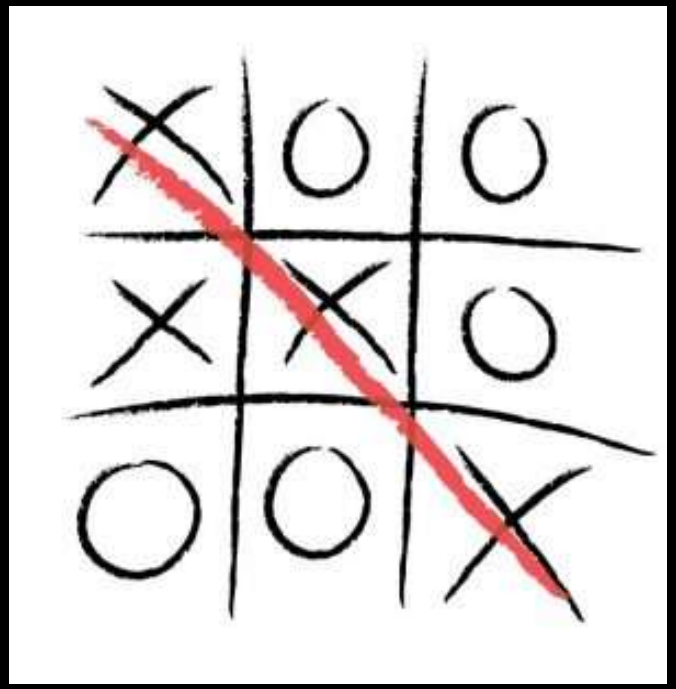
Tic Tac Toe (pygame)



Ομάδα 50:

Αναστάσης Κιουτσιούκης

Βασίλης Καρατράντος

Γεωργία Μπασαγιάννη

Χρήστος-Άρης Κοκόσης

Κατερίνα Τζούλα

Ελένη Σακελλαρίου-Μάτση

Το πρόβλημα και οι αρμοδιότητες μας

Δημιουργία του παιχνιδιού tic tac toe (τρίλιζα) με τη βιβλιοθήκη pygame της python.

- Γεωργία: Δημιουργία πίνακα, τετραγώνων, επιλογή σχημάτων, tkinter
- Βασίλης: Δημιουργία συνάρτησης που ελέγχει ποιος έχει κερδίσει
- Χρήστος: Δημιουργία συνάρτησης, η οποία αξιοποιεί τη συνάρτηση του Βασίλη και τυπώνει ποιος έχει κερδίσει αναλόγως
- Ελένη: Συνάρτηση που εναλλάσσει τους παίκτες και τυπώνει το αντίστοιχο σύμβολο ανάλογα τον παίκτη, tkinter, βοήθησε στην οργάνωση του κώδικα
- Κατερίνα: Συνάρτηση που καθορίζει την τυχαία θέση που θα παίξει ο υπολογιστής στην κλάση AI
- Αναστάσης: Δημιουργία των κλάσεων, οργάνωση του κώδικα, δημιουργία επιπρόσθετων συναρτήσεων για να συνδεθούν οι παραπάνω κώδικες, tkinter

Ο κώδικας

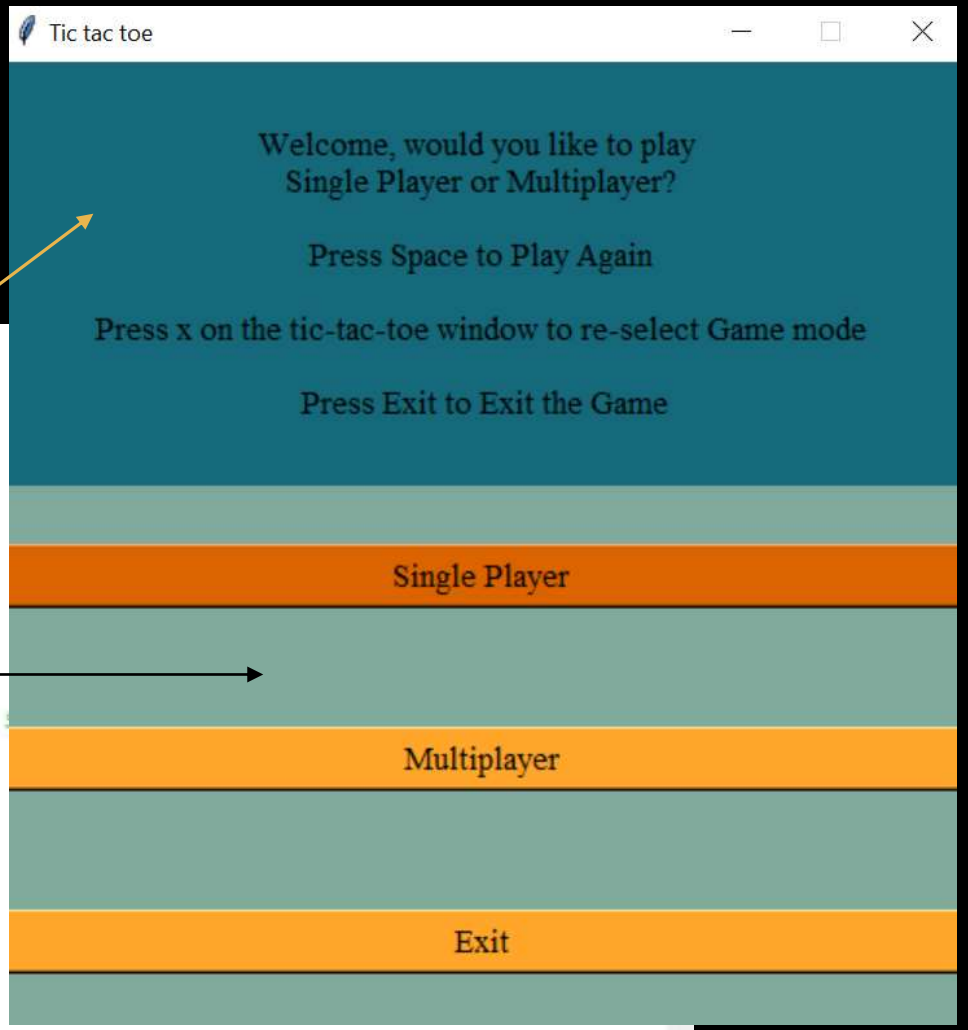
Ο χρήστης έχει τη δυνατότητα να επιλέξει αν θα παίξει εναντίον του υπολογιστή ή εναντίον κάποιου άλλου παίκτη.

Ο κώδικας αποτελείται από 4 κλάσεις:

1) **Class Handler** : Με την εκκίνηση του κώδικα καλείται η συγκεκριμένη κλάση, η οποία δημιουργεί το παράθυρο tkinter που περιέχει το μενού του παιχνιδιού.
2) **Class TicTacToe** : Είναι η μητρική κλάση, η οποία έχει τις κοινές μεθόδους που υπάρχουν στις δύο παρακάτω κλάσεις.
3) **Class AI**: Καλείται αν ο χρήστης επιλέξει να παίξει εναντίον του υπολογιστή.
4) **Class Game**: Καλείται όταν ο χρήστης επιλέξει να παίξουν δύο παίκτες.

class Handler

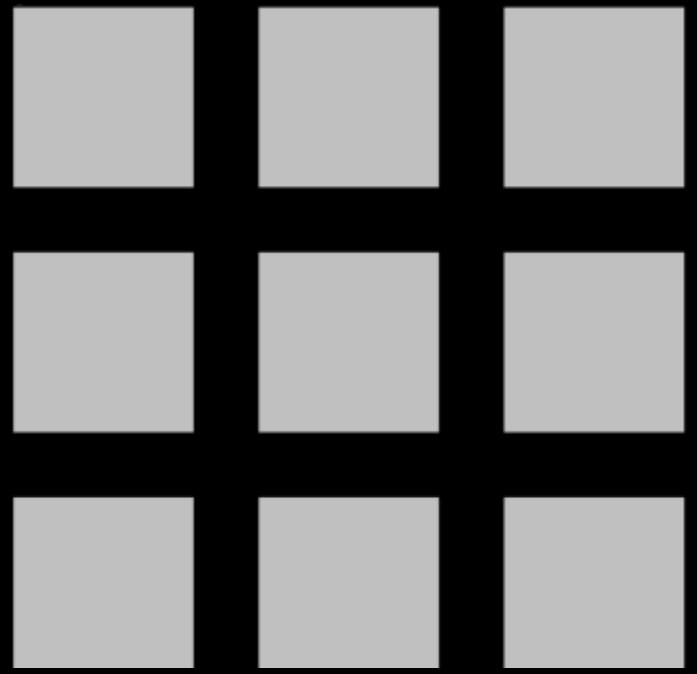
```
class Handler():  # creates the tkinter window
    def __init__(self, root, screen, x, o):
        self.screen = screen
        self.x = x
        self.o = o
        self.root = root
        self.root.title('Tic tac toe')
        self.root.resizable(False, False)
        self.root.geometry('500x500')
        self.mywidgets()

    def mywidgets(self): #creates the widgets
        f=tk.Frame(root,bg='#7fa99b')
        f.pack(expand=1,fill='both')
        l=tk.Label(f,text='Welcome, would you like to play \n Single Player or Multiplayer? \n\n Press
                   bg='#16697a',font=('Times',13))
        l.pack(expand=1,fill='both')
        b=tk.Button(f,text='Single Player',bg='#db6400',font=('Times',13), command=self.single_player)
        b.pack(expand=1,fill='x')
        b2=tk.Button(f,text='Multiplayer',bg='#ffa62b',font=('Times',13), command=self.multiplayer)
        b2.pack(expand=1,fill='x')
        b3=tk.Button(f,text='Exit',bg='#ffa62b',font=('Times',13), command=self.destroy_window)
        b3.pack(expand=1,fill='x')

    def single_player(self): #καλεί την κλάση AI
        AI(screen, x, o)

    def multiplayer(self): #καλεί την κλάση Game
        Game(screen, x, o)

    def destroy_window(self): #κλείνει τα παράθυρα των tkinter και pygame
        pygame.quit()
        self.root.destroy()
```



class TicTacToe (μητρική κλάση)

χρώμα διαστάσεις

```
def square_maker(self):  # Δημιουργεί τα τετράγωνα
    self.first = pygame.draw.rect(self.screen, (192, 192, 192), (43, 30, 180, 180))
    self.second = pygame.draw.rect(self.screen, (192, 192, 192), (288, 30, 180, 180))
    self.third = pygame.draw.rect(self.screen, (192, 192, 192), (533, 30, 180, 180))
    self.fourth = pygame.draw.rect(self.screen, (192, 192, 192), (43, 275, 180, 180))
    self.fifth = pygame.draw.rect(self.screen, (192, 192, 192), (288, 275, 180, 180))
```



```
def open_positions(self):  # Defining the booleans that determine if a block is open
    self.first_open = True
    self.second_open = True
```

Μόνο αν είναι ανοικτό μπορεί ο παίκτης να το επιλέξει

```
def win_check(self, num): #Ελέγχει αν έχει κερδίσει κάποιος από τους δύο παίκτες
    for row in self.board:
        for tile in row:
            if tile == num:
                continue
            else:
                break
        else:
            return True
```

Αντίστοιχη διαδικασία για τις στήλες και τις διαγώνιους του πίνακα

```
def draw_line(self):  # This is a function that draws a line according to whoever won the game and which win condition did they meet
    if self.board[0][0] == self.board[0][1] == self.board[0][2] == 1 or self.board[0][0] == self.board[0][1] == self.board[0][2] == 2:
        pygame.draw.line(self.screen, (0, 0, 0), (30, 120), (720, 120), 10)
    if self.board[1][0] == self.board[1][1] == self.board[1][2] == 1 or self.board[1][0] == self.board[1][1] == self.board[1][2] == 2:
```

class AI (player vs computer)

```
class AI(TicTacToe):  # Creating the class of the AI Game
    def __init__(self, screen, x, o):
        self.screen = screen
        self.x = x
        self.o = o
        self.square_maker()
        self.game = True
        self.won = False
        self.run = True
        self.open_positions()
        self.starting_status()
        self.board = [[0, 0, 0], [0, 0, 0], [0, 0, 0]]
        self.runner()

    def starting_status(self):# A function that contains the status of the game
        self.current_player = 1
        self.font = pygame.font.Font('freesansbold.ttf', 26)
        self.text = self.font.render("It's your turn.", True, (215, 236, 255))
        self.textRect = self.text.get_rect()
        self.screen.blit(self.text, self.textRect)
        self.rects = {"first": (70, 55), "second": (315, 55), "third": (560, 55
                      "fifth": (315, 300),
                      "sixth": (560, 300), "seventh": (70, 545), "eighth": (315
```

ο χρήστης παίζει πρώτος

at the start

συντεταγμένες που θα ζωγραφιστούν τα αντίστοιχα σύμβολα

Κάθε φορά που παίζει ο παίκτης, εφόσον δεν έχει κερδίσει κάποιος, καλείται η μέθοδος που καθορίζει τις κινήσεις του υπολογιστή.

Επανεκκίνηση του παιχνιδιού

```
def runner(self):
    while self.run:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                self.screen.fill((0, 0, 0), (0, 0, 750, 30))
                self.run = False

            if event.type == pygame.KEYDOWN:
                if event.key == pygame.K_SPACE:
                    self.game = True
                    self.won = False
                    self.board = [[0, 0, 0], [0, 0, 0], [0, 0, 0]]
                    self.screen.fill((0, 0, 0), (0, 0, 750, 750))
                    self.starting_status()
                    self.open_positions()
                    self.square_maker()

            if event.type == pygame.MOUSEBUTTONUP:
                pos = pygame.mouse.get_pos()
                if self.draw_symbol(pos):
                    self.check()
                    self.computer_move(2)
                    self.check()

        pygame.display.update()
```

παίρνει τις συντεταγμένες του κλικ

```
def draw_symbol(self, pos):  # Function that draws x in the selected square and deletes the same square from the dictionary
    if not self.won and self.game:
        if self.current_player == 1:
            if self.first.collidepoint(pos) and self.first_open:
                screen.blit(x, (70, 55))
                self.board[0][0] = 1
                del self.rects["first"]
                self.first_open = False
                return True
```

ώστε ο υπολογιστής να μην μπορεί να παίξει στην ίδια θέση

Το αντίστοιχο τετράγωνο στον board παίρνει την τιμή του current player ώστε να βγει ο νικητής.

```
def computer_move(self, player):
    if self.game and not self.won:
        if player == 2:
            self.list_square = [x for x in self.rects.keys()]
            if len(self.list_square) == 0:
                self.game = False
                pass
            else:
                i = random.choice(self.list_square)
                self.screen.blit(o, self.rects[i])
                self.board_adder(i)
                self.computer_bool_changer(i)
                self.current_player = 1
                del self.rects[i]
```

```
def computer_bool_changer(self, key):
    if str(key) == "first":
        self.first_open = False
    elif key == "second":
        self.second_open = False
```

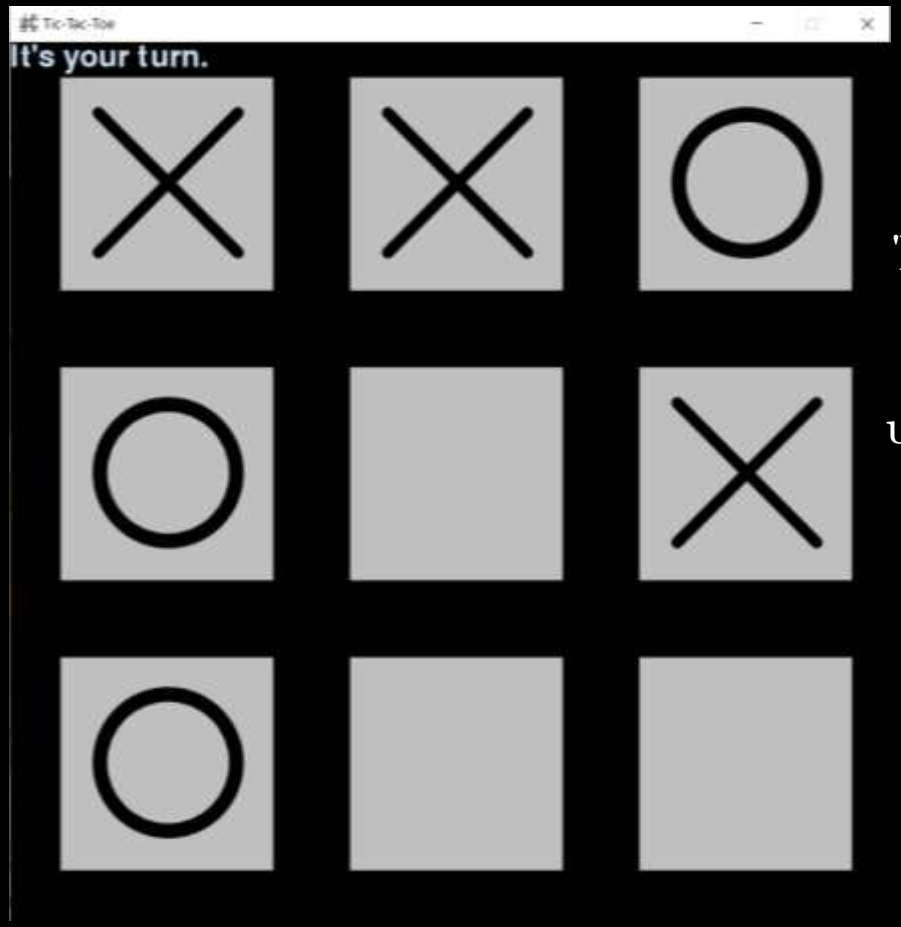
Κλειδώνει το τετράγωνο

```
def board_adder(self, key):
    if str(key) == "first":
        self.board[0][0] = 2
    elif key == "second":
        self.board[0][1] = 2
```

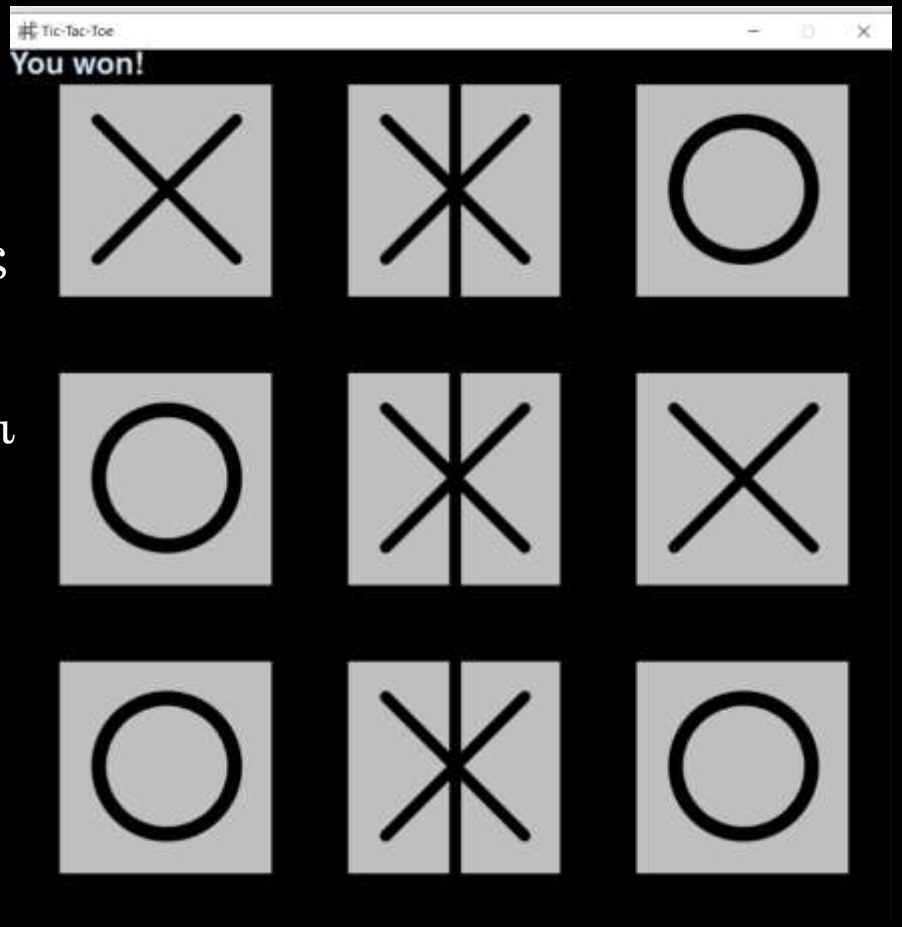
```
def check(self):
    if self.win_check(1):
        self.won = True
        self.game = False
        self.screen.fill((0, 0, 0), (0, 0, 750, 30))
        self.text = self.font.render('You won!', True, (215, 236, 255))
        self.screen.blit(self.text, self.textRect)
        self.draw_line()
```

Παρόμοιες διαδικασίες για την νίκη του υπολογιστή και την ισοπαλία.

Single Player



Έτσι, όταν ο παίκτης κάνει κλικ σε ένα τετράγωνο, ο υπολογιστής επιλέγει τυχαία ένα άλλο μέχρι να νικήσει κάποιος το παιχνίδι ή να καταλήξει σε ισοπαλία.



Ο παίκτης έχει το x κι ο υπολογιστής το o.

Class Game (player 1 vs player 2)

```
class Game(TicTacToe):
    def __init__(self, screen, x, o):  # The variables used in the class
        self.screen = screen
        self.x = x
        self.o = o
        self.square_maker()
        self.starting_status()
        self.game = True
        self.won = False
        self.run = True
        self.open_positions()
        self.board = [[0, 0, 0], [0, 0, 0], [0, 0, 0]]
        self.cell_rects = {"first":self.first,"second": self.second,"third": self.third,"fourth": self.fourth,"fifth": self.fifth,\
                           "sixth":self.sixth,"seventh": self.seventh,"eighth":self.eighth,"ninth": self.ninth}
        self.rect_info = {"first": [(70, 55),(0,0)], "second": [(315, 55),(0,1)], "third": [(560, 55),(0,2)],
                          "fourth": [(70, 300),(1,0)], "fifth": [(315, 300),(1,1)], "sixth": [(560, 300),(1,2)],
                          "seventh": [(70, 545),(2,0)], "eighth": [(315, 545),(2,1)], "ninth": [(560, 545),(2,2)]}
        self.init_list_open()
        self.runner()

    def init_list_open(self):
        self.rects_open = {"first": self.first_open, "second": self.second, "thi
                           "fourth": self.fourth_open, "fifth": self.fifth_open,
                           "sixth": self.sixth_open, "seventh": self.seventh_ope
                           "ninth": self.ninth_open}


    def starting_status(self):  # A function that contains the status of the game at the start
        self.player = 0
        self.font = pygame.font.Font('freesansbold.ttf', 26)
        self.text = self.font.render("Player 1's turn", True, (215, 236, 255))
        self.textRect = self.text.get_rect()
        self.screen.blit(self.text, self.textRect)
```

Λεξικό το οποίο αντιστοιχίζει κάθε τετράγωνο στις συντεταγμένες που θα ζωγραφιστεί το σχήμα και στην αντίστοιχη θέση στον πίνακα.

Class Game (player 1 vs player 2)

Το πρώτο μέρος της συνάρτησης **runner** είναι παρόμοιο με της AI

```
if event.type == pygame.MOUSEBUTTONUP:
    pos = pygame.mouse.get_pos()
    for cell in self.cell_rects:
        if self.cell_rects.get(cell).collidepoint(pos) and self.game and self.rects_open.get(cell):
            self.draw_mark(self.player, cell)
            self.switch_players()
            self.check()
            break
```

Η self.check είναι αντίστοιχη με αυτή της AI

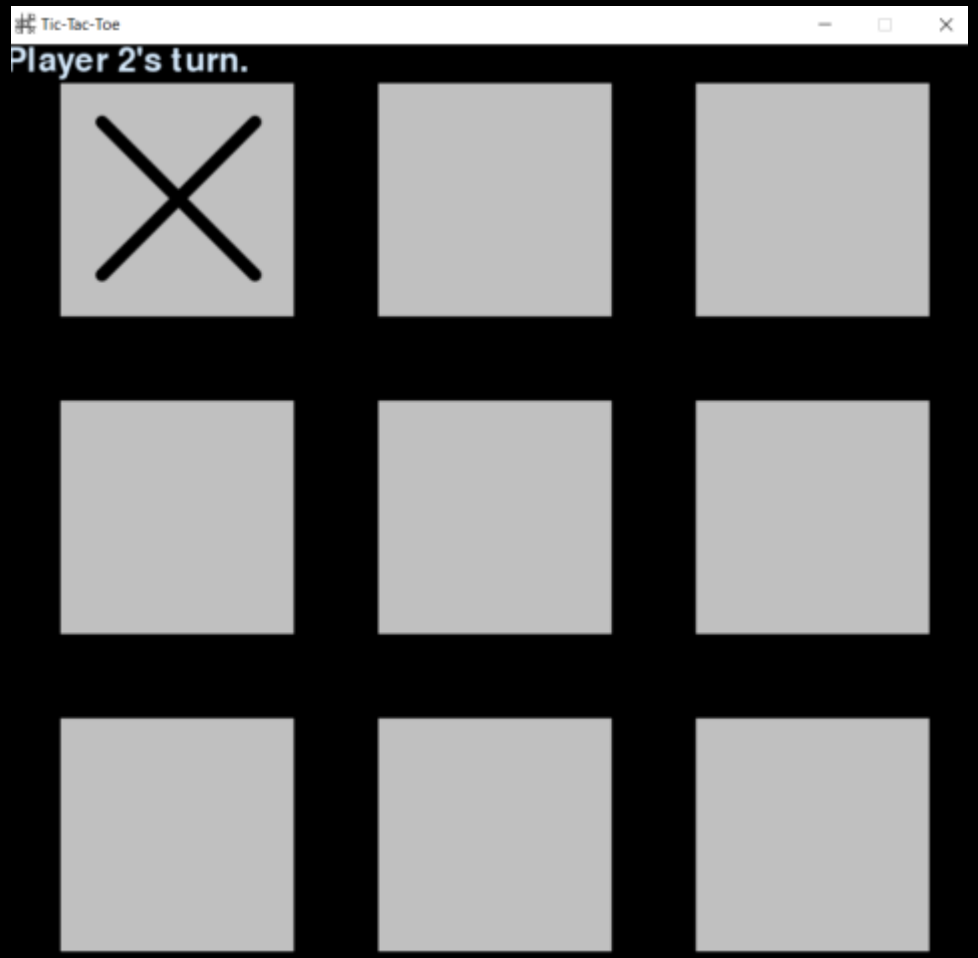
```
def switch_players(self):  # This function prints a different text according to which player is playing
    if self.player == 0:
        self.text = self.font.render("Player 2's turn.", True, (215, 236, 255), (0, 0, 0))
        self.textRect = self.text.get_rect()
        self.screen.blit(self.text, self.textRect)
        self.player = 1
    elif self.player == 1:
        self.text = self.font.render("Player 1's turn.", True, (215, 236, 255), (0, 0, 0))
        self.textRect = self.text.get_rect()
        self.screen.blit(self.text, self.textRect)
        self.player = 0
```

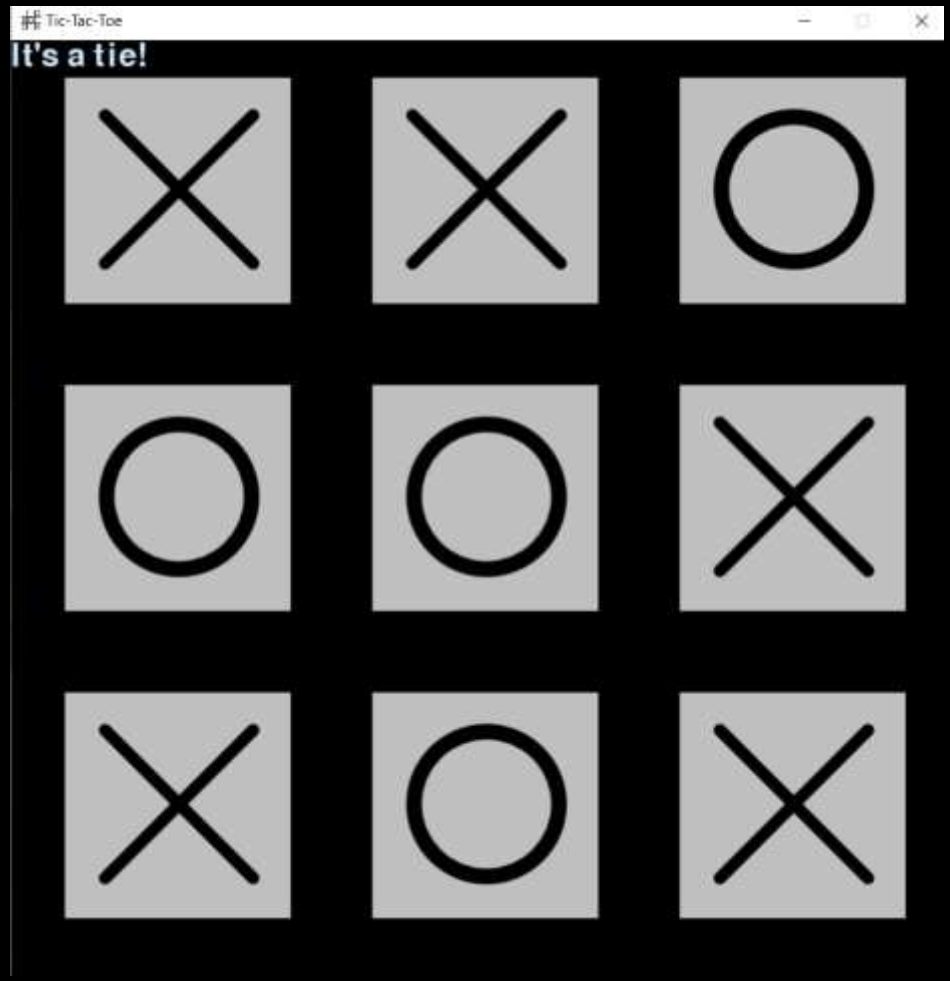
κι εναλλάσσει τους παίκτες

```
def draw_mark(self, current_player,key):
    if not self.won:
        if self.rects_open.get(key):
            self.rects_open[key]= False
            self.screen.blit(symbols[current_player], self.rect_info.get(key)[0])
            board_pos = self.rect_info.get(key)[1]
            self.board[board_pos[0]][board_pos[1]] = p_values[current_player]
```

Ανάλογα με τον παίκτη ζωγραφίζει το κατάλληλο σχήμα στο τετράγωνο που έγινε το κλικ και βάζει το σωστό νούμερο στο αντίστοιχο σημείο του πίνακα.

Player 1 vs Player 2





Ο παίκτης 1 έχει το x και ο 2 το ο.

Main Program

```
import pygame
import random
import tkinter as tk

# Creating the board and uploading the images
pygame.init()
screen = pygame.display.set_mode((750, 750))
pygame.display.set_caption("Tic-Tac-Toe")
x = pygame.image.load('cancel.png')
o = pygame.image.load('circle-ring.png')
symbols=[x,o]
p_values=[1,2]
icon = pygame.image.load('tic-tac-toe.png')
pygame.display.set_icon(icon)
```

Modules που χρησιμοποιήσαμε

Εισάγονται οι εικόνες που έχουν τα σύμβολα

```
root = tk.Tk()
root.title('Tic-Tac-Toe')
mygame = Handler(root, screen, x, o)
root.mainloop()
```

Δημιουργείται το παράθυρο tkinter
και καλείται η κλάση Handler

Ευχαριστούμε για την προσοχή σας!

Βιβλιογραφία:

- Βιβλίο Python Εισαγωγή στους Υπολογιστές
- Σημειώσεις από upatras eclass για το μάθημα
- Εικόνες google
- https://www.pygame.org/docs/
- https://www.python.org/
- https://www.youtube.com/watch?v=Gv8hsNsX5G4 και τα tutorials 2,3,4