

**Московский
государственный
технический университет
им. Н.Э. Баумана**

**Факультет «Информатика и
системы управления»
Кафедра ИУ5 «Системы обработки
информации и управления»**

**Курс «Парадигмы и конструкции языков
программирования» Отчет по
Лабораторной работе №4**

Выполнил:
студент группы ИУ5-34Б
Файзуллин Камиль
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.
Подпись и дата:

Москва, 2023 г.

Задание:

Игра “Змейка”

Код программы:

```
from tkinter import *
import time
import random

f_width = 500
f_height = 500
snake_item = 20
snake_color1 = "black"
snake_color2 = "red"

f_vision_x = f_width//snake_item
f_vision_y = f_height//snake_item

snake_x=f_vision_x//2
snake_y=f_vision_y//2
snake_x_navigat = 0
snake_y_navigat = 0

game_run = True

snake_list = []
snake_size = 5

tk = Tk()
tk.title("Змейка на Python")
tk.resizable(0, 0)
tk.wm_attributes("-topmost", 1)

field = Canvas(tk, width=f_width, height=f_height, bd=0,
highlightthickness=0)
field.pack()
tk.update()

def snake_print_item(field, x, y):
    global snake_list
    id1 = field.create_rectangle(x*snake_item, y*snake_item,
x*snake_item+snake_item, y*snake_item+snake_item, fill = snake_color1)
    id2 = field.create_rectangle(x*snake_item+2, y*snake_item+2,
x*snake_item+snake_item-2, y*snake_item+snake_item-2, fill = snake_color2)
    snake_list.append([x, y, id1, id2])
    print(snake_list)

snake_print_item(field, snake_x, snake_y)

def check():
    if len(snake_list) >= snake_size:
        temp = snake_list.pop(0)
        print(temp)
        field.delete(temp[2])
        field.delete(temp[3])

def check_food():
    global snake_size
    for i in range(len(food_list)):
```

```

        if food_list[i][0] == snake_x and food_list[i][1] == snake_y:
            #print("eat")
            snake_size += 1
            field.delete(food_list[i][2])
            field.delete(food_list[i][3])

food_list = []
food_count = 15
food_color1 = "yellow"
food_color2 = "green"
for i in range(food_count):
    temp_food_x = random.randrange(f_vision_x)
    temp_food_y = random.randrange(f_vision_y)
    id1 = field.create_oval(temp_food_x*snake_item, temp_food_y*snake_item,
temp_food_x*snake_item+snake_item, temp_food_y*snake_item+snake_item, fill =
food_color1)
    id2 = field.create_oval(temp_food_x*snake_item+2,
temp_food_y*snake_item+2, temp_food_x*snake_item+snake_item-2,
temp_food_y*snake_item+snake_item-2, fill = food_color2)
    food_list.append([temp_food_x, temp_food_y, id1, id2])

def snake_move(do):
    global snake_x
    global snake_y
    global snake_x_navigat
    global snake_y_navigat
    if do.keysym == "Left":
        snake_x_navigat = -1
        snake_y_navigat = 0
        check()

    elif do.keysym == "Right":
        snake_x_navigat = 1
        snake_y_navigat = 0
        check()

    elif do.keysym == "Up":
        snake_x_navigat = 0
        snake_y_navigat = -1
        check()

    elif do.keysym == "Down":
        snake_x_navigat = 0
        snake_y_navigat = 1
        check()

    snake_x += snake_x_navigat
    snake_y += snake_y_navigat
    snake_print_item(field, snake_x, snake_y)
    check_food()

field.bind_all("<KeyPress-Left>", snake_move)
field.bind_all("<KeyPress-Right>", snake_move)
field.bind_all("<KeyPress-Up>", snake_move)
field.bind_all("<KeyPress-Down>", snake_move)

def loos():
    global game_run
    game_run = False

def check_bd():
    if snake_x>=f_vision_x or snake_y>=f_vision_y:
        loos()

```

```

def check_myself(f_x, f_y):
    global game_run
    if not(snake_x_navigat == 0 and snake_y_navigat == 0):
        for i in range(len(snake_list)):
            if snake_list[i][0] == f_x and snake_list[i][1] == f_y:
                print("myself")
                game_run = False

while game_run:
    check()
    check_food()
    check_bd()
    check_myself(snake_x_navigat+snake_x, snake_y+snake_y_navigat)
    snake_x += snake_x_navigat
    snake_y += snake_y_navigat
    snake_print_item(field, snake_x, snake_y)
    tk.update_idletasks()
    tk.update()
    time.sleep(0.1)

def stop():
    pass
field.bind_all("<KeyPress-Left>", stop)
field.bind_all("<KeyPress-Right>", stop)
field.bind_all("<KeyPress-Up>", stop)
field.bind_all("<KeyPress-Down>", stop)

mainloop()

```

Результат:



