



# Projecte: Dames

Metodologia de la Programació

Curs 2024 - 2025

# Determinar els moviments vàlids d'una peça

Tipus i color de la peça:  
O: Fitxa normal blanca  
X: Fitxa normal negra  
D: Dama blanca  
R: Dama negra

8		X		D				X
7							X	
6								
5	R							
4				X				O
3	O		X					
2		O						X
1	O		O		D			
	a	b	c	d	e	f	g	h

Posició inicial fitxa: b2

Moviments vàlids = [] → Inicialitzem la llista de moviments vàlids

Moviment actual = [] → Inicialitzem el moviment actual

Posicions Valides(b2) = [] → Recuperem a quines posicions ens podem moure des de la posició actual

Moviments vàlids: [] → Si no ens podem moure a cap posició acabem i retornem la llista de moviments vàlids

# Determinar els moviments vàlids d'una peça

Tipus i color de la peça:  
O: Fitxa normal blanca  
X: Fitxa normal negra  
D: Dama blanca  
R: Dama negra

8		X		D				X
7							X	
6								
5	R							
4				X				O
3	O							
2		O						X
1	O		O		D			
	a	b	c	d	e	f	g	h

Posició inicial fitxa: b2

Moviments vàlids = [] → Inicialitzem la llista de moviments vàlids

Moviment actual = [] → Inicialitzem el moviment actual

Posicions Valides(b2) = [c3] → Recuperem a quines posicions ens podem moure des de la posició actual

Moviment actual = [c3] → Afegim la posició al moviment actual

Posicions Valides(c3) = [] → Repetim amb la nova posició fins que no ens podem moure a noves posicions vàlides

Moviments vàlids: [[c3]]

# Determinar els moviments vàlids d'una peça

Tipus i color de la peça:  
O: Fitxa normal blanca  
X: Fitxa normal negra  
D: Dama blanca  
R: Dama negra

8		X		D				X
7							X	
6								
5	R							
4								O
3	O		X					
2								X
1	O		O		D			
	a	b	c	d	e	f	g	h

Posició inicial fitxa: b2

Moviments vàlids = []

Moviment actual = []

Posicions Valides(b2) = [d4]

Moviment actual = [d4]

Posicions Valides(d4) = []

Moviments vàlids: [[d4]]

# Determinar els moviments vàlids d'una peça

Tipus i color de la peça:  
O: Fitxa normal blanca  
X: Fitxa normal negra  
D: Dama blanca  
R: Dama negra

Posició inicial fitxa: b2

8		X		D				X
7							X	
6								
5	R				X			
4								O
3	O		X					
2		O						X
1	O		O		D			
	a	b	c	d	e	f	g	h

Moviments vàlids = []

Moviment actual = []

Posicions Valides(b2) = [d4]

Moviment actual = [d4]

Posicions Valides(d4) = [f6]

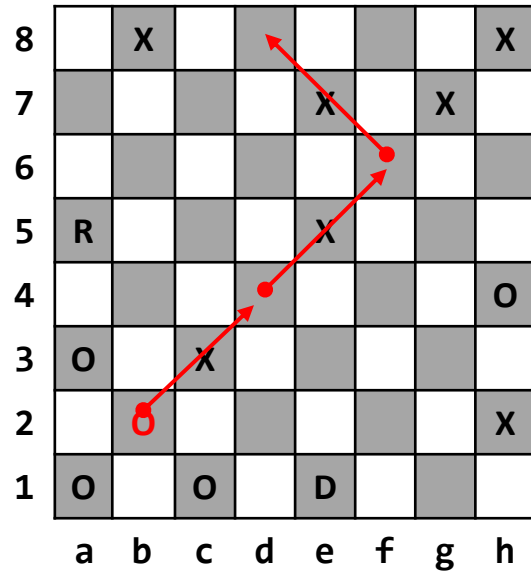
Moviment actual = [d4, f6]

Posicions Valides(f6) = []

Moviments vàlids: [[d4, f6]]

# Determinar els moviments vàlids d'una peça

Tipus i color de la peça:  
O: Fitxa normal blanca  
X: Fitxa normal negra  
D: Dama blanca  
R: Dama negra



Posició inicial fitxa: b2

Moviments vàlids = []

Moviment actual = []

Posicions Valides(b2) = [d4]

Moviment actual = [d4]

Posicions Valides(d4) = [f6]

Moviment actual = [d4, f6]

Posicions Valides(f6) = [d8]

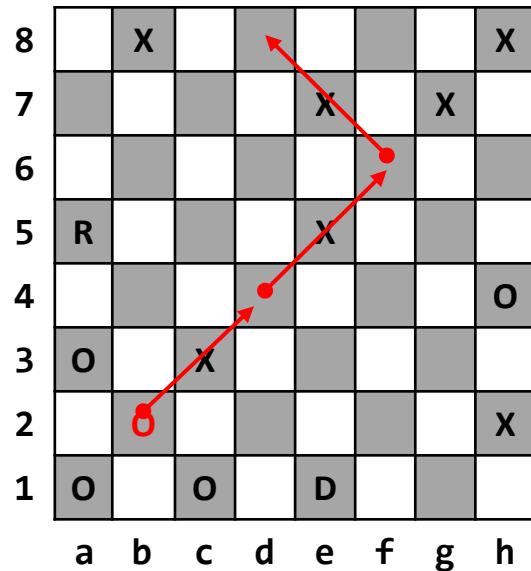
Moviment actual = [d4, f6, d8]

Posicions Valides(d8) = []

Moviments vàlids: [[d4, f6, d8]]

# Determinar els moviments vàlids d'una peça

Tipus i color de la peça:  
O: Fitxa normal blanca  
X: Fitxa normal negra  
D: Dama blanca  
R: Dama negra



Posició inicial fitxa: b2

Moviments vàlids = []

Moviment actual = []

Posicions Valides(b2) = [d4]

Moviment actual = [d4]

Posicions Valides(d4) = [f6]

Moviment actual = [d4, f6]

Posicions Valides(f6) = [d8]

Moviment actual = [d4, f6, d8]

Posicions Valides(d8) = []

Moviments vàlids: [[d4, f6, d8]]

**Algorisme** (si només hi ha un camí possible)

```
movimentsValids = []
movimentActual = []
posicioActual = posicio de la fitxa
posValides = PosicionsValides(posicioActual)
while (posValides != [])
{
    afegir a movimentActual la posició vàlida
    posicioActual = última posició de movimentActual
    posValides = PosicionsValides(posicioActual)
}
if (movimentActual != [])
    afegir movimentActual a movimentsValids
```

# Determinar els moviments vàlids d'una peça

Tipus i color de la peça:  
O: Fitxa normal blanca  
X: Fitxa normal negra  
D: Dama blanca  
R: Dama negra

Què passa si tenim més d'un moviment possible?

8		X		D				X
7							X	
6								
5	R							
4				X				O
3								
2		O						X
1	O		O		D			
	a	b	c	d	e	f	g	h

Posició inicial fitxa: b2

Moviments vàlids = []

Moviment actual = []

Posicions Valides(b2) = [a3, c3]

Tenim dos possibles moviments que hem de processar

a3

c3

Moviment actual = [a3]

Posicions Valides(a3) = []

Moviments vàlids: [[a3]]

Moviment actual = [c3]

Posicions Valides(c3) = []

Moviments vàlids: [[a3], [c3]]

Processem un dels dos moviments i quan acabem hem de tornar enrera a procesar el que ha quedat pendent



# Determinar els moviments vàlids d'una peça

Tipus i color de la peça:  
O: Fitxa normal blanca  
X: Fitxa normal negra  
D: Dama blanca  
R: Dama negra

## Què passa si tenim més d'un moviment possible?

8		X		D				X
7							X	
6								
5	R							
4				X				O
3								
2								X
1	O		O		D			
	a	b	c	d	e	f	g	h

Quan acabem de processar un moviment recuperem el primer moviment de la llista de moviments pendents per continuar processant-lo

Posició inicial fitxa: b2

Moviments vàlids = []

Moviment actual = []

Posicions Valides(b2) = [a3, c3]

a3

Moviment actual = [a3]

Posicions Valides(a3) = []

Moviments vàlids: [[a3]]



Moviment actual = [c3]

Posicions Valides(c3) = []

Moviments vàlids: [[a3], [c3]]

Utilitzem una llista de moviments pendents de processar

Moviments pendents: []

Afegim el segon possible moviment (pendent de procesar) a la llista de moviments pendents

c3

Moviment pendent = [c3]

Moviments pendents: [[c3]]

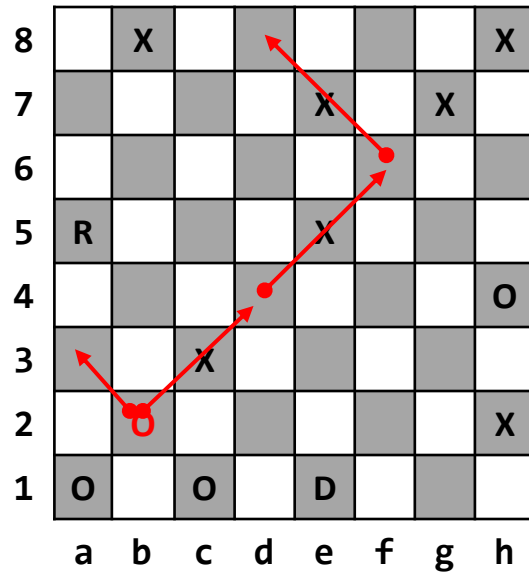


Moviments pendents: []

# Determinar els moviments vàlids d'una peça

Tipus i color de la peça:  
O: Fitxa normal blanca  
X: Fitxa normal negra  
D: Dama blanca  
R: Dama negra

Posició inicial fitxa: b2



Moviments vàlids: []

Moviment actual = []

Posicions Valides(b2) = [a3, d4]

Moviment actual = [a3]

Posicions Valides(a3) = []

Moviments vàlids: [[a3]]

Moviment actual = [d4]

Posicions Valides(d4) = [f6]

Moviment actual = [d4, f6]

Posicions Valides(f6) = [d8]

Moviment actual = [d4, f6, d8]

Posicions Valides(d8) = []

Moviments vàlids: [[a3], [d4, f6, d8]]

Moviments pendents: []

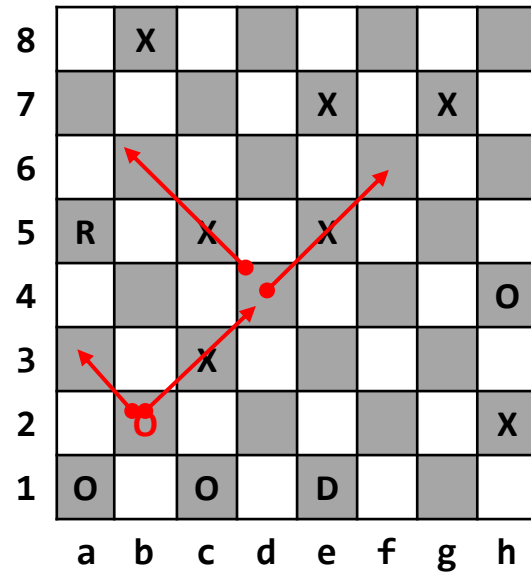
Moviment pendent = [d4]

Moviments pendents: [[d4]]

Moviments pendents: []

# Determinar els moviments vàlids d'una peça

Tipus i color de la peça:  
O: Fitxa normal blanca  
X: Fitxa normal negra  
D: Dama blanca  
R: Dama negra



Posició inicial fitxa: b2

Moviments vàlids: []

Moviment actual = []

Posicions Valides(b2) = [a3, d4]

Moviment actual = [a3]

Posicions Valides(a3) = []

Moviments vàlids: [[a3]]

Moviment actual = [d4]

Posicions Valides(d4) = [f6, b6]

Moviment actual = [d4, f6]

Moviments pendents: []

Moviment pendent = [d4]

Moviments pendents: [[d4]]

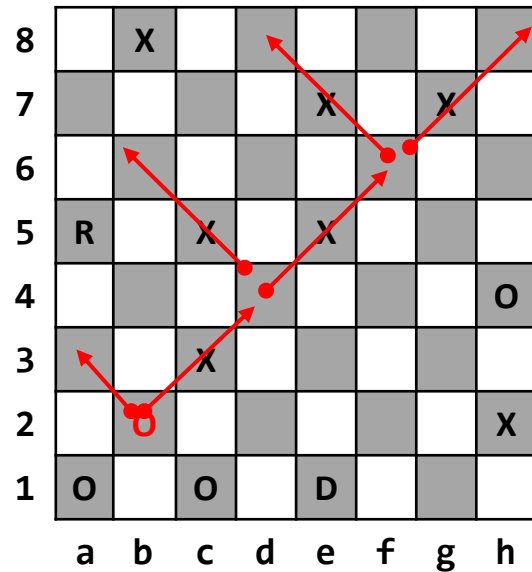
Moviments pendents: []

Moviment pendent = [d4, b6]

Moviments pendents: [[d4, b6]]

# Determinar els moviments vàlids d'una peça

Tipus i color de la peça:  
O: Fitxa normal blanca  
X: Fitxa normal negra  
D: Dama blanca  
R: Dama negra



Moviment actual = [d4, f6]

Moviment pendent = [d4, b6]

Moviments pendents: [[d4, b6]]

Posicions Valides(f6) = [h8, d8]

Moviment actual = [d4, f6, h8]

Moviment pendent = [d4, f6, d8]

Posicions Valides(h8) = []

Moviments pendents: [[d4, b6], [d4, f6, d8]]

Moviments vàlids: [[a3], [d4, f6, h8]]

Moviment actual = [d4, b6]

Moviments pendents: [[d4, b6], [d4, f6, d8]]

Posicions Valides(b6) = []

Moviments pendents: [[d4, f6, d8]]

Moviments vàlids: [[a3], [d4, f6, h8],  
[d4, b6]]

Moviment actual = [d4, f6, d8]  
Posicions Valides(d8) = []

Moviments pendents: []

Moviments vàlids: [[a3], [d4, f6, h8],  
[d4, b6], [d4, f6, d8]]

# Determinar els moviments vàlids d'una peça

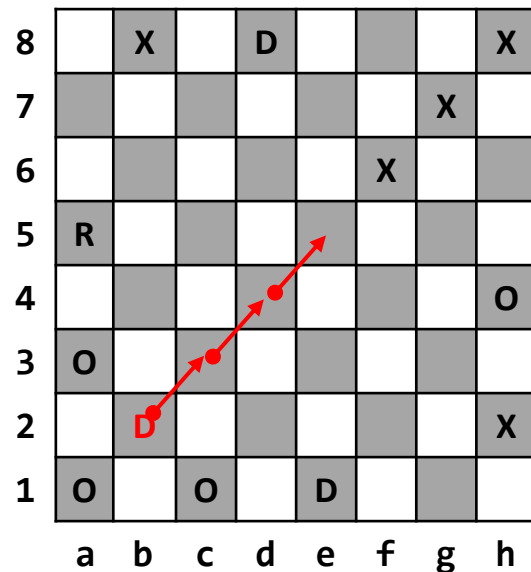
## Algorisme (vàlid per tots els casos)

```
movimentsValids = []
movimentsPendants = []
movimentActual = []
afegir movimentActual a movimentsPendants
posicioActual = posicio de la Fitxa
do
{
    movimentActual = primer element de movimentsPendants
    posValides = PosicionsValides(posicioActual)
    while (posValides != [])
    {
        afegir a movimentActual la primera posicio valida
        per la resta de posicions valides
        movimentPendent = duplicaMoviment(movimentActual)
        afegir la posicio a movimentPendent
        afegir movimentPendent a movimentsPendants
        posicioActual = primera posicio valida
        posValides = PosicionsValides(posicioActual)
    }
    if (movimentActual != [])
        afegir movimentActual a movimentsValids
} while movimentsPendants != []
```

# Determinar els moviments vàlids d'una peça

Tipus i color de la peça:  
O: Fitxa normal blanca  
X: Fitxa normal negra  
D: Dama blanca  
R: Dama negra

## Moviments de les dames



Posició inicial fitxa: b2

Moviments vàlids: []

Moviment actual = []

Posicions Valides(b2) = [c3]

Moviment actual = [a3]

Posicions Valides(a3) = [d4]

Moviment actual = [a3, d4]

Posicions Valides(d4) = [e5]

Moviment actual = [a3, d4, e5]

Posicions Valides(e5) = []

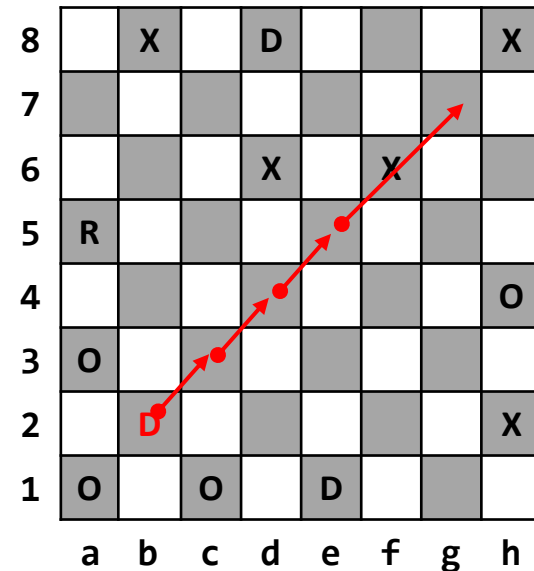
Moviments vàlids: [[c3, d4, e5]]

Moviments pendants: []

# Determinar els moviments vàlids d'una peça

Tipus i color de la peça:  
O: Fitxa normal blanca  
X: Fitxa normal negra  
D: Dama blanca  
R: Dama negra

## Moviments de les dames



Posició inicial fitxa: b2

Moviments vàlids: []

Moviment actual = []

Posicions Valides(b2) = [c3]

Moviment actual = [a3]

Posicions Valides(a3) = [d4]

Moviment actual = [a3, d4]

Posicions Valides(d4) = [e5]

Moviment actual = [a3, d4, e5]

Posicions Valides(e5) = [g7]

Moviment actual = [a3, d4, e5, g7]

Posicions Valides(g7) = []

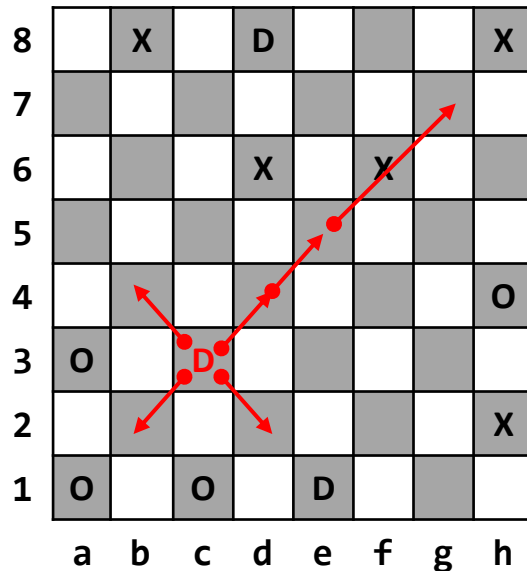
Moviments vàlids: [[c3, d4, e5, g7]]

Moviments pendants: []

# Determinar els moviments vàlids d'una peça

Tipus i color de la peça:  
O: Fitxa normal blanca  
X: Fitxa normal negra  
D: Dama blanca  
R: Dama negra

## Moviments de les dames



Posició inicial fitxa: c3

Moviments vàlids: []

Moviment actual = []

Posicions Valides(c3) = [d4, d2, b2, b4]

Moviment actual = [d4]

Posicions Valides(d4) = [e5]

Moviment actual = [d4, e5]

Posicions Valides(e5) = [g7]

Moviment actual = [d4, e5, g7]

Posicions Valides(g7) = []

Moviments vàlids: [[d4, e5, g7]]

Moviments pendents: []

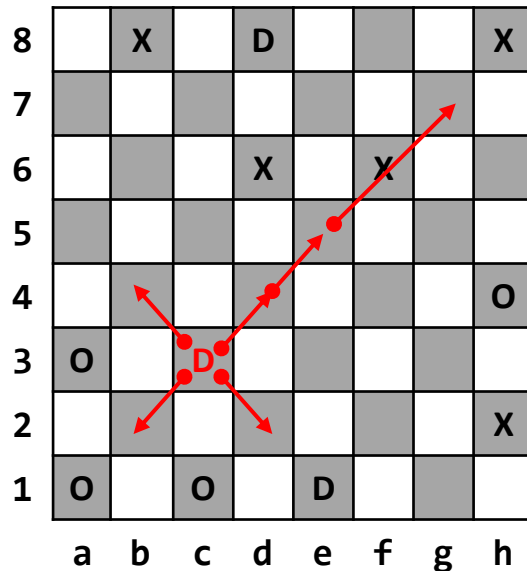
Moviments pendents: [[d2], [b2], [b4]]



# Determinar els moviments vàlids d'una peça

Tipus i color de la peça:  
O: Fitxa normal blanca  
X: Fitxa normal negra  
D: Dama blanca  
R: Dama negra

## Moviments de les dames



Moviments vàlids: [[d4, e5, g7]]

Moviment actual = [d2]

Posicions Valides(d2) = []

Moviments vàlids: [[d4, e5, g7], [d2]]

Moviment actual = [b2]

Posicions Valides(b2) = []

Moviments vàlids: [[d4, e5, g7], [d2], [b2]]

Moviment actual = [b4]

Posicions Valides(b4) = []

Moviments vàlids: [[d4, e5, g7], [d2], [b2], [b4]]

Moviments pendants: [[d2], [b2], [b4]]

Moviments pendants: [[b2], [b4]]

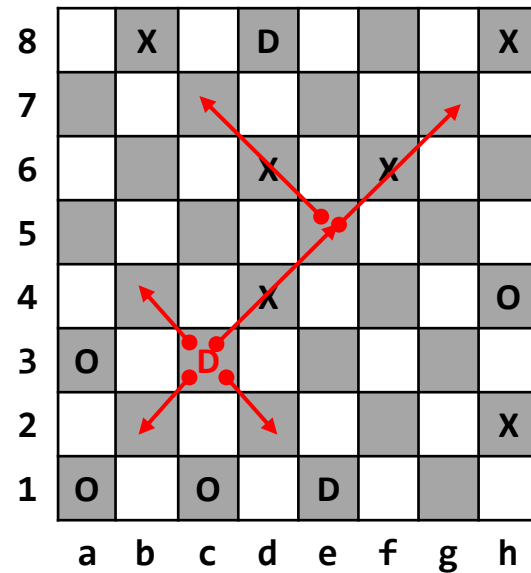
Moviments pendants: [[b4]]

Moviments pendants: []

# Determinar els moviments vàlids d'una peça

Tipus i color de la peça:  
O: Fitxa normal blanca  
X: Fitxa normal negra  
D: Dama blanca  
R: Dama negra

## Moviments de les dames



Posició inicial fitxa: c3

Moviments vàlids: []

Moviment actual = []

Posicions Valides(c3) = [e5, d2, b2, b4]

Moviment actual = [e5]

Posicions Valides(e5) = [g7, c7]

Moviment actual = [e5, g7]

Posicions Valides(g7) = []

Moviments vàlids: [[e5, g7]]

Moviments pendents: []

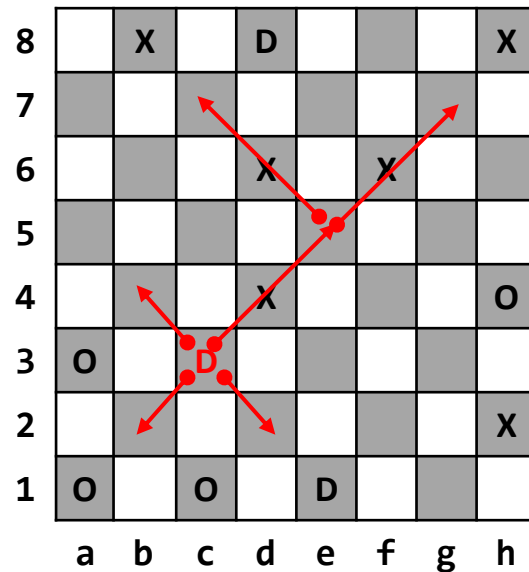
Moviments pendents: [[d2], [b2], [b4]]

Moviments pendents: [[d2], [b2], [b4],  
[e5, c7]]

# Determinar els moviments vàlids d'una peça

Tipus i color de la peça:  
O: Fitxa normal blanca  
X: Fitxa normal negra  
D: Dama blanca  
R: Dama negra

## Moviments de les dames



Moviments vàlids: [[e5, g7]]

Moviment actual = [d2]

Posicions Valides(d2) = []

Moviments vàlids: [[e5, g7], [d2]]

Moviment actual = [b2]

Posicions Valides(b2) = []

Moviments vàlids: [[e5, g7], [d2], [b2]]

Moviment actual = [b4]

Posicions Valides(b4) = []

Moviments vàlids: [[d4, e5, g7], [d2], [b2], [b4]]

Moviment actual = [e5, c7]

Posicions Valides(c7) = []

Moviments vàlids: [[d4, e5, g7], [d2], [b2], [b4], [e5, c7]]

Moviments pendants: [[d2], [b2], [b4], [e5, c7]]

Moviments pendants: [[b2], [b4], [e5, c7]]

Moviments pendants: [[b4], [e5, c7]]

Moviments pendants: [[e5, c7]]

Moviments pendants: []

# Primera versió del projecte

## Estructura de classes

### Tauler

<code>Fitxa m_tauler[N_FILES][N_COLUMNS];</code>
<code>void inicialitza(const string&amp; nomFitxer);</code> <code>void actualitzaMovimentsValids();</code> <code>void getPosicionsPossibles(const Posicio&amp; origen,</code> <code>                            int&amp; nPosicions, Posicio posicionsPossibles[]);</code> <code>bool mouFitxa(const Posicio&amp; origen, const Posicio&amp; desti);</code> <code>string toString() const;</code>

### Fitxa

...
...

### Moviment

...
...

### Posicio

...
<code>Posicio(const string&amp; posicio);</code> <code>bool operator==(const Posicio&amp; posicio) const;</code> ...

# Primera versió del projecte

## Classe Fitxa

- Guarda la informació d'una peça del tauler de joc
- Haureu de pensar quins atributs fan falta per guardar el color (de quin jugador és), el tipus de la peça i el conjunt de moviments vàlids que pot fer la peça en cada moment
- **Haurà de tenir algun atribut per guardar tots els moviments vàlids de la fitxa en un moment determinat de la partida, utilitzant la classe Moviment**
- Afegiu tots els mètodes que facin falta: constructor, getters/setters necessaris, mètodes per gestionar els moviments vàlids de la peça, convertir la fitxa en dama, ...
- Per codificar el tipus i el color de les peces us donem ja declarats aquests dos tipus enumerate:

```
typedef enum
{
    TIPUS_NORMAL,
    TIPUS_DAMA,
    TIPUS_EMPTY
} TipusFitxa;
```

```
typedef enum
{
    COLOR_NEGRE,
    COLOR_BLANC,
} ColorFitxa;
```

# Primera versió del projecte

## Classe Moviment

- Ha de servir per guardar la informació dels moviments que poden fer les peces del tauler
- Haureu de pensar quins atributs fan falta per poder guardar tota la informació del moviment, tenint en compte que en un sol moviment la peça pot saltar per diverses posicions del tauler i matar diverses fitxes contràries.
- Tingueu en compte que en un moviment s'han de poder guardar totes les posicions per les que ha de passar la fitxa.
- A part de les posicions del moviment, pot ser convenient guardar informació addicional (si serveix per matar o no, quantes fitxes contràries es maten, quines fitxes són, ...), depenent de com plantegeu el vostre programa.
- Haureu d'afegir tots els mètodes que facin falta per poder gestionar qualsevol tipus de moviment

# Primera versió del projecte

## Classe Posicio

- Ha de servir per codificar qualsevol de les posicions del tauler
- Penseu els atributs que ha de tenir per poder guardar una posició del tauler
- Afegiu els mètodes que facin falta: constructor, getters/setters necessaris i també algun mètode per convertir la posició en un string o inicialitzar-la a partir d'un string.
- Per poder executar el test que us donarem a Gradescope:
  - Cal que tingui un constructor que rebi com a paràmetre un string en el format que hem indicat abans quan hem explicat la inicialització de la partida.
  - Cal que tingui definida la sobrecàrrega de l'operador == per poder comparar si dues posicions són iguals o no.
- Us podeu inspirar en la classe Posicio que vam fer a la sessió 10 del tema 1.

### Posicio

...

```
Posicio(const string& posicio);  
bool operator==(const Posicio& posicio) const;  
...
```

# Primera versió del projecte

## Classe Tauler

- Guarda la informació del tauler del joc
- Té un atribut per guardar una matriu on cada posició correspon a una de les caselles del joc i conté la informació de la peça que hi ha en aquella posició.
- A les posicions on no hi ha cap peça es pot guardar una peça de tipus FITXA\_EMPTY
- Haureu d'implementar els mètodes públics de la classe Tauler. Per implementar-los haureu d'implementar altres mètodes privats auxiliars que us ajudin a estructurar el codi seguint els principis de la descomposició modular.

### Tauler

```
Fitxa m_tauler[N_FILES][N_COLUMNS];

void inicialitza(const string& nomFitxer);
void actualitzaMovimentsValids();
bool mouFitxa(const Posicio& origen, const Posicio& desti);
void getPosicionsPossibles(const Posicio& origen,
                           int& nPosicions, Posicio posicionsPossibles[]);
string toString() const;
```



# Primera versió del projecte

## Classe Tauler: mètodes

`void inicialitza(const string& nomFitxer)`

- Inicialitza el tauler de joc a partir de la informació d'un fitxer de text en el format que s'ha explicat abans

Tipus i color de la peça:  
O: Fitxa normal blanca  
X: Fitxa normal negra  
D: Dama blanca  
R: Dama negra

O a1  
O c1  
D e1  
O b2  
O h4  
D d8  
X b8  
X h8  
R a5  
X g7  
X f6  
X h2

8		X		D				X
7						X		
6					X			
5	R							
4							O	
3								
2		O						X
1	O		O		D			
	a	b	c	d	e	f	g	h

Posició dins del tauler

Tauler

```
Fitxa m_tauler[N_FILES][N_COLUMNS];
```

```
void inicialitza(const string& nomFitxer);
```

```
void actualitzaMovimentsValida();
```

```
void getPosicionsPossibles(const Posicio& origen,  
                           int& nPosicions, Posicio posicionsPossibles[]);
```

```
bool mouFitxa(const Posicio& origen, const Posicio& desti);
```

```
string toString() const;
```

# Primera versió del projecte

## Classe Tauler: mètodes

`const` actualitzaMovimentsValids() `const`

- Actualitzar tots els moviments vàlids que pot qualsevol de les peces del tauler
- Per guardar cadascun dels moviments es pot fer servir la **classe Moviment**.
- Penseu com podeu guardar tots els moviments de totes les peces del tauler utilitzant les classes Fitxa, Moviment i Posicio
- Per implementar aquest mètode penseu bé quins mètodes auxiliars us fan falta a la classe Tauler i també quins mètodes hauríeu d'afegir a les classes Fitxa i Moviment

### Tauler

```
Fitxa m_tauler[N_FILES][N_COLUMNS];

void inicialitza(const string& nomFitxer);
void actualitzaMovimentsValids();
void getPosicionsPossibles(const Posicio& origen,
                           int& nPosicions, Posicio posicionsPossibles[]);
bool mouFitxa(const Posicio& origen, const Posicio& desti);
string toString() const;
```

# Primera versió del projecte

## Tauler

```
Fitxa m_tauler[N_FILES][N_COLUMNS];

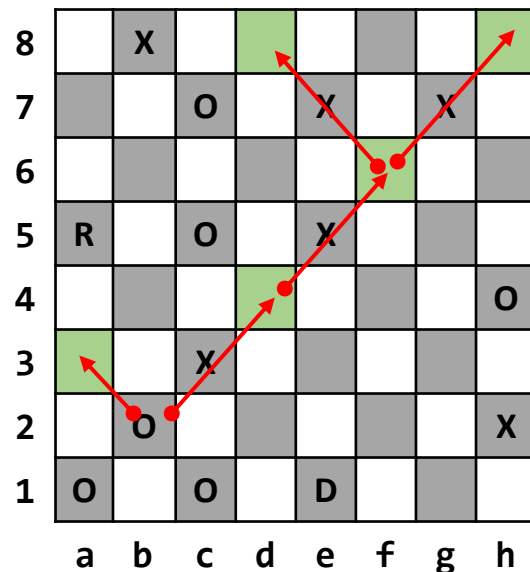
void inicialitza(const string& nomFitxer);
void actualitzaMovimentsValids();
void getPosicionsPossibles(const Posicio& origen,
                           int& nPosicions, Posicio posicionsPossibles[]);
bool mouFitxa(const Posicio& origen, const Posicio& desti);
string toString() const;
```

## Classe Tauler: mètodes

```
void getPosicionsPossibles(const Posicio& origen, int& nPosicions,
                           Posicio posicionsPossibles[])
```

- El mètode retorna a l'array posicionsPossibles totes les posicions on es podria moure la peça que ocupa la posició origen, tenint en compte tots els passos de qualsevol moviment vàlid de la peça. nPosicions indica el nº total de posicions possibles.

## Exemple:



Posicions possibles de la posició 'b2':  
[ 'a3', 'd4', 'f6', 'd8', 'h8' ]

# Primera versió del projecte

Tauler

```
Fitxa m_tauler[N_FILES][N_COLUMNS];

void inicialitza(const string& nomFitxer);
void actualitzaMovimentsValids();
void getPosicionsPossibles(const Posicio& origen,
                           int& nPosicions, Posicio posicionsPossibles[]);
bool mouFitxa(const Posicio& origen, const Posicio& desti);
string toString() const;
```

## Classe Tauler: mètodes

`bool` mouFitxa (`const Posicio&` origen, `const Posicio&` desti)

- Mou la peça que ocupa la posició del paràmetre origen a la posició del paràmetre desti.
- Ha de comprovar que la posició destí estigui dins dels moviments vàlids de la peça.
- Si no ho està no fa el moviment i retorna false.
- A part de moure la peça de la posició origen a la posició destí ha de fer el següent:
  - Eliminar del tauler totes les peces que es maten fent el moviment.
  - Si la peça arriba al final del tauler, convertir-la en una dama.
  - Si el moviment no ha matat una peça contrària quan hi ha altres moviments que sí permeten fer-ho o el moviment no ha matat el màxim de peces contràries possibles, *bufar* una peça del jugador (eliminar la peça del tauler) , seguint els criteris que s'han explicat a les regles del joc.

# Primera versió del projecte

## Tauler

```
Fitxa m_tauler[N_FILES][N_COLUMNES];
```

```
void inicialitza(const string& nomFitxer);  
void actualitzaMovimentsValids();  
void getPosicionsPossibles(const Posicio& origen,  
                           int& nPosicions, Posicio posicionsPossibles[]);  
bool mouFitxa(const Posicio& origen, const Posicio& desti);  
string toString() const;
```

## Classe Tauler: mètodes

`string ToString() const`

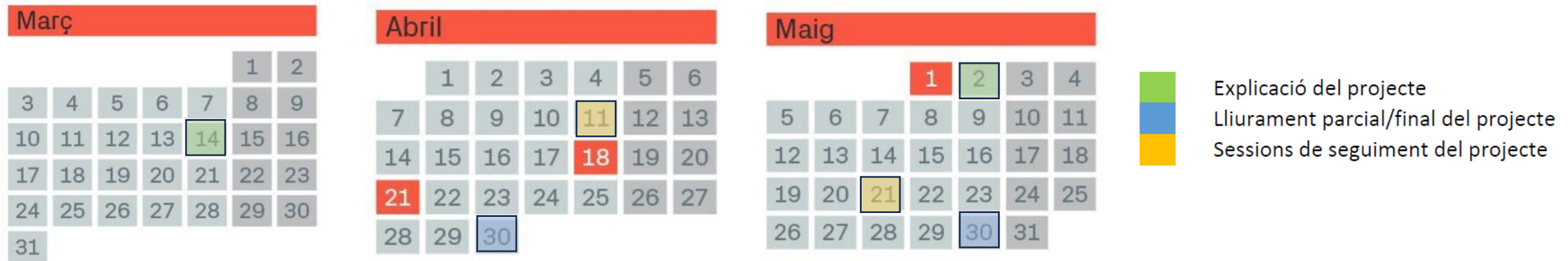
- Genera un string amb l'estat actual del tauler de joc en el format que especifiquem a continuació

8		X		D				X
7							X	
6								
5	R							
4								O
3			O				O	
2		O						X
1	O		O		D			
	a	b	c	d	e	f	g	h



```
8: _ X _ D _ _ _ X  
7: _ _ _ _ _ X _  
6: _ _ _ _ _ _ _  
5: R _ _ _ _ _ _  
4: _ _ _ _ _ O  
3: _ _ O _ _ O _  
2: _ O _ _ _ _ X  
1: O _ O _ D _ _ _  
   A B C D E F G H
```

# Organització del projecte



- La propera **sessió de seguiment** del projecte serà el **divendres 11 d'abril**.
  - Revisió del que heu avançat en el projecte, de com plantegeu el disseny de les classes (atributs i mètodes de cada classe) i la implementació dels mètodes de la classe Tauler (**passos 1-5 descrits a la següent diapositiva**)
  - Comptarà un **20% de la nota del lliurament parcial**.
- El lliurament parcial del projecte amb la implementació de la primera versió serà el dimecres **30 d'abril**. Farem una avaluació online individual del lliurament de cada grup.
- A Gradescope tindreu un **test d'autoavaluació** dels mètodes públics de la classe Tauler que heu d'implementar.
- Us demanarem crear un **repositori a GitHub** amb el codi del projecte i que ens hi doneu accés per poder veure i avaluar el vostre codi.

# Propers passos: alguns consells

1. Penseu quins mètodes us poden fer falta a la **classe Posicio**. Feu una primera **implementació** de la classe.
2. Penseu quins atributs i mètodes us poden fer falta a la classe Fitxa. Feu la **declaració de la classe** (no cal que implementeu els mètodes, de moment)
3. Penseu quins atributs i mètodes us poden fer falta a la **classe Moviment**. Feu la **declaració de la classe** (no cal que implementeu els mètodes, de moment)
4. Feu la **implementació del mètode inicialitza** de la **classe Tauler** per poder inicialitzar el tauler de fitxer. **Implementeu els mètodes que us facin falta** a les classes **Fitxa i Posicio**, Si fa falta, modifiqueu la declaració inicial que hagueu fet d'aquestes classes.
5. Penseu com podeu **descomposar el codi del mètode actualitzaMovimentsValids** amb funcions més petites i simples.
  - Penseu bé com podeu organitzar el codi del mètode per poder implementar de la forma més simple possible i evitant duplicitats, el codi per recuperar qualsevol tipus de moviment vàlid per cada qualsevol tipus de peça de qualsevol color
6. Comenceu a implementar el codi del mètode actualitzaMoviments, pels moviments més simples (desplaçaments sense matar de fitxes normals i desplaçaments matant només una fitxa contrària).