

# ИТОГОВЫЙ ПРОЕКТ ПО КУРСУ "2"

Тема: Создание игры на Python  
"КРЕСТИКИ-НОЛИКИ"

Выполнил студенты группы ИС-  
27: Потемкин Матвей

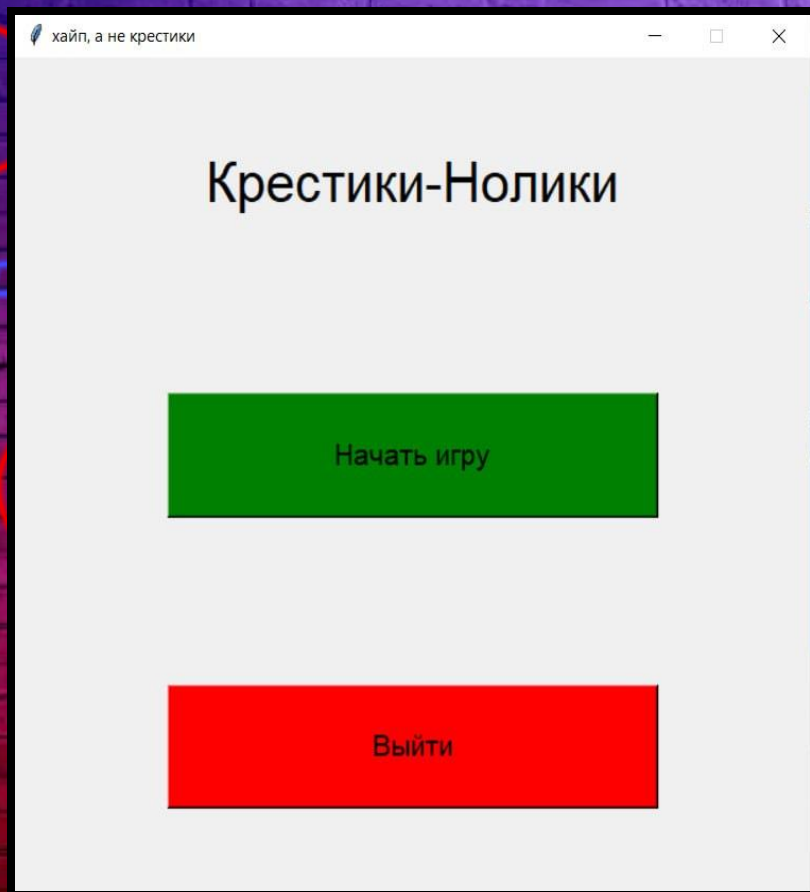
Гусенко Дмитрий

Руководитель проекта :  
Манакова Ольга Петровна

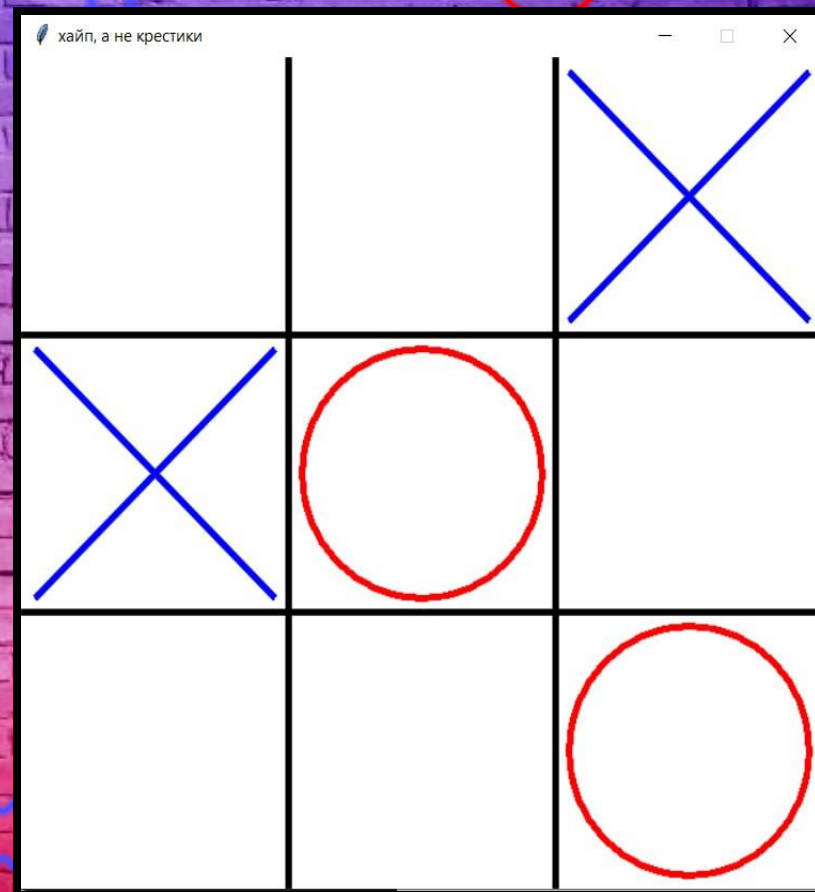


# ЦЕЛИ ПРОЕКТНОЙ РАБОТЫ

1) Сделать вступительное меню



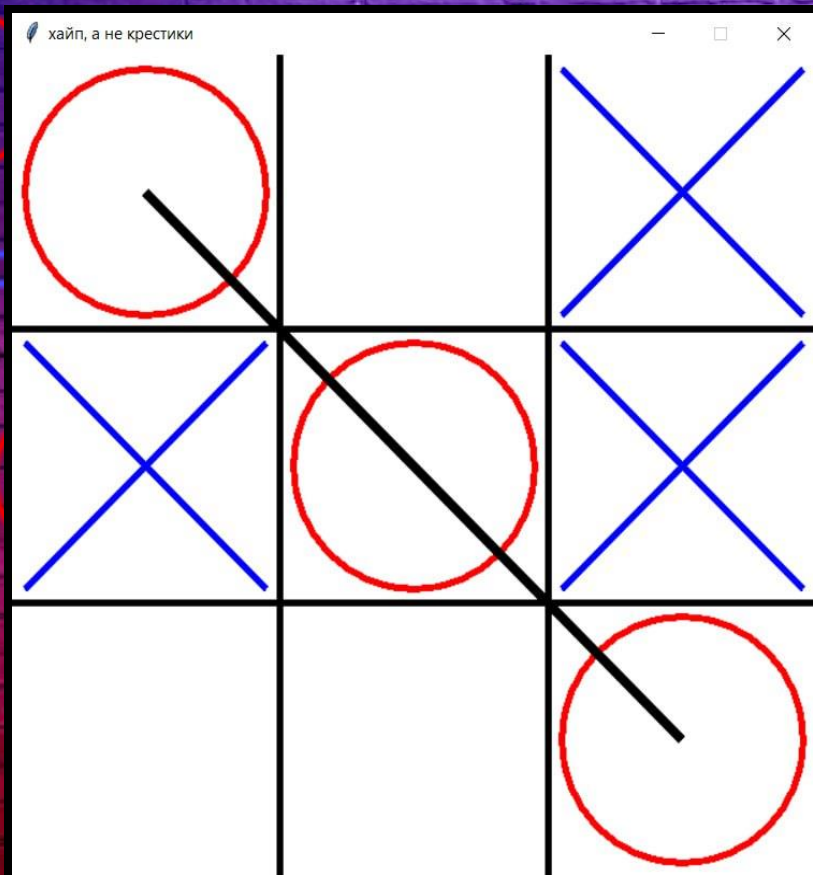
2) Сделать игровое поле





# ЦЕЛИ ПРОЕКТНОЙ РАБОТЫ

3) Сделать отображение победы



4) Сделать заключительное окно





# ЗАДАЧИ ПРОЕКТНОЙ РАБОТЫ

- 1. Обсуждение создания игры
- 2. Изучение библиотеки Tkinter
- 3. Создание дизайна (Визуализация)
- 4. Написание функционального кода (бэкенд)
- 5. Тестировка и исправление ошибок
- 6. Защита





# ОБСУЖДЕНИЕ

- Мы решили разработать игру "крестики-нолики", так это довольно простой, но в то же время очень интересный проект, так как включает в себя разные области программирования, связанные с разработкой дизайна и обеспечением функционала программы





# ИЗУЧЕНИЕ БИБЛИОТЕКИ TKINTER

tkinter это

Tkinter— это кроссплатформенная библиотека для разработки графического интерфейса на языке Python. Tkinter расшифровывается как Tk interface, и является интерфейсом к Tcl/Tk. Tkinter входит в стандартный дистрибутив Python.

## Pack Geometry Manager

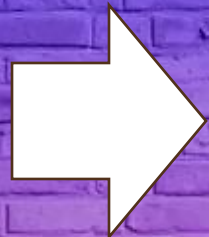
```
import tkinter as tk
root = tk.Tk()
lb1 = tk.Label(root, bg='red',text="Hello, World !")
lb2 = tk.Label(root,bg='yellow',text="Hello, World !")
lb1.pack(side="left")
lb2.pack(side="right")
root.mainloop()
```





# РАЗРАБОТКА КОДА

Импортируем библиотеку и создаем поле для нашей игры



```
import tkinter as tk
import time
```

```
window = tk.Tk()
window.geometry('600x600')
window.resizable(False, False)
window.title('хайп, а не крестики')
```

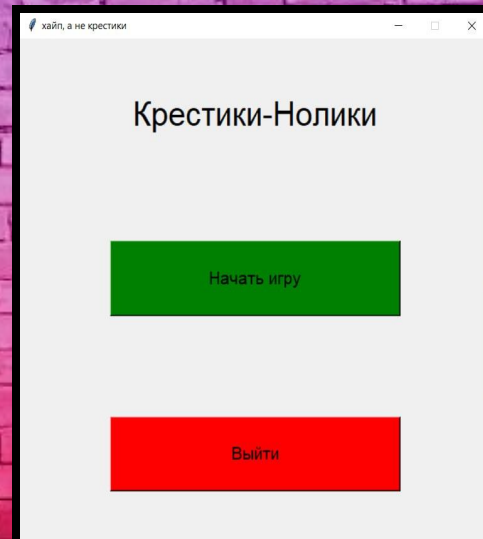
```
title_label = tk.Label(window, text="Крестики-Нолики", font=
("SANS", 30))
title_label.pack(pady=65,)
```

```
start_button = tk.Button(window, text="Начать игру",
command=start_game, width=30, height=3, font=("SANS", 16),
bg='green')
start_button.pack(pady=60)
```

```
quit_button = tk.Button(window, text="Выйти",
command=window.quit, width=30, height=3, font=("SANS", 16),
bg='red')
quit_button.pack(pady=60)
```



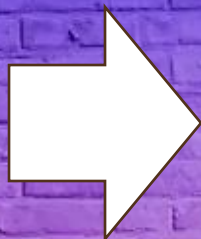
Создаем стартовое меню





# РАЗРАБОТКА КОДА

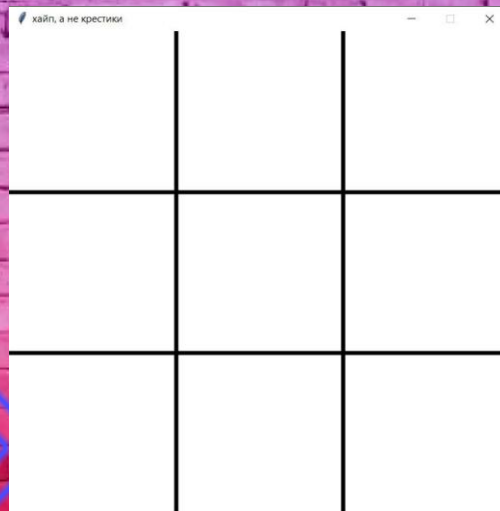
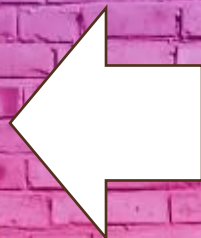
Эта функция скрывает стартовое меню и запускает сами крестики-нолики



```
def start_game():  
    draw_board()  
    start_button.pack_forget()  
    quit_button.pack_forget()  
    title_label.pack_forget()
```

Рисуем разметку крестиков-ноликов

```
c.create_line(200, 0, 200, 600, width=5)  
c.create_line(400, 0, 400, 600, width=5)  
c.create_line(0, 200, 600, 200, width=5)  
c.create_line(0, 400, 600, 400, width=5)  
window.update()
```





# РАЗРАБОТКА КОДА

Создаем квадратики (секции) в каждой ячейке поля, а затем, при нажатии срабатывает функция, описанная ниже

```
def draw_board():
    q_1 = c.create_rectangle(0, 0, 200, 200, fill='white', outline='')
    q_2 = c.create_rectangle(200, 0, 400, 200, fill='white', outline='')
    q_3 = c.create_rectangle(400, 0, 600, 200, fill='white', outline='')
    q_4 = c.create_rectangle(0, 200, 200, 400, fill='white', outline='')
    q_5 = c.create_rectangle(200, 200, 400, 400, fill='white',
    outline='')
    q_6 = c.create_rectangle(400, 200, 600, 400, fill='white',
    outline='')
    q_7 = c.create_rectangle(0, 400, 200, 600, fill='white', outline='')
    q_8 = c.create_rectangle(200, 400, 400, 600, fill='white',
    outline='')
    q_9 = c.create_rectangle(400, 400, 600, 600, fill='white',
    outline='')

```

```
c.tag_bind(q_1, "<Button-1>", lambda event: quadro(0))
c.tag_bind(q_2, "<Button-1>", lambda event: quadro(1))
c.tag_bind(q_3, "<Button-1>", lambda event: quadro(2))
c.tag_bind(q_4, "<Button-1>", lambda event: quadro(3))
c.tag_bind(q_5, "<Button-1>", lambda event: quadro(4))
c.tag_bind(q_6, "<Button-1>", lambda event: quadro(5))
c.tag_bind(q_7, "<Button-1>", lambda event: quadro(6))
c.tag_bind(q_8, "<Button-1>", lambda event: quadro(7))
c.tag_bind(q_9, "<Button-1>", lambda event: quadro(8))

```

```
def quadro(n):
    global current_player, step, play_button, goaway_button,
    game_end
    if not check_win() and board[n] not in ['X', 'O']:
        board[n] = current_player
        if current_player == 'X':
            current_player = 'O'
        else:
            current_player = 'X'
        step += 1
        update_board()

    pobeditelb = check_win()
    if pobeditelb:
        y1 = pobeditelb[0] // 3 * 200 + 100
        y2 = pobeditelb[2] // 3 * 200 + 100
        c.create_line(pobeditelb[0] % 3 * 200 + 100, y1,
        pobeditelb[2] % 3 * 200 + 100, y2, width=7)
        window.update()

    time.sleep(2)

```

Тут идет проверка  
победной комбинации  
через множество, если да,  
игра окончена и  
появляется другое меню  
спустя 2 секунды, иначе  
ставится крестик или  
нолик в зависимости от  
хода,



# РАЗРАБОТКА КОДА

Когда побеждает игрок,  
появляется полоса,  
которая зачеркивает  
победную комбинацию



```
def update_board():
    global docuvoditel
    for i in docuvoditel:
        c.delete(i)
    docuvoditel = []
    for y in range(3):
        for x in range(3):
            if board[y * 3 + x] == 'X':
                el1 = c.create_line(x * 200 + 10, y * 200 + 10, (x + 1) *
                200 - 10, (y + 1) * 200 - 10, width=5,
                fill='blue')
                el2 = c.create_line(x * 200 + 10, (y + 1) * 200 - 10, (x + 1) *
                200 - 10, y * 200 + 10, width=5,
                fill='blue')
                docuvoditel.append(el1)
                docuvoditel.append(el2)
            elif board[y * 3 + x] == 'O':
                el1 = c.create_oval(x * 200 + 10, y * 200 + 10, (x + 1) *
                200 - 10, (y + 1) * 200 - 10, width=5,
                fill='white', outline='red')
                docuvoditel.append(el1)
    window.update()
```

```
c.delete("all")
c.create_text(300, 50, text="победили " +
str(board[pobeditelb[0]]) + " !!!", anchor=tk.CENTER,
font=('SANS', 20))

play_button = tk.Button(window, text="Начать заново",
command=start_game, width=30, height=4,
font=('SANS', 16), bg='yellow')
play_button.place(x=100, y=200)

goaway_button = tk.Button(window, text="Выйти",
command>window.quit, width=30, height=4, font=('SANS', 16),
bg='red')
goaway_button.place(x=100, y=400)
game_end = True

window.update()
```



По завершению игры  
появляется следующее меню



# РАЗРАБОТКА КОДА

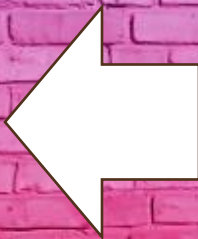
Если игрок нажимает  
"начать заново", заново  
создается игровое поле



```
def start_game():  
    global play_button, goaway_button, current_player, step, board,  
    docuvoditel  
    current_player = 'X'  
    step = 1  
    board = [1, 2, 3, 4, 5, 6, 7, 8, 9]  
    docuvoditel = []  
    c.delete("all")  
    if game_end:  
        play_button.destroy()  
        goaway_button.destroy()  
        draw_board()
```

start\_game()

Directed by  
ROBERT B. WEIDE



Если игрок нажимает  
"Выйти", то происходит  
выход из программы



# ТЕСТИРОВКА

Во время тестировки мы  
столкнулись с небольшими  
проблемами оформления  
фигур, а так всё работает  
исправно





# ВЫВОД

Подводя итоги, можно сказать, что в ходе работы над своим проектом мы изучили библиотеку Python – Tkinter.

Мы ознакомились с её методами и применили их в создании игры "Крестики-нолики".





# ИСПОЛЬЗУЕМАЯ ЛИТЕРАТУРА

<https://docs.python.org/3/library/tkinter.html>

<https://www.youtube.com/watch?v=mLySBcS-6p0&list=PLQAt0m1f90Hsd6U5okp1XLoYyQR0oBjMM>

<https://python-scripts.com/tkinter?ysclid=lxanemxjla491778415>