



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat d'Informàtica de Barcelona



TRES EN RATLLA

Practica Programació amb Pygame Programació i Algorísmia 1

Inteligencia Artificial

Autores

Artur Aubach Altes
César Mejía Rota

Grup 11

Professors Jordi Delgado i José Luis Balcázar
Quadrimestre Tardor 2022/2023

CONTINGUTS

1. INTRODUCCIÓN	2
1.1. PYTHON	2
1.2. PYGAME	2
2. NORMES DEL NOSTRE JOC I ABAST	4
3. QUE CONTÉ EL NOSTRE TREBALL	5
3. 1. IMATGES	5
3. 2. FUNCIONS IMPORTANT	5
3. 1. 1. <i>CONDICIÓ DE VICTORIA</i>	5
3. 1. 2. <i>BOTONS</i>	6
3. 1. 3. <i>MANEIG DE PÀGINES</i>	7
3.1.3.1 MAIN MENU	7
3.1.3.2 PLAY	8
3.1.3.3 GAME	11
4. DIFICULTATS	16
5. REPARTIMENT DE RESPONSABILITATS	17
6. VALORACIÓ DEL APRENENTATGE	18
7. REFERENCIAS	19
8. CÓDIGO	20

1. INTRODUCCIÓN

L'objectiu d'aquest treball ha sigut el desenvolupament d'un tres en ratlla mitjançant el llenguatge de programació python i la seva llibreria pygame.

1.1. PYTHON

Python és un llenguatge de programació interpretat àmpliament utilitzat en el desenvolupament de software, aplicacions web, ciència de dades i Machine Learning. És eficient i es pot executar en moltes plataformes diferents. Consta d'un codi molt llegible i fàcil d'entendre, fet que simplifica el seu aprenentatge: és per això que és un molt bon punt de partida per aprendre programació.

En aquest treball utilitzarem Python per a desenvolupar un joc de 3 en ratlla, i per a fer això ens ajudarem d'una de les llibreries de Python: pygame.

Una llibreria, en programació, és un conjunt d'arxius de codi usats per al desenvolupament de software. És a dir, són eines que faciliten el treball i el desenvolupament proporcionant funcions que ja han sigut resoltes per altres programadors. Per exemple, la llibreria "math" inclou funcions com 'sqrt' (square root, arrel quadrada) que no existeixen com a funcions bàsiques i que faciliten, en aquest cas, càlculs matemàtics més complexos que els que ofereix la versió 'base' (sense la llibreria) de Python.

1.2. PYGAME

Pygame, com s'ha mencionat abans, és una llibreria de python que té com a objectiu facilitar la implementació de jocs en dues dimensions. Aquesta llibreria està basada en 'SDL', que és una llibreria que proporciona accés de baix nivell a certs components del sistema on s'executa el programa, com bé podrien ser teclat, ratolí, àudio i sistema gràfic.

A pygame, l'estructura d'un videojoc es basa en un bucle principal, on succeeix tot el codi del joc (entrarem en detalls pròximament), amb funcions que inicien el joc i s'encarreguen de detectar que està fent l'usuari, és a dir, quins esdeveniments passen i que fer segons el que passi. Conjuntament amb el bucle principal trobem una funció que finalitza el joc.

El joc constarà d'un programa principal (main) i un o més arxius que contenen funcions implementades per a un millor funcionament del programa principal. Aquests arxius

contenidors són coneguts com a 'drivers' i es pot accedir al seu contingut important-los com si fossin una llibreria al principi de l'arxiu principal del programa ('main').

2. NORMES DEL NOSTRE JOC I ABAST

Les normes del joc són les pròpies d'un 3 en ratlla normal: el primer jugador que aconsegueix col·locar 3 peces pròpies formant una línia, sigui de manera horitzontal, vertical o diagonal, guanya.

Cada jugador té el mateix nombre de peces, i es juga de manera que col·loquen peces al taulell de manera alternada fins que tots dos es queden sense peces. En aquest moment, cada jugador selecciona la fitxa que vol moure i aquesta es mourà a la posició no ocupada.

En un exemple on juguem el clàssic 3 en ratlla (3 en ratlla per defecte), tots 2 jugadors tenen 4 peces i les posen al taulell de manera intercalada fins que queda una casella buida. Ara, de manera alternada un altre cop, els jugadors escullen la peça que volen moure a la posició sense ocupar fins que un dels dos obté 3 en ratlla.

L'abast podria ser infinit, però pel que fa a jugabilitat seria poc pràctic i visualment no es veuria pràcticament res, és per això que hem decidit limitar la grandària del tauler seleccionable.

Es poden escollir a l'hora de jugar la dimensió del tauler i la quantitat de fitxes en línia que calen per guanyar, és a dir, la condició de victòria.

Pel que fa a limitacions, l'única restricció que té el joc és que no es pot seleccionar un mode de joc de grandària superior al del mateix taulell, ja que si no seria impossible terminar la partida.

3. QUE CONTÉ EL NOSTRE TREBALL

La nostra entrega consta de tres carpetes diferents. La carpeta 'PROJECTE' conté tots els arxius amb els codis i les diferents funcions implementades i la carpeta 'IMATGES' emmagatzema totes les imatges utilitzades per a la interfície. Per tant, hem d'executar el nostre codi principal des del directori que conté aquestes dues carpetes. La comanda per executar el programa és:

```
python .\PROJECTE\tauler_abstracte.py
```

El codi del programa esta escrit en el arxiu 'tauler abstracte (dins la carpeta PROJECTE) ;, des del qual s'accedeix a la resta de 'drivers'.

3. 1. IMATGES

Amb la funció 'pygame.image.load' carreguem els arxius .png (imatges). Cal assegurar-se de que al executar el codi la direcció de les imatges, com ara pot ser FONS_TITOL.png, estiguin ben senyalitzades ja que poden variar la orientació i les dimensions depenent de l'entorn on es crida l'imatge.

3. 2. FUNCIONS IMPORTANT

3. 1. 1. CONDICIÓN DE VICTORIA

Aquesta funció detecta quan hi ha X elements en línia. Això ho fa detectant quins valors tenen els elements dins la llista de llistes. Primer comprova les files, després les columnes i, finalment, les diagonals.

```
def check_winner(matrix, X):  
  
    rows = len(matrix)  
    cols = rows  
  
    # Check rows  
    for row in range(rows):  
        for col in range(cols - X + 1):  
            if matrix[row][col] != 0:  
                player = matrix[row][col]  
                win = True  
                for i in range(1, X):  
                    if matrix[row][col + i] != player:  
                        win = False  
                        break  
                if win:  
                    return player  
  
    # Check columns  
    for col in range(cols):
```

```

    for row in range(rows - X + 1):
        if matrix[row][col] != 0:
            player = matrix[row][col]
            win = True
            for i in range(1, X):
                if matrix[row + i][col] != player:
                    win = False
                    break
            if win:
                return player

# Check diagonals
for row in range(rows - X + 1):
    for col in range(cols - X + 1):
        if matrix[row][col] != 0:
            player = matrix[row][col]
            win = True
            for i in range(1, X):
                if matrix[row + i][col + i] != player:
                    win = False
                    break
            if win:
                return player

for row in range(X - 1, rows):
    for col in range(cols - X + 1):
        if matrix[row][col] != 0:
            player = matrix[row][col]
            win = True
            for i in range(1, X):
                if matrix[row - i][col + i] != player:
                    win = False
                    break
            if win:
                return player
# If no player wins
return False

```

3. 1. 2. BOTONS

Aquesta funció defineix la classe ‘Button’:

```

class Button():
    def __init__(self, image, pos, text_input, font, base_color, hovering_color):
        self.image = image
        self.x_pos = pos[0]
        self.y_pos = pos[1]
        self.font = font
        self.base_color, self.hovering_color = base_color, hovering_color
        self.text_input = text_input
        self.text = self.font.render(self.text_input, True, self.base_color)
        if self.image is None:
            self.image = self.text
        self.rect = self.image.get_rect(center=(self.x_pos, self.y_pos))
        self.text_rect = self.text.get_rect(center=(self.x_pos, self.y_pos))

    def update(self, screen):
        if self.image is not None:
            screen.blit(self.image, self.rect)
            screen.blit(self.text, self.text_rect)

```

```

def checkForInput(self, position):
    if position[0] in range(self.rect.left, self.rect.right) and position[1] in
range(self.rect.top, self.rect.bottom):
        return True
    return False

def changeColor(self, position):
    if position[0] in range(self.rect.left, self.rect.right) and position[1] in
range(self.rect.top, self.rect.bottom):
        self.text = self.font.render(self.text_input, True,
self.hovering_color)
    else:
        self.text = self.font.render(self.text_input, True, self.base_color)

```

3. 1. 3. MARCAR A TABLERO

Aquestes funcions marquen a la matriu un 1 o un 2 depenent de quin ‘player’ estigui jugant.

```

# Marcar tablero
def Marcar_On_QUI_1(row, col, player, matriu):
    matriu[row][col] = player

def abiable_sqare_1(row, col, matriu):
    if matriu[row][col] == 0:
        return True
    else:
        return False

def Marcar_On_QUI_2(row, col, modo_de_taula, matriu):
    for i in range(modo_de_taula):
        for j in range(modo_de_taula):
            if matriu[i][j] == 0:
                matriu[i][j], matriu[row][col] = matriu[row][col], matriu[i][j]
    return matriu

def abiable_sqare_2(row, col, player_torn, matriu):
    if matriu[row][col] == 1 and player_torn == 1:
        return True
    elif matriu[row][col] == 2 and player_torn == 2:
        return True
    else:
        return False

```

3. 1. 3. MANEIG DE PÀGINES

Tenim 3 pàgines, la del ‘Main menu’, la del ‘play’ i la del ‘game’, representades totes amb les seves respectives funcions:

3.1.3.1 MAIN MENU

Aquesta funció descriu la pantalla que podem veure un cop obrim el joc. Conté el boto del play, que ens condueix a la següent pàgina, la pàgina del ‘Play’. Com be esta descrit en el propi codi, el primer que es fa es establir el nom de la finestra (‘pygame.display.set_caption’). Tot seguit detecta la posició del ratolí amb la funció ‘pygame.mouse.get_pos’. Un cop fet

això, col·loca els fons de pantalla i el títol (arxius .png) i assigna els botons del play i el de sortir del joc. Estableix al final que passa si pressiones el boto play (canvia de pantalla a ‘play’) o si has pressionat el de sortit (pygame.QUIT).

```
def main_menu():
    # Nom del "MAIN MENU" (nom del joc)
    pygame.display.set_caption("RES EN RATLLA")

    #BUCLE GENERAL
    while True:

        #DETECTAR OPSICCIO DE RATOLI
        MENU_MOUSE_POS = pygame.mouse.get_pos()

        # "IMATGES" PANTALLA
        ## Fons de pantalla GENERAL
        screen.blit(fons, (0, 0))
        ## Títol
        screen.blit(fons_titol, (0, 0))

        #BOTONS
        ##PLAY
        PLAY = pygame.image.load("IMATGES/PLAY.png")
        quadre_play = pygame.transform.scale(PLAY, (250, 100))
        MENU_PLAY_BUTTON = Button(image=quadre_play, pos=(
            400, 325), text_input="PLAY", font=main_font, base_color=(255, 255,
            255), hovering_color=(128, 0, 128))

        for button in [MENU_PLAY_BUTTON]:
            button.changeColor(MENU_MOUSE_POS)
            button.update(screen)
        # EVENTOS QUE PODEN PASSAR DINS DE MAIN MENU
        ## Registra tots els "eventos" que passa a la pantalla
        for event in pygame.event.get():
            ### Sortir al clicar la X
            if event.type == pygame.QUIT:
                pygame.quit()
                sys.exit()
            ### AL CLICAR
            if event.type == pygame.MOUSEBUTTONDOWN:
                # Si has clicat a play
                if MENU_PLAY_BUTTON.checkForInput(MENU_MOUSE_POS):
                    # Canviar de pantalla a play
                    play()

        # FPS
        temps.tick(FPS)

        # Actualitzar (repetir comando de adalt)
        pygame.display.update()
```

3.1.3.2 PLAY

Com podem veure, amb aquesta funció definim el mode de joc i la condició de victòria, conjuntament amb un botó de play que ens envia a l'últim menú, el menú on pròpiament es juga. El primer que fem un altre cop es posar-li nom a la finestra (pygame.display.set_caption) i després iniciem les variables globals per al mode de taula, mode de joc, la matriu, el torn i les respectives victòries de cada jugador. El següent a fer, ja dins el bucle general, es detectar la posició del ratolí i establir el fons de pantalla. Tot seguit s'estableixen les strings que surten en la pròpia pantalla (tamany de la taula i mode de joc), juntament amb les funcions amb variables que canvien segons el tamany i el mode.

S'estableixen els botons amb les strings '>' i '<' que pujaran i baixaran respectivament el valor de les variables de la taula i el mode de joc. L'últim a fer es la funció que detecta els esdeveniments que passen a la finestra, es a dir, en aquest cas, on cliques amb el ratolí. Això fa que si has pressionat '<' o '>' el numero varia, no nomes internament sinó a la vista de l'usuari també, i que si pressiones 'play' o 'exit' t'envii a la següent pantalla o surti respectivament.

```
def play():
    # Nom de la finestra "PLAY"
    pygame.display.set_caption("Play")

    #BUCLE GENERAL
    while True:
        global MODO_DE_TAUULA
        global MODO_DE_JOC
        global MATRIU
        global TORN
        global R_WIN_X
        global R_WIN_Y
        global PLAYER_TORN

        #DETECTAR OPSICCIO DE RATOLI
        PLAY_MOUSE_POS = pygame.mouse.get_pos()

        # "IMATGES" PANTALLA
        ## Fons de pantalla GENERAL
        screen.blit(fons, (0, 0))

        ## Temany taula
        TEXT_TAUULA = main_font3.render("Tauler de joc", True, (0, 0, 0))
        screen.blit(TEXT_TAUULA, (100, 100))

        ### NUM DEL Temany taula
        mod_taula = main_font3.render(str(MODO_DE_TAUULA), True, (0, 0, 0))
        screen.blit(mod_taula, (195, 165))

        ## Modo de joc en pantalla
        TEXT_MODAL = main_font4.render("Condicció de victòria", True, (0, 0, 0))
        screen.blit(TEXT_MODAL, (450, 110))

        ### NUM DEL Modo de joc
        TEXT_MODAL = main_font3.render(str(MODO_DE_JOC), True, (0, 0, 0))
        screen.blit(TEXT_MODAL, (570, 165))

        #BOTONS
        ##PLAY
        MENU_PLAY2_BUTTON = pygame.image.load("IMATGES/PLAY.png")
        quadre_play2 = pygame.transform.scale(MENU_PLAY2_BUTTON, (375, 100))

        MENU_PLAY2_BUTTON = Button(image=quadre_play2, pos=(
            400, 350), text_input="COMENÇAR", font=main_font, base_color=(255, 255,
255), hovering_color=(128, 0, 128))

        ##REDUIR TEMANY TAUULA
        TAUULA_MENYS = Button(image=None, pos=(150, 185),
            text_input="<", font=main_font, base_color="White",
            hovering_color=(128, 0, 128))
```

```

    ##INCREMENTAR TEMANY TAULA
    TAULA_MES = Button(image=None, pos=(260, 185),
                        text_input=">", font=main_font, base_color="White",
                        hovering_color=(128, 0, 128))

    ##REDUIR MODO DE JOC
    MODO_MENYS = Button(image=None, pos=(525, 185),
                        text_input="<", font=main_font, base_color="White",
                        hovering_color=(128, 0, 128))

    ##INCREMENTAR MODO DE JOC
    MODO_MES = Button(image=None, pos=(635, 185),
                        text_input=">", font=main_font, base_color="White",
                        hovering_color=(128, 0, 128))

    ##GEENERAL
    for button in [MENU_PLAY2_BUTTON, TAULA_MENYS, TAULA_MES, MODO_MENYS,
MODO_MES]:
        button.changeColor(PLAY_MOUSE_POS)
        button.update(screen)

    # EVENTOS QUE PODEN PASSAR DINS DE GAME
    ## Registra tots els "eventos" que passa a la pantalla
    for event in pygame.event.get():
        ###Sortir al clicar la X
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()
        ### AL CLCAR

        if event.type == pygame.MOUSEBUTTONDOWN:
            # Si has clicat a play
            if MENU_PLAY2_BUTTON.checkForInput(PLAY_MOUSE_POS):
                if MODO_DE_JOC > MODO_DE_TAUOLA:
                    print("Error: MODO DE JOC > TAULER")
                # Canviar de pantalla a play
            else:
                #GENERAR MATRIU TABLERO
                MATRIU = [[0] * MODO_DE_TAUOLA for i in
range(MODO_DE_TAUOLA)]

                #CONFIGURACIO DEL JOC
                TORN = 1
                R_WIN_X = 0
                R_WIN_Y = 0
                PLAYER_TORN = 1
                game()

            #RESTRICCIÓ DE CONDICIÓN DE VICTÒRIA - TAMANY TABLERO
            if TAULA_MENYS.checkForInput(PLAY_MOUSE_POS):
                if MODO_DE_TAUOLA > 3:
                    MODO_DE_TAUOLA -= 1

            if TAULA_MES.checkForInput(PLAY_MOUSE_POS):
                if 12 > MODO_DE_TAUOLA:
                    MODO_DE_TAUOLA += 1

            if MODO_MENYS.checkForInput(PLAY_MOUSE_POS):
                if MODO_DE_JOC > 2:
                    MODO_DE_JOC -= 1

            if MODO_MES.checkForInput(PLAY_MOUSE_POS):
                if 12 > MODO_DE_JOC:
                    MODO_DE_JOC += 1

    #FPS
    temps.tick(FPS)

    # Actualitzar (repetir comando de adalt)

```

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

3.1.3.3 GAME

En aquest fragment de codi veiem com un altre cop el primer que fem al iniciar la funció es establir el nom de la finestra, i tot seguit iniciem el bucle general. Veiem la funció de detectar la posició del ratolí un altre cop i la carrega de les imatges en pantalla, en aquest cas el fons. Trobem les posicions inicials, juntament amb el canvi de color i l'altura i l'amplada en funció d'una variable de nom X. Per últim, la funció que dibuixa el rectangle que serà la quadrícula:

```
def game():
    # Nom del "MAIN MENU" (nom del joc)
    pygame.display.set_caption("GAME")

    #BUCLE GENERAL
    while True:
        global MODO_DE_TAUOLA
        global MODO_DE_JOC
        global MATRIU
        global PLAYER_TORN
        global TORN
        global R_WIN_X
        global R_WIN_Y

        #DETECTAR OPSICCIO DE RATOLI
        GAME_MOUSE_POS = pygame.mouse.get_pos()

        # "IMATGES" PANTALLA
        ## Fons de pantalla GENERAL
        screen.blit(fons, (0, 0))

        # #RACHA DE VICTORIES
        # X
        rect_color_X = (0, 255, 0)

        pos_x_inicial_X = (SCREEN_X/14)
        pos_y_inicial_X = (SCREEN_Y/7)*2

        allargada_X = (SCREEN_X/14)*2
        altura_X = (SCREEN_Y/7)*3
        '''
        pygame.draw.rect(screen, rect_color_X, (pos_x_inicial_X,
                                                pos_y_inicial_X, allargada_X, altura_X))
        '''
```

Veiem aquí les funcions de les icones, tant de X com de O (les dimensions son iguals), tenim la funció del tauler i les funcions de les ratxes de victories:

```
### ICONO X
```

```

X_RACHA_IMATGE = pygame.image.load("IMATGES/X.png")
x_racha_imatge = pygame.transform.scale(X_RACHA_IMATGE, (allargada_X,
allargada_X))
screen.blit(x_racha_imatge, (pos_x_inicial_X, pos_y_inicial_X))

### ICONO NUM X
X_RACHA_ESCRIT = main_font2.render("VICTORIES: " + str(R_WIN_X), True, (0,
0, 0))
screen.blit(X_RACHA_ESCRIT, (pos_x_inicial_X, 275))

## O
rect_color_O = (0, 255, 0)

pos_x_inicial_O = SCREEN_X-allargada_X-pos_x_inicial_X
pos_y_inicial_O = pos_y_inicial_X

allargada_O = allargada_X
altura_O = altura_X
'''
pygame.draw.rect(screen, rect_color_O, (pos_x_inicial_O,
pos_y_inicial_O, allargada_O, altura_O))
'''

### ICONO O
O_RACHA_IMATGE = pygame.image.load("IMATGES/O.png")
o_racha_imatge = pygame.transform.scale(O_RACHA_IMATGE, (allargada_O,
allargada_O))
screen.blit(o_racha_imatge, (pos_x_inicial_O, pos_y_inicial_O))

### ICONO NUM O
X_RACHA_ESCRIT = main_font2.render("VICTORIES: " + str(R_WIN_Y), True, (0,
0, 0))
screen.blit(X_RACHA_ESCRIT, (pos_x_inicial_O, 275))

## TABLERO
rect_color_TABLERO = (150, 150, 150)

allargada_TABLERO = 350
altura_TABLERO = 350

pos_x_inicial_TABLERO = (SCREEN_X-allargada_TABLERO)/2
pos_y_inicial_TABLERO = (SCREEN_Y-altura_TABLERO)/2

```

Com podem veure, aquesta funció es la més extensa de totes i conte tot el necessari per jugar, com el dibuix del tauler amb les línies, la redimensió de les fitxes i la col·locació de les mateixes en el tauler. Trobem també la funció que dibuixa X i O en el tauler, les funcions de la primera i la segona fase de l partida, detecció del final de la partida, la notificació de que un jugador ha guanyat i la opció de jugar de nou. Veiem també el botó de tornar enrere i el de 'new game' quan acaba la partida:

```

'''
pygame.draw.rect(screen, rect_color_TABLERO, (pos_x_inicial_TABLERO,
pos_y_inicial_TABLERO, allargada_TABLERO, altura_TABLERO))
'''

#QUADRICULA
line_color = (0, 0, 0)
line_grux = 2

##LINIES VERTICALS
VarX_TABLERO = allargada_TABLERO/MODO_DE_TAULA
pos_y_inicial_L_V = pos_y_inicial_TABLERO

```

```

pos_y_final_L_V = pos_y_inicial_TABLERO+altura_TABLERO

for i in range(MODO_DE_TAULA-1):
    pos_x_inicial_L_MODIF = pos_x_inicial_TABLERO+VarX_TABLERO*(i+1)
    pygame.draw.line(screen, line_color, (pos_x_inicial_L_MODIF,
                                           pos_y_inicial_L_V),
(pos_x_inicial_L_MODIF, pos_y_final_L_V), line_grux)

##LINIES HORITZONTAL
pos_x_inicial_L_H = pos_x_inicial_TABLERO
VarY_TABLERO = VarX_TABLERO

pos_x_final_L_H = pos_x_inicial_TABLERO+allargada_TABLERO

for i in range(MODO_DE_TAULA-1):
    pos_y_inicial_L_MODIF = pos_y_inicial_TABLERO+VarY_TABLERO*(i+1)
    pygame.draw.line(screen, line_color, (pos_x_inicial_L_H,
                                           pos_y_inicial_L_MODIF),
(pos_x_final_L_H, pos_y_inicial_L_MODIF), line_grux)

#DIBUIXAR X i O A TABLERO
n = int(VarX_TABLERO-line_grux)
X_redimensionada = pygame.transform.scale(X, (n, n))
O_redimensionada = pygame.transform.scale(O, (n, n))
mitad_del_gruix = line_grux/2
Num_fila = 0

for i in MATRIU:
    CORDENADES_X_per_pecas_MOD = pos_x_inicial_TABLERO+VarX_TABLERO*Num_fila
    Num_column = 0

    for j in i:
        CORDENADES_y_per_pecas_MOD =
pos_y_inicial_TABLERO+VarY_TABLERO*Num_column
        if j == 1:
            screen.blit(
                X_redimensionada,
                (CORDENADES_X_per_pecas_MOD+mitad_del_gruix,
                CORDENADES_y_per_pecas_MOD+mitad_del_gruix))
        elif j == 2:
            screen.blit(
                O_redimensionada,
                (CORDENADES_X_per_pecas_MOD+mitad_del_gruix,
                CORDENADES_y_per_pecas_MOD+mitad_del_gruix))
            Num_column += 1

    Num_fila += 1

'''
#RAYAA !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
line_color=(0,0,0)

pos_x_inicial_L = 400
pos_y_inicial_L = 0

pos_x_final_L = 400
pos_y_final_L = 450

line_grux = 15

pygame.draw.line(screen, line_color, (pos_x_inicial_L,
                                     pos_y_inicial_L), (pos_x_final_L, pos_y_final_L),
line_grux)

'''

```

```

#ICONO EXIT
EXIT = pygame.image.load("IMATGES/EXIT.png")
exit_t = pygame.transform.scale(EXIT, (40, 40))
EXIT_BUTTON = Button(image=exit_t, pos=(
    750, 50), text_input="", font=main_font2, base_color=(0, 0, 0),
    hovering_color=(50, 50, 50))

# ICONO RESTART GAME
RESTART_GAME = pygame.image.load("IMATGES/RESTART_GAME.png")
restart_game = pygame.transform.scale(RESTART_GAME, (190, 60))
RESTART_GAME_BUTTON = Button(image=restart_game, pos=(675, 400),
    text_input="NEW GAME", font=main_font2, base_color=(0, 0, 0), hovering_color=(255,
    255, 255))

##GEENERAL
for button in [EXIT_BUTTON]:
    button.changeColor(GAME_MOUSE_POS)
    button.update(screen)

#IMATGE WIN O
WIN_O = pygame.image.load("IMATGES/WIN_O.png")
wins_o = pygame.transform.scale(WIN_O, (400, 200))

# IMATGE WIN X
WIN_X = pygame.image.load("IMATGES/WIN_X.png")
wins_x = pygame.transform.scale(WIN_X, (400, 200))

#MOSTRAR PER PANTALLA QUI GUANYA
if check_winner(MATRIU, MODO_DE_JOC) in (1, 2):
    RESTART_GAME_BUTTON.changeColor(GAME_MOUSE_POS)
    RESTART_GAME_BUTTON.update(screen)

    if check_winner(MATRIU, MODO_DE_JOC) == 1:
        screen.blit(wins_x, (200, 75))

    else:
        screen.blit(wins_o, (200, 75))

# EVENTOS QUE PODEN PASSAR DINS DE GAME
# Registra tots els "eventos" que passa a la pantalla
for event in pygame.event.get():
    ### Sortir al clicar la X
    if event.type == pygame.QUIT:
        pygame.quit()
        sys.exit()
    ### AL CLCAR
    if event.type == pygame.MOUSEBUTTONDOWN:
        if EXIT_BUTTON.checkForInput(GAME_MOUSE_POS):
            #Tmany taula predeterminat
            MODO_DE_TAUOLA = 3
            #Modo de joc predeterminat
            MODO_DE_JOC = 3
            #Jugador que començara
            play()

        #TORNAR A JUGAR
        if check_winner(MATRIU, MODO_DE_JOC) == 1:
            if RESTART_GAME_BUTTON.checkForInput(GAME_MOUSE_POS):
                PLAYER_TORN = 2
                R_WIN_X += 1
                TORN = 1
                MATRIU = [[0] * MODO_DE_TAUOLA for i in
range(MODO_DE_TAUOLA)]

            elif check_winner(MATRIU, MODO_DE_JOC) == 2:
                PLAYER_TORN = 1
                R_WIN_Y += 1

```

```

TORN = 1
MATRIU = [[0] * MODO_DE_TAULA for i in
range(MODO_DE_TAULA)]

#MARCAR TABLERO
ratX = event.pos[0] #cordenada X al clicar
ratY = event.pos[1] #cordenades Y al clicar

if (pos_x_inicial_TABLERO < ratX <
pos_x_inicial_TABLERO+allargada_TABLERO) and (pos_y_inicial_TABLERO < ratY <
pos_y_inicial_TABLERO+altura_TABLERO):

    #Quina fila ha clicat
    valor_clic_files = int((-pos_x_inicial_TABLERO+ratX)//
VarX_TABLERO)

    #Quina columna ha clicat
    valor_clic_columnes = int((-pos_y_inicial_TABLERO+ratY)//
VarY_TABLERO)

    #PRIMERA FASE JOC
    if abiable_sqare_1(valor_clic_files,valor_clic_columnes,MATRIU)
and TORN<(MODO_DE_TAULA*MODO_DE_TAULA) and (check_winner(MATRIU,
MODO_DE_JOC)==False):
        if PLAYER_TORN == 1:
            PLAYER_TORN = 2
            TORN +=1
            Marcar_On_Qui_1(valor_clic_files,valor_clic_columnes,
1, MATRIU)

        elif PLAYER_TORN == 2:
            PLAYER_TORN = 1
            TORN +=1
            Marcar_On_Qui_1(valor_clic_files,valor_clic_columnes,
2, MATRIU)

    #SEGONA FASE JOC
    elif abiable_sqare_2(valor_clic_files, valor_clic_columnes,
PLAYER_TORN, MATRIU) and TORN >= (MODO_DE_TAULA*MODO_DE_TAULA) and
(check_winner(MATRIU, MODO_DE_JOC)==False):
        if PLAYER_TORN == 1:
            PLAYER_TORN = 2
            TORN +=1
            Marcar_On_Qui_2(valor_clic_files,valor_clic_columnes,
MODO_DE_TAULA, MATRIU)

        elif PLAYER_TORN == 2:
            PLAYER_TORN = 1
            TORN +=1
            Marcar_On_Qui_2(valor_clic_files,valor_clic_columnes,
MODO_DE_TAULA, MATRIU)

# FPS
temps.tick(FPS)

# Actualitzar (repetir comando de adalt)
pygame.display.update()

#-----
#INICIAR AMB "MAIN MENU"

main_menu()

```


4. DIFICULTATS

Una de les majors dificultats a les quals ens hem hagut d'enfrontar ha sigut programar i configurar el menú, ja que hem hagut de dimensionar i col·locar manualment tot el que apareix en els diferents menús, cosa que ens ha consumit molt de temps, sumat al fet de col·locar i configurar tots els botons que apareixen en pantalla en cada menú, sent la configuració dels botons tant o més complicada que la mateixa configuració del menú. Cal mencionar també la dificultat que ens ha generat el fet de crear un programa capaç de variar en funció de la grandària del taulell i de la condició de victòria, i ja no només això, sinó que fos capaç de canviar en funció a aquests dos paràmetres de manera independent, tot limitant un paràmetre en funció de l'altre. Aquesta dificultat ve possiblement lligada al fet que vam començar el treball des de la part del pygame, ja que era la part en la qual més dubtes teníem i volíem entendre com més aviat millor l'entorn en el qual hauríem de fer el treball. Això va provocar que no tinguéssim una versió jugable del joc en Python3, i ens va complicar les coses a l'hora d'implementar el joc en si en l'entorn de pygame que ja teníem dissenyat, ja que a les problemàtiques incloses en el fet de programar un 3 en ratlla s'hi va sumar la necessitat de fer-ho directament en un entorn amb què no estàvem tan familiaritzats com era el de pygame. En el cas d'haver de fer un altre cop un projecte similar, creiem que caldria primerament assentar les bases del projecte en l'entorn principal en què ja sabem treballar com és el Python i fer després una conversió i sofisticació d'aquesta idea principal de codi per implementar-la amb pygame. Si bé és cert que ens ha generat dificultats fer-ho d'aquesta manera, ens ha servit per aprendre més sobre l'ús de pygame.

És digne de menció també els problemes que ens van donar certes funcions, com ara la funció de condició de victòria, a causa d'errors petits o descuits dels quals no ens vam adonar.

5. REPARTIMENT DE RESPONSABILITATS

En un primer moment, vam treballar conjuntament i mentre l'Artur feia les bases de la interfície a nivell visual, el César buscava les imatges a utilitzar (format .png) i les editava amb Photoshop.

Un cop vam tenir ja les imatges, mentre l'Artur acabava de fer la interfície, el César pensava i començava a implementar la 'condició de victòria'. Vam discutir sobre com volíem fer el programa final i quina forma volíem que tingués; quines pantalles tindria i que hi hauria a cada una d'elles. Un cop decidit això, i amb la primera pantalla 'Main Menu' feta per l'Artur, vam fer la pantalla 'Game', que es la del joc en sí pròpiament. Entre els 2, vam escriure la funció que creava una matriu buida de tamany X per X. Finalment, mentre l'Artur va fer la resta del codi (funcions de PLAY, GAME i MAIN MENU), el César va redactar l'informe.

6. VALORACIÓ DEL APRENENTATGE

Creiem que ha sigut una experiència molt enriquidora a nivell de coneixement sobre programació en un entorn més 'real' que el que ofereixen els problemes de classe, i ens ha ajudat a afrontar problemes aparentment senzills des de punts de vista molt diversos en busca de la millor solució possible.

Podem afirmar que tots 2 hem après ha superar els problemes que ha generat el fet de treballar en un entorn gairebé nou com era pygame i gràcies als errors que han anat sorgint i a la resolució dels mateixos, hem obtingut coneixement sobre com encarar els problemes en properes ocasions i com plantejar futures practiques per a obtenir un millor resultat final i un treball en grup més satisfactori. Tot i així, estem orgullosos del treball que hem fet i el resultat final obtingut.

7. REFERENCIAS

BaralTech. (2022, 5 enero). HOW TO MAKE A MENU SCREEN IN PYGAME! - YouTube.

<https://www.youtube.com/watch?v=GMBqjxcKogA>

Coding Spot. (2020, 17 agosto). Tic Tac Toe Using Python and Pygame (Part 4) | Winning and Restart Function. - YouTube.

<https://www.youtube.com/watch?v=dNJtgXzdxfo>

BaralTech. (2021, 21 abril). EASY Way to Make BUTTONS for Python/PyGame Projects. - YouTube.

https://www.youtube.com/watch?v=al_V4OGSvFU

Curso de pygame 2020. (s. f.). - YouTube.

https://www.youtube.com/playlist?list=PLuB3bC9rWQAu6cGeRo_I6QV8cz1_2V6uM

Curso PYGAME de cero. (s. f.). - YouTube.

<https://www.youtube.com/playlist?list=PLVzwufPir356RMxSsOccc38jmxfxqfBdp>

Chat Online – Chat GPT

<https://chat.openai.com>

Santander Universidades. (2022, 16 septiembre). ¿Qué es Python? | Blog. Becas Santander.

<https://www.becas-santander.com/es/blog/python-que-es.html>

KeepCoding, R. (2022, 9 junio). ¿Qué es Pygame? KeepCoding Tech School.

<https://keepcoding.io/blog/que-es-pygame/>

Despoudis, T. (2021, 22 marzo). How to Use PyGame for Game Development. ActiveState.

<https://www.activestate.com/blog/how-to-use-pygame-for-game-development/>

BaralTech. (2021b, abril 21). EASY Way to Make BUTTONS for Python/PyGame Projects.

YouTube. https://www.youtube.com/watch?v=al_V4OGSvFU

BaralTech. (2022b, enero 5). HOW TO MAKE A MENU SCREEN IN PYGAME! - YouTube.

<https://www.youtube.com/watch?v=GMBqjxcKogA>

8. CÓDIGO

```
#Inici

# Marcar tablero
from marcar_a_tablero import *

# Comprovar Comprovar guanyador
from comprobar_guanyador import *

# Botons
from botons import *

import pygame
import sys
pygame.init()
#-----
#GENERAL

## Crear finestra
SCREEN_X = 800
SCREEN_Y = 450

size = (SCREEN_X, SCREEN_Y)
screen = pygame.display.set_mode(size)

# Fons de pantalla general
fons = pygame.image.load("IMATGES/FONS.png")

# Fons de pantalla TITOL
fons_titol = pygame.image.load("IMATGES/FONS_TITOL.png")

# Icono
icono = pygame.image.load("IMATGES/ICONO.png")
pygame.display.set_icon(icono)

# Jugadors
O = pygame.image.load("IMATGES/O.png")
X = pygame.image.load("IMATGES/X.png")

# Tipologia de lletra
main_font = pygame.font.SysFont("cambria", 50)
main_font2 = pygame.font.SysFont("cambria", 20)
main_font3 = pygame.font.SysFont("cambria", 40)
main_font4 = pygame.font.SysFont("cambria", 30)

#Temany taula predeterminat
MODO_DE_TAUULA = 3

#Modo de joc predeterminat
MODO_DE_JOC = 3

# FPS
FPS = 60 # FPS
temps = pygame.time.Clock()

#-----
## PANTALLA DEL "MAIN MENU"
def main_menu():
    # Nom del "MAIN MENU" (nom del joc)
    pygame.display.set_caption("RES EN RATLLA")

    #BUCLE GENERAL
    while True:

        #DETECTAR OPSICCIO DE RATOLI
        MENU_MOUSE_POS = pygame.mouse.get_pos()

        # "IMATGES" PANTALLA
        ## Fons de pantalla GENERAL
        screen.blit(fons, (0, 0))
        ## Titol
        screen.blit(fons_titol, (0, 0))

        #BOTONS
        ##PLAY
        PLAY = pygame.image.load("IMATGES/PLAY.png")
        quadre_play = pygame.transform.scale(PLAY, (250, 100))
        MENU_PLAY_BUTTON = Button(image=quadre_play, pos=(
```

```

        400, 325), text_input="PLAY", font=main_font, base_color=(255, 255, 255),
        hovering_color=(128, 0, 128))

    for button in [MENU_PLAY_BUTTON]:
        button.changeColor(MENU_MOUSE_POS)
        button.update(screen)
    # EVENTOS QUE PODEN PASSAR DINS DE MAIN MENU
    ## Registra tots els "eventos" que passa a la pantalla
    for event in pygame.event.get():
        ### Sortir al clicar la X
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()
        ### AL CLCAR
        if event.type == pygame.MOUSEBUTTONDOWN:
            # Si has clicat a play
            if MENU_PLAY_BUTTON.checkForInput(MENU_MOUSE_POS):
                # Canviar de pantalla a play
                play()

    # FPS
    temps.tick(FPS)

    # Actualitzar (repetir comando de adalt)
    pygame.display.update()

# -----
#PLAY (tot)

## PANTALLA DEL "PLAY"
def play():
    # Nom de la finestra "PLAY"
    pygame.display.set_caption("Play")

    #BUCLE GENERAL
    while True:
        global MODO_DE_TAUULA
        global MODO_DE_JOC
        global MATRIU
        global TORN
        global R_WIN_X
        global R_WIN_Y
        global PLAYER_TORN

        #DETECTAR OPSICCIO DE RATOLI
        PLAY_MOUSE_POS = pygame.mouse.get_pos()

        # "IMATGES" PANTALLA
        ## Fons de pantalla GENERAL
        screen.blit(fons, (0, 0))

        ## Temany taula
        TEXT_TAUULA = main_font3.render("Tauler de joc", True, (0, 0, 0))
        screen.blit(TEXT_TAUULA, (100, 100))

        ### NUM DEL Temany taula
        mod_taula = main_font3.render(str(MODO_DE_TAUULA), True, (0, 0, 0))
        screen.blit(mod_taula, (195, 165))

        ## Modo de joc en pantalla
        TEXT_MODAL = main_font4.render("Condicció de victòria", True, (0, 0, 0))
        screen.blit(TEXT_MODAL, (450, 110))

        ### NUM DEL Modo de joc
        TEXT_MODAL = main_font3.render(str(MODO_DE_JOC), True, (0, 0, 0))
        screen.blit(TEXT_MODAL, (570, 165))

        #BOTONS
        ##PLAY
        MENU_PLAY2_BUTTON = pygame.image.load("IMATGES/PLAY.png")
        quadre_play2 = pygame.transform.scale(MENU_PLAY2_BUTTON, (375, 100))

        MENU_PLAY2_BUTTON = Button(image=quadre_play2, pos=(
            400, 350), text_input="COMENÇAR", font=main_font, base_color=(255, 255, 255),
            hovering_color=(128, 0, 128))

```

```

        ##REDUIR TEMANY TAULA
        TAULA_MENYS = Button(image=None, pos=(150, 185),
                               text_input="<", font=main_font, base_color="White",
                               hovering_color=(128, 0, 128))

        ##INCREMENTAR TEMANY TAULA
        TAULA_MES = Button(image=None, pos=(260, 185),
                               text_input=">", font=main_font, base_color="White", hovering_color=(128,
0, 128))

        ##REDUIR MODO DE JOC
        MODO_MENYS = Button(image=None, pos=(525, 185),
                               text_input="<", font=main_font, base_color="White", hovering_color=(128,
0, 128))

        ##INCREMENTAR MODO DE JOC
        MODO_MES = Button(image=None, pos=(635, 185),
                               text_input=">", font=main_font, base_color="White", hovering_color=(128,
0, 128))

        ##GEENERAL
        for button in [MENU_PLAY2_BUTTON, TAULA_MENYS, TAULA_MES, MODO_MENYS, MODO_MES]:
            button.changeColor(PLAY_MOUSE_POS)
            button.update(screen)

        # EVENTOS QUE PODEN PASSAR DINS DE GAME
        ## Registra tots els "eventos" que passa a la pantalla
        for event in pygame.event.get():
            ##Sortir al clicar la X
            if event.type == pygame.QUIT:
                pygame.quit()
                sys.exit()
                ### AL CLCAR

            if event.type == pygame.MOUSEBUTTONDOWN:
                # Si has clicat a play
                if MENU_PLAY2_BUTTON.checkForInput(PLAY_MOUSE_POS):
                    if MODO_DE_JOC > MODO_DE_TAUOLA:
                        print("Error: MODO DE JOC > TAULER")
                    # Canviar de pantalla a play
                    else:
                        #GENERAR MATRIU TABLERO
                        MATRIU = [[0] * MODO_DE_TAUOLA for i in range(MODO_DE_TAUOLA)]
                        #CONFIGURACIO DEL JOC
                        TORN = 1
                        R_WIN_X = 0
                        R_WIN_Y = 0
                        PLAYER_TORN = 1
                        game()

                #RESTRICCIÓ DE CONDICIÓN DE VICTÒRIA - TAMANY TABLERO
                if TAULA_MENYS.checkForInput(PLAY_MOUSE_POS):
                    if MODO_DE_TAUOLA > 3:
                        MODO_DE_TAUOLA -= 1

                if TAULA_MES.checkForInput(PLAY_MOUSE_POS):
                    if 12 > MODO_DE_TAUOLA:
                        MODO_DE_TAUOLA += 1

                if MODO_MENYS.checkForInput(PLAY_MOUSE_POS):
                    if MODO_DE_JOC > 2:
                        MODO_DE_JOC -= 1

                if MODO_MES.checkForInput(PLAY_MOUSE_POS):
                    if 12 > MODO_DE_JOC:
                        MODO_DE_JOC += 1

        #FPS
        temps.tick(FPS)

        # Actualitzar (repetir comando de adalt)
        pygame.display.update()

# -----
#GAME (tot)

## PANTALLA DEL "GAME"
def game():
    # Nom del "MAIN MENU" (nom del joc)
    pygame.display.set_caption("GAME")

```

```

#BUCLE GENERAL
while True:
    global MODO_DE_TAUOLA
    global MODO_DE_JOC
    global MATRIU
    global PLAYER_TORN
    global TORN
    global R_WIN_X
    global R_WIN_Y

    #DETECTAR OPSICCIO DE RATOLI
    GAME_MOUSE_POS = pygame.mouse.get_pos()

    # "IMATGES" PANTALLA
    ## Fons de pantalla GENERAL
    screen.blit(fons, (0, 0))

    # #RACHA DE VICTORIES
    # X
    rect_color_X = (0, 255, 0)

    pos_x_inicial_X = (SCREEN_X/14)
    pos_y_inicial_X = (SCREEN_Y/7)*2

    allargada_X = (SCREEN_X/14)*2
    altura_X = (SCREEN_Y/7)*3
    '''
    pygame.draw.rect(screen, rect_color_X, (pos_x_inicial_X,
                                             pos_y_inicial_X, allargada_X, altura_X))
    '''

    ### ICONO X
    X_RACHA_IMATGE = pygame.image.load("IMATGES/X.png")
    x_racha_imatge = pygame.transform.scale(X_RACHA_IMATGE, (allargada_X, allargada_X))
    screen.blit(x_racha_imatge, (pos_x_inicial_X, pos_y_inicial_X))

    ### ICONO NUM X
    X_RACHA_ESCRIT = main_font2.render("VICTORIES: " + str(R_WIN_X), True, (0, 0, 0))
    screen.blit(X_RACHA_ESCRIT, (pos_x_inicial_X, 275))

    ## O
    rect_color_O = (0, 255, 0)

    pos_x_inicial_O = SCREEN_X-allargada_X-pos_x_inicial_X
    pos_y_inicial_O = pos_y_inicial_X

    allargada_O = allargada_X
    altura_O = altura_X
    '''
    pygame.draw.rect(screen, rect_color_O, (pos_x_inicial_O,
                                             pos_y_inicial_O, allargada_O, altura_O))
    '''

    ### ICONO O
    O_RACHA_IMATGE = pygame.image.load("IMATGES/O.png")
    o_racha_imatge = pygame.transform.scale(O_RACHA_IMATGE, (allargada_O, allargada_O))
    screen.blit(o_racha_imatge, (pos_x_inicial_O, pos_y_inicial_O))

    ### ICONO NUM O
    X_RACHA_ESCRIT = main_font2.render("VICTORIES: " + str(R_WIN_Y), True, (0, 0, 0))
    screen.blit(X_RACHA_ESCRIT, (pos_x_inicial_O, 275))

    ## TABLERO
    rect_color_TABLERO = (150, 150, 150)

    allargada_TABLERO = 350
    altura_TABLERO = 350

    pos_x_inicial_TABLERO = (SCREEN_X-allargada_TABLERO)/2
    pos_y_inicial_TABLERO = (SCREEN_Y-altura_TABLERO)/2
    '''
    pygame.draw.rect(screen, rect_color_TABLERO, (pos_x_inicial_TABLERO, pos_y_inicial_TABLERO,
allargada_TABLERO, altura_TABLERO))
    '''

```



```

#QUADRICULA
line_color = (0, 0, 0)
line_grux = 2

##LINIES VERTICALS
VarX_TABLERO = allargada_TABLERO/MODO_DE_TAUOLA
pos_y_inicial_L_V = pos_y_inicial_TABLERO

pos_y_final_L_V = pos_y_inicial_TABLERO+altura_TABLERO

for i in range(MODO_DE_TAUOLA-1):
    pos_x_inicial_L_MODIF = pos_x_inicial_TABLERO+VarX_TABLERO*(i+1)
    pygame.draw.line(screen, line_color, (pos_x_inicial_L_MODIF,
                                           pos_y_inicial_L_V), (pos_x_inicial_L_MODIF,
pos_y_final_L_V), line_grux)

##LINIES HORITZONTAL
pos_x_inicial_L_H = pos_x_inicial_TABLERO
VarY_TABLERO = VarX_TABLERO

pos_x_final_L_H = pos_x_inicial_TABLERO+allargada_TABLERO

for i in range(MODO_DE_TAUOLA-1):
    pos_y_inicial_L_MODIF = pos_y_inicial_TABLERO+VarY_TABLERO*(i+1)
    pygame.draw.line(screen, line_color, (pos_x_inicial_L_H,
                                           pos_y_inicial_L_MODIF), (pos_x_final_L_H,
pos_y_inicial_L_MODIF), line_grux)

#DIBUIXAR X i O A TABLERO
n = int(VarX_TABLERO-line_grux)

X_redimensionada = pygame.transform.scale(X, (n, n))
O_redimensionada = pygame.transform.scale(O, (n, n))

mitad_del_gruix = line_grux/2

Num_fila = 0

for i in MATRIU:
    CORDENADES_X_per_pecas_MOD = pos_x_inicial_TABLERO+VarX_TABLERO*Num_fila
    Num_column = 0

    for j in i:
        CORDENADES_y_per_pecas_MOD = pos_y_inicial_TABLERO+VarY_TABLERO*Num_column
        if j == 1:
            screen.blit(
                X_redimensionada, (CORDENADES_X_per_pecas_MOD+mitad_del_gruix,
CORDENADES_y_per_pecas_MOD+mitad_del_gruix))
        elif j == 2:
            screen.blit(
                O_redimensionada, (CORDENADES_X_per_pecas_MOD+mitad_del_gruix,
CORDENADES_y_per_pecas_MOD+mitad_del_gruix))
            Num_column += 1

    Num_fila += 1

'''
#RAYAA !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
line_color=(0,0,0)

pos_x_inicial_L = 400
pos_y_inicial_L = 0

pos_x_final_L = 400
pos_y_final_L = 450

line_grux = 15

pygame.draw.line(screen, line_color, (pos_x_inicial_L,
pos_y_inicial_L), (pos_x_final_L, pos_y_final_L), line_grux)

'''

#ICONO EXIT
EXIT = pygame.image.load("IMATGES/EXIT.png")
exit_t = pygame.transform.scale(EXIT, (40, 40))
EXIT_BUTTON = Button(image=exit_t, pos=(

```

```

750, 50)), text_input="", font=main_font2, base_color=(0, 0, 0), hovering_color=(50, 50,
50))

# ICONO RESTART GAME
RESTART_GAME = pygame.image.load("IMATGES/RESTART_GAME.png")
restart_game = pygame.transform.scale(RESTART_GAME, (190, 60))
RESTART_GAME_BUTTON = Button(image=restart_game, pos=(675, 400), text_input="NEW GAME",
font=main_font2, base_color=(0, 0, 0), hovering_color=(255, 255, 255))

##GEENERAL
for button in [EXIT_BUTTON]:
    button.changeColor(GAME_MOUSE_POS)
    button.update(screen)

#IMATGE WIN O
WIN_O = pygame.image.load("IMATGES/WIN_O.png")
wins_o = pygame.transform.scale(WIN_O, (400, 200))

# IMATGE WIN X
WIN_X = pygame.image.load("IMATGES/WIN_X.png")
wins_x = pygame.transform.scale(WIN_X, (400, 200))

#MOSTRAR PER PANTALLA QUI GUANYA
if check_winner(MATRIU,MODO_DE_JOC) in (1, 2):
    RESTART_GAME_BUTTON.changeColor(GAME_MOUSE_POS)
    RESTART_GAME_BUTTON.update(screen)

    if check_winner(MATRIU,MODO_DE_JOC) == 1:
        screen.blit(wins_x, (200, 75))

    else:
        screen.blit(wins_o, (200, 75))

# EVENTOS QUE PODEN PASSAR DINS DE GAME
# Registra tots els "eventos" que passa a la pantalla
for event in pygame.event.get():
    ### Sortir al clicar la X
    if event.type == pygame.QUIT:
        pygame.quit()
        sys.exit()
    ### AL CLCAR
    if event.type == pygame.MOUSEBUTTONDOWN:
        if EXIT_BUTTON.checkForInput(GAME_MOUSE_POS):
            #Temany taula predeterminat
            MODO_DE_TAUOLA = 3
            #Modo de joc predeterminat
            MODO_DE_JOC = 3
            #Jugador que començara
            play()

#TORNAR A JUGAR
if check_winner(MATRIU,MODO_DE_JOC) == 1:
    if RESTART_GAME_BUTTON.checkForInput(GAME_MOUSE_POS):
        PLAYER_TORN = 2
        R_WIN_X += 1
        TORN = 1
        MATRIU = [[0] * MODO_DE_TAUOLA for i in range(MODO_DE_TAUOLA)]

elif check_winner(MATRIU,MODO_DE_JOC) == 2:
    PLAYER_TORN = 1
    R_WIN_Y += 1
    TORN = 1
    MATRIU = [[0] * MODO_DE_TAUOLA for i in range(MODO_DE_TAUOLA)]

#MARCAR TABLERO
ratX = event.pos[0] #cordenada X al clicar
ratY = event.pos[1] #cordenades Y al clicar

    if (pos_x_inicial_TABLERO < ratX < pos_x_inicial_TABLERO+allargada_TABLERO) and
(pos_y_inicial_TABLERO < ratY < pos_y_inicial_TABLERO+altura_TABLERO):

        #Quina fila ha clicat
        valor_clic_files = int((-pos_x_inicial_TABLERO+ratX)// VarX_TABLERO)

        #Quina columna ha clicat
        valor_clic_columnes = int((-pos_y_inicial_TABLERO+ratY)// VarY_TABLERO)

        #PRIMERA FASE JOC
        if abiable_sqare_1(valor_clic_files,valor_clic_columnes,MATRIU) and
TORN<(MODO_DE_TAUOLA*MODO_DE_TAUOLA) and (check_winner(MATRIU, MODO_DE_JOC)==False):

```

```

        if PLAYER_TORN == 1:
            PLAYER_TORN = 2
            TORN +=1
            Marcar_On_Qui_1(valor_clic_files,valor_clic_columnes, 1, MATRIU)

        elif PLAYER_TORN == 2:
            PLAYER_TORN = 1
            TORN +=1
            Marcar_On_Qui_1(valor_clic_files,valor_clic_columnes, 2, MATRIU)

        #SEGONA FASE JOC
        elif abiable_sqare_2(valor_clic_files, valor_clic_columnes, PLAYER_TORN, MATRIU)
and TORN >= (MODO_DE_TAULA*MODO_DE_TAULA) and (check_winner(MATRIU, MODO_DE_JOC)==False):
            if PLAYER_TORN == 1:
                PLAYER_TORN = 2
                TORN +=1
                Marcar_On_Qui_2(valor_clic_files,valor_clic_columnes, MODO_DE_TAULA,
MATRIU)

            elif PLAYER_TORN == 2:
                PLAYER_TORN = 1
                TORN +=1
                Marcar_On_Qui_2(valor_clic_files,valor_clic_columnes, MODO_DE_TAULA,
MATRIU)

        # FPS
        temps.tick(FPS)

        # Actualitzar (repetir comando de adalt)
        pygame.display.update()

#-----
#INICIAR AMB "MAIN MENU"

main_menu()

```