

Documentació per al primer lliurament

Hi ha bastant a millorar, sobretot aspectes de modularitat.

PARTIDA

Descripció general: Partida és la classe que actua com a gestor del joc (creació de tauler, control de moviments de les peces, salvaments necessaris...).

Operacions

`Partida(nomFitxer: String, esCarregada: boolean)`

Pre: Nom de fitxer en el qual les dades es troben en JSON i conté les normes de la partida.

Post: S'ha creat una partida nova amb la normativa per defecte o es carrega una partida si és carregada.

`void iniciarPartida()`

Pre: Normes inciades o precarregades.

Post: Es juga mentre no s'acabi la partia o es pari per a guardar.

`void guardarPartida()`

Pre: ---

Post: S'han guardat tots els torns referents a la partida actual

`void referTorn()`

Pre: ---

Post: S'ha refet el torn

`void desferTorn()`

Pre: `#torns > 0` ?

Post: S'ha desfet el torn.

`void escollirJugadors()`

Pre: Partida iniciada no carregada.

Post: S'escull si la partida serà entre CPUs, CPU vs Jugador o entre Jugadors i els colors.

On es guarda? Millor passar la ruta del fitxer com a paràmetre, i tenir el diàleg a un altre nivell.

La interacció de l'usuari hauria de separar-se de la lògica del joc. Res a dir si aquesta és la classe que fa la interacció amb l'usuari, però llavors no hauria de rebre el nom de cap fitxer en el constructor. En qualsevol cas, l'entrada/sortida hauria de concentrar-se en un mòdul. Mireu l'exemple de les Dames que hi ha al Moodle (en mode text).

TAULER

Descripció general: El tauler està format per un nombre donat de caselles representades per una matriu. Aquest tauler controla també les

Operacions:

`Tauler(nomFitxer: String)`

Pre: El fitxer conté les **se us ha tallat**

Post: S'ha creat un tauler amb les peces corresponents

boolean esLimitEscacsSeguits() operació de tauler o de partida?

Pre: --

Post: Retorna si s'ha superat el límit d'escacs seguits

boolean esLimitTornsinactivacio() operació de tauler o de partida?

Pre: --

Post: Retorna si s'ha superat el límit de torns d'inactivitat

boolean ultimMovimientPromocionable() operació de tauler o de partida?

Pre: --

Post: Retorna si l'últim moviment és promocionable.

void mourePeça(filaOrigen: int, columnaOrigen: int, filaDesti: int, columnaDesti: int)

Pre: Posicions vàlides.

Post: S'ha col·locat la peça de (filaOrigen, columnaOrigen) a (filaDesti, columnaDesti), incrementat els límits per taules (si és necessari), capture la peça si és possible.

boolean escaciMat()

Pre: #torns > 0 ?

Post: Retorna cert si el rei del jugador que ha de fer el torn es troba en escaciMat

int fiPartida() potser hauria de ser operació de Partida, sembla que

Pre: #torns > 0 ? barregeu coses

Post: Retorna -1 en cas que no s'acabi la partida en el corresponent torn, 0 en cas d'empat (taules) i 1 en cas victoria blanques i 2 en cas de victoria negres.

TORN

Descripció sospitosa de concepte poc clar

Descripció general: Es realitza el moviment d'una peça amb les seves respectives validacions i reaccions, alhora que s'emmagatzema tot. Segurament no té prou entitat per ser un mòdul
Operacions:

Torn(jugador Jugador, tauler Tauler)

Pre: ---

Post: S'inicia el torn del jugador

boolean ferJugada()

Pre: --

Concentrar la interacció en un mòdul

Post: Demana a l'usuari/CPU corresponent al torn la peça a moure i executa la jugada. Retorna fals en cas que l'usuari demani taules

JUGADOR

Concentrar la interacció en un mòdul

Descripció: Jugador controla els moviments del jugador i, en cas de la CPU, realitza els càculs necessaris per a pensar i executar la següent jugada.

Operacions:

Jugador()

Pre:--

Post: S'ha creat un jugador

abstract void ferJugada()

Pre: Torn del jugador. Modularitat!

Post: S'ha escollit la peça i el moviment a fer en el torn.

USUARI refina Jugador

Descripció: Usuari controla les accions que es duran a terme en cas que el jugador sigui un usuari físic.

Operacions:

Jugador()

Pre: ---

Post: Inicialitza el jugador

int escollirColor()

Pre: ---

Concentrar la interacció en un mòdul

Post: Retorna el color escollit per a l'usuari

void ferJugada()

Pre: ---

Post: S'ha escollit la peça a moure i la seva nova posició o es retornen taules.

CPU refina Jugador

Descripció: CPU serà la classe que controlarà les accions en el joc de la part no controlada per l'usuari, és a dir, de manera automàtica.

Operacions:

CPU()

Pre: ---

Post: Inicialitza l'usuari

No té informació de la partida ni del tauler?!

void ferJugada()

Pre: ---

Post: Es pensa la jugada i es s'escull la peça a moure i la seva nova posició

CONEIXEMENT

Pensar en objectes: què és un "Coneixement"?

Descripció: ~~Extensió de la classe CPU en cas que s'entrin les dades que atorguen coneixement d'una jugada a la CPU~~

Operacions:

Coneixement(coneixement: fitxerJSON);

Pre: --

Post: Construeix l'objecte creixement a partir del fitxer entrat

void ferJugada(filaO: int, columnaO: int, filaD: int, columnaD: int)

Pre: **El moviment pertany a l'últim torn. Modularitat!**

Post: Farà un nou moviment seguint el coneixement, o, **en cas que no es pugui seguir, se'n farà un altre** ?

PEÇA

Descripció: Peça serà d'un tipus amb uns moviments i propietats predefinits amb les regles del joc. Serà la classe encarregada de validar els moviments que vulgui fer.

Operacions:

Peça(propietatsJSON: string) **Descriure**

Pre: ---

Post: S'ha creat una peça amb uns moviments i **propietats específiques**.

boolean esInvulnerable()

Pre: ---

Post: Retorna si és invulnerable

boolean esPromocionable()

Pre: ---

Post: Retorna si és promocionable

boolean capturarSaltant()

Pre: ---

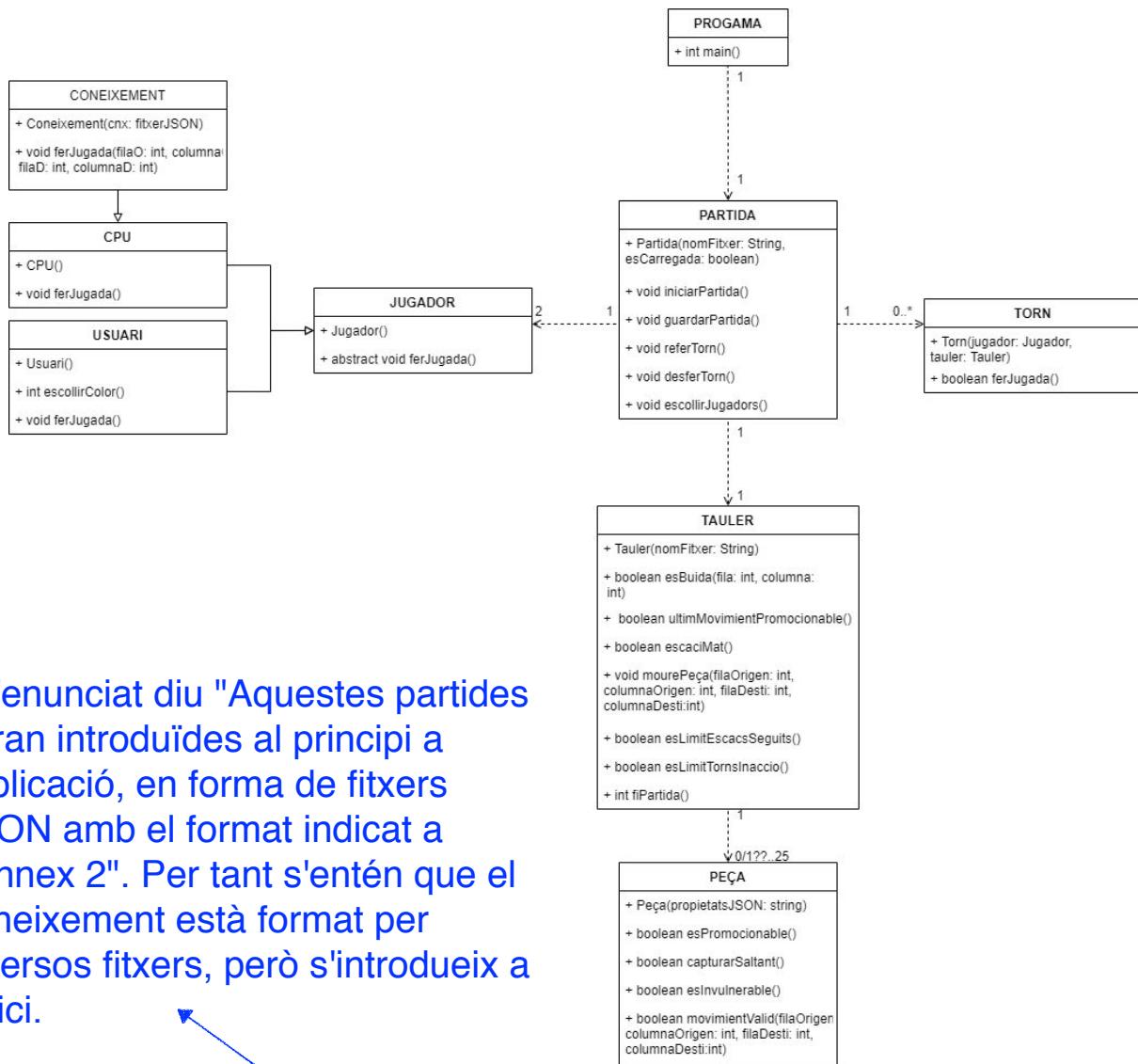
Post: Retorna si potCapturarSaltant

boolean movimentValid(filaOrigen:int, columnaOrigen:int, filaDesti:int, columnaDesti:int)

En quin tauler?

Pre: ---

Post: Retorna si el moviment és vàlid



A l'enunciat diu "Aquestes partides seran introduïdes al principi a l'aplicació, en forma de fitxers JSON amb el format indicat a l'Annex 2". Per tant s'entén que el coneixement està format per diversos fitxers, però s'introdueix a l'inici.

DUBTES

El coneixement ha de carregar la informació només un cop al començament de la partida o ha de poguer carregar múltiples cops a través per exemple d'un botó ?.

No sabem aquest del projecte si una classe casella seria útil per poder guardar informació sobre si el rei està amenaçat o no. De moment no tenim cap objecte casella i la nostra idea és fer-ho tot en una matriu de peces.

Ja aneu bé. D'entrada Casella no sembla necessària. Però pot ser que més endavant quan feu al interfície gràfica d'usuari us faci falta. Ja l'afegireu quan toqui.

En principi això hauria de poder-se controlar d'altres maneres (no justifica l'existencia d'una classe)