

**Joc(main)**

Descripció general: Classe que contindrà l'entrada al programa.

Operacions:

- nova\_partida()
  - Pre: ---
  - Post: Crea una nova partida.
- carregar\_partida()
  - Pre: ---
  - Post: Carrega una partida existent.
- jugar()
  - Pre: ---
  - Post: Demana les comandes per jugar.

**Partida**

Descripció general: ~~Classe que gestiona les operacions durant la partida.~~

Operacions: **Una partida d'escacs**

- partida()
  - Pre: ---
  - Post: ~~Constructor de partida.~~
- carregar(File fitxer) **String fitxer**
  - Pre: fitxer!=null
  - Post: Nova partida carregada des de fitxer.
- guardar()
  - Pre: ---
  - Post: Guarda l'estat de la partida. **On el guarda?**
- refer\_jugada()
  - Pre: Hi ha una jugada per refer.
  - Post: Torna a fer la jugada següent.
  - Excepcions: [PENDENT DE DEFINIR TIPUS] si no té jugada.
- boolean pot\_refer()
  - Pre: ---
  - Post: Cert si hi ha una jugada per refer, fals altrament.
- desfer\_jugada()
  - Pre: Hi ha una jugada per desfer.
  - Post: Desfà la jugada actual.
  - Excepcions: [PENDENT DE DEFINIR TIPUS] si no té jugada.
- boolean pot\_desfer()
  - Pre: ---
  - Post: Cert si hi ha una jugada per desfer, fals altrament.
- fer\_jugada(String j)
  - Pre: ---
  - Post: El jugador que té el torn fa la jugada especificada per j.
- Jugada comprovar\_validesa()
  - Pre: ---
  - Post: Retorna **la Jugada demandada** si aquesta és vàlida, null altrament.

Fusionar?

**Descriure el format de j**

?

- inicialitza\_jugador1()
  - Pre: ---
  - Post: Inicialitza el jugador de les blanques.
- inicialitza\_jugador2()
  - Pre: ---
  - Post: Inicialitza el jugador de les negres.

## Jugada

Descripció general: ~~Classe que guarda una jugada.~~ Una jugada del joc d'escacs

Operacions:

- jugada() Una jugada buida no pot ser
  - Pre: ---
  - Post: Constructor de jugada.

Falten operacions. Si és buida no té raó d'existir.

## Jugador

Descripció general: ~~Classe que guarda el jugador, sigui blanques o negres.~~

Operacions:

Un jugador d'escacs

- jugador()
  - Pre: ---
  - Post: ~~Constructor de jugador.~~ Jugador buit?

## Huma

Descripció general: ~~Classe d'un jugador humà.~~

Operacions:

- huma()
  - Pre: ---
  - Post: ~~Constructor de huma.~~ "Constructor" no és cap condició

## CPU

Descripció general: ~~Classe d'un jugador no humà.~~

Operacions:

- huma() cpu()
  - Pre: ---
  - Post: ~~Constructor de cpu.~~

Falten operacions que heu indicat en el diagrama

## Tauler

Descripció general: Un conjunt de caselles.

Operacions:

- tauler() Indicar dimensions
  - Pre: ---
  - Post: ~~Constructor de tauler.~~
- carregar\_peces()
  - Pre: ---
  - Post: Carrega les peces al tauler. Quines peces?

Falten operacions: peça en una posició, ...

No està clar que calgui, a no ser que pensem en la interfície.

Casella Podríem treballar directament amb peces dins el tauler.

Descripció general: Una casella del tauler.

Operacions:

- casella():
  - Pre: ---
  - Post: Crea una casella.
- char te\_peca()
  - Pre: ---
  - Post: Retorna un char que correspon a la representació de la peça. Si no hi ha peça retorna '0'.
- donam\_peca()
  - Pre: La casella conté una peça
  - Post: La casella ja no té cap peça.
  - Excepcions: [PENDENT DE DEFINIR TIPUS] si no tenia peça.
- agafa\_peca(char p)
  - Pre: La casella no conté cap peça
  - Post: La casella conté la peça p
  - Excepcions: [PENDENT DE DEFINIR TIPUS] si ja tenia peça.

## Tipus\_Peca

Descripció general: La configuració d'una peça del joc.

Operacions:

- peca() **Posar paràmetres en el constructor (color, nom, valor, ...)**
  - Pre: ---
  - Post: Crea una peça **buida?!**
- boolean ~~diguem\_color()~~ **esBlanca()**
  - Pre: ---
  - post: Retorna el color de la peça. true = blanca, false = negre
- String ~~diguem\_nom()~~
  - Pre: ---
  - Post: Retorna el nom de la peça.
- boolean ~~estas\_capturada()~~ **No està clar que això sigui una propietat de la peça**
  - Pre: ---
  - Post: Retorna un booleà en funció de si la peça ha estat capturada
- int ~~diguem\_valor()~~
  - Pre: ---
  - Post: Retorna el valor de la peça.
- pren\_moviment(Moviment m)
  - Pre: m != null
  - Post: La peça permet fer el moviment m.

## Moviment

Descripció general: Moviment que pot fer una peça

Operacions:

- moviment(String dades)
  - Pre: dades segueix el format xml per la configuració dels moviments.  
Aquest format es pot consultar a l'annex de l'enunciat del projecte.
  - Post: Crea el moviment segons la configuració.



Millor desglossar el moviment en les seves parts. Separar el format de lectura del moviment en sí.