

### Tipus JocsEscacs (main file)

**Descripció general:** És l'encarregat de inicialitzar, gestionar els arguments en els que hi haurà la mida del tauler i es podrà triar entre mode gràfic i mode text.

#### Operacions:

void main();

### Tipus PartidaText

**Descripció general:** Administrador de la interfície del joc en mode text

#### Operacions:

juga(Partida p);  
void mostrarTaulell()

### Tipus PartidaGrafica

**Descripció general:** Administrador de la interfície del joc en mode gràfic

#### Operacions:

juga(Partida p);  
void mostrarTaulell()

### Tipus Partida

**Descripció general:** És l'encarregat de gestionar tota la partida

#### Operacions:

void CambiarTorn() //Crida els mètodes necessàries per a canviar de torn  
void MostrarTaulell() //Mostra l'interfície del taulell per a la visualització per pantalla  
void RealitzarTirada( Tirada t) //Realitza la tirada que ja s'ha comprovat que es valida  
previament.  
void AcabarPartida(string manera) //Executa el codi per a acabar la partida  
    *void Taules(int nMoviments) //Després de n moviments sense que es mori una peça  
s'acaba la partida*  
    *void TaulesAcordades(bool dacord) //Els dos jugadors s'han posat d'acord en acabar  
en taules.*

### Tipus PecaGrafica

**Descripció general:** Administrador de la interfície de la peça (quin aspecte tindrà en mode gràfic).

#### Operacions:

PecaGrafica()

### Tipus Rajola

**Descripció general:** Administrador de la interfície de la rajola.

**Operacions:**

Rajola( lmatge ima, int amplada);

**Tipus Taulell**

**Descripció general:** Matriu que defineix les dimensions que tindrà el taulell durant una partida.

**Operacions:**

Taulell(int fil, int col)

Pre: nombres enters entre 4 i 16.

Post: taulell creat de dimensió fil x col.

bool ocupat(Posicio p)

Pre: ---

Post: Retorna true si la posició està ocupada, fals altrament

void afegir\_peça(Peça p, Posicio p)

Pre: --

Post: S'ha afegit la peça al taulell

bool fora\_de\_rang(Posicio p)

Pre: --

Post: Retorna true si la posicio entrada està fora del rang del taulell.

bool color\_correcte(Posicio p, Jugador j)

Pre: --

Post: Retorna true si la peça de la Posicio p es del color del jugador.

void matar(Posicio p)

Pre: --

Post: La peça ha deixat d'estar en joc

**Tipus Posicio**

**Descripció general:** Conte la lògica de les posicions

**Operacions:**

Posicio(string fila, string columna)

Pre: string valid

Post: S'ha crear la posició(fila,columna)

**Tipus Jugador (és una classe abstracta, té dades a dintre)**

**Descripció general:** Gestiona les tirades de la partida

### **Operacions:**

Jugador()

Pre:---

Post: Jugador creat

void decisió()

Pre:---

Post: S'ha realitzat una tirada, s'executen mètodes privats de taules i retiraar-se si el jugador actual ho ha demanat.

### **Tipus Persona (implementa els mètodes de la interfície de Jugador)**

**Descripció general:** Classe heretada de Jugador que gestiona les tirades d'una persona física

### **Operacions:**

*Persona()*

*Pre:---*

*Post: Jugador creat*

*void decisió()*

*Pre:---*

*Post: Ha proposat una decisió el jugador actual.*

**\*Els mètodes en cursiva són implementats dels mètodes de la interfície.**

### **Tipus CPU (implementa els mètodes de la interfície de Jugador)**

**Descripció general:** Gestiona totes les jugades que fa la CPU, tant predir jugades com agafar-ne de partides passades per a mirar de guanyar al adversari

### **Operacions:**

*CPU()*

*Pre: --*

*Post: S'ha creat un jugador CPU amb la dificultat predeterminada*

*void decisió()*

*Pre: --*

*Post: La CPU ha calculat la jugada que considera més adient per a aquell torn.*

*void estudiarTirada()*

*Pre: --*

*Post: La CPU considera quin és el tipus de jugada que ha d'afrontar en aquell torn*

### **Detallar els tipus de tirades**

void tiradaDefensiva()

Pre: --

Post: La CPU calcula la jugada defensiva més adient en aquell torn

void tiradaOfensiva()

Pre: --

Post: La CPU calcula la jugada ofensiva més adient en aquell torn.

**\*Els mètodes en cursiva són implementats dels mètodes de la interfície.**

### **Tipus Peça**

**Descripció general:** Defineix les propietats generals d'una peça.

**Operacions:**

Peca()

Pre: --

Post: Inicialitza una peça

Peca(int nlcone, string nom, char caracter)

Pre: Icone vàlid.

Post: S'ha creat una peça amb atributs personalitzats.

Posició posicióActual()

Pre: La peça està viva

Post: Retorna la posició actual de la peça

bool sha\_mogut()

Pre:--

Post: Diu si la peça s'ha mogut almenys una vegada

bool color()

Pre:--

Post: retorna true si la peça és de color negre, fals si és blanc

### **Tipus Moviment**

**Descripció general:** Administra els possibles moviments de cada peça

**Operacions:**

Moviment(char vertical, char horitzontal, int saltar, int matar)

pre: --

post: s'ha definit un moviment

int potSaltar()  
pre: --  
post: retorna 0 si no pot saltar, 1 si pot saltar i 2 si salta matant

int potMatar()  
pre: --  
post; retorna la classe de forma de matar que pot executar.

### Tipus Historial

**Descripció general:** Guarda el conjunt de moviments realitzats durant la partida i ens permet desfer i refer

#### Operacions:

Historial()  
pre: --  
post: Inicialitza l'historial.

void afegirTirada(Tirada t)  
pre: --  
post: afegim el moviment que s'està executant.

Tirada desfer()  
pre: --  
post: realitza l'opció de desfer un moviment.

Tirada refer()  
pre: --  
post: realitza l'opció de refer un moviment.

Tirada Mostrar\_tirada\_anterior()  
pre--  
post: Mostra l'ultima tirada realitzada

### Tipus SituacioPassada

**Descripció general:** Comprova situacions passades per a treure'n avantatge a la partida actual. Té accés a un conjunt de partides que han estat guanyades prèviament.

#### Operacions:

bool situacioPassada()  
pre: --  
post: realitza comprovacions de situacions passades si coincideix la posició de totes les peces. Retorna cert si troba alguna que coincideixi, fals altrament.

string coincidenciaPassada()

pre: Hem trobat una coincidència de la posició de les peces d'una partida anterior en comparació amb l'actual

post: Retorna la decisió que es va fer en la partida anterior.

### Tipus TipusPeça

#### Operacions:

void afegirMoviment(Moviment m)

Pre: Moviment vàlid

Post: Moviment m afegit a la peça, contador nMoviments++

bool moviment\_valid(Moviment m)

pre: --

post: mira si el moviment m és vàlid per aquesta peça.

### Tipus LlegirPartides

**Descripció general:** Entrada dels fitxers de partides.

#### Operacions:

void llegirPartides(string color);

Pre: --

Post: S'han llegit els fitxers de partides i s'han guardat les que són útils per a la CPU. Es guarden les que son del mateix color de la CPU i han guanyat.

### Tipus LlegirRegles

**Descripció general:** Entrada del fitxer de les regles de la partida que s'inicialitza.

#### Operacions:

void llegirRegles();

Pre: --

Post: s'ha llegit el fitxer de regles i s'han guarda per a la partida actual.

### Tipus Tirada

**Descripció general:** Entrada d'una tirada que es realitza en un torn

#### Operacions:

Tirada()

Pre: --

Post: Inicialitzada la tirada

### Tipus AltresAccions

**Descripció general:** Tracta les accions que no son tirades

**Operacions:**

void taules()

Pre:--

Post: El jugador ha demanat per fer taules.

void retirar-se()

Pre:--

Post: El jugador ha decidit retirar-se de la partida

## Tipus CarregarPartida

**Descripció general:** Inicialitzador d'una partida nova o a partir d'una anterior.

**Operacions:**

CarregarPartida()

Pre: --

Post: Inicialitza una partida nova o carregar partida.

void carregarPartidaComencada()

Pre: --

Post: Es carrega una partida que ja tenim guardada prèviament i que no esta acabada

void carregarPartidaNoComencada()

Pre: --

Post: Es carrega una partida nova.

## Tipus TancarPartida

**Descripció general:** Guarda una partida en un fitxer. (acabada o no)

**Operacions:**

TancarPartida()

Pre: --

Post: Guarda una partida. (acabada o no acabada) Hi haurà un caràcter al principi del nom del fitxer que mostrarà si esta acabada o no.