

Puissance 4 : version graphique

Le jeu



Puissance 4 est un jeu de stratégie combinatoire qui se joue exclusivement à deux joueurs.

Les règles de Puissance 4

Début de partie

Pour commencer une partie de puissance 4, on désigne le joueur qui commence.

Déroulement de la partie

Il met un de ses jetons de couleur dans l'une des colonnes de son choix. Le jeton se retrouve en bas de la colonne. Le deuxième joueur insère à son tour son jeton, de l'autre couleur dans la colonne de son choix. Les joueurs répètent la manœuvre jusqu'à ce qu'un joueur aligne 4 pions.

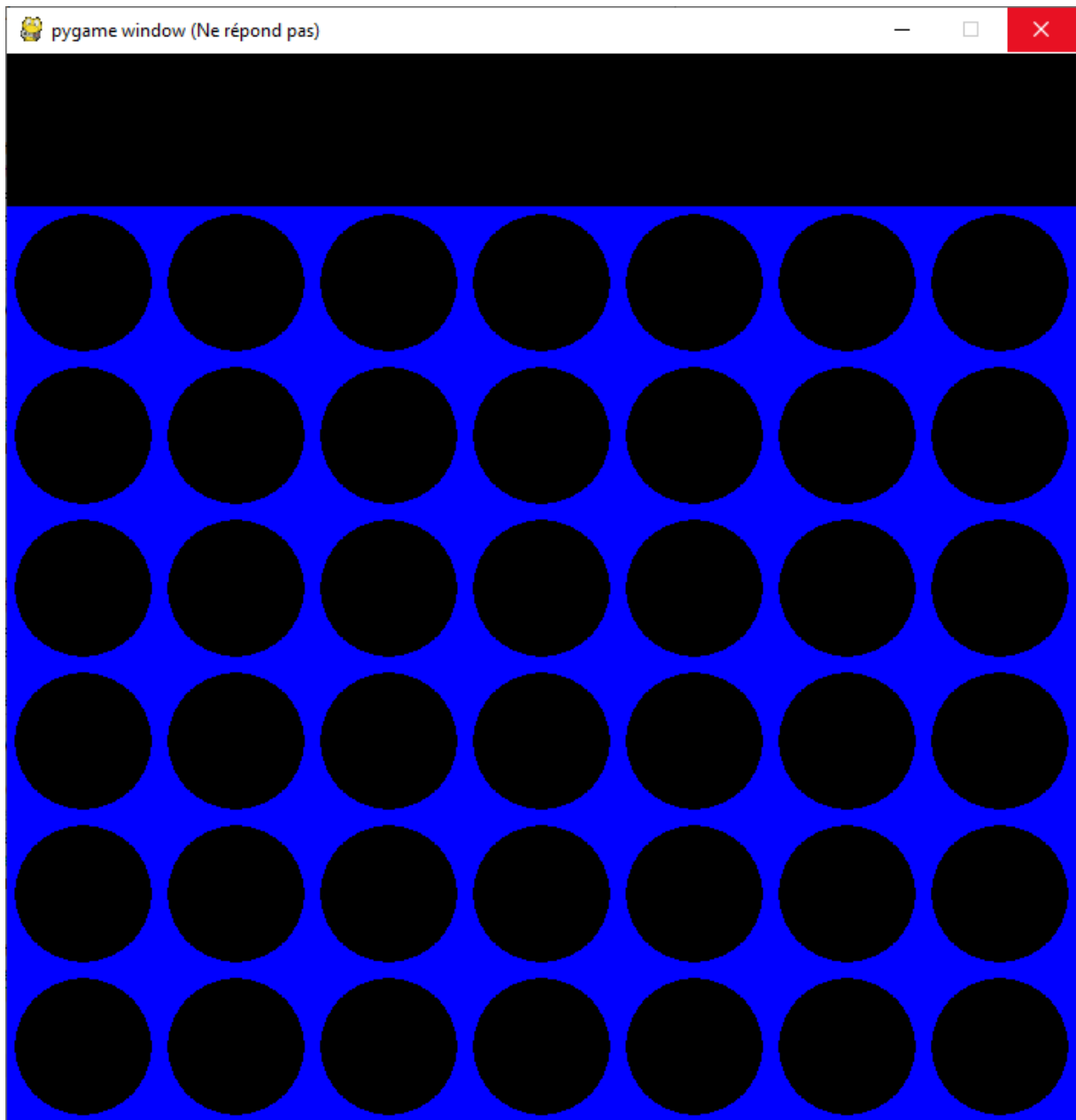
Fin de la partie

Le premier joueur à aligner 4 pions, que ce soit horizontalement, verticalement ou en diagonale, remporte la partie.

Nous partons de la version console du jeu, qui implémente tout le moteur du jeu.

Il ne nous reste qu'à ajouter la partie graphique :

Affichage du plateau de jeu



Le plateau bleu et la première ligne noire

Le plateau est constitué de 7 lignes et 7 colonnes. La ligne du haut est laissée vide pour y afficher le moment venu le jeton à jouer.

La zone bleue est un rectangle commençant en $x=0$, $y = \text{LARGEUR_CASE}$ (la première ligne est laissée en noire), de largeur $= \text{NB_COLONNE} * \text{LARGEUR_CASE}$ et de hauteur $= (\text{NB_LIGNE}+1) * \text{LARGEUR_CASE}$ (nous avons une ligne de plus en haut du plateau).

La documentation de la fonction pygame permettant de tracer un rectangle donne :

```
pygame.draw.rect(surface, color, rect)
```

Draws a rectangle on the given surface.

surface (Surface) -- surface to draw on

color (Color or int or tuple(int, int, int, [int])) -- color to draw with, the alpha value is optional if using a tuple (RGB[A])

rect (Rect) -- rectangle to draw, position and dimensions

Un rectangle est défini par :

```
Rect(left, top, width, height)
```

Les ronds noirs indiquant les places libres

Si une case du plateau est libre, la case correspondante dans la matrice plateau contient la valeur 0.

Chaque emplacement libre est représenté par un cercle noir.

La fonction pygame permettant de tracer un cercle est :

```
pygame.draw.circle(surface, color, center, radius)
```

center : tuple donnant les coordonnées du centre du cercle

Les coordonnées du centre de chaque cercle se calcule par rapport à la case du plateau sur laquelle on doit le dessiner.

Si c est le numéro de colonne et r le numéro de la ligne, de la case où l'on doit dessiner le cercle alors les coordonnées du centre de ce cercle sont :

$X = \text{int}(c * \text{LARGEUR_CASE} + \text{LARGEUR_CASE} / 2)$: on ne prend que la partie entière de l'opération.

Si $c = 0$, $x = \text{LARGEUR_CASE} / 2$

$Y = \text{int}(r * \text{LARGEUR_CASE} + \text{LARGEUR_CASE} / 2 + \text{LARGEUR_CASE})$

On doit ajouter une case en hauteur, car on a une ligne vide en haut du plateau.

Le rayon d'un cercle est :

$\text{RAYON} = \text{int}(\text{LARGEUR_CASE} / 2) - 4$

Les cercles ne doivent pas toucher les bords des cases : on laisse 2 pixels vides autour des cercles.

Test

Reprendre le programme puissance4Console, et l'enregistrer en puissance4Graphique.

Juste au dessus des déclarations des fonctions ajouter :

```
#-----partie console-----
```

Au dessus de cette ligne, ajouter :

```
import pygame
```

#Les constantes

```
LARGEUR_CASE = 100
```

```
BLEU =(0,0,255)
```

```
ROUGE = (255,0,0)
```

```
JAUNE = (255,255,0)
```

```
NOIR = (0,0,0)
```

```
BLANC =(255,255,255)
```

```
RAYON=int(LARGEUR_CASE/2)-4
```

```
NB_LIGNE = 6
```

```
NB_COLONNE=7
```

```
#-----partie pygame-----
```

#dessiner le plateau de jeu

```
def dessiner_plateau(plateau):
```

```
    pygame.draw.rect(ecran,BLEU,(0,LARGEUR_CASE,NB_COLONNE *  
LARGEUR_CASE,(NB_LIGNE+1) * LARGEUR_CASE))
```

```
    for c in range (NB_COLONNE):
```

```
        for r in range(NB_LIGNE):
```

```
            pygame.draw.circle(ecran,NOIR,(int(c*LARGEUR_CASE+LARGEUR_CASE/2),  
int(r*LARGEUR_CASE+LARGEUR_CASE/2+LARGEUR_CASE)),RAYON)
```

```
            pygame.display.update()
```

```
#-----partie console-----
```

Dans la partie programme principale, avant la boucle de jeu :

```
#=====
pygame.init()

largeur = NB_COLONNE * LARGEUR_CASE
hauteur = (NB_LIGNE+1) * LARGEUR_CASE
taille = (largeur, hauteur)

ecran = pygame.display.set_mode(taille)

plateau = creer_plateau()

dessiner_plateau(plateau)

game_over = True

tour = 0

afficher_plateau(plateau)
```

Nous affectons la valeur True à **game_over** de façon à ne pas entrer dans la boucle de jeu.

A la toute fin du programme, en dehors de la boucle ajouter :

```
pygame.time.wait(3000)

pygame.quit()
```

La première instruction arrête le programme pendant 3 secondes, afin de nous permettre de voir si le plateau est correctement créé.

Affichage des jetons rouges ou jaunes une fois qu'un joueur a joué

Dans la fonction dessiner_plateau, nous avons affiché toutes les places des jetons en noir.

Nous allons compléter cette fonction en testant la matrice plateau. Si une case de cette matrice contient la valeur 1, nous devons afficher un jeton rouge dans la case correspondante du plateau. Si cette valeur est 2, nous devons afficher un jeton jaune.

Il suffit d'ajouter dans cette fonction :

```
for c in range(NB_COLONNE):
    for r in range(NB_LIGNE):
        if plateau[r][c] == 1:
            pygame.draw.circle(ecran, ROUGE, (int(c*LARGEUR_CASE+LARGEUR_CASE/2),
            hauteur- int(r*LARGEUR_CASE+LARGEUR_CASE/2)), RAYON)
```

```

        if plateau[r][c] == 2:

            pygame.draw.circle(ecran,JAUNE,(int(c*LARGEUR_CASE+LARGEUR_CASE/2),
hauteur- int(r*LARGEUR_CASE+LARGEUR_CASE/2)), RAYON)

    pygame.display.update()

```

Tester :

Nous allons simplement remplir la matrice tableau avec quelques valeurs et faire afficher le plateau de jeu pour vérifier que les jetons rouge ou jaune se placent correctement.

Le code avant la boucle de jeu :

```

#=====

pygame.init()

largeur = NB_COLONNE * LARGEUR_CASE
hauteur = (NB_LIGNE+1) * LARGEUR_CASE
taille = (largeur, hauteur)
ecran = pygame.display.set_mode(taille)
plateau = creer_plateau()

plateau[0][3]=1
plateau[1][3]=2
plateau[0][2]=1
plateau[2][3]=2

dessiner_plateau(plateau) #voir résultat ci-dessous

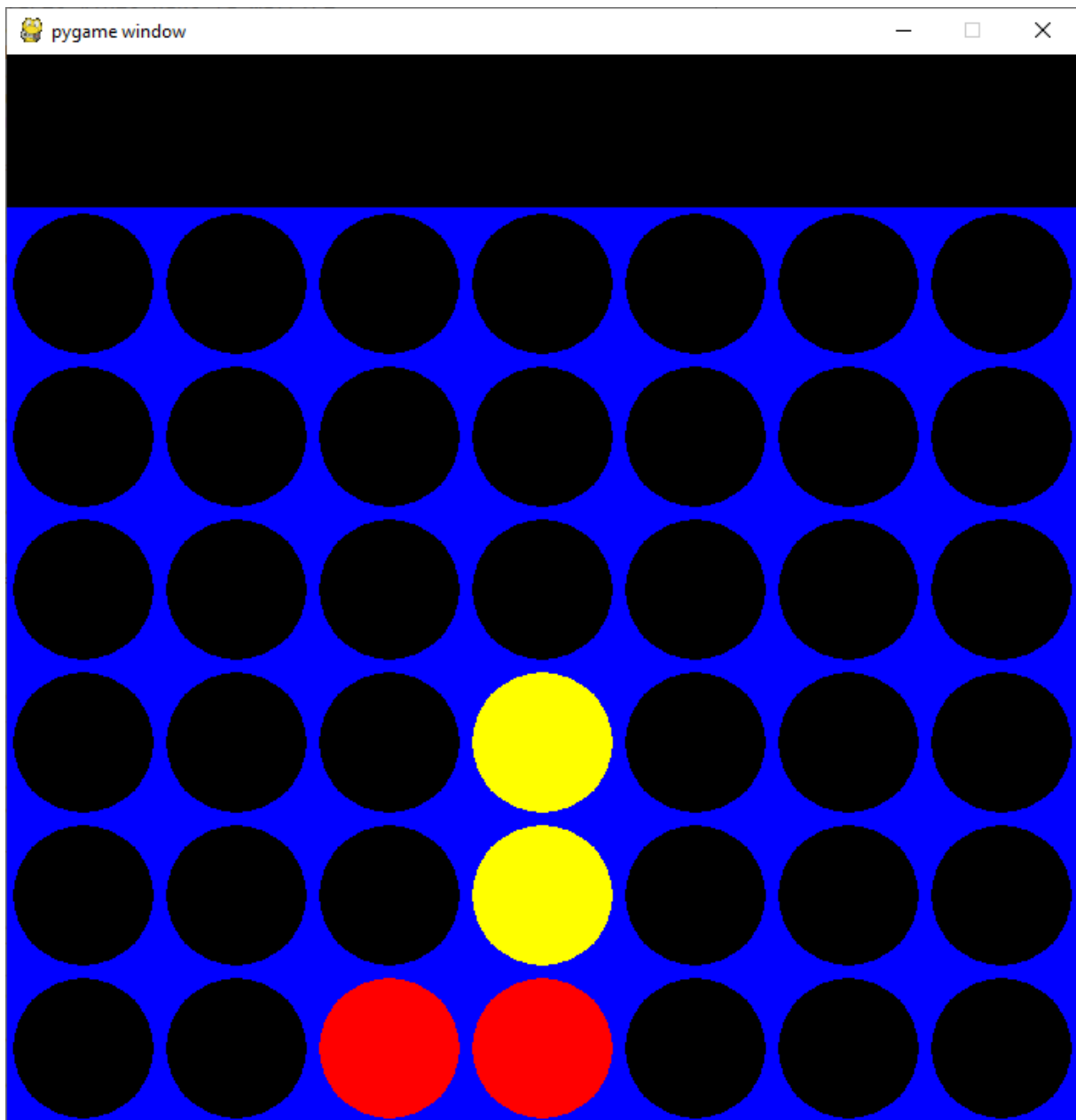
game_over = True
#game_over = False

tour = 0

afficher_plateau(plateau)

[0.  0.  0.  0.  0.  0.  0.  0.]
[0.  0.  0.  0.  0.  0.  0.  0.]
[0.  0.  0.  0.  0.  0.  0.  0.]
[0.  0.  0.  2.  0.  0.  0.  0.]
[0.  0.  0.  2.  0.  0.  0.  0.]
[0.  0.  1.  1.  0.  0.  0.  0.]

```



Prise en compte du choix d'un joueur

Le joueur doit pouvoir indiquer dans quelle colonne il veut placer son jeton. Le plus simple est de cliquer sur la colonne choisie.

Nous devons donc dans la boucle de jeu, capturer l'évènement `pygame.MOUSEBUTTONDOWN`

Dans le traitement de cet évènement, nous devons déterminer quelle est la colonne choisie.

L'évènement `MOUSEBUTTONDOWN` fournit les coordonnées de la souris au moment du clic. `event.pos[0]` donne l'abscisse et `event.pos[1]` l'ordonnée.

Le numéro de la colonne sur laquelle le joueur a cliqué, peut être calculée par :

#calcul du numéro de colonne correspondant à l'abscisse de la souris

```
colonne = int(math.floor(posx/LARGEUR_CASE))
```

La documentation de math.floor indique :

math.floor(x)

Return the floor of x as a float, the largest integer value less than or equal to x.

Il faut importer le module math au début du programme : **import math**

Nous modifions la partie principale du jeu de la façon suivante :

```
#=====
```

```
pygame.init()
```

```
largeur = NB_COLONNE * LARGEUR_CASE
```

```
hauteur = (NB_LIGNE+1) * LARGEUR_CASE
```

```
taille = (largeur, hauteur)
```

```
ecran = pygame.display.set_mode(taille)
```

```
plateau = creer_plateau()
```

```
dessiner_plateau(plateau)
```

```
game_over = False
```

```
tour = 0
```

```
while not game_over:
```

```
    for event in pygame.event.get():
```

```
        if event.type==pygame.QUIT:
```

```
            game_over = True
```

```
            pygame.quit()
```

```
    #le joueur a cliqué avec le bouton de la souris
```

```
    if event.type == pygame.MOUSEBUTTONDOWN:
```

```
        #joueur 1
```

```
        if tour==0:
```

```
            #position de la souris au moment du clic
```



```

posx = event.pos[0]

#choix du numéro de colonne par le joueur 1

#calcul du numéro de colonne correspondant à l'abscisse de la souris
colonne = int(math.floor(posx/LARGEUR_CASE))

if emplacement_valide(plateau,colonne):

    #trouver la ligne libre dans la colonne choisie

    ligne = trouver_Ligne_Vide(plateau,colonne)

    #mettre dans la matrice la valeur 1 dans la ligne, colonne

    lacher_jeton(plateau, ligne, colonne,1)

    #tester si le joueur 1 a gagné

    if coup_gagnant(plateau,1):

        print("Le joueur 1 a gagné!!! Bravo")

        game_over = True

    tour = 1

    dessiner_plateau(plateau)

#joueur 2

else:

    #position de la souris au moment du clic

    posx = event.pos[0]

    #choix du numéro de colonne par le joueur 2

    #calcul du numéro de colonne correspondant à l'abscisse de la souris
    colonne = int(math.floor(posx/LARGEUR_CASE))

    if emplacement_valide(plateau,colonne):

        #trouver la ligne libre dans la colonne choisie

        ligne = trouver_Ligne_Vide(plateau,colonne)

        #mettre dans la matrice la valeur 2 dans la ligne, colonne

        lacher_jeton(plateau, ligne, colonne,2)

        #tester si le joueur 2 a gagné

        if coup_gagnant(plateau,2):

```

```

        print("Le joueur 2 a gagné!!! Bravo")

        game_over = True

    tour = 0

    dessiner_plateau(plateau)

    if partie_nulle(plateau):
        print("Partie nulle")
        game_over = True

pygame.time.wait(3000)
pygame.quit()

```

Afficher sur la première ligne du plateau le jeton à jouer

A chaque tour de jeu, nous afficherons le jeton à jouer afin d'indiquer à qui est le tour.

Sachant que maintenant dans la première ligne du plateau, il va y avoir affiché le jeton à jouer, lorsque le joueur clique dans une colonne, nous allons commencer par effacer ce jeton, puisque celui-ci tombe dans la colonne.

Avant de passer la main à l'autre joueur nous devons afficher son jeton dans cette première ligne.

La prise en compte de l'évènement MOUSEBUTTONDOWN devient :

#le joueur a cliqué avec le bouton de la souris

```

if event.type == pygame.MOUSEBUTTONDOWN:

    #effacer la première ligne du plateau
    pygame.draw.rect(ecran,NOIR,(0,0,largeur,LARGEUR_CASE))

    #joueur 1
    if tour==0:

        #position de la souris au moment du clic
        posx = event.pos[0]

        #choix du numéro de colonne par le joueur 1

        #calcul du numéro de colonne correspondant à l'abscisse de la souris

```

```

colonne = int(math.floor(posx/LARGEUR_CASE))
if emplacement_valide(plateau,colonne):
    #trouver la ligne libre dans la colonne choisie
    ligne = trouver_Ligne_Vide(plateau,colonne)
    #mettre dans la matrice la valeur 1 dans la ligne, colonne
    lacher_jeton(plateau, ligne, colonne,1)
    #tester si le joueur 1 a gagné
    if coup_gagnant(plateau,1):
        print("Le joueur 1 a gagné!!! Bravo")
        game_over = True
    tour = 1
    #afficher le jeton du joueur 2
    pygame.draw.circle(ecran, JAUNE, (int(LARGEUR_CASE/2), int(LARGEUR_CASE/2)),
RAYON)
    dessiner_plateau(plateau)
#joueur 2
else:
    #effacer la première ligne du plateau
    pygame.draw.rect(ecran,NOIR,(0,0,largeur,LARGEUR_CASE))
    #position de la souris au moment du clic
    posx = event.pos[0]
    #choix du numéro de colonne par le joueur 2
    #calcul du numéro de colonne correspondant à l'abscisse de la souris
    colonne = int(math.floor(posx/LARGEUR_CASE))
    if emplacement_valide(plateau,colonne):
        #trouver la ligne libre dans la colonne choisie
        ligne = trouver_Ligne_Vide(plateau,colonne)
        #mettre dans la matrice la valeur 2 dans la ligne, colonne
        lacher_jeton(plateau, ligne, colonne,2)

```

```

#tester si le joueur 2 a gagné

if coup_gagnant(plateau,2):

    print("Le joueur 2 a gagné!!! Bravo")

    game_over = True

tour = 0

#afficher le jeton du joueur 1

pygame.draw.circle(ecran,ROUGE, (int(LARGEUR_CASE/2), int(LARGEUR_CASE/2)),
RAYON)

dessiner_plateau(plateau)

```

Déplacer le jeton à jouer avec la souris

Le jeton affiché sur la première ligne du plateau, doit suivre les mouvements de la souris (juste en ligne droite), jusqu'à ce que le joueur clique sur une colonne.

Nous devons prendre en compte l'évènement pygame.MOUSEMOTION

Cet évènement s'accompagne des coordonnées de la souris.

```

#on déplace le jeton avec la souris

if event.type == pygame.MOUSEMOTION:

    #effacer la première ligne du plateau

    pygame.draw.rect(ecran,NOIR,(0,0,largeur,LARGEUR_CASE))

    #position de la souris

    posx = event.pos[0]

    #si joueur 1 on affiche un jeton rouge, à la position x de la souris

    if tour==0:

        pygame.draw.circle(ecran,ROUGE,(posx,int(LARGEUR_CASE/2)),RAYON)

    #si joueur 2 on affiche un jeton jaune, à la position x de la souris

    else:

        pygame.draw.circle(ecran,JAUNE,(posx,int(LARGEUR_CASE/2)),RAYON)

    #on met l'écran à jour

    pygame.display.update()

```

Afficher qui a gagné

Lorsque l'un des joueurs a gagné, nous afficherons dans la première ligne le texte : Joueur x a gagné.

Nous devons ajouter **avant la boucle** la définition d'une police :

```
myfont = pygame.font.SysFont("comicansms", 40)
```

et dans la boucle, remplacer les lignes `print(" Le joueur x a gagné ")` ou `print(" Partie nulle ")` par :

#tester si le joueur 1 a gagné

```
if coup_gagnant(plateau,1):
```

```
    texte_image = myfont.render("Le joueur 1 a gagné", True, ROUGE)
    ecran.blit(texte_image, [50, 10])
    game_over = True
```

Idem pour le joueur 2 :

#tester si le joueur 2 a gagné

```
if coup_gagnant(plateau,2):
```

```
    texte_image = myfont.render("Le joueur 2 a gagné", True, JAUNE)
    ecran.blit(texte_image, [50, 10])
    game_over = True
```

Et si une partie est nulle :

```
if partie_nulle(plateau):
```

```
    texte_image = myfont.render("Partie Nulle", True, BLANC)
    ecran.blit(texte_image, [50, 10])
    game_over = True
```

