

Projecte: Tetris

Metodologia de la Programació

Curs 2023 - 2024

Objectius de la sessió

1. Presentació i organització del projecte
2. Explicar les tasques a realitzar a la primera versió del projecte

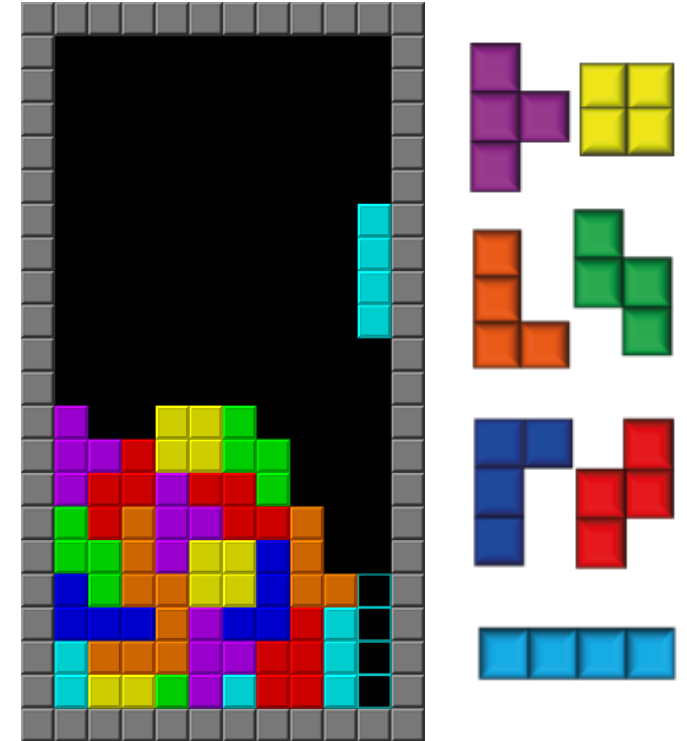
Organització del projecte

- **Grups de 2 persones:** trobareu a Caronte una inscripció a grups perquè pugueu indicar amb qui fareu el projecte.
 - Apunteu-vos a un dels subgrups corresponents al vostre grup de classe.
- El projecte és part del **treball autònom** de l'assignatura. Durant les hores de classe dedicarem algunes sessions a seguiment i avaluació del treball.

Tetris

- El joc consta d'un tauler rectangular de dimensions $F \times C$, on F és el nombre de files i C el nombre de columnes, en el que aniran caient figures.
- Les peces es poden rotar i desplaçar horitzontalment segons calgui per tal que encaixin amb les peces prèviament col·locades.
- Sempre que les peces col·locades emplenin una o varies files senceres, aquestes s'eliminaran fent que totes les cel·les del damunt, es desplacin cap avall. Per cada fila eliminada, s'incrementa la puntuació del jugador. Si les files eliminades són 3 de seguides, el jugador rep puntuació extra.
- Cada vegada que es sobrepassi un cert llindar de puntuació, el nivell del joc augmentarà. És a dir, les peces cauran més ràpid.
- El joc finalitza quan les peces col·locades sobrepassen el nombre de files (F) del tauler.

<https://en.wikipedia.org/wiki/Tetris>



Objectiu del projecte

Disseny i desenvolupament d'una versió completa del joc del **tetris** original, posant en pràctica els conceptes que anem explicant a les sessions de classe.

El vostre programa final haurà de permetre (com a mínim):

- **Jugar una partida**
 - Generació de les figures que van apareixent.
 - Permetre el moviment de la figura (dreta, esquerra i girs en les dues direccions) mentre va baixant pel tauler de joc.
 - Encaixar la figura amb les altres figures ja col·locades.
 - Eliminar les files completades i fer baixar les que estan per sobre.
 - Actualitzar la puntuació de la partida a mesura que es van eliminant files.
 - Incrementar el nivell (incrementar la velocitat del joc) en funció de la puntuació aconseguida.
 - Comprovar el final de la partida.
- **Visualització gràfica** de la partida i interacció amb el jugador per fer els moviments

Podreu afegir **funcionalitats extres**, si voleu, que comptaran positivament com a **punts addicionals** a la **nota final** del projecte.

Planificació del projecte

El projecte el desenvoluparem en dues fases, que es correspondran amb el lliurament parcial i el lliurament final del projecte.

Primera versió del projecte:

- Mostrar una figura al tauler de joc.
- Moviment de la figura (dreta, esquerra, avall i girs en les dues direccions), comprovant que el moviment és vàlid (no col·lisiona amb altres figures ja col·locades o amb els límits del tauler).
- Encaixar la figura amb les altres figures ja col·locades.
- Eliminar les files completades i fer baixar les que estan per sobre.
- Guardar l'estat actual del tauler en un fitxer.
- En aquesta primer versió treballarem sense visualització gràfica. Tindreu un test d'autoavaluació a Caronte per poder validar el correcte funcionament de les diferents funcionalitats.

Planificació del projecte

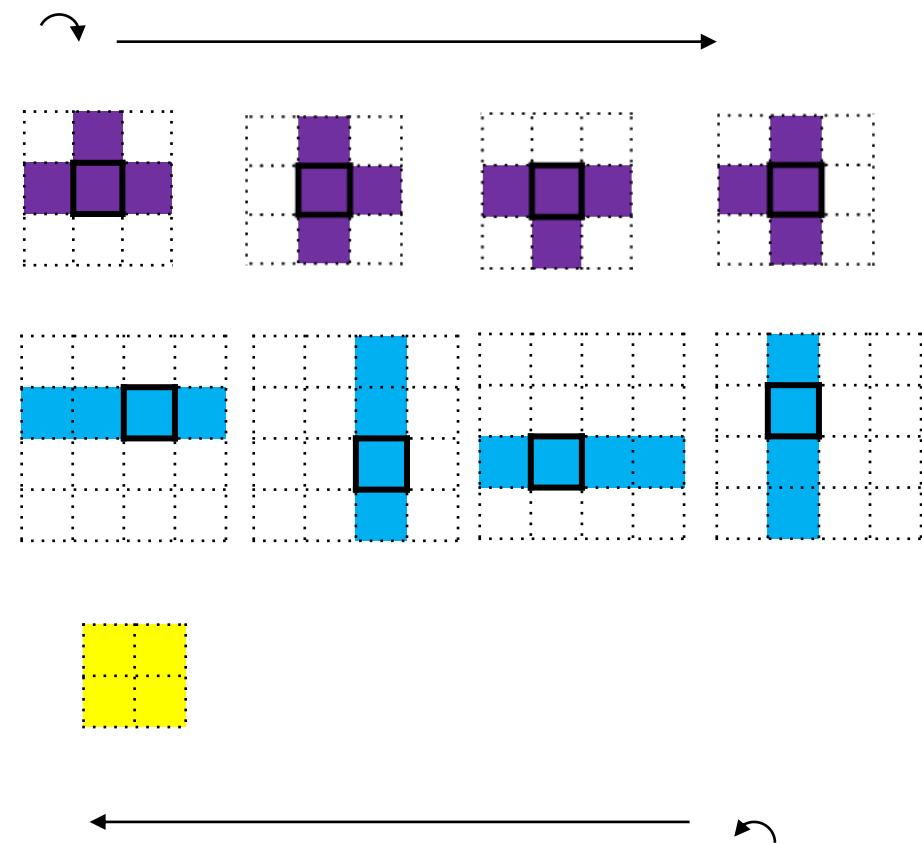
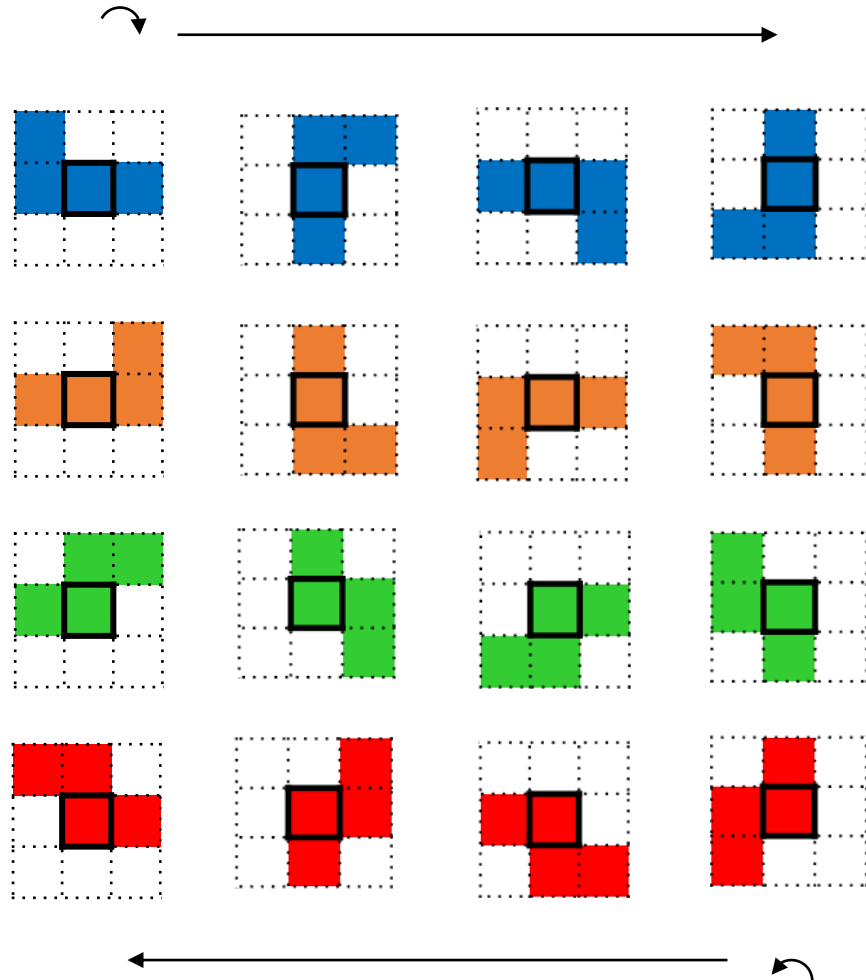
El projecte el desenvoluparem en dues fases, que es correspondran amb el lliurament parcial i el lliurament final del projecte.

Segona versió del projecte:

- Implementar la part gràfica del joc i la interacció del jugador amb el tauler.
- Implementar el desenvolupament complet d'una partida:
 - Generació de les noves figures que van apareixent.
 - Actualitzar la puntuació de la partida a mesura que es van eliminant files.
 - Incrementar el nivell (incrementar la velocitat del joc) en funció de la puntuació aconseguida.
 - Comprovar el final de la partida.
- Possibilitat d'afegir funcionalitats extra.

Funcionament del joc

Girs de les figures

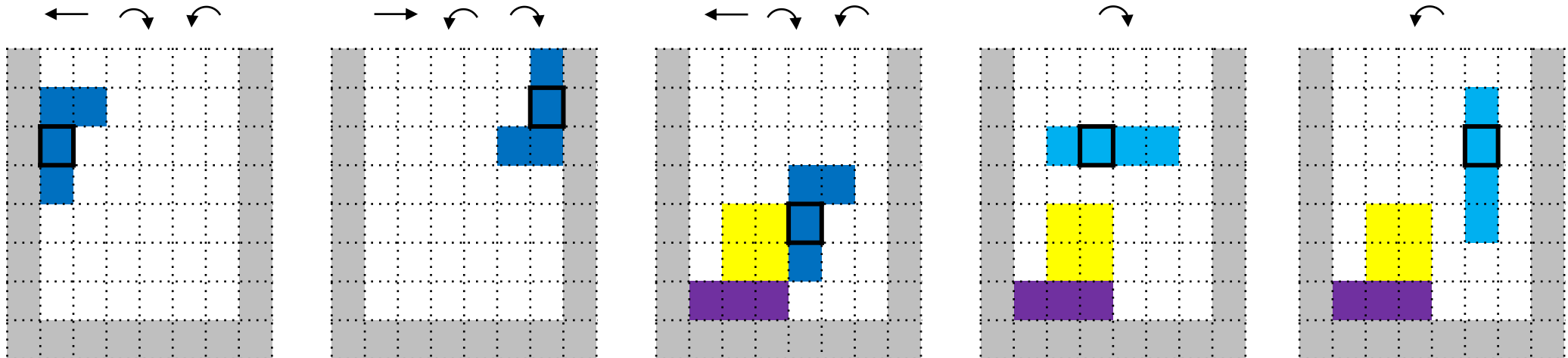


Funcionament del joc

Girs i moviments de les figures

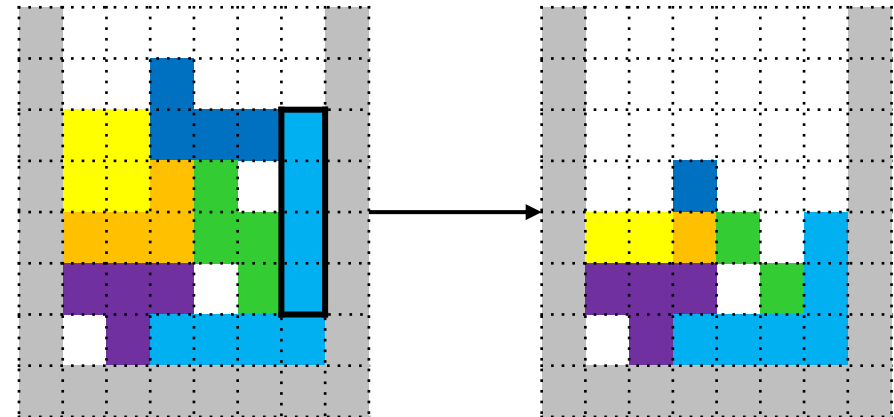
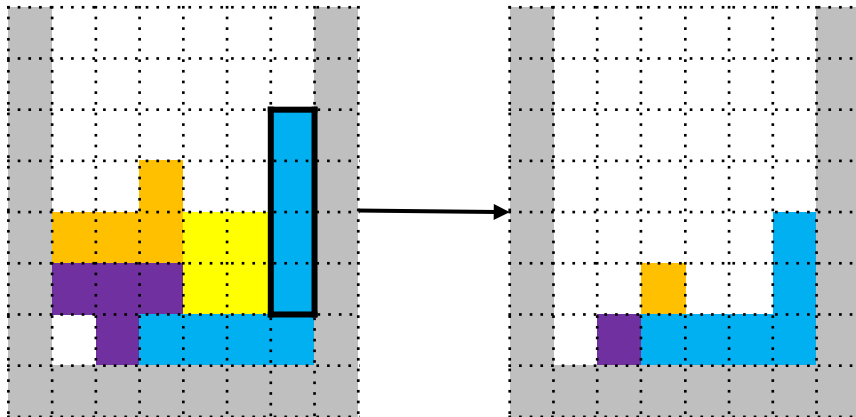
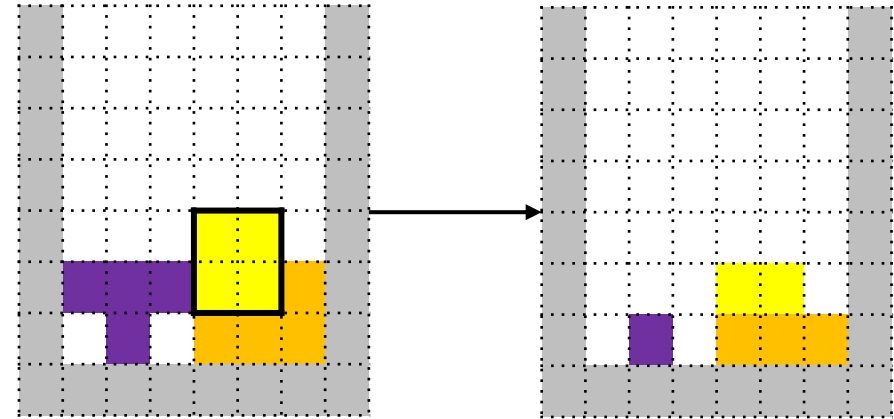
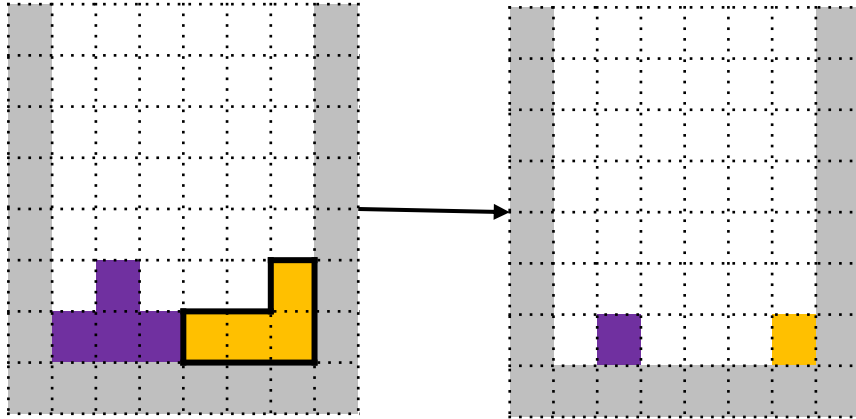
- Els girs i desplaçaments laterals de la figura només seran vàlids si la figura no col·lisiona ni amb els límits laterals del tauler ni amb les altres figures que ja s'han col·locat prèviament al tauler.
- En els girs hem de tenir en compte que la casella de referència queda fixa i són les altres caselles les que es mouen.

Exemples de girs i moviments no vàlids



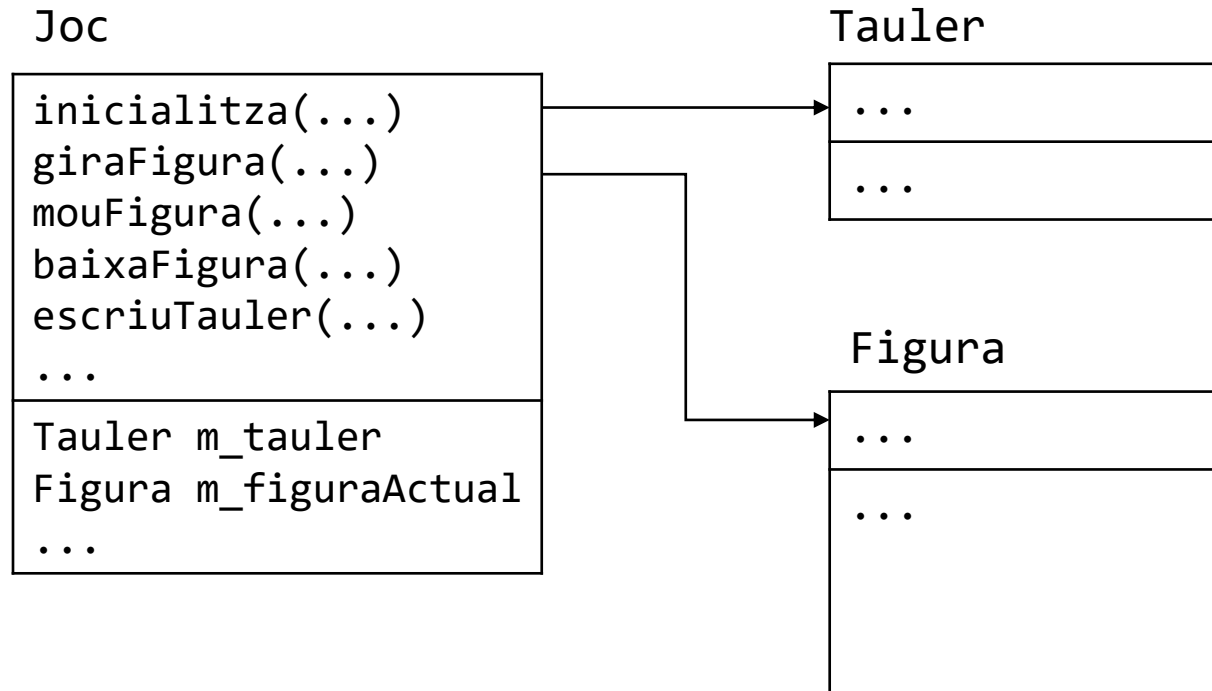
Funcionament del joc

Eliminació de files



Primera versió del projecte: implementació

Estructura de classes



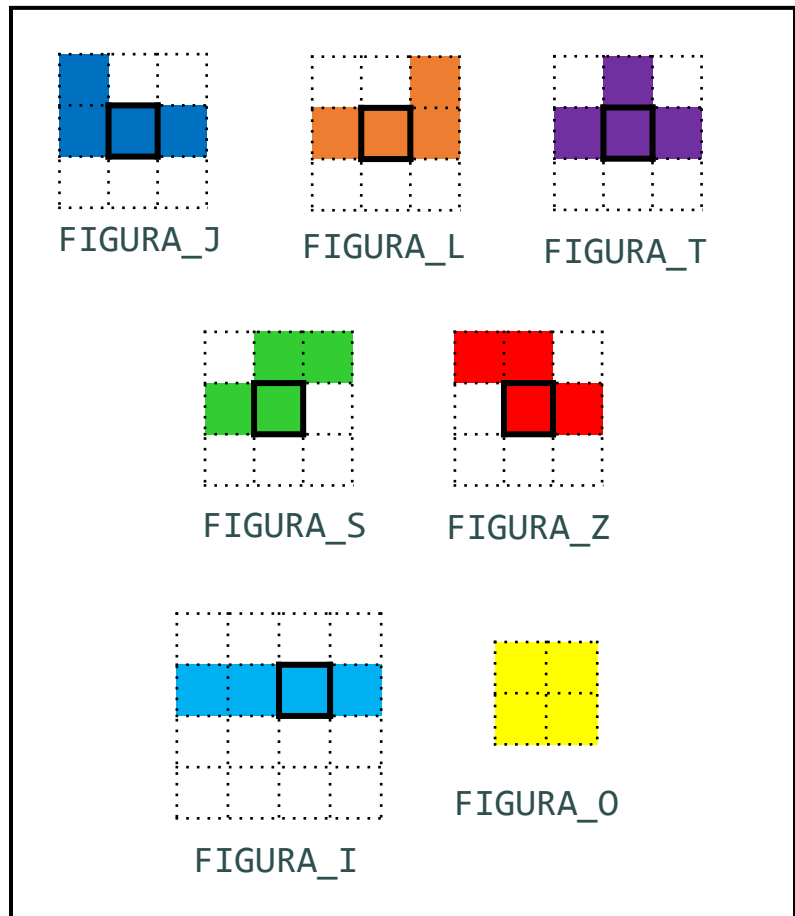
Primera versió del projecte: implementació

Classe Figura

- Ha de servir per poder gestionar figures que van caient.
- Ha de poder guardar tota la informació necessària per gestionar la caiguda de la figura: tipus i color de la figura, posició actual, forma actual de la figura.
- Ha de tenir mètodes que permetin fer el moviment i el gir de la figura i la interacció amb el tauler:
 - Desplaçar-se lateralment
 - Baixar dins del tauler
 - Girar
 - Recuperar la forma actual de la figura
 - ...

Primera versió del projecte: implementació

Classe Figura: Tipus i color de les figures



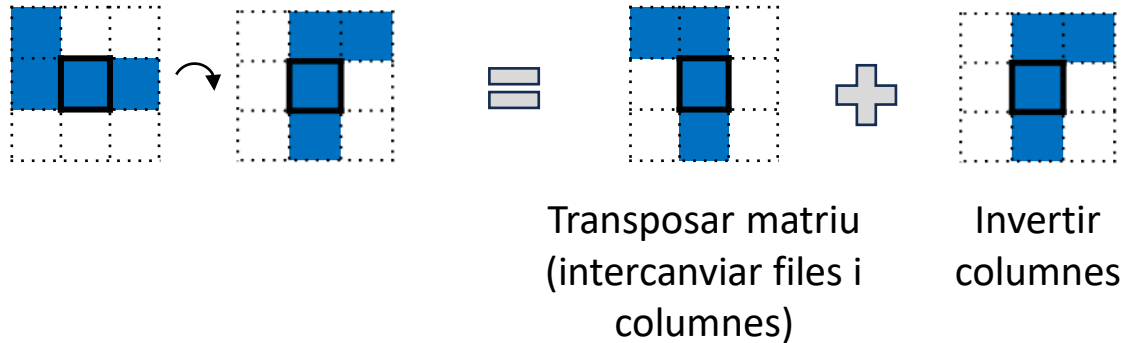
```
typedef enum
{
    NO_FIGURA = 0,
    FIGURA_O,
    FIGURA_I,
    FIGURA_T,
    FIGURA_L,
    FIGURA_J,
    FIGURA_Z,
    FIGURA_S
} TipusFigura;
```

```
typedef enum
{
    COLOR_NEGRE = 0,
    COLOR_GROC,
    COLOR_BLAUCEL,
    COLOR_MAGENTA,
    COLOR_TARONJA,
    COLOR_BLAUFOSC,
    COLOR_VERMELL,
    COLOR_VERD,
    NO_COLOR
} ColorFigura;
```

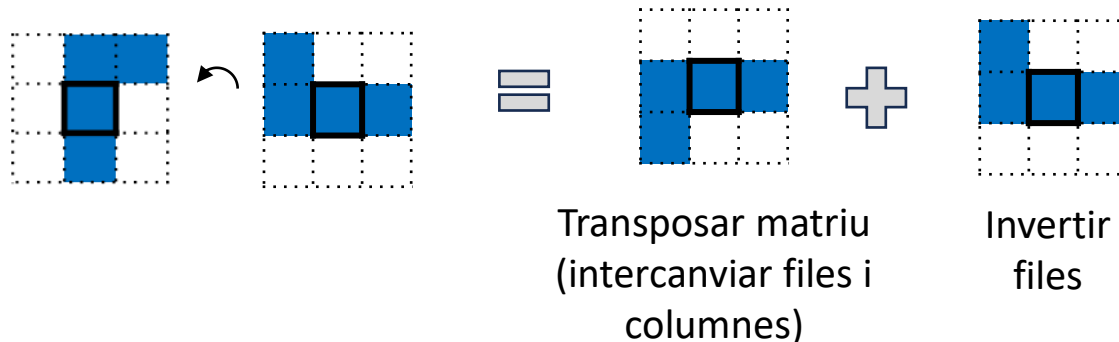
Primera versió del projecte: implementació

Classe Figura: Implementació dels girs

GIR HORARI



GIR ANTI-HORARI



Primera versió del projecte: implementació

Classe Tauler

- Ha de servir per poder gestionar el tauler de joc.
- Ha de poder guardar quines caselles del tauler estan ocupades per figures ja col·locades i per la figura actual que està caient.
- Ha de tenir tots els mètodes necessaris per poder comprovar que els moviments i girs de la figura són vàlids, col·locar la figura a la seva posició final i eliminar les files que es van completant.
- Penseu bé quins mètodes públics ha de tenir aquesta classe. Poseu només com a mètodes públics els que sigui necessari utilitzar des de la classe Joc.
- Si és necessari, descomposeu els mètodes públics en altres mètodes més simples. Definiu com a privats aquests mètodes auxiliars.

Tauler

...

ColorFigura m_tauler[N_FILES][N_COLUMNS]

...

Primera versió del projecte: implementació

Classe Joc

- Ha de servir per poder gestionar el funcionament global del joc
- Ha de poder guardar i gestionar el tauler de joc (utilitzant la classe Tauler), la figura actual que està caient (utilitzant la classe Figura), la informació de l'estat de la partida (puntuació, nivell, etc.) i qualsevol altra informació que considereu necessària per gestionar la partida.
- Ha de tenir aquesta interfície pública:

```
void inicialitza(const string& nomFitxer);  
bool giraFigura(DireccioGir direccio);  
bool mouFigura(int dirX);  
int baixaFigura();  
void escriuTauler(const string& nomFitxer);
```

Joc

```
inicialitza(...)  
giraFigura(...)  
mouFigura(...)  
baixaFigura(...)  
escriuTauler(...)  
...
```

```
Tauler m_tauler  
Figura m_figura  
...
```


Primera versió del projecte: implementació

Classe Joc: mètodes

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

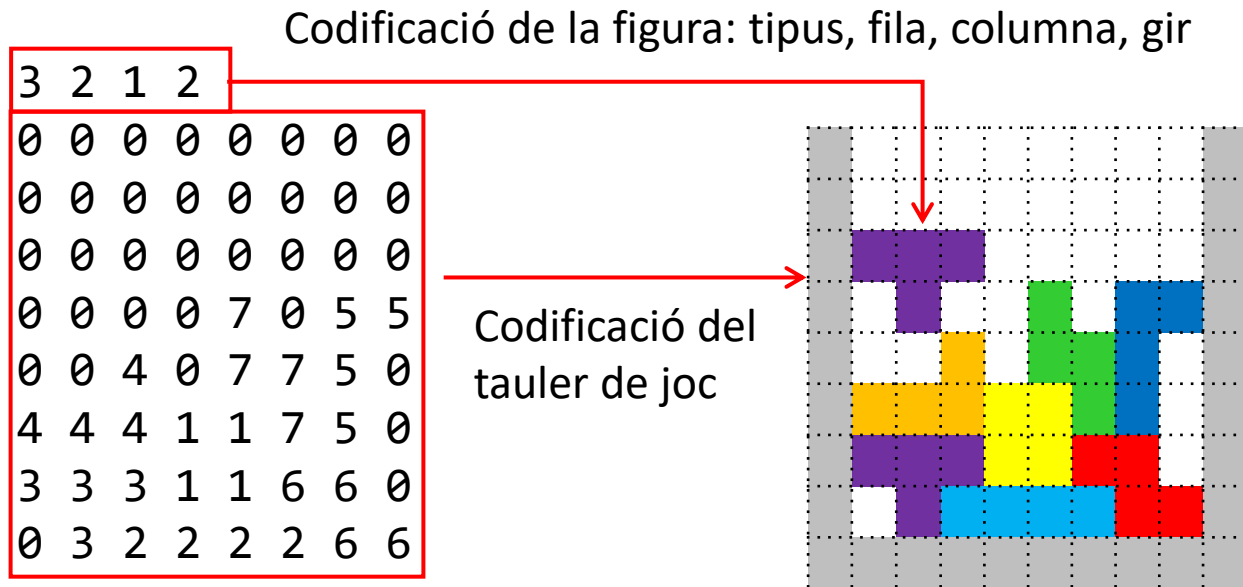
- Aquest mètode el farem servir només pel test automàtic del lliurament parcial.
- Servirà per inicialitzar la partida a partir de la informació que llegirem del fitxer que es passa com a paràmetre.
- Permetrà inicialitzar el tauler de joc amb algunes caselles ja ocupades per figures col·locades i fixar el tipus, forma i posició de la figura actual que està caient.

Primera versió del projecte: implementació

Classe Joc: mètodes

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

- El fitxer tindrà aquest format:



Codi Gir

0 - 3, segons ordre dels girs en sentit horari a la descripció de les figures (pàgina 8)

Codi Tauler

POSICIO_BUIDA = 0,
COLOR_GROC = 1
COLOR_BLAUCEL = 2
COLOR_MAGENTA = 3
COLOR_TARONJA = 4
COLOR_BLAUFOSC = 5
COLOR_VERMELL = 6
COLOR_VERD = 7

Codi Figura

FIGURA_O = 1
FIGURA_I = 2
FIGURA_T = 3
FIGURA_L = 4
FIGURA_J = 5
FIGURA_Z = 6
FIGURA_S = 7

Primera versió del projecte: implementació

Classe Joc: mètodes

```
bool giraFigura(DireccioGir direccio);
```

- Gira la figura actual en la direcció indicada, només si el gir és vàlid (la figura no col·lisiona ni amb els límits del tauler ni amb altres figures ja col·locades al tauler).
- Retorna un booleà en funció de si el gir és vàlid segons la posició de la figura i el contingut del tauler.

```
typedef enum  
{  
    GIR_HORARI = 0,  
    GIR_ANTI_HORARI  
} DireccioGir;
```

Primera versió del projecte: implementació

Classe Joc: mètodes

```
bool mouFigura(int dirX);
```

- Desplaça la figura lateralment en la direcció indicada (+1 cap a la dreta, -1 cap a l'esquerra), només si el moviment és vàlid (la figura no col·lisiona ni amb els límits del tauler ni amb altres figures ja col·locades al tauler).
- Retorna un booleà en funció de si el moviment és vàlid segons la posició de la figura i el contingut del tauler.

Primera versió del projecte: implementació

Classe Joc: mètodes

```
int baixaFigura();
```

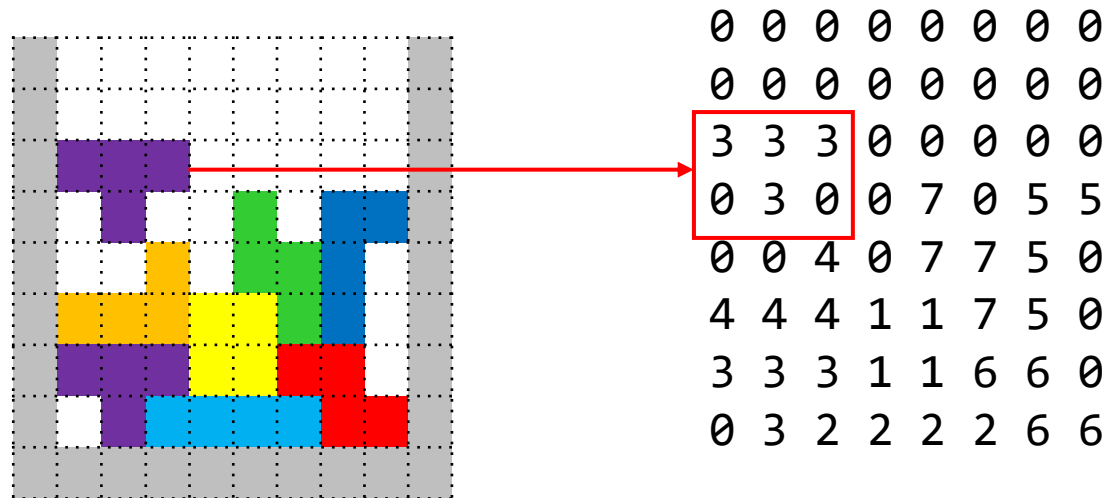
- Desplaça la figura verticalment una posició cap a baix.
- Si el desplaçament provoca que la figura col·lisió amb altres figures ja col·locades al tauler s'ha de col·locar la figura al tauler, comprovar si es completa alguna fila i, si és el cas, eliminar les files completades.
- Retorna el nombre de files completades.

Primera versió del projecte: implementació

Classe Joc: mètodes

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

- Ha de escriure l'estat actual del tauler en el fitxer que es passa com a paràmetre, seguint el mateix format que hem descrit pel mètode `inicialitza`. Només ha d'escriure els valors actuals del tauler, no la informació de la figura actual.
- La figura s'ha de mostrar com si estigués col·locada al tauler a la seva posició actual.



Organització del projecte

març						
L	M	X	J	V	S	D
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

abril						
L	M	X	J	V	S	D
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

maig						
L	M	X	J	V	S	D
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

juny						
L	M	X	J	V	S	D
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30



Explicació del projecte
Lliurament parcial/final del projecte
Sessions de seguiment del projecte

- La propera sessió de seguiment del projecte serà el **dimecres 17 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 Joc (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 divendres **3 de maig**. Farem una avaluació online individual del lliurament de cada grup.
- A Caronte tindreu un **test d'autoavaluació** dels mètodes públics de la classe Joc 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 atributs i mètodes us poden fer falta a la classe `Figura`. Feu una primera implementació de la classe.
2. Feu la declaració bàsica de la classe `Tauler` (només atributs i constructor).
3. Feu la implementació del mètode `inicialitza` de la classe `Joc` per poder inicialitzar el tauler de joc des de fitxer. Penseu quin mètode o mètodes us fa falta afegir a la classe `Tauler`, i implementeu-los. Si us fa falta afegir algun mètode de la classe `Figura` afegiu-lo a la definició i implementació de la classe.
4. Feu la implementació del mètode `escriuTauler` de la classe `Joc` per poder guardar el resultat final del tauler en un fitxer. Penseu si us fa falta afegir algun mètode a la classe `Tauler`, i implementeu-los. Si us fa falta afegir o modificar algun mètode a la classe `Figura` afegiu-lo també.
5. Penseu com podeu implementar els mètodes `giraFigura`, `mouFigura` i `baixaFigura` de la classe `Joc` a partir de crides a mètodes de la classe `Tauler` i a la classe `Figura`. Penseu bé com definir els mètodes de les classes `Tauler` per poder implementar fàcilment la col·lisió de la figura actual amb el tauler i l'eliminació de les files del tauler.
6. Feu la implementació del codi de tots els mètodes de les classes `Joc`, `Tauler` i `Figura` necessaris per implementar els mètodes anteriors