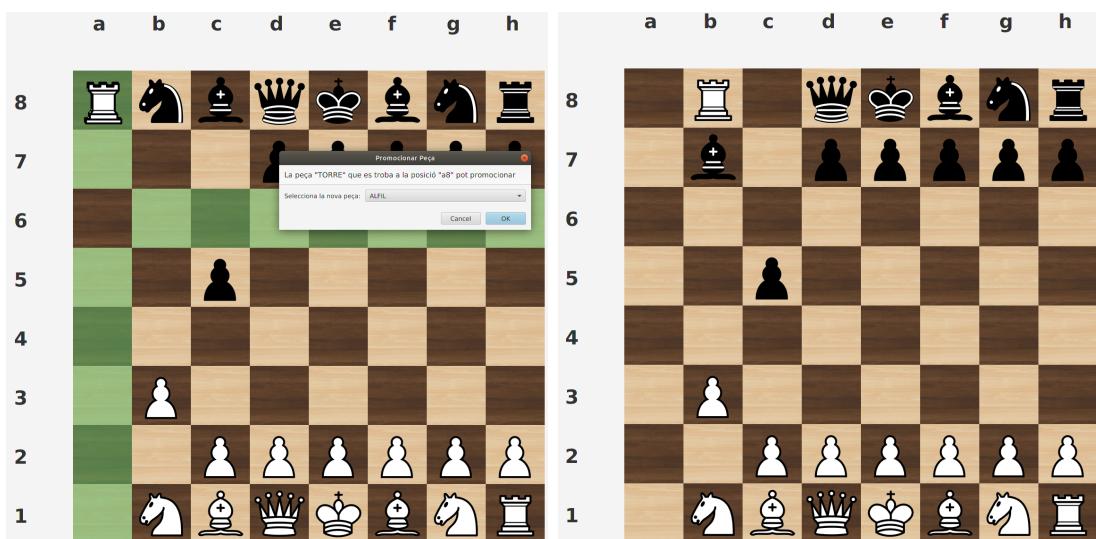
Una tasca que realitza el tauler cada vegada que un jugador efectua una jugada ordinària és comprovar si aquesta permet fer una promoció.

Perquè existeixi una promoció no és suficient amb què la peça pugui promocionar i es trobi a l'última fila segons el seu color. També és necessari que la peça provingui d'una fila diferent perquè si no significa que ja ha estat presentada per a promocionar en alguna jugada anterior.

Tot seguit es mostren dos exemples; en el primer es pot promocionar però en el segon no:



En aquest cas el peó blanc de *f7* capture al cavall de *f8*. Just en aquell moment se li pregunta al jugador si vol promocionar. La seva resposta és "Sí, per una dama".



En aquest segon exemple, la torre té la propietat de promocionar. Quan la torre blanca arriba a *a8* decideix no promocionar. A la següent jugada de les blanques, de *a8* a *b8*, la torre continua a la mateixa fila, per tant no se li permet promocionar.

6.4 Detecció d'escac

Amb aquest apartat s'inicien les proves que afecten l'estat del rei en el tauler.

Recordem que el rei es troba en situació d'escac si alguna peça de l'adversari l'amenaça de forma immediata. No obstant això, el rei pot escapar d'aquest perill si el jugador realitza algunes de les accions que s'han descrit a la pàgina 24 .

A continuació es mostren un seguit d'escacs realitzats al llarg de diferents partides disputades:

Com es pot apreciar a la imatge de la dreta, el rei de les negres es troba en perill. Les peces utilitzades són les típiques dels escacs convencionals, per tant es pot observar com la torre blanca de *c5* podria *capturar* al rei negre. La raó per la qual no és un escac i mat és perquè hi ha varíes maneres de salvar el rei en la propera jugada. Per exemple, el rei pot fugir a una casella segura o, fins i tot, la torre que hi ha a la casella *a6* es podria sacrificar pel seu rei. Tanmateix, no es nega la possibilitat que a la següent tirada de les negres tornés a haver-hi escac.

A la següent imatge, on de nou les peces són les de tota la vida, torna a haver-hi un escac en contra de les negres. Qui amenaça directament al rei negre és la dama blanca segons el seu moviment diagonal. Aquí els únics moviments permesos són els que salven al rei. En aquest cas hi ha tres possibilitats. Per una banda el rei pot fugir a una casella segura. Per una altra, la dama negra podria situar-se a *e6* per salvar al seu rei -tot i que a la següent ronda es repetiria l'escac-. Finalment, potser la millor opció, és que la dama negra mati a la dama blanca posicionant-se a la seva casella.

En aquest exemple també es treballa amb les peces dels escacs convencionals. A diferència de la resta, aquí l'escac és contra les peces blanques, doncs podem comprovar com la dama negra posa en perill al rei blanc. Tanmateix, no es tracta d'un escac i mat perquè el rei podria fugir o directament matar-la, ja que la posició de la reina no és perillosa o, fins i tot, una opció seria efectuar l'enroc amb la torre de la casella *h1*.



6.5 Detecció d'escac i mat

L'escac i mat, com indica el seu nom, suposa que el rei d'algun dels dos bàndols es troba en situació d'amenaça immediata, és a dir, en escac, però aquest rei no té escapatòria i ha de sucumbir al perill, essent així com acaba la partida. Remarco que una de les regles dels escacs és que un rei mai pot ser capturat, és a dir, mai es veurà una jugada en què el rei desapareix del tauler. Tan bon punt es detecta la situació d'escac i mat, la partida ha de finalitzar.

Tot seguit es presenten imatges, accompanyades d'un petit comentari, sobre taulers en què un dels reis ha patit un escac i mat:

Per escalfar motors començarem amb un tauler que conté les típiques peces dels escacs. En aquest tauler s'ha modelat la partida segons el famós *mat del pastor*¹. A l'última jugada de les blanques aquestes han optat per moure la dama que es troava a la posició *h5* fins a la *f7*. Ràpidament s'ha detectat escac i mat perquè el rei no es pot moure a cap posició segura i tampoc hi ha cap manera de salvar-lo matant a la dama.



Un altre escac i mat reconegut al món dels escacs és el *mat de legal*². Aquest escac i mat s'efectua en set tirades i el tauler resultant és el de la dreta.

Després de l'escac fet per l'alfil blanc, el rei negre s'ha vist obligat a desplaçar-se fins a *e7*, la seva única escapatòria. Per mala sort, el cavall blanc ha passat de la posició *c3* a la *d5*, condemnant així el rei negre a escac i mat.

¹Mat del pastor

²Mat de Legal



Podríem seguir fent proves en el tauler amb diferents escacs i mats que ja existeixen. Tanmateix, també és interessant treballar amb altres peces, com són les del fitxer de regles de l'annex A.

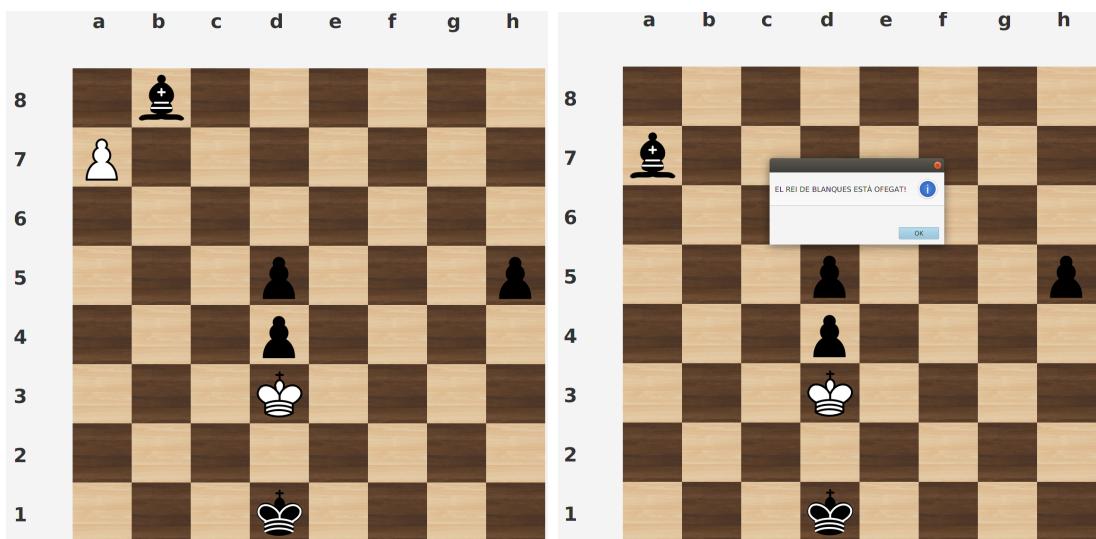
A la imatge de la dreta podem comprovar una situació d'escac i mat. En aquest cas ha estat per part de les peces negres. La torre que hi ha a la posició *a1* té la propietat de moure's com les típiques torres dels escacs amb una petita variant: que pot saltar capturant i també capturar a destí. Per aquesta raó pot arribar fins a la casella *f1* encara que hi hagi peces entre el rei i ella. Per contra, el rei no té possibilitat de salvar-se, només podria moure's a la casella *e1* però aquesta també està sota amenaça de la torre. A més, cap peça de les blanques pot matar a la torre. Així, doncs, ens trobem en un escac i mat.



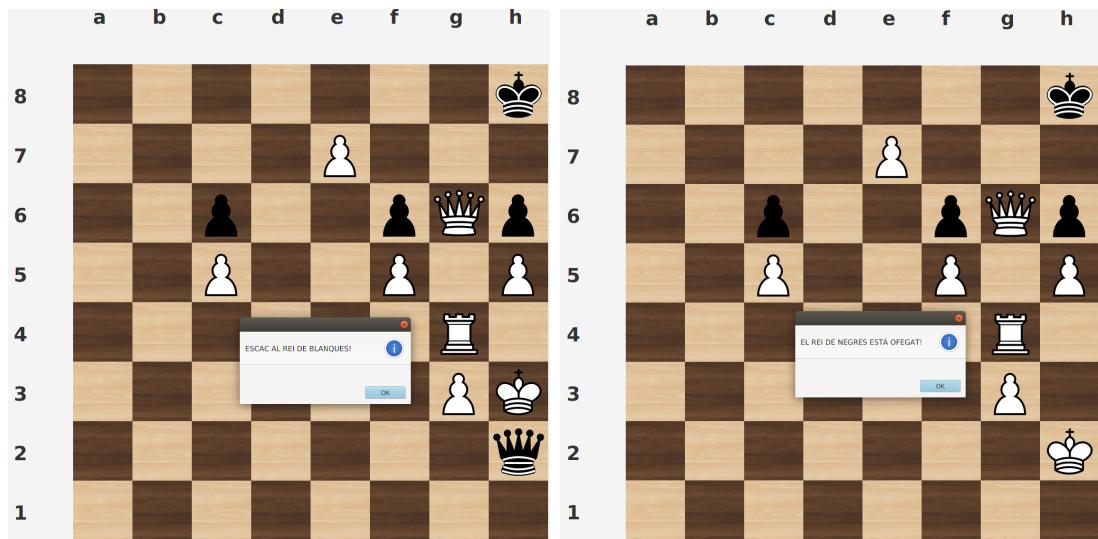
6.6 Detecció de rei ofegat

Per posar fi a les situacions del rei, n'existeix una de poc usual però no per això menys important anomenada rei ofegat. Això succeeix quan el rei no es pot moure a cap posició, sigui perquè no hi ha caselles o perquè aquestes estan amenaçades, i totes les altres peces tampoc tenen moviments possibles. Així, doncs, el jugador no pot efectuar cap mena de jugada i la partida finalitza per taules.

Dit això, posem a prova el tauler donades dues situacions de rei ofegat. Per no complicar-ho, el nombre de peces serà reduït i s'utilitzaran les peces dels escacs quotidians:



Com es pot veure a la imatge anterior, el jugador negre decideix moure l'alfil a la posició $a7$, acabant així la partida en taules per rei ofegat, ja que el rei blanc no es pot moure a cap posició del tauler.



L'anterior situació és interessant perquè a la primera imatge es pot observa com la dama negra deixa el rei blanc en situació d'escac. No obstant això, quan el rei blanc surt de l'escac, capturant a la dama, el bàndol de les negres es queda sense cap classe de moviment possible. Així, doncs, acaba la partida amb taules per rei ofegat.

7 Proves en el jugador CPU

El jugador CPU es regeix segons un conegut algoritme de decisió usat en molts jocs de taula anomenat *minimax*³. Aquest algoritme, en poques paraules, es basa en la creació d'un arbre de joc. Els diferents nodes representen cada tauler després d'efectuar una jugada i a partir de cadascun es generen tots els possibles taulers aplicant les jugades ordinàries i les jugades enroc del següent jugador. Les fulles de l'arbre són els taulers d'una partida finalitzada.

L'algoritme, doncs, el que pretén és escollir el valor del tauler que, seguint el trajecte de jugades, millor representi el màxim benefici pel jugador considerant que el contrari sempre triarà les jugades menys favorables. Gràcies a aquest valor, que s'ha obtingut de les fulles i s'ha transportat fins a l'arrel, és possible coneixer la jugada que s'acosta al tauler desitjat. En això es basa l'algoritme *minimax*: maximitzar els beneficis però minimitzant les pèrdues.

Tothom coincidirà que el cost de temps per crear l'arbre és inimaginable. Per aquesta raó, l'exploració en el nostre cas sempre és de dos nivells i s'ha usat la poda *alfa-beta*⁴ per a reduir les branques a visitar.

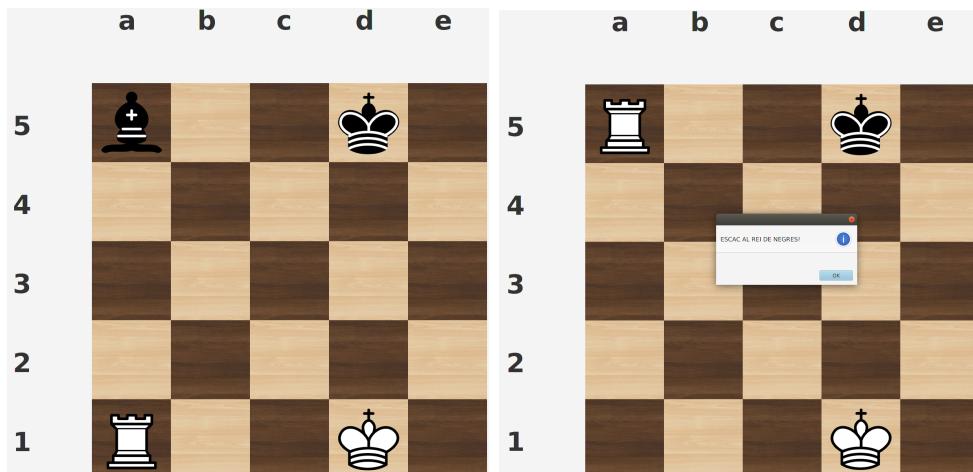
Comentat això posem a prova les capacitats del jugador CPU quan ha d'escollir la millor jugada, acceptar o denegar taules i seleccionar la peça per a la qual vol promocionar.

7.1 Selecció de la millor jugada

En aquest apartat es vol posar a prova la capacitat de decisió del jugador CPU en funció de les jugades.

Per valorar si és cert que escull la jugada que permet "maximitzar el benefici però minimitzar les pèrdues", el millor és treballar amb poques peces. Les peces usades en la següent prova són les típiques dels escacs.

El tauler inicial es mostra a continuació, acompanyat del tauler resultant després d'efectuar la jugada de la CPU:



³Minimax

⁴Poda alfa-beta

A la imatge anterior podem veure com el jugador CPU ha decidit efectuar la jugada de $a1$ a $a5$. Els heurístics retornats en el primer nivell, després de minimitzar les jugades de l'oponent per cadascun dels taulers amb les tirades del jugador CPU, són: **2, 2, 2, 2, 5, 2, 2, 2, 2, -1073740822, 2**.

Les conclusions que podem extreure d'aquests valors són que el jugador CPU pot escollir entre 11 jugades possibles. D'aquestes onze, considerant la pitjor resposta del contrincant, nou jugades possibles deixaran un tauler amb les mateixes peces que l'inicial. Per una altra banda, una tirada del jugador CPU permetrà al contrincant fer-li un escac, d'aquí el valor -1073740822. La tirada que ho permet és la següent, ja que si l'alfil es col·loca a $d2$ hi haurà escac:

5		a			r	
4						
3						
2						
1		T		R		
	a	b	c	d	e	

Per últim, existeix una jugada que pot conduir a un tauler amb valor 5. De fet, aquesta és la jugada que la CPU efectua, ja que en el primer nivell s'escull, d'entre els valors rebuts al minimitzar, el màxim. El valor cinc és perquè l'afil desapareix del tauler.

Una possible pregunta és: haurà escollit aquesta jugada perquè efectua un escac al rei de negres? La resposta és no, el tauler que s'avalua a les fulles conté dues tirades, primer la del jugador CPU i a continuació la de l'adversari. Per tant, el tauler que es puntuja ja no conté l'escac perquè l'adversari haurà hagut de salvar al seu rei.

En definitiva, es pot veure com l'algoritme funciona tal com és d'esperar. Més endavant es parlarà sobre com interacciona dins d'una partida un jugador CPU.

7.2 Selecció de taules

El jugador CPU té la capacitat d'acceptar o denegar taules. En realitat, la metodologia emprada és exactament la mateixa. En aquest cas, també s'explora un arbre de dos nivells però tenint en compte que qui primer realitzarà la jugada serà el **jugador contrari**. Llavors, si l'heurístic del millor tauler és negatiu optarà per acceptar-les, mentre que si és positiu les denegarà. Vegem tot seguit dos exemples de sol·licitud de taules amb resultats diferents:



Taules acceptades pel jugador CPU

En el primer exemple, el jugador CPU representa les peces negres. S'ha simulat el famós *mat del pastor* fins a la penúltima jugada. Ara és el torn de les blanques per a moure la dama de *h5* fins a *f7*, que condirà la partida a escac i mat. No obstant això, s'ha decidit demanar taules. El jugador CPU les ha acceptat.

La raó és perquè, per a la jugada guanyadora, únicament ha pogut explorar fins al primer nivell, ja que escac i mat és sinònim de fi de partida. El jugador CPU no pot fer cap jugada com a resposta.

Llavors, l'heurístic d'aquest tauler és el més negatiu de tots perquè conté una jugada molt dolenta pel jugador CPU. Per tant, quan es minimitzen les pèrdues d'entre totes les jugades del contrincant, aquesta és la pitjor i el jugador CPU prefereix rendir-se.



Taules denegades pel jugador CPU

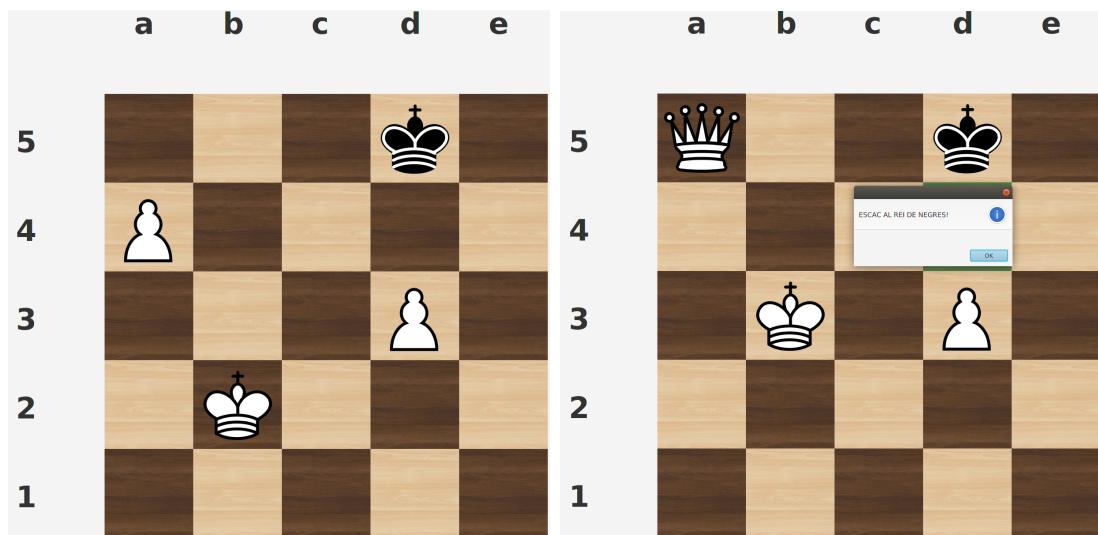
Ara toca invertir els papers: la CPU és el jugador de blanques. Ens trobem a l'antepenúltima tirada, quan encara el cavall de $g7$ no s'ha mogut. Llavors, se li sol·liciten taules però el jugador CPU les denega.

Té sentit perquè encara que reproduueixi totes les tirades del contrincant i després les seves, no troba un tauler que li suposa una partida totalment perduda. En altres paraules, no es troba que a la següent jugada li faran un escac i mat. Tampoc que tots els taulers de les fulles continguin peces del jugador CPU molt més desfavorables que les del contrincant. De fet, quan es mogui el cavall, ell optarà per fer un escac i mat. Així, doncs, no decideix acceptar-les.

7.3 Selecció de promoció

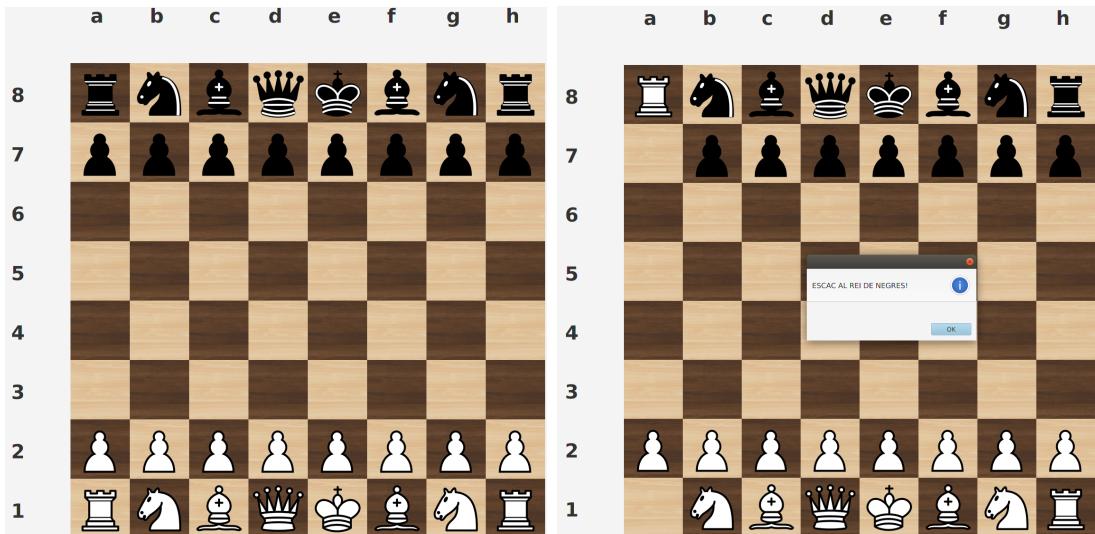
A l'hora de promocionar se segueixen unes pautes similars. El jugador CPU rep la llista de les peces per a les quals pot promocionar. Llavors, simula per cada peça un tauler i per cada tauler tots els diferents taulers que existeixen com a resposta de l'oponent. Per tant, agafa primer els valors mínims de les fulles i en el nivell superior els màxims per escollir la peça per a la qual promocionar.

En el següent cas les peces són les dels escacs convencionals i el jugador CPU controla les blanques. Posem-lo a prova amb dues situacions:



Ell decideix efectuar la jugada del peó i promocionar-lo per una dama. Escull la dama perquè és la peça que li aporta més valor al tauler. També la dama és qui té els moviments més valuosos i li permet fer un escac.

Ara imaginem-nos que les típiques torres tenen capacitat de promocionar i també poden saltar capturant. El jugador CPU controla les peces blanques de nou.



Tan bon punt comença la partida, mou la torre de *a1* a *a8*, matant un peó i una torre de l'adversari. Tanmateix, escull no promocionar.

L'explicació és perquè si decideix promocionar per qualsevol altra peça, el jugador enemic, pensant en el pitjor cas, optarà per fer-li un escac amb la torre negra. En canvi, si no promociona, el jugador negre es veurà obligat a treure el seu rei de l'escac matant la torre blanca. En aquest cas, el valor del tauler estarà a favor del jugador CPU només per una unitat, doncs recordem que amb la primera jugada també ha capturat un peó de les negres. Tanmateix, és millor aquest resultat que una situació d'escac en contra seva.

7.4 Jugador humà vs Jugador CPU

La millor manera de posar a prova les qualitats del jugador CPU és jugant directament contra ell. El més sorprenent és que encara que només explori les jugades a dos nivells de profunditat acostumen a ser molt encertades.

En molts casos, depenent de les capacitats dels moviments, el jugador CPU opta per jugades que capturen les peces més valuoses o el màxim de peces sense derivar en situacions de perill extrem. A més, acostuma a decantar-se per jugades que conduiran a l'escac i un possible escac i mat. Fins i tot és difícil d'escapar de les taules per escacs continus si no es juga amb serietat. No és de més dir que mentre juga també realitza enrocs entre les peces i promocións.

Per tal d'evitar que el jugador CPU es quedi en bucle entre dues caselles durant una part de la partida, s'ha optat per afegir un factor d'aleatorietat. D'entre les n millor jugades, escull una a l'atzar. Això també és interessant perquè si es decideix desfer una jugada, la següent no té perquè ser exactament la mateixa que l'anterior. Així, doncs, la partida pot acabar completament diferent.

7.5 Jugador CPU vs Jugador CPU

També és possible disputar una partida on els dos jugadors són regits per la CPU. A l'annex C se'n pot veure el desenvolupament d'una d'elles on guanyen les blanques després d'efectuar repetits escacs. A més també apareix un enroc per part de les negres.

Quan es tracta de dos jugadors CPU, es pot veure com el factor d'aleatorietat permet estalviar-se una partida on els dos jugadors quedin limitats a dues posicions i aquesta acabi amb taules per inacció.

De fet, vàries partides entre jugadors CPU acaben amb escacs continus si el valor atorgat al límit és massa petit. Tanmateix, amb valors iguals a 6, per exemple, es poden presenciar partides que finalitzen amb la victòria d'un dels dos jugadors. També és interessant el fet de poder aturar i reprendre l'activitat dels jugadors CPU, ja que a mesura que disminueixen les peces el temps de resposta es redueix molt.

7.5.1 Temps de resposta

El marge de resposta del jugador CPU varia en funció dels següents factors:

- Les mides del tauler.
- El nombre de peces que hi ha al tauler.
- Els moviments possibles per peça. Per exemple, quan hi ha escac, el jugador CPU acostuma a respondre ràpidament perquè els únics destins que se li permeten són els que treuen al rei de l'escac.

Cal dir, però, que el temps de decisió oscil·la entre les mil·lèsimes de segon i els 3 segons si es considera un tauler 8 x 8 amb peces convencionals. Aquests valors poden augmentar o disminuir en funció de les condicions esmentades. A més, no cal dir que també depèn del processador.

En referència al primer punt, per exemple, si el tauler passa a ser de 16 x 16, la primera jugada tarda 3 minuts, essent un escac al rei de negres. També, si mantenim el tauler de 8 x 8 però fem la prova amb el fitxer de l'annex A, la primera jugada tarda 7 segons i no 2, ja que les peces tenen molts més destins i amb efectes molt diferents. Tanmateix, les respostes quan hi ha escac són més ràpides perquè només es permeten moviments que salven al rei.

Cal no descuidar que s'ha de conservar una relació estable entre temps i potència, és a dir, no és satisfactori un jugador CPU brillant però que tarda molt ni un sense criteri que respongui a l'instant.

Respecte al nostre, si es considera que en mitjana els jugadors optaran per taules de mides properes a 8 x 8, el temps del jugadorCPU en relació amb la intel·ligència és més que satisfactori.

A Fitxer de regles

```
{  
    "nFiles": 8,  
    "nCols": 8,  
    "peces": [  
        {  
            "nom": "REI",  
            "simbol": "R",  
            "imatgeBlanca": "reiB.png",  
            "imatgeNegra": "reiN.png",  
            "valor": 100,  
            "moviments": [  
                [  
                    0,  
                    1,  
                    1,  
                    0  
                ],  
                [  
                    0,  
                    -1,  
                    1,  
                    0  
                ],  
                [  
                    -1,  
                    0,  
                    1,  
                    0  
                ],  
                [  
                    1,  
                    0,  
                    1,  
                    0  
                ]  
            ],  
            "movimentsIniciais": [],  
            "promocio": false,  
            "invulnerabilitat": false  
        },  
        {
```

```

"nom": "ALFIL",
"simbol": "A",
"imatgeBlanca": "alfilB.png",
"imatgeNegra": "alfilN.png",
"valor": 10,
"moviments": [
    [
        "a",
        "a",
        1,
        1
    ],
    [
        "a",
        "-a",
        1,
        1
    ]
],
"movimentsIniciais": [
    [
        3,
        0,
        0,
        2
    ]
],
"promocio": false,
"invulnerabilitat": false
},
{
"nom": "TORRE",
"simbol": "T",
"imatgeBlanca": "torreB.png",
"imatgeNegra": "torreN.png",
"valor": 15,
"moviments": [
    [
        0,
        "a",
        1,
        2
    ],

```

```

[  

  "a",  

  0,  

  1,  

  2  

],  

"movimentsIniciais": [],  

"promocio": false,  

"invulnerabilitat": false  

},  

{  

  "nom": "DAMA",  

  "simbol": "D",  

  "imatgeBlanca": "damaB.png",  

  "imatgeNegra": "damaN.png",  

  "valor": 20,  

  "moviments": [  

    [  

      "n",  

      "n",  

      1,  

      1  

    ],  

    [  

      "n",  

      "-n",  

      1,  

      1  

    ],  

    [  

      0,  

      "m",  

      2,  

      1  

    ]  

  ],  

  "movimentsIniciais": [],  

  "promocio": false,  

  "invulnerabilitat": false  

},  

{  

  "nom": "CAVALL",

```

```

"simbol": "C",
"imatgeBlanca": "cavallB.png",
"imatgeNegra": "cavallN.png",
"valor": 10,
"moviments": [
    [
        "n",
        "-1",
        1,
        1
    ],
    [
        "-n",
        "1",
        1,
        1
    ]
],
"movimentsIniciais": [
    [
        0,
        "n",
        1,
        2
    ]
],
"promocio": false,
"invulnerabilitat": true
},
{
"nom": "PEO",
"simbol": "P",
"imatgeBlanca": "peoB.png",
"imatgeNegra": "peoN.png",
"valor": 1,
"moviments": [
    [
        2,
        0,
        0,
        2
    ],
    [

```

```

        1,
        1,
        2,
        0
    ],
    [
        1,
        -1,
        2,
        0
    ]
],
"movimentsIniciais": [
    [
        "3",
        "0",
        1,
        1
    ]
],
"promocio": true,
"invulnerabilitat": false
}
],
"posInicial": [
    "TORRE",
    "ALFIL",
    "CAVALL",
    "",
    "",
    "REI",
    "ALFIL",
    "DAMA",
    "DAMA",
    "",
    "",
    "PEO",
    "PEO",
    "PEO",
    "PEO",
    "CAVALL"
],
"limitEscacsSeguits": 8,

```

```

"limitTornsInaccio": 20,
"enrocs": [
{
    "peçaA": "REI",
    "peçaB": "TORRE",
    "quiets": true,
    "buitAlMig": true
},
{
    "peçaA": "REI",
    "peçaB": "ALFIL",
    "quiets": false,
    "buitAlMig": false
},
{
    "peçaA": "ALFIL",
    "peçaB": "DAMA",
    "quiets": true,
    "buitAlMig": false
}
]
}

```

B Fitxer de desenvolupament d'una partida ajornada

```

{
"fitxerRegles": "nouRegles.json",
"posIniBlanques": [
{
    "pos": "b1",
    "tipus": "ALFIL",
    "moguda": "false"
},
{
    "pos": "a1",
    "tipus": "TORRE",
    "moguda": "false"
},
{
    "pos": "a2",

```

```

    "tipus": "DAMA",
    "moguda": "false"
},
{
    "pos": "h1",
    "tipus": "DAMA",
    "moguda": "false"
},
{
    "pos": "g1",
    "tipus": "ALFIL",
    "moguda": "false"
},
{
    "pos": "h2",
    "tipus": "CAVALL",
    "moguda": "false"
},
{
    "pos": "f1",
    "tipus": "REI",
    "moguda": "false"
},
{
    "pos": "g2",
    "tipus": "PEO",
    "moguda": "false"
},
{
    "pos": "f2",
    "tipus": "PEO",
    "moguda": "false"
},
{
    "pos": "e2",
    "tipus": "PEO",
    "moguda": "false"
},
{
    "pos": "c1",
    "tipus": "CAVALL",
    "moguda": "false"
},

```

```

{
  "pos": "d2",
  "tipus": "PEO",
  "moguda": "false"
}
],
"posIniNegres": [
{
  "pos": "h7",
  "tipus": "DAMA",
  "moguda": "false"
},
{
  {
    "pos": "h8",
    "tipus": "TORRE",
    "moguda": "false"
  },
  {
    "pos": "g8",
    "tipus": "ALFIL",
    "moguda": "false"
  },
  {
    "pos": "f8",
    "tipus": "CAVALL",
    "moguda": "false"
  },
  {
    "pos": "e7",
    "tipus": "PEO",
    "moguda": "false"
  },
  {
    "pos": "d7",
    "tipus": "PEO",
    "moguda": "false"
  },
  {
    "pos": "c7",
    "tipus": "PEO",
    "moguda": "false"
  },
  {
    "pos": "b7",
    "tipus": "PEO",
    "moguda": "false"
  }
]

```

```

    "pos": "c8",
    "tipus": "REI",
    "moguda": "false"
},
{
    "pos": "b7",
    "tipus": "PEO",
    "moguda": "false"
},
{
    "pos": "b8",
    "tipus": "ALFIL",
    "moguda": "false"
},
{
    "pos": "a7",
    "tipus": "CAVALL",
    "moguda": "false"
},
{
    "pos": "a8",
    "tipus": "DAMA",
    "moguda": "false"
}
],
"proper_torn": "BLANQUES",
"tirades": [
{
    "torn": "BLANQUES",
    "origen": "b1",
    "desti": "h7",
    "resultat": ""
},
{
    "torn": "NEGRES",
    "origen": "h8",
    "desti": "h1",
    "resultat": "ESCAC"
},
{
    "torn": "BLANQUES",
    "origen": "a1",
    "desti": "h1",

```

```

    "resultat": "",

},
{
    "torn": "NEGRES",
    "origen": "f8",
    "desti": "e8",
    "resultat": "ESCAC"
},
{
    "torn": "BLANQUES",
    "origen": "f1",
    "desti": "e1",
    "resultat": ""

},
{
    "torn": "NEGRES",
    "origen": "a8",
    "desti": "h1",
    "resultat": "ESCAC"
},
{
    "torn": "BLANQUES",
    "origen": "",
    "desti": "",
    "resultat": "AJORNAMENT"
}
],
"resultat_final": "PARTIDA AJORNADA"
}

```

C Partida entre dos jugadors CPUs

```

{
"fitxerRegles": "nouRegles.json",
"posIniBlanques": [
{
    "pos": "b1",
    "tipus": "ALFIL",
    "moguda": "false"
},
{
    "pos": "a1",

```

```

    "tipus": "TORRE",
    "moguda": "false"
},
{
    "pos": "a2",
    "tipus": "DAMA",
    "moguda": "false"
},
{
    "pos": "h1",
    "tipus": "DAMA",
    "moguda": "false"
},
{
    "pos": "g1",
    "tipus": "ALFIL",
    "moguda": "false"
},
{
    "pos": "h2",
    "tipus": "CAVALL",
    "moguda": "false"
},
{
    "pos": "f1",
    "tipus": "REI",
    "moguda": "false"
},
{
    "pos": "g2",
    "tipus": "PEO",
    "moguda": "false"
},
{
    "pos": "f2",
    "tipus": "PEO",
    "moguda": "false"
},
{
    "pos": "e2",
    "tipus": "PEO",
    "moguda": "false"
}
,
```

```

{
  "pos": "c1",
  "tipus": "CAVALL",
  "moguda": "false"
},
{
  "pos": "d2",
  "tipus": "PEO",
  "moguda": "false"
}
],
"posIniNegres": [
{
  "pos": "h7",
  "tipus": "CAVALL",
  "moguda": "false"
},
{
  "pos": "h8",
  "tipus": "DAMA",
  "moguda": "false"
},
{
  "pos": "g7",
  "tipus": "PEO",
  "moguda": "false"
},
{
  "pos": "g8",
  "tipus": "ALFIL",
  "moguda": "false"
},
{
  "pos": "f7",
  "tipus": "PEO",
  "moguda": "false"
},
{
  "pos": "f8",
  "tipus": "REI",
  "moguda": "false"
}
]
}

```

```

    "pos": "e7",
    "tipus": "PEO",
    "moguda": "false"
},
{
    "pos": "d7",
    "tipus": "PEO",
    "moguda": "false"
},
{
    "pos": "c8",
    "tipus": "CAVALL",
    "moguda": "false"
},
{
    "pos": "b8",
    "tipus": "ALFIL",
    "moguda": "false"
},
{
    "pos": "a7",
    "tipus": "DAMA",
    "moguda": "false"
},
{
    "pos": "a8",
    "tipus": "TORRE",
    "moguda": "false"
}
],
"proper_torn": "BLANQUES",
"tirades": [
{
    "torn": "BLANQUES",
    "origen": "a1",
    "desti": "a8",
    "resultat": "ESCAC"
},
{
    "torn": "NEGRES",
    "origen": "c8",
    "desti": "a8",
    "resultat": ""
}
]

```

```

},
{
  "torn": "BLANQUES",
  "origen": "h2",
  "desti": "g7",
  "resultat": "ESCAC"
},
{
  "torn": "NEGRES",
  "origen": "f8 - b8",
  "desti": "d8 - e8",
  "resultat": ""
},
{
  "torn": "BLANQUES",
  "origen": "c1",
  "desti": "e1",
  "resultat": "ESCAC"
},
{
  "torn": "NEGRES",
  "origen": "d8",
  "desti": "c8",
  "resultat": ""
},
{
  "torn": "BLANQUES",
  "origen": "e1",
  "desti": "d7",
  "resultat": "ESCAC"
},
{
  "torn": "NEGRES",
  "origen": "c8",
  "desti": "b8",
  "resultat": ""
},
{
  "torn": "BLANQUES",
  "origen": "g1",
  "desti": "h2",
  "resultat": "ESCAC"
},

```

```

{
  "torn": "NEGRES",
  "origen": "b8 - e8",
  "desti": "d8 - c8",
  "resultat": ""

},

{
  "torn": "BLANQUES",
  "origen": "d7",
  "desti": "e3",
  "resultat": "ESCAC"

},

{
  "torn": "NEGRES",
  "origen": "d8",
  "desti": "e8",
  "resultat": ""

},

{
  "torn": "BLANQUES",
  "origen": "h1",
  "desti": "c6",
  "resultat": "ESCAC I MAT"

}

],

"resultat_final": "BLANQUES GUANYEN"
}

```