

Projecte de Programació
Grau en Enginyeria Informàtica
Curs 2019-2020

ESCACS GENÈRICS

INTRODUCCIÓ

El joc dels escacs, àmpliament conegut arreu del món fins al punt d'haver assolit la categoria d'esport, és originari de l'Índia.¹ Existeixen unes normes que defineixen l'estàndard actual del joc. Aquestes regles les podeu trobar al document que us adjuntem al Moodle de l'assignatura.

Existeixen diverses modalitats del joc dels escacs, algunes de les quals les podem trobar al món de la ficció, com ara els escacs 3D de les sèries *Star Trek* o *The Big Bang Theory* (encara que, per cert, a la realitat també existeixen variants d'escacs en 3 dimensions https://es.wikipedia.org/wiki/Ajedrez_tridimensional).



Fig. 1: Escacs 3D a *The Big Bang Theory*

L'objectiu d'aquest projecte és el de desenvolupar una aplicació que permeti configurar diferents variants del joc d'escacs i, després, jugar segons les regles d'aquestes variants. Concretant una mica més, es pretén que l'aplicació que desenvolupeu:

- a) Permeti llegir la definició d'una modalitat del joc d'escacs, provinent d'un fitxer de text.
- b) Possibiliti jugar una partida seguint les regles establertes (punt a).
- c) Permeti desfer/refer tirades prèvies.
- d) Admeti una partida entre un jugador "humà" i un jugador "CPU", i, també, entre dos jugadors "CPU".
- e) Admeti interacció i visualització de la partida en mode text i també en mode gràfic.

¹ <https://ca.wikipedia.org/wiki/Escacs>

- f) Generi un *fitxer de desenvolupament de partida* que, entre altres coses, permeti, en cas de partida no acabada, la continuació de la mateixa.
- g) Permeti *jugar a partir de l'estat d'una partida començada*, que serà carregada d'un fitxer de text.
- h) (només per a grups de 3 alumnes) Admeti que es pugui *introduir "coneixement" sobre situacions i seqüències de tirades conegudes*, de cares a millorar el joc del jugador "CPU".

Tot seguit detallarem aquests requeriments.

EL FITXER DE DEFINICIÓ DE JOC

L'aplicació haurà de llegir, a partir d'un fitxer en format JSON (vegeu definició del format JSON a <https://json.org/json-es.html>), la definició d'una variant d'escacs. A aquesta definició hi apareixeran bàsicament les dimensions del tauler, la tipologia de peces amb les característiques de cada tipus i la disposició inicial de les peces al tauler.

ASPECTES NO CONFIGURABLES

En primer lloc, citarem els aspectes que han de ser comuns a qualsevol de les variants del joc dels escacs contemplades en aquest projecte:

- El tauler tindrà forma rectangular. Serà una matriu de caselles en la qual el nombre de files i columnes estarà entre 4 i 16.
- Jugaran blanques contra negres. Es jugarà per torns, començant sempre la partida el jugador que tingui les blanques.
- Mai no hi podrà haver més d'una peça en una casella.
- Cada equip tindrà un i només un rei. L'objectiu del joc serà el mateix que el dels escacs convencionals, és a dir, fer escac i mat al rei de l'altre equip. Qui ho aconsegueixi, guanyarà la partida.
- Un jugador, quan té el torn, podrà optar per rendir-se. En aquest cas, el jugador contrari serà el guanyador de la partida.
- Una partida pot acabar en taules (empat). Hi ha dos desencadenants de taules que cal considerar en qualsevol de les variants del joc: (1) que el jugador que té el torn demani taules i l'altre, a continuació, les accepti, i (2) que el jugador que té el torn no pugui moure cap peça (rei ofegat); recordem que no està permès que el rei es mogui a cap casella on estigui amenaçat (escac) per una altra peça.

ASPECTES CONFIGURABLES

Un fitxer d'entrada, que serà llegit per l'aplicació, definirà el reglament d'una variant concreta dels escacs. Els aspectes configurables seran els següents:

Dimensions del tauler

S'indicarà el nombre de files del tauler, i també el de columnes. Podran no ser iguals (tauler rectangular), i hauran d'estar entre 4 i 16.

Peces

Com hem indicat, l'única peça que ha d'aparèixer obligatòriament és el rei (que quedarà identificat amb el nom "REI"). A partir d'aquí, el fitxer definirà un cert nombre de tipus de peça (màxim 25 tipus, per poder assignar a cada tipus una lletra A..Z identificadora). Per a cada tipus de peça definit, s'indicaran les següents característiques:

- *Nom* del tipus de peça.
- *Caràcter* amb què es representarà en mode text. Es farà servir una lletra majúscula A...Z. D'aquesta forma, es podran diferenciar les peces blanques de les negres fent servir les lletres majúscules per a unes i les corresponents minúscules per a les altres.
- *Imatges* amb què es representarà en mode gràfic (ruta del fitxer d'imatge). En faran falta dues, una per a la peça blanca i una altra per a la negra.
- *Valor* del tipus de peça (enter positiu). Sempre, el rei serà la peça més valuosa. Li serà assignat un valor arbitrari fent que el valor de les altres peces sigui imperativament inferior.
- *Moviments permesos*

Cada tipus de peça tindrà associada una llista de moviments vàlids, i cada moviment vindrà definit amb un vector (A,B,C,D) en què la primera component A correspondrà a l'*increment de fila* (positiu: cap amunt, negatiu: cap avall), la segona B a l'*increment de columna* (positiu: cap a la dreta, negatiu: cap a l'esquerra), la C a la *capacitat de matar* i la D a la *capacitat de saltar* (aquestes dues capacitats seran detallades més endavant). Cal remarcar que els conceptes d'esquerra, dreta, amunt i avall seran relatius a la vista del tauler del jugador que fa la tirada.

En funció de les components A i B, distingirem 3 tipus de moviments:

1) Moviment rectilini

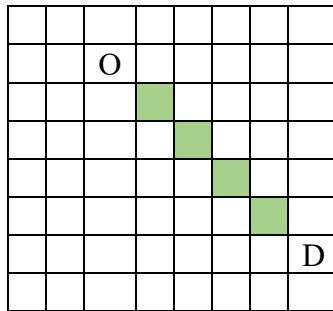
- és el que es realitza al llarg d'una fila o d'una columna, com per exemple el que presenta la torre dels escacs estàndard
- el representarem de manera que una de les components A,B sigui zero i l'altra no.
- els moviments rectilinis poden tenir o no la propietat de saltar sobre les peces (del mateix bàndol o no) del camí entre la casella origen i la casella destí; cas que el salt sigui permès, també poden tenir la propietat de capturar les peces contràries que hagin saltat

		O				D	

Moviment rectilini (0,4). O: casella origen; D: casella destí; en verd, el camí seguit

2) Moviment diagonal

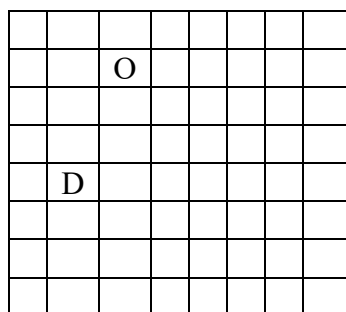
- és el que es realitza al llarg d'una diagonal, com per exemple el que presenta l'alfil dels escacs estàndard
- el representarem de manera que les components A,B tinguin el mateix valor absolut
- igual que els rectilinis, els moviments diagonals poden tenir o no la propietat de saltar sobre les peces (del mateix bàndol o no) del camí entre la casella origen i la casella destí; cas que el salt sigui permès, també poden tenir la propietat de capturar les peces contràries que hagin saltat



Moviment diagonal (-5, 5). O: casella origen; D: casella destí; en verd, el camí seguit

3) Moviment combinat

- els moviments que no siguin rectilinis ni diagonals els anomenarem combinats; un exemple seria el que presenta el cavall als escacs convencionals
- el representarem de manera que les components A,B no valdran zero ni tindran el mateix valor absolut
- la noció de camí no és fàcil de definir als moviments combinats; degut a aquest fet, i per simplificar, considerarem que els moviments combinats sempre admeten el salt però sense capturar cap peça



Moviment combinat (-3, -1). O: casella origen; D: casella destí; no considerarem cap camí en aquests casos

- Els valors A,B (vector de moviment) poden ser:
 - *un valor enter determinat*: 2, -1, 0, 3, etc.
 - *un literal del conjunt {a,b,m,n}*, on *a* i *b* representaran valors enters diferents de 0, mentre que *m* i *n* representaran valors naturals diferents de 0
- El valor C indica si en aquest moviment la peça pot capturar en destí. Si val 0 vol dir que no pot, si val 1 indica que sí que pot, i si val 2 vol dir que el moviment **només és possible si hi ha captura**, és a dir, *si la casella destí es troba ocupada per una peça rival*. En aquestes condicions, la peça rival serà eliminada del joc i el seu lloc serà ocupat per la peça que l'ha capturada.
- El valor D fa referència a la propietat de saltar. Si val 0, significa que la peça no pot saltar usant aquest moviment. Si val 1, significa que pot saltar. Si val 2, vol dir que, a més de saltar, capturarà les peces rivals sobre les que hagi saltat.

NOTA IMPORTANT: si el moviment és combinat (ni horitzontal ni diagonal), el valor de D sempre serà 1, que significa salt permès però sense captura.

Cada tipus de peça definit tindrà assignada una llista no buida de moviments permesos.

RESTRICCIÓ IMPORTANT: Un tipus de peça no haurà de tenir solapaments entre els seus moviments. És a dir, **no podran existir dos o més moviments que permetin el mateix vector desplaçament** (dit d'altra forma, una peça no podrà assolir una certa casella usant més d'un moviment).

Si, en llegir els moviments del fitxer JSON, s'observa la circumstància descrita, el moviment que la produeixi serà ignorat, i per tant no serà enregistrat al tipus de peça. El fet serà indicat pel canal d'error.

Vegem-ne alguns exemples de moviments vàlids per a diferents tipus de peça:

Moviments de la torre dels escacs estàndard: $[(0, a, 1, 0), (a, 0, 1, 0)]$

→ el primer ítem correspon al moviment horitzontal de la torre, i el segon al vertical; notem que en ambdós casos la torre pot capturar, i també que s'admeten els dos sentits del moviment, tant en horitzontal com en vertical. El salt no és permès.

Moviments de l'alfil dels escacs estàndard: $[(a, a, 1, 0), (a, -a, 1, 0)]$

→ el primer ítem correspon al moviment en diagonal /, mentre que el segon correspon al moviment en diagonal \; notem que les dues components del moviment han de ser iguals (en valor absolut), i que als alfils també els és permès capturar. Salt no permès.

Moviments del peó dels escacs estàndard: $[(1, 0, 0, 0), (1, 1, 2, 0), (1, -1, 2, 0)]$

→ el primer ítem correspon a l'avançament del peó una casella endavant, en què no pot matar, mentre que el segon i el tercer corresponen als moviments en diagonal, en què és obligat matar. Notem que el moviment inicial dos caselles endavant serà definit a l'apartat següent: "moviments inicials". Salt no permès.

Moviments d'una "nova" peça que pot matar en diagonal cap a endavant, permetent el salt i a més matant les peces rivals sobre les quals ha saltat. A més pot retrocedir una fila enrere però sense matar:
 $[(n, n, 1, 2), (n, -n, 1, 2), (-1, 0, 0, 0)]$

→ notem que n ha de ser un valor estrictament positiu

Moviments d'una "nova" peça que pot anar en vertical endavant i en diagonal 1 fila enrere, sent obligat matar en el primer tipus de moviment però no en el segon: $[(n, 0, 2, 0), (-1, 1, 1, 0), (-1, -1, 1, 0)]$

→ notem que aquesta peça no pot saltar en cap cas

- *Moviments inicials:*
Cada tipus de peça pot tenir també una llista de moviments inicials, és a dir, moviments que només pot dur a terme el primer cop que es mou, a l'inici de la partida. Es defineixen de la mateixa forma que els moviments permesos. Si la peça no té cap moviment inicial especial, aquesta llista serà buida.
- *Capacitat de promocionar (true|false)*, és a dir, de transformar-se en una altra peça triada pel jugador (qualsevol excepte el rei) en arribar a la darrera fila. Cal remarcar que no cal que la peça triada hagi estat capturada prèviament; en altres paraules, en escacs convencionals un bàndol podria jugar amb dues dames o amb tres torres, per exemple.
- *Invulnerabilitat (true|false)*. Una peça invulnerable no pot ser mai capturada. Notem que una peça invulnerable no serà capturada ni si es troba a la casella final del moviment (en aquest cas el moviment no serà viable) ni tampoc si és saltada per sobre per una peça amb capacitat de capturar saltant.

Il·lustrem-ho amb la possible definició dels tipus de peça per a la variant estàndard dels escacs:²

```
{
  "peces": [
    {
      "nom": "ALFIL",
      "simbol": "A",
      "imatgeBlanca": "alfilB.png",
      "imatgeNegra": "alfilN.png",
      "valor": 3,
      "moviments": [
        ["a", "a", 1, 0],
        ["a", "-a", 1, 0]
      ],
      "movimentsInicials": [],
      "promocio": false,
      "invulnerabilitat": false
    },
    {
      "nom": "TORRE",
      "simbol": "T",
      "imatgeBlanca": "torreB.png",
      "imatgeNegra": "torreN.png",
      "valor": 5,
      "moviments": [
        [0, "a", 1, 0],
        ["a", 0, 1, 0]
      ],
      "movimentsInicials": [],
      "promocio": false,
      "invulnerabilitat": false
    },
    {
      "nom": "DAMA",
      "simbol": "D",
      "imatgeBlanca": "damaB.png",
      "imatgeNegra": "damaN.png",
      "valor": 9,
      "moviments": [
        [0, "a", 1, 0],
        ["a", 0, 1, 0],
        ["a", "a", 1, 0],
        ["a", "-a", 1, 0]
      ],
      "movimentsInicials": [],
      "promocio": false,
      "invulnerabilitat": false
    },
    {
      "nom": "CAVALL",
      "simbol": "C",
      "imatgeBlanca": "cavallB.png",
      "imatgeNegra": "cavallN.png",
      "valor": 3,
      "moviments": [
        [2, 1, 1],
        [2, -1, 1],
        [-2, 1, 1],
        [-2, -1, 1],
        [1, 2, 1],
        [1, -2, 1],
        [-1, 2, 1],
        [-1, -2, 1]
      ],
      "movimentsInicials": [],
      "promocio": false,
      "invulnerabilitat": false
    },
    {
      "nom": "PEO",
      "simbol": "P",
      "imatgeBlanca": "peoB.png",
      "imatgeNegra": "peoN.png",
      "valor": 1,
      "moviments": [
        [1, 0, 0, 0],
        [1, 1, 2, 0],
        [1, -1, 2, 0]
      ],
      "movimentsInicials": [
        [2, 0, 0, 0]
      ],
      "promocio": true,
      "invulnerabilitat": false
    },
    {
      "nom": "REI",
      "simbol": "R",
      "imatgeBlanca": "reiB.png",
      "imatgeNegra": "reiN.png",
      "valor": 100,
    }
  ]
}
```

² S'ha usat un format JSON. A l'annex 1 es pot veure la descripció sencera del fitxer *regles.json*, amb una indentació més clara.

```

"moviments": [[1,0,1,0], [1,1,1,0], [1,-1,1,0], [0,1,1,0], [0,-1,1,0], [-1,-1,1,0], [-1,0,1,0],
[-1,1,1,0]],
"movimentsInicials": [],
"promocio": false,
"invulnerabilitat": false}]
}

```

El rei

Al fitxer de regles ha d'haver-hi un tipus de peça anomenat “REI”, de la qual cada bàndol ha de tenir una unitat. És una peça fonamental al joc, ja que *per guanyar una partida cal fer escac i mat al rei rival*, sigui quina sigui la variant d'escacs definida a les regles.

Els moviments i peculiaritats del rei, però, seran definits al fitxer de regles. Els vistos a l'apartat previ serien els corresponents al rei dels escacs estàndard.

Posició inicial

Després de la definició de les peces, s'indicarà la posició inicial corresponent a la variant de joc definida. Aquesta posició inicial determina el nombre i tipus de peces. La posició *serà indicada per a les blanques, i les negres la replicaran de forma simètrica, com a la versió estàndard dels escacs*:



Fig. 2. Font: <https://www.ajedrezdetorneo.com/posicion-inicial-piezas/>

Les fileres s'aniran omplint començant per la casella inferior esquerra. Al cas dels escacs estàndard, la posició inicial vindria indicada de la següent forma:

```

"posInicial":
["TORRE", "CAVALL", "ALFIL", "DAMA", "REI", "ALFIL", "CAVALL", "TORRE", "PEO", "PEO", "PEO", "PEO", "PEO", "PEO", "PEO", "PEO"]

```

Tanmateix, és possible configurar posicions inicials en què les peces no siguin correlatives. Les caselles buides seran indicades amb “” (string buit). Per exemple, la configuració següent correspondria a la d'una partida normal però sense cavalls.

```

"posInicial":
["TORRE", "", "ALFIL", "DAMA", "REI", "ALFIL", "", "TORRE", "PEO", "PEO", "PEO", "PEO", "PEO", "PEO", "PEO", "PEO"]

```

Definició de situacions que impliquen taules

A part de les situacions estàndard (descrites prèviament) que generen taules en qualsevol variant dels escacs, hi ha un parell de casos addicionals que podran ser configurats al fitxer (ambdós casos són simplifications de regles oficials aplicades només en competicions).

- a) *Taules per escac continuat*: quan un bàndol és capaç de forçar escac un cert nombre consecutiu de vegades, seran decretades taules de forma immediata.
Si es vol ometre aquesta regla, s'indicarà al fitxer un valor molt gran, inabastable a la pràctica.

Per exemple: "limitEscacsSeguits": 3

Vegem-ne un exemple:



*Les negres poden fer escac i mat a la propera tirada, movent la dama a g2. Però és torn de les blanques.
Les blanques, per evitar el mat, fan escac al rei negre movent la dama a b5. El rei negre fuig a a7, però la dama mou a a5 i torna a fer escac. El rei escapa a b7, i llavors la dama mou a b5 i de nou fa escac...
S'ha arribat al límit de 3 escacs seguits i per tant es decreten taules. Les blanques han evitat la derrota.*

<https://ajedrezcastellon.com/cursos-y-teoria/2010/01/14/leccion-19-las-tablas-por-repeticion-de-jugadas/>

- b) *Taules per inacció*: quan s'ha arribat a un cert nombre preestablert de torns seguits (és a dir, de parells de tirades consecutives) sense que s'hagi capturat cap peça. El nombre màxim de torns s'establirà també a l'arxiu de regles. Igual que a l'apartat anterior, si es vol ometre aquesta regla s'indicarà un nombre de torns molt elevat.

Per exemple: "limitTornsInaccio": 50

Moviments especials

Les regles estàndard dels escacs permeten dos moviments especials: l'*enroc* i la *captura al pas*. No considerarem aquesta darrera, però sí l'enroc.

L'*enroc* consisteix en un moviment en què dues peces, el rei i una torre, es mouen alhora en una tirada. Requereix que no s'hagin mogut de la seva posició inicial i que no hi hagi cap altra peça enmig. Si és així, el jugador pot enrocar, movent el rei dues caselles cap a la posició de la torre i movent la torre a la casella adjacent a la nova posició del rei (és com si rei i torre es creuessin).

Volem permetre un *enroc generalitzat* en què dues peces, independentment de quins siguin els seus moviments ordinaris, puguin creuar-se horitzontalment (dins una mateixa fila). Aquest moviment pot requerir o no que les caselles del mig siguin buides i/o que les peces no s'hagin mogut inicialment. La primera peça anirà a parar a la casella del mig (o una més enllà, si no hi ha un mig "exacte"), mentre que la segona quedarà al costat de la primera després d'haver-se creuat. Vegem com es definiria al fitxer l'enroc convencional:

```
"enrocs": [{"peçaA": "REI", "peçaB": "TORRE", "quiets": true, "buitAlMig": true}]
```


Imaginem que hi volguéssim afegir un nou enroc entre dama i cavall, permetent fer-ho fins i tot si s'han mogut prèviament, però no si hi ha peces al mig. Resultaria:

```
"enrocs": [{"peçaA": "REI", "peçaB": "TORRE", "quiets": true, "buitAlMig": true},  
            {"peçaA": "DAMA", "peçaB": "CAVALL", "quiets": false, "buitAlMig": true}]
```

Cal remarcar en un enroc no es permetrà capturar cap peça rival: serà simplement una acció de desplaçament de peces. Per tant, si alguna de les caselles destí està ocupada, l'enroc no serà possible.

Exemple

A l'Annex 1 podem veure el fitxer JSON corresponent a una possible configuració estàndard del joc d'escacs amb regles tradicionals incloent-hi dues situacions extra de taules.

JUGAR

L'aplicació ha de permetre jugar als escacs. S'hauran de gestionar els torns, de forma que cada jugador pugui indicar allà on vol tirar. Es farà servir una nomenclatura tradicional, on les files s'identifiquen amb números i les columnes amb lletres:

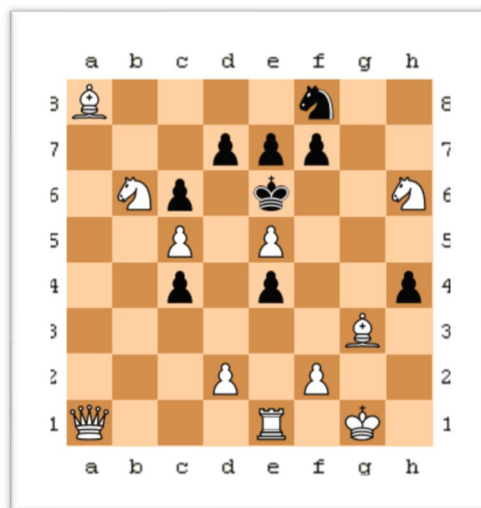


Fig. 3. (llicència Wikimedia Commons)

Per indicar (en mode text) la tirada en què la torre blanca captura el peó negre de la seva columna, indicarem simplement la casella inicial i la final del moviment:

Blanques: e1 e4

I el fet que el peó és capturat ja quedarà deduït. Si es vol indicar una jugada tipus enroc, en què dues peces de la mateixa fila es mouen, ho indicarem amb el símbol '-':

Blanques: a1 - e1

(enroc no vàlid amb les regles estàndard dels escacs, ja que entre d'altres coses la peça a la casella a1 és la dama, i no el rei).

Si una jugada resulta en escac al rei, l'aplicació ho haurà d'indicar d'una forma o altra. Seguint aquest exemple (en mode text):

Blanques: d2 d4
Negres: f8 g6
Blanques: d4 d5
ESCAC AL REI NEGRE !
Negres:

NOTA: recordem que, en cas d'escac, el jugador que té amenaçat el seu rei està obligat a intentar sortir de la situació d'escac (i si no pot és escac i mat, i ha perdut la partida).

Si una jugada no és vàlida segons el reglament establert, l'aplicació ho haurà d'indicar convenientment i haurà de demanar una nova tirada:

Blanques: a8 d5
TIRADA INVÀLIDA !
Blanques: ...

Si és produeix escac i mat, l'aplicació també ho haurà d'indicar, així com si es produeix una situació que provoca taules. En aquests casos, la partida es donarà per acabada i s'indicarà el resultat final.

Dins la comanda que realitza un jugador a cada tirada es pot incloure la sol·licitud de taules, que podrà o no ser acceptada per l'altre:

Blanques: taules
Negres: acceptades
TAULES

Blanques: taules
Negres: no acceptades
Blanques: ...

Finalment, també s'hi pot incloure la petició d'ajornar la partida. En aquest cas, el joc s'interromprà i el seu estat quedarà reflectit al fitxer de partida, on s'indicarà que el torn correspon al jugador que ha demanat ajornament:

Negres: ajornar
PARTIDA AJORNADA

Remarquem que els exemples vistos en aquest apartat corresponen al mode text. Totes aquestes accions de l'usuari han de ser també viables usant la interfície gràfica, quan es jugui en mode gràfic.

DESFER I REFER TIRADES PRÈVIES

La interfície de l'aplicació ha de permetre *desfer* tirada. L'acció s'ha de poder dur a terme sia en mode text amb la comanda “desfer”, o en mode gràfic (per exemple amb un botó).

Aquesta acció desfarà la darrera tirada realitzada. A més, si escau, es podrà demanar varies vegades seguides, *possibilitant que la partida retrocedeixi a un estat previ qualsevol*.

Anàlogament, s'ha de permetre *refer* cada una de les tirades desfetes, tant en mode text (comanda “refer”) o en mode gràfic. Aquesta comanda *revertirà l'efecte de la darrera acció desfer*, si existeix. També podrà ser aplicada varies vegades seguides. Lògicament, no es podrà aplicar una acció refer si immediatament abans no hi ha hagut una acció desfer.

Quan aquestes accions (desfer i refer) no siguin aplicables, el programa simplement notificarà aquest fet.

JOC DE LA CPU

Cal permetre que un dels dos jugadors (o bé tots dos) sigui regit per la CPU, que haurà de triar, quan sigui el seu torn, la tirada a realitzar, dins el conjunt de tirades possibles. Naturalment, la tria haurà de ser realitzada amb un cert criteri, no és suficient que la CPU esculli simplement una tirada a l'atzar entre totes les possibles.

Aquest “joc CPU” pot estar basat en esquemes algorísmics i/o estratègies que hàgiu vist en alguna assignatura prèvia (particularment EDA). Ens referim en concret a esquemes tipus *backtracking* i a estratègies basades en *heurístiques* (greedy, ...).

Tanmateix, teniu llibertat de basar el joc CPU en altres algorismes o estratègies que conegueu. Només heu d'ajustar-vos a un parell de requeriments:

- No ha de ser trivial derrotar la CPU. És a dir, les tirades del jugador CPU han de tenir una mínima intencionalitat de guanyar la partida.
- El temps de resposta de cada tirada de la CPU ha de ser raonable, evitant esperes excessives.

INTERFÍCIE: MODES TEXT I GRÀFIC³

L'aplicació ha de permetre la interacció i visualització en mode text i també en mode gràfic. L'usuari indicarà al principi quin dels dos modes vol utilitzar.

MODE TEXT

L'execució es realitzarà a la consola, usant caràcters ordinaris. Vegem-ne un exemple de visualització d'un tauler 8x8, corresponent a l'anterior figura:

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

Fig. 4. Possible representació d'una partida en mode text.

Noteu que aquí s'han usat majúscules per representar les figures blanques, i minúscules per a les negres (noteu que els símbols de cada peça apareixen en majúscules al fitxer JSON de definició) . Per la seva banda, la representació de les divisions entre caselles pot no ser exactament reproduïble en mode text.

MODE GRÀFIC

Demanem també que aquesta aplicació permeti una visualització en mode gràfic. Caldrà que proporcioni una interfície gràfica, encara que sigui senzilla, que permeti configurar el joc (carregar fitxers, etc.) i jugar amb una certa comoditat.

L'eina més adient per desenvolupar aquesta mena d'interfícies, usant Java, és la llibreria JavaFX. L'ús d'aquesta llibreria us serà introduït al seu moment.

Noteu que a la definició de cada tipus de peça al fitxer JSON (Annex 1) apareix la ruta dels fitxers de les imatges corresponents a la peça en mode gràfic (en versions peça blanca i peça negra).

FITXER DE DESENVOLUPAMENT DE PARTIDA

Per tal d'enregistrar el desenvolupament d'una partida, cada cop que es jugui es generarà un fitxer que descriurà aquest desenvolupament. Aquest fitxer serà gravat quan la partida acabi o bé sigui ajornada, i durà la següent informació:

- *Nom del fitxer de regles del joc.*
- *Posició inicial de les fitxes.* Notem que no necessàriament ha de coincidir amb la posició inicial indicada al fitxer de regles. Aquest aspecte es comenta al proper apartat.
- *Bàndol a qui correspon el proper torn* (tot i que aquesta informació només serà necessària si la seq. de tirades és buida).

³ L'exemple del joc de dames que veurem a classe de pràctiques us pot servir d'orientació per a aquest apartat.

- *Seqüència de tirades.* Se seguirà la mateixa nomenclatura usada a l'apartat "Jugar", indicant per a cada tirada qui la fa (blanques o negres), la casella origen i la casella destí de la tirada (en cas d'enroc, s'indicaran dues caselles origen i dues caselles destí, separades amb un guió '-' i, a més, en un camp anomenat "resultat" (que podrà ser buit), qualsevol possible situació a la qual s'hagi arribat a rel de l'acció del jugador que ha tirat:
 - "ESCAC"
 - "ESCAC I MAT"
 - "PROMOCIÓ: tipus_vell – tipus_nou"
 - "TAULES PER REI OFEGAT"
 - "TAULES PER ESCAC CONTINU"
 - "TAULES PER INACCIÓ"
 - "TAULES SOL·LICITADES"
 - "TAULES ACCEPTADES"
 - "RENDICIÓ"
 - "AJORNAMENT"

NOTA: Noteu que als quatre darrers casos, els camps corresponents a les posicions origen i destí seran buits, ja que no correspondran a una tirada "ordinària".

A aquest registre de tirades no s'han d'indicar les tirades no vàlides ni les corresponents a accions de desfer i refer.

- *Estat final de la partida.* Pot ser un d'aquests quatre: "BLANQUES GUANYEN", "NEGRES GUANYEN", "TAULES", "PARTIDA AJORNADA".

La posició de les fitxes vindrà indicada per una llista de peces, quedant cada peça definida per tres camps: el primer correspondrà a la posició, el segon al tipus de peça i el tercer serà un booleà que indicarà si la peça ja s'ha mogut (true) o no (false).

L'Annex 2 mostra el format que ha de tenir aquest fitxer (JSON), amb un exemple corresponent a una partida acabada amb el famós "mat del pastor"⁴ (seguint el reglament dels escacs convencionals).

⁴ https://ca.wikipedia.org/wiki/Mat_del_pastor

CÀRREGA D'UNA PARTIDA COMENÇADA

L'aplicació ha de permetre carregar un fitxer de partida del tipus descrit a l'apartat anterior. D'aquesta manera es pot reprendre una partida que s'hagi deixat pendent (per manca de temps o per qualsevol altre motiu).

Un cop carregada la partida, l'aplicació demanarà qui juga amb blanques i qui amb negres. Aquí és on es decidirà si tots dos jugadors són humans, o CPU, o un de cada. A partir d'aquest punt, es jugarà la partida de forma normal (vegeu apartat "Jugar").

Val la pena remarcar que aquesta opció pot servir no només per continuar una partida prèviament ajornada, sinó també per jugar una partida a partir d'una certa posició inicial de les peces, a l'estil dels típics problemes d'escacs que apareixen als diaris. Notem que en aquest darrer cas (en què el fitxer haurà estat editat manualment) l'apartat del fitxer corresponent a les tirades prèvies serà buit, ja que només interessarà la posició inicial i el bàndol a qui toca tirar.

INTRODUCCIÓ DE CONEIXEMENT (només per a grups de 3 alumnes)

El joc de la CPU es pot veure enriquit mitjançant l'ús d'informació provinent de partides (o fragments de partides) prèvies. Aquestes partides seran introduïdes al principi a l'aplicació, en forma de fitxers JSON amb el format indicat a l'Annex 2. Cal remarcar que aquests fitxers:

- Ofereixen, com a informacions aprofitables per a aquest apartat, una posició inicial de les peces i una seqüència de tirades.
- El seu estat final correspondrà normalment a una victòria de les blanques o bé de les negres. També podria correspondre a una situació de taules, o a una situació no final però considerada avantatjosa per a les blanques o bé les negres (major nombre de peces, peces de més valor, ...).
- Poden haver estat generats per la mateixa aplicació i/o editats manualment (p.ex. es pot crear un fitxer per a cada una de les obertures clàssiques que condueixen a un escac i mat ràpid, com ara el mat del pastor, el de legal, ...⁵).

La idea base és la següent: el jugador CPU haurà de mirar, a cada torn, si la situació de la partida coincideix amb algun dels estats d'alguna de les partides "subministrades" inicialment a l'aplicació. Si és així, la CPU, si ho troba convenient per als seus interessos, pot "enganxar-se" a la partida en qüestió i començar a emular les tirades d'un dels dos bàndols.

El comportament d'emulació es mantindrà mentre les tirades del rival coincideixin amb les de la partida emulada; tan bon punt el rival mogui de forma distinta, la CPU es *desenganxarà* de la partida i tornarà a buscar coincidències amb les partides emmagatzemades.

Vegem-ne tot seguit un exemple usant el fitxer de l'Annex 2, corresponent al desenvolupament del mat del pastor.

Suposem una partida (amb les regles estàndard definides a l'Annex 1) en què les blanques són governades per la CPU i les negres per un jugador humà. La CPU, abans de fer la primera tirada, veu que l'estat de la partida (peces als seus llocs inicials) coincideix amb el del fitxer del mat pastor, i per tant tria la mateixa estratègia:

Blanques (CPU): e2 e4

Les negres responen fent la mateixa jugada enregistrada al fitxer del mat del pastor:

⁵ <https://www.chess.com/es/blog/ala984/mates-rapidos-en-ajedrez>

Negres: e7 e5

Per tant, la CPU segueix la tirada indicada al fitxer del mat pastor que està emulant:

Blanques (CPU): d1 h5

Però llavors el jugador que mou les negres veu les “intencions” de la CPU i amenaça la dama blanca amb un peó, impedit el desenvolupament del mat del pastor:

Negres: g7 g6

A partir d'aquí la CPU haurà de veure que les negres no segueixen la tirada esperada, i per tant abandonarà l'emulació del mat pastor...

TRACTAMENT D'ANOMALIES A LES DADES D'ENTRADA

L'aplicació haurà de verificar la correcció de les dades d'entrada, tant les introduïdes per l'usuari com les contingudes als diferents fitxers que siguin carregats. En particular, caldrà verificar:

- Que els formats dels fitxers s'ajustin al format especificat als annexos.
- Que les dimensions del tauler estiguin dins els límits establerts.
- Que no hi hagi noms repetits de peça.
- Que cada bàndol tingui un rei.
- Que el rei tingui un valor superior al de les altres peces.
- Que els moviments permesos a les peces tinguin un format correcte.
- Que, per a un tipus de peça, no hi hagi solapament de moviments (2 o més moviments que puguin produir el mateix desplaçament)
- Que a la posició inicial només apareguin tipus de peça del fitxer de definició.
- Que, cas de partida començada, no hi hagi varies peces a la mateixa casella.

Cal que tracteu les anomalies, preferentment, amb el mecanisme d'excepcions que Java proporciona. Noteu que el comportament raonable a la majoria dels casos és avortar l'execució tot mostrant un missatge d'error entenedor. Tanmateix, en algunes situacions serà possible ignorar l'anomalia i continuar llegint les dades d'entrada.

ASPECTES IMPORTANTS

Tot i que el document “*normativa del projecte*” que us facilitem al Moodle incidim en els aspectes més importants que heu de tenir en compte en el desenvolupament del projecte, tot seguit us n’indiquem breument els punts principals pel que fa al resultat final (és a dir, el que lliurareu a l’acabament).

- L’aplicació ha de ser el màxim de modular possible, de manera que resulti fàcil incorporar-hi modificacions i/o reutilitzar els mòduls en què estigui descomposta.
- Cal que relacioneu els mòduls de forma adequada: associació, subtipatge,..., i que feu servir les eines que Java proporciona per treballar amb mòduls: classes abstractes, interfaces, ...
- Pel que fa a la implementació, cal que trieu les estructures de dades i algorismes més adequats, considerant d’una banda l’eficiència però també la simplicitat i claredat del codi.
- El codi ha de ser llegible. Heu de procurar que segueixi les normes d’estil indicades al Moodle (apartat “Normes bàsiques”).
- Cada mòdul ha de ser degudament documentat. Vegeu l’apartat “Exemple de documentació” del Moodle.
- Heu de realitzar i documentar un joc de proves complet, que inclogui com a mínim dos fitxers de regles de joc diferents, amb diferents proves que facin servir cada un d’ells (amb i sense jugador CPU, carregant una partida començada, etc.). Cada prova haurà de ser degudament documentada, amb els fitxers d’entrada usats, el fitxer de descripció de la partida resultant, alguna captura de pantalla, etc.

DIFERENCIACIÓ ENTRE GRUPS DE DOS I GRUPS DE TRES ALUMNES

Com a aspecte diferenciador més important, els grups de dos alumnes queden exempts de fer l’apartat “Introducció de coneixement”.

També, pel que fa al joc de la CPU, el nivell d’exigència per als grups de dos serà menys elevat. Serà suficient que la CPU avaluï quina és la seva millor tirada possible tenint només en compte el resultat de la seva tirada i el de la potencial rèplica (propera tirada) del rival. Els grups de tres, en canvi, hauran d’implementar un joc CPU més sofisticat.

Pel que fa a la resta d’aspectes (lectura del fitxer de regles, jugar, visualització, generació del fitxer de partida, càrrega de partida començada i tractament de dades anòmales), les exigències seran les mateixes independentment del nombre de membres del grup.

ANNEX 1: Fitxer de definició corresponent a les regles estàndard dels escacs

```
{
  "nFiles": 8,
  "nCols": 8,
  "peces": [
    {
      "nom": "REI",
      "simbol": "R",
      "imatgeBlanca": "reiB.png",
      "imatgeNegra": "reiN.png",
      "valor": 100,
      "moviments": [
        [
          1,
          0,
          1,
          0
        ],
        [
          1,
          1,
          1,
          0
        ],
        [
          1,
          -1,
          1,
          0
        ],
        [
          0,
          1,
          1,
          0
        ],
        [
          0,
          -1,
          1,
          0
        ],
        [
          -1,
          -1,
          1,
          0
        ],
        [
          -1,
          0,
          1,
          0
        ],
        [
          -1,
          1,
          1,
          0
        ]
      ],
      "movimentsInicials": [],
      "promocio": false,
      "invulnerabilitat": false
    },
    {
      "nom": "ALFIL",
      "simbol": "A",
      "imatgeBlanca": "alfilB.png",
      "imatgeNegra": "alfilN.png",
      "valor": 3,
```

```

"moviments": [
  [
    "a",
    "a",
    1,
    0
  ],
  [
    "a",
    "-a",
    1,
    0
  ]
],
"movimentsInicials": [],
"promocio": false,
"invulnerabilitat": false
},
{
  "nom": "TORRE",
  "simbol": "T",
  "imatgeBlanca": "torreB.png",
  "imatgeNegra": "torreN.png",
  "valor": 5,
  "moviments": [
    [
      0,
      "a",
      1,
      0
    ],
    [
      "a",
      0,
      1,
      0
    ]
  ],
  "movimentsInicials": [],
  "promocio": false,
  "invulnerabilitat": false
},
{
  "nom": "DAMA",
  "simbol": "D",
  "imatgeBlanca": "damaB.png",
  "imatgeNegra": "damaN.png",
  "valor": 9,
  "moviments": [
    [
      0,
      "a",
      1,
      0
    ],
    [
      "a",
      0,
      1,
      0
    ],
    [
      "a",
      "a",
      1,
      0
    ],
    [
      "a",
      "-a",
      1,
      0
    ]
  ],
  "movimentsInicials": [],
  "promocio": false,

```

```

    "invulnerabilitat": false
  },
  {
    "nom": "CAVALL",
    "simbol": "C",
    "imatgeBlanca": "cavallB.png",
    "imatgeNegra": "cavallN.png",
    "valor": 3,
    "moviments": [
      [
        2,
        1,
        1,
        1
      ],
      [
        2,
        -1,
        1,
        1
      ],
      [
        -2,
        1,
        1,
        1
      ],
      [
        -2,
        -1,
        1,
        1
      ],
      [
        1,
        2,
        1,
        1
      ],
      [
        1,
        -2,
        1,
        1
      ],
      [
        -1,
        2,
        1,
        1
      ],
      [
        -1,
        -2,
        1,
        1
      ]
    ],
    "movimentsInicials": [],
    "promocio": false,
    "invulnerabilitat": false
  },
  {
    "nom": "PEO",
    "simbol": "P",
    "imatgeBlanca": "peoB.png",
    "imatgeNegra": "peoN.png",
    "valor": 1,
    "moviments": [
      [
        1,
        0,
        0,
        0
      ],

```

```

    [
      1,
      1,
      2,
      0
    ],
    [
      1,
      -1,
      2,
      0
    ]
  ],
  "movimentsInicials": [
    [
      2,
      0,
      0,
      0
    ]
  ],
  "promocio": true,
  "invulnerabilitat": false
}
],
"posInicial": [
  "TORRE",
  "CAVALL",
  "ALFIL",
  "DAMA",
  "REI",
  "ALFIL",
  "CAVALL",
  "TORRE",
  "PEO",
  "PEO",
  "PEO",
  "PEO",
  "PEO",
  "PEO",
  "PEO",
  "PEO"
],
"limitEscacsSeguits": 3,
"limitTornsInaccio": 50,
"enrocs": [
  {
    "peçaA": "REI",
    "peçaB": "TORRE",
    "quiets": true,
    "buitAlMig": true
  }
]
}

```

ANNEX 2: Fitxer de desenvolupament d'una partida

```
{
  "fitxerRegles": "regles.json",
  "posIniBlanques": [
    {
      "pos": "a2",
      "tipus": "PE0",
      "moguda": false
    },
    {
      "pos": "b2",
      "tipus": "PE0",
      "moguda": false
    },
    {
      "pos": "c2",
      "tipus": "PE0",
      "moguda": false
    },
    {
      "pos": "d2",
      "tipus": "PE0",
      "moguda": false
    },
    {
      "pos": "e2",
      "tipus": "PE0",
      "moguda": false
    },
    {
      "pos": "f2",
      "tipus": "PE0",
      "moguda": false
    },
    {
      "pos": "g2",
      "tipus": "PE0",
      "moguda": false
    },
    {
      "pos": "h2",
      "tipus": "PE0",
      "moguda": false
    },
    {
      "pos": "a1",
      "tipus": "TORRE",
      "moguda": false
    },
    {
      "pos": "b1",
      "tipus": "CAVALL",
      "moguda": false
    },
    {
      "pos": "c1",
      "tipus": "ALFIL",
      "moguda": false
    },
    {
      "pos": "d1",
      "tipus": "DAMA",
      "moguda": false
    },
    {
      "pos": "e1",
      "tipus": "REI",
      "moguda": false
    },
    {
      "pos": "f1",
      "tipus": "ALFIL",
      "moguda": false
    },
    {

```

```

    "pos": "g1",
    "tipus": "CAVALL",
    "moguda": false
  },
  {
    "pos": "h1",
    "tipus": "TORRE",
    "moguda": false
  }
],
"posIniNegres": [
  {
    "pos": "a7",
    "tipus": "PE0",
    "moguda": false
  },
  {
    "pos": "b7",
    "tipus": "PE0",
    "moguda": false
  },
  {
    "pos": "c7",
    "tipus": "PE0",
    "moguda": false
  },
  {
    "pos": "d7",
    "tipus": "PE0",
    "moguda": false
  },
  {
    "pos": "e7",
    "tipus": "PE0",
    "moguda": false
  },
  {
    "pos": "f7",
    "tipus": "PE0",
    "moguda": false
  },
  {
    "pos": "g7",
    "tipus": "PE0",
    "moguda": false
  },
  {
    "pos": "h7",
    "tipus": "PE0",
    "moguda": false
  },
  {
    "pos": "a8",
    "tipus": "TORRE",
    "moguda": false
  },
  {
    "pos": "b8",
    "tipus": "CAVALL",
    "moguda": false
  },
  {
    "pos": "c8",
    "tipus": "ALFIL",
    "moguda": false
  },
  {
    "pos": "d8",
    "tipus": "DAMA",
    "moguda": false
  },
  {
    "pos": "e8",
    "tipus": "REI",
    "moguda": false
  },
],

```

```

{
  "pos": "f8",
  "tipus": "ALFIL",
  "moguda": false
},
{
  "pos": "g8",
  "tipus": "CAVALL",
  "moguda": false
},
{
  "pos": "h8",
  "tipus": "TORRE",
  "moguda": false
}
],
"proper_torn": "BLANQUES",
"tirades": [
  {
    "torn": "BLANQUES",
    "origen": "e2",
    "desti": "e4",
    "resultat": ""
  },
  {
    "torn": "NEGRES",
    "origen": "e7",
    "desti": "e5",
    "resultat": ""
  },
  {
    "torn": "BLANQUES",
    "origen": "d1",
    "desti": "h5",
    "resultat": ""
  },
  {
    "torn": "NEGRES",
    "origen": "b8",
    "desti": "c6",
    "resultat": ""
  },
  {
    "torn": "BLANQUES",
    "origen": "f1",
    "desti": "c4",
    "resultat": ""
  },
  {
    "torn": "NEGRES",
    "origen": "g8",
    "desti": "f6",
    "resultat": ""
  },
  {
    "torn": "BLANQUES",
    "origen": "h5",
    "desti": "f7",
    "resultat": "ESCAC I MAT"
  }
],
"resultat_final": "BLANQUES GUANYEN"
}

```