ЛАБОРАТОРНАЯ РАБОТА «СОЗДАНИЕ ИГРЫ НА PYTHON»

Прочитайте о классе Canvas:

https://younglinux.info/tkinter/canvas

https://metanit.com/python/tkinter/7.1.php

Задание 1. Нарисуйте с помощью **Canvas несколько несложных фигур: круг, квадрат, треугольник**

Задание 2. Выведите с помощью **Canvas время работы окна в секундах. Попробуйте настроить автоматическое закрытие окна через 10 секунд**

Тут можно использовать модуль time

```
import time
```

```
start_time = time.time()
time1 = round(time.time()-start_time)
```

Задание 3.

Запустите игру, код которой приведен ниже. Доработайте игру так, чтобы помимо попаданий вычислялись промахи.

Выведите в окне игры время игры в секундах.

Придумайте правила игры: когда игра может закончится? Реализуйте свою идею.

```
from random import randrange as rnd, choice
from tkinter import *
from tkinter import messagebox
import time

root = Tk()
root.geometry('800x600')

canv = Canvas(root, bg='white')
canv.pack(fill=BOTH, expand=1)

colors = ['red', 'orange', 'yellow', 'green', 'blue']
k = 0

def new_ball():
    global x, y, r
    canv.delete(ALL)
    x = rnd(100, 700)
    y = rnd(100, 500)
    r = rnd(30, 50)
    canv.create_text(180, 20, text=f'Попаданий: {str(k)}', font='Arial 20')
    canv.create_oval(x - r, y - r, x + r, y + r, fill=choice(colors), width=1)
    root.after(1000, new_ball)

def click(event):
    global k, promah
    if (event.y - y) ** 2 + (event.x - x) ** 2 <= r ** 2:
        k += 1
```

```
canv.bind('<Button-1>', click)
new_ball()

mainloop()
```

Задание 4*. Запустите игру. Постарайтесь доработать игру, добавив таймер и окончание игры. Измените фон, найдите свои файлы для фона, персонажа и т.д.

```
from tkinter import *

from random import randrange as rnd, choice
import time

root = Tk()
canvas = Canvas(root, width=500, height=500, bg="white")
canvas.pack()
x = 100
y = 350


label1 = Label(root, justify='center')
label1.pack()

label2 = Label(root, justify='left')
label2.pack()


fon = PhotoImage(file="fon.png")  # Вставляем фон
id_fon = canvas.create_image(0, 0, anchor=NW, image=fon)  # 0, 0 - координаты
левого верхнего угла картинки

fish_2 = PhotoImage(file="2fish.png")  # Вставляем персонажа игры
id_fish = canvas.create_image(10, 10, anchor=NW, image=fish_2)

fish_o = PhotoImage(file="fish.png")  # подготовить файл с изображением
id_img = canvas.create_image(x, y, anchor=NW, image=fish_o)

k = 0


def new_fish():
    global x, y, k

    x = rnd(0, 450)
    y = rnd(0, 10)
    id_fish = canvas.create_image(x, y, anchor=NW, image=fish_2)
    k2 = 500 - 200


    for _ in range(550):
        canvas.move(id_fish, 0, 0.005 * k2)
        canvas.update()
        time.sleep(0.01)
        loc_fihs = canvas.coords(id_fish)
        location = canvas.coords(id_img)
        if (loc_fihs[0] - location[0]) ** 2 + (loc_fihs[1] - location[1]) ** 2
< 1000:
            print('see on fish')
            k += 1
            m = str(k) + " fish"
            #canvas.delete(id_text)
            #id_text = canvas.create_text(40, 20, text=str(k) + ' fish',
font='Arial 20')
```

```
            label2.config(text=m, font='times 25')
            canvas.delete(id_fish)
            break

    root.after(10, new_fish)


for i in range(100):
    canvas.move(id_fish, 0, 100)


def move_pers(event):
    # global k, promah

    location = canvas.coords(id_img)
    if event.keysym == 'Up' and location[1] > 0:
        canvas.move(id_img, 0, -2)

    elif event.keysym == 'Down' and location[1] < 350:
        canvas.move(id_img, 0, 2)
    elif event.keysym == 'Left' and location[0] > 0:
        canvas.move(id_img, -4, 0)
    elif event.keysym == 'Right' and location[0] < 420:
        canvas.move(id_img, 4, 0)


canvas.bind_all("<KeyPress-Up>", move_pers)
canvas.bind_all("<KeyPress-Down>", move_pers)
canvas.bind_all("<KeyPress-Left>", move_pers)
canvas.bind_all("<KeyPress-Right>", move_pers)

new_fish()




root.mainloop()
```