

Урок 38. Гра “Морський бій”. Етап 4

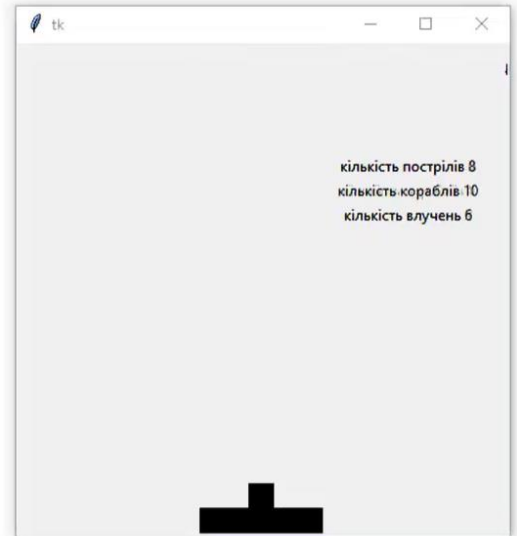
Вивчення нового матеріалу

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.

```
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)          # видалення ядра
    canvas.delete(p)          # видалення вибуху
    canvas.delete(s)          # видалення корабля
    break                     # завершення руху корабля та ядра
```

примусова зупинка
виконання циклу

Слайд
№ 3

Ще потрібно оголосити змінну **yb** у функції **ball** як глобальну, оскільки у цій функції та в основній програмі це має бути та сама змінна.

```
# ядро
def ball(event):
    global b, shots, yb
    if shots<10:
        yb=460
        ...
    for z in range(10):
```

оголошення змінної **yb**
як глобальної

якщо ядра ще не
закінчилися,
встановити ядро в
початкове положення



Якщо цього не зробити, то присвоєння **yb=460** не матиме ніякого впливу на основну програму.

Слайд
№ 4

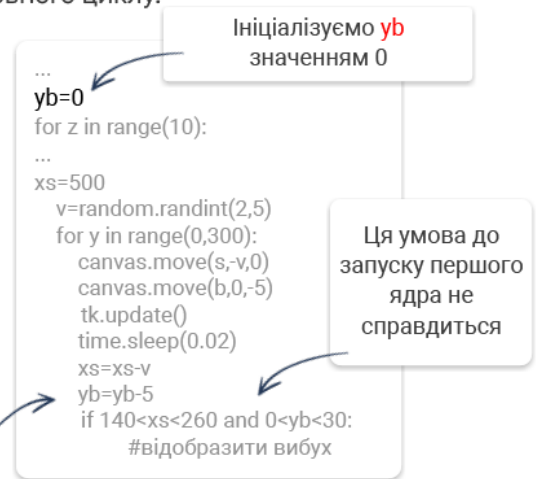
Зауважте, що присвоєння **yb=yb-5** в нашій програмі виконуватиметься ще до того, як буде запущено перше ядро і змінна **yb** буде ініціалізована значенням **460**. Отже, змінну **yb** треба ініціалізувати також перед початком основного циклу.

На початку програми змінній **yb** можна присвоїти **0**.

Тоді до запуску першого ядра **yb** набуватиме від'ємних значень, умова

0<yb<30

не справджуватиметься і не відбудеться «віртуального влучення» ще не існуючого ядра в корабель.



Якщо не зробити цього присвоєння, то під час виконання програми виникне помилка «Спроба використати невизначену змінну **yb** в команді **yb=yb-5**».

Слайд
№ 5

Отже, для відображення вибуху потрібно:

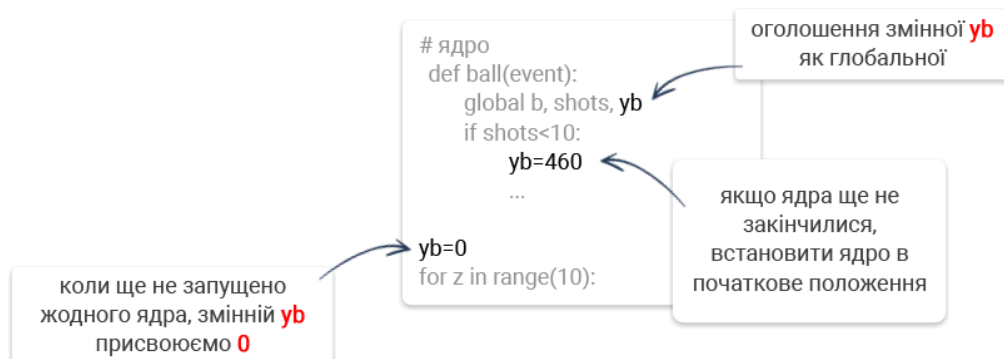
1. Проаналізувати координати корабля та ядра й відобразити вибух.

```
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)          # видалення ядра
    canvas.delete(p)          # видалення вибуху
    canvas.delete(s)          # видалення корабля
    break                    # завершення руху корабля та ядра
```

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

Слайд
№ 6

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.



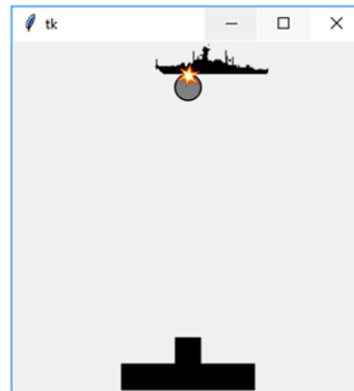
Присвоєння **yb=460** імітує запуск ядра і тому його слід виконувати тільки коли натиснуто «пробіл» і ядра ще не закінчилися.

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)
        canvas.delete(s)
        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.

Готово