

Урок 29. Гра “Морський бій”. Етап 1

Вивчення нового матеріалу

Слайд № 1

Сьогодні ми розпочнемо створювати справжню комп’ютерну гру “Морський бій”.

Правила гри

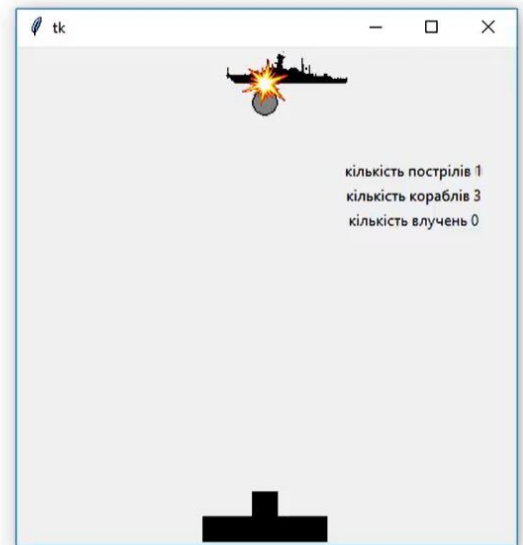
Посередині нижньої межі вікна розміщено гармату, яка по натисканні клавіші **Пробіл** стріляє снарядом вертикально вгору.

Уздовж верхньої межі екрана справа наліво рухається корабель.

Коли або снаряд влучить у корабель, або корабель допливе до лівої межі екрана, впливає наступний корабель.

Протягом гри пропливає **10 кораблів**, а гравцеві дається **10 снарядів**.

Метою є підбити якомога більше кораблів.



Створимо справжню комп’ютерну гру!

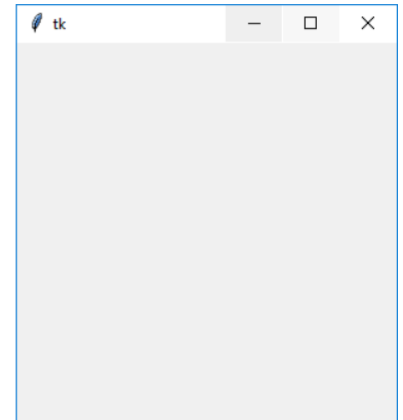
Розпочнемо зі створення вікна програми та полотна, на якому можна буде виводити всі потрібні зображення.

Вправи

Вправа № 1



Вправа 1. У новому файлі уведіть код створення вікна програми та розміщення в ньому полотна для малювання розміром **500x500**.



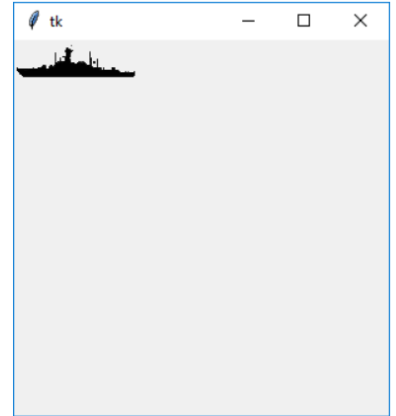
У середовищі Python уведіть код створення вікна програми та розміщення в ньому полотна для малювання.

Вправа № 2



Вправа 2. Додайте до коду програми команди виведення зображення на полотні. Завантажте файл зображення з онлайн-підручника.

Запустіть програму та переконайтеся, що зображення виведено згідно зразка.



```
ship_image = PhotoImage(file = "ship.gif")  
s = canvas_create_image(0, 0, anchor = NW, image = ship_image)
```

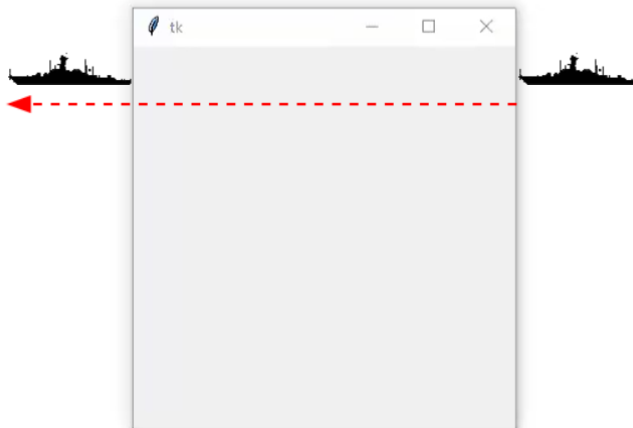


Для відображення зображення на полотні додайте вказаний код у середовищі Python.

Вивчення нового матеріалу

Слайд № 2

Тепер змінимо код так, щоб корабель рухався з крайнього правого положення (з-за меж вікна) вліво до повного зникнення з вікна.



Створимо анімацію руху корабля.

Слайд № 3

Для переміщення корабля через кожні 0.02 сек. застосуємо такий код:

Модуль використання часу

canvas.move(s,dx,dy) — функція переміщення графічного об'єкта **s** на **dx** по горизонталі та **dy** по вертикалі. У нашому випадку **s** — це корабель.

Команда примусового оновлення (перемальовки) зображення на екрані. Без цієї команди зображення не буде змінюватися.

import time

```
from tkinter import *
tk=Tk()
canvas=Canvas(tk, width=500, height=500)
canvas.pack()
ship_image=PhotoImage(file='ship.gif')
s=canvas.create_image(500, 0, anchor=NW, image=ship_image)
```

for y in range (200):

canvas.move(s,-3,0)

tk.update()

time.sleep(0.02)

Затримка виконання програми на вказану кількість секунд



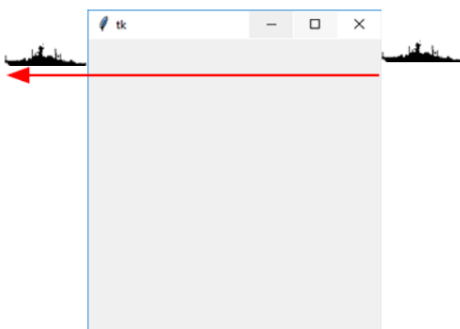
Запам'ятайте команди для руху зображення.

Вправа

Вправа № 3



Вправа 3. Додайте код для руху корабля. Запустіть програму на виконання. У результаті корабель має рухатися від правого до лівого краю вікна.



import time

```
from tkinter import *
tk=Tk()
canvas=Canvas(tk, width=500, height=500)
canvas.pack()
ship_image=PhotoImage(file='ship.gif')
s=canvas.create_image(500, 0, anchor=NW, image=ship_image)
```

for y in range (200):

canvas.move(s,-3,0)

tk.update()

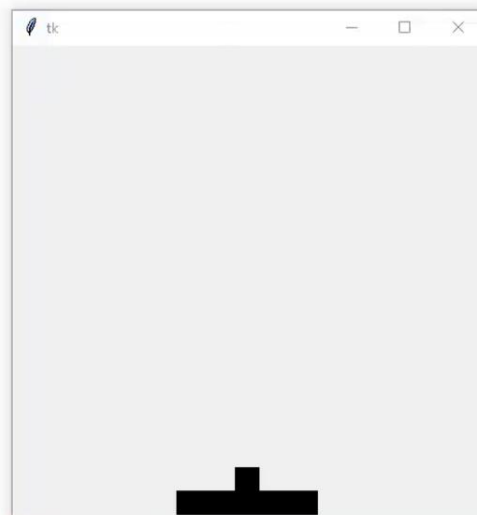
time.sleep(0.02)



Для відтворення руху корабля додайте вказаний код у середовищі Python.

Слайд № 4

Тепер намалюємо гармату і ядро, а також створимо код, завдяки якому ядро буде рухатися, коли натискають клавішу "Пробіл".



Гармату намалюємо на цьому уроці, а ядро та його рух зімітуємо на наступному.

Слайд № 5

Із наведеного зображення видно, що гармата складається з двох прямокутників.

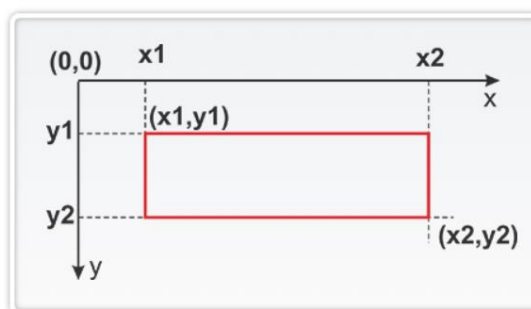


Для побудови прямокутника в Python використовують таку команду:

```
canvas.create_rectangle(x1,y1,x2,y2,fill=color)
```

x1,y1 та **x2,y2** — координати верхнього лівого та нижнього правого кутів відповідно.
fill=color — колір заливки прямокутника.

Замість слова **color** вказують назву кольору в лапках, наприклад **'black'**.



Запам'ятайте команду побудови прямокутника.



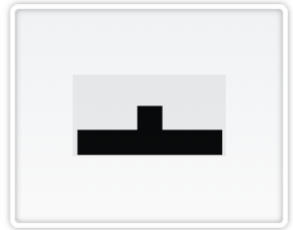
Вправа 4. Додайте команди малювання гармати.

#низ гармати

```
canvas.create_rectangle(190, 480, 310, 500, fill = 'black')
```

дуло гармати

```
canvas.create_rectangle(240, 460, 260, 480, fill = 'black')
```



Запустіть програму та перевірте правильність побудови зображення.