

Урок 32. Гра “Морський бій”. Етап 4

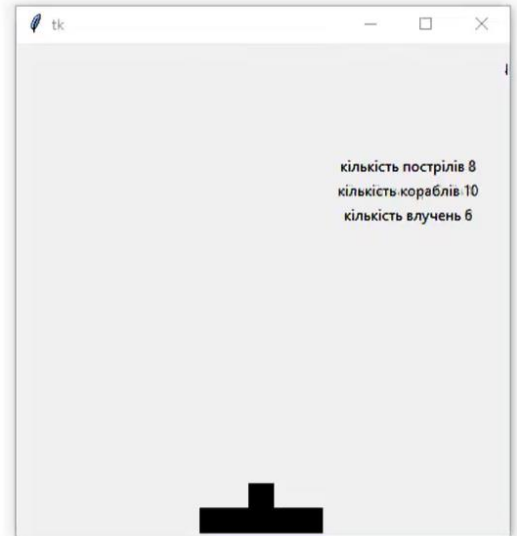
Вивчення нового матеріалу

Слайд № 1

Сьогодні ми виконаємо такі завдання:

- 1) відобразимо кількість кораблів, що пройшли;
- 2) відтворимо момент зіткнення корабля і ядра;
- 3) підрахуємо кількість таких зіткнень.

В момент зіткнення буде з'являтися зображення вибуху і через деякий час корабель та ядро зникатимуть.



Продовжимо створювати комп'ютерну гру “Морський бій”.

Вправа

Вправа № 1



Вправа 1. Додайте до програми команди виведення повідомлення про кількість переміщених кораблів.

Запустіть програму і перевірте правильність її виконання.

```
ship_image=PhotoImage(file='ship.gif')
for z in range(10):
    s=canvas.create_image(500, 0,anchor=NW, image=ship_image)
    v=random.randint(2,5)
    canvas.create_text(420, 120, text='кількість кораблів'+str(z), fill='#F0F0ED')
    canvas.create_text(420, 120, text='кількість кораблів'+str(z+1))
    for y in range(300):
        canvas.move(s,-v,0)
        canvas.move(b,0,-5)
    tk.update()
    time.sleep(0.02)
```



Додайте команди виведення повідомлення про кількість кораблів.

Слайд
№ 2

Зверніть увагу на використання в коді оператора **break**. Він примусово завершує виконання циклу. У нашому випадку, якщо ядро та корабель зіткнулися, їх рух припиняється.

примусова зупинка
роботи циклу

```
if 140<xs<260 and 0<yb<30:
    # зображення вибуху
    pow_image=PhotoImage(file= 'pow.gif' )
    p=canvas.create_image(230,10,anchor=NW,image=pow_image)
    tk.update()
    time.sleep(2)
    canvas.delete(b)
    canvas.delete(p)
    break
```

видалення ядра
видалення вибуху
завершення руху корабля та ядра



Оператор **break** зупиняє найбільш внутрішній цикл, у якому він розташований.

Слайд
№ 4

Отже, для відображення вибуху потрібно:

1. Проаналізувати координати корабля та ядра й відобразити вибух.

```
if 140<xs<260 and 0<yb<30:
    # зображення вибуху
    pow_image=PhotoImage(file= 'pow.gif' )
    p=canvas.create_image(230,10,anchor=NW,image=pow_image)
    tk.update()
    time.sleep(2)
    canvas.delete(b)
    canvas.delete(p)
    break
```

видалення ядра
видалення вибуху
завершення руху корабля та ядра

2. У циклі **for z in range(10)**: присвоїти змінній **xs** початкове значення **500**.

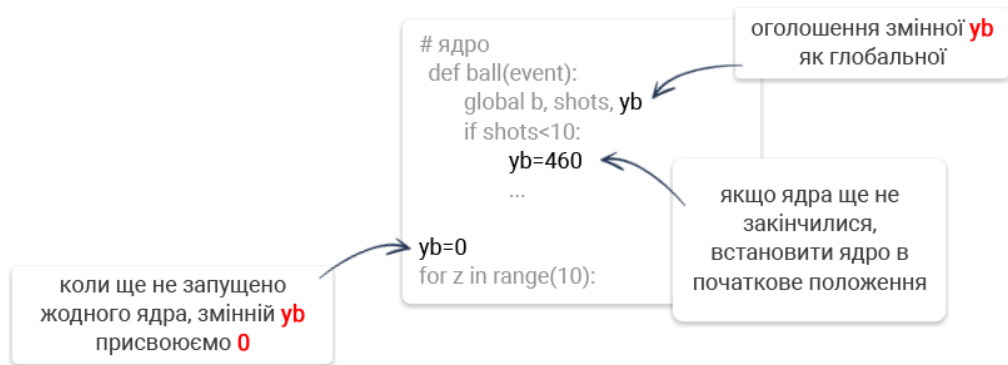


Оператор **break** зупиняє найбільш внутрішній цикл, у якому він розташований.

Далі

Слайд
№ 5

3. Оголосити змінну **yb** у функції **ball** як глобальну та присвоїти їй початкове значення **460** у функції **ball** та значення **0** на початку основної програми.



Присвоєння **yb=460** імітує запуск ядра і тому його слід виконувати тільки коли натиснуто «пробіл» і ядра ще не закінчилися.

Вправи

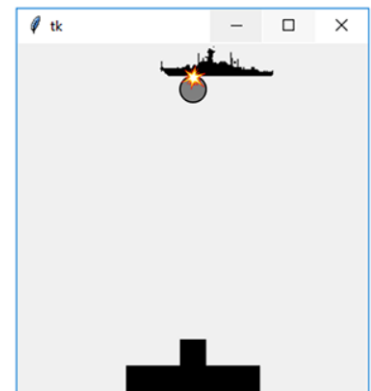
Вправа № 2



Вправа 2. Додайте до програми код, що враховує та відображає момент зіткнення корабля та ядра.

Завантажте з електронного підручника файл **pow.gif** до папки з програмою.

Запустіть програму та перевірте правильність її виконання.



Додайте команди, що відображають момент зіткнення.

Вправа № 3



Вправа 3. Додайте команди виведення повідомлення про кількість влучень.

Запустіть програму та перевірте правильність її виконання.

```
k=0
canvas.create_text(420, 140, text='кількість влучень '+str(k))
for z in range(10):
    ...
    if 140<xs<260 and 0<yb<30:
        pow_image=PhotoImage(file='pow.gif')
        p=canvas.create_image(230, 10, anchor=NW, image=pow_image)
        tk.update()
        time.sleep(2)
        canvas.delete(b)
        canvas.delete(p)
        k=k+1
        canvas.create_text(420, 140, text='кількість влучень '+str(k-1), fill='F0F0ED')
        canvas.create_text(420, 140, text='кількість влучень '+str(k))
        break
```



Додайте команди виведення повідомлення про кількість влучень.