Урок 36. Гра “Морський бій”. Етап 2

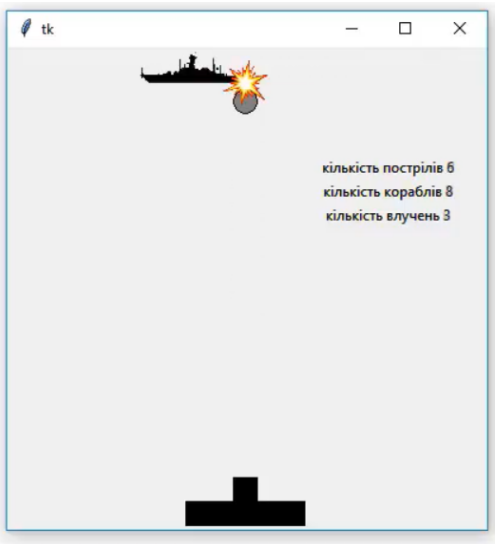
Вивчення нового матеріалу

Слайд № 1

Продовжимо створювати гру “Морський бій”.

Сьогодні ми запрограмуємо рух снаряду. Він має вилітати з гармати по натисканні клавіші пробіл і рухатися вертикально вгору.

Оскільки снаряд має круглу форму, назвемо його ядром.



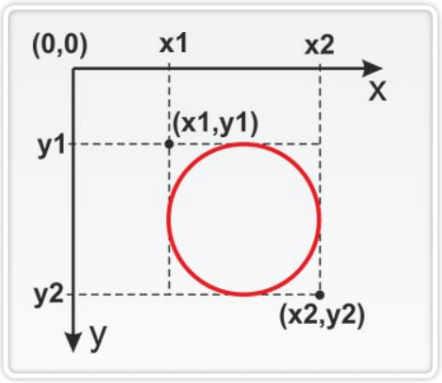
Корабель у нас вже пливе — тепер запрограмуємо постріл.

Слайд № 2

Команда побудови круга в Python:

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

x1,y1 та x2,y2 — координати верхнього лівого та нижнього правого кутів квадрата, описаного навколо круга.



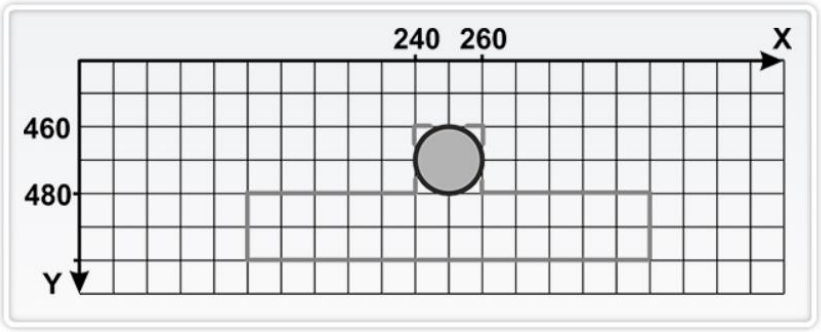
Замість слова color записують назву кольору заливки.

Вправа

Вправа № 1	

Вправа 1. Додайте до програми команду малювання ядра.

```
canvas.create_oval(240, 460, 260, 480, fill = 'gray')
```



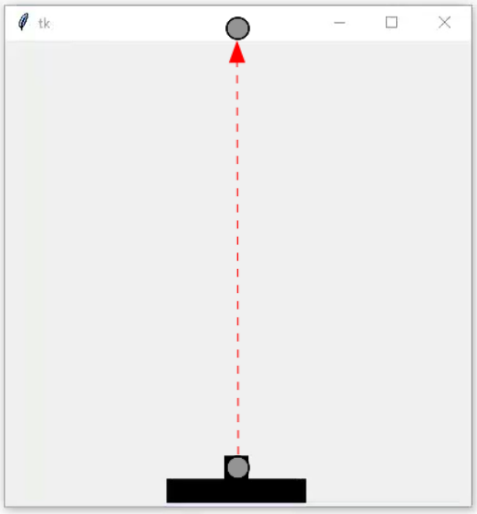
Запустіть програму та перевірте правильність побудови зображення.

Вивчення нового матеріалу

Слайд № 3	

Додамо до програми команди руху ядра від гармати і аж до виходу за верхній край вікна.

Ядро має рухатися одночасно з кораблем.



Додамо рух ядра.

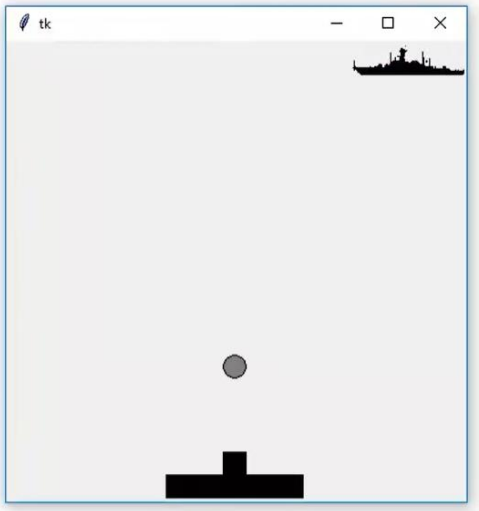
Вправа

Вправа № 2

Вправа 2. Доповніть код командами руху ядра.

```
...
ship_image=PhotoImage(file='ship.gif')
s=canvas.create_image(500, 0, anchor=NW, image=ship_image)
```

```
b=canvas.create_oval(240,460,260,480,fill='gray')
for y in range (200):
    canvas.move(s,-3,0)
    canvas.move(b,0,-5)
    tk.update()
    time.sleep(0.02)
```



Запустіть програму та перевірте правильність її виконання.

Вивчення нового матеріалу

Слайд № 4

Однак нам потрібно, щоб ядро рухалося не само по собі, а коли натиснуто клавішу **Пробіл**. Для цього до події натискання пробілу ми маємо прив'язати деяку **функцію**.

Ось як це зробити.

Сама функція

```
def ім'я функції(event):
    тіло функції
```

довільне ім'я

Прив'язка функції

```
canvas.bind_all(подія, ім'я функції)
```

команда прив'язки

Зауважте, що до подій натискання клавіш функції прив'язують не так, як до подій елементів керування.

Слайд № 5

У нашому випадку у функції може створюватися об'єкт ядра:

```
def ball(event):
    b=canvas.create_oval(240, 460, 260, 480, fill='gray')
```

А ось прив'язка функції до події натискання клавіші **Пробіл**:

У нашому випадку назва клавіші – space (пробіл).
Її треба взяти в гострі дужки та в лапки: '<space>'.

Вправа

Вправа № 3

Вправа 3. Змініть код малювання ядра, перетворивши його на функцію. Додайте код прив'язки цієї функції до події натискання клавіші пробіл.

```
...
ship_image=PhotoImage(file='ship.gif')
s=canvas.create_image(500, 0, anchor=NW, image=ship_image)
```

```
def ball(event):
    b=canvas.create_oval(240,460,260,480,fill='gray')
canvas.bind_all('<space>', ball)
```

```
for y in range (200):
    canvas.move(s,-3,0)
    canvas.move(b,0,-5)
    tk.update()
    time.sleep(0.02)
```

Перевірте, чи працює програма.

Вивчення нового матеріалу

Слайд № 6

Отже, змінна **b**, якій присвоюється об'єкт ядра під час виконання функції, є **локальною**, тобто існує лише в тілі функції.

Змінна **b**, яка використовується в тілі програми, є **глобальною** і її значення в даному випадку є невизначеним.

локальна змінна

```
def ball(event):
    b=canvas.create_oval(240,460,260,480,fill='gray')
canvas.bind_all('<space>', ball)
```

```
for y in range (200):
    canvas.move(s,-3,0)
    canvas.move(b,0,-5)
    tk.update()
    time.sleep(0.02)
```

глобальна змінна

Змінні, що оголошені в тілі функції, — локальні, в основній програмі — глобальні.

Слайд № 7

Щоб змінна **b** у функції була глобальною, це необхідно вказати після заголовку функції.

Крім того, оскільки команда **canvas.move(b,0,-5)** може виконуватися ще до того, як створено ядро, змінну **b** потрібно ініціалізувати, тобто присвоїти їй якесь початкове значення.

ініціалізація змінної

оголошення змінної в межах функції як **глобальної**

```
b=0
def ball(event):
    global b
    b=canvas.create_oval(240,460,260,480,fill='gray')
canvas.bind_all('<space>', ball)
for y in range (200):
    canvas.move(s,-3,0)
    canvas.move(b,0,-5)
    tk.update()
    time.sleep(0.02)
```

Зауважте, що в програмі необхідно встановити початкове значення змінної **b**.

Вправа

Вправа № 4

Вправа 4. Додайте до програми команди ініціалізації змінної b та оголошення її як глобальної.

```
...
ship_image=PhotoImage(file='ship.gif')
s=canvas.create_image(500, 0, anchor=NW,
```

```
b=0
def ball(event):
    global b
    b=canvas.create_oval(240,460,260,480,fill='gray')
canvas.bind_all('<space>', ball)
for y in range (200):
    canvas.move(s,-3,0)
    canvas.move(b,0,-5)
    tk.update()
    time.sleep(0.02)
```

Запустіть програму та натисніть клавішу **Пробіл** декілька разів.

Вивчення нового матеріалу

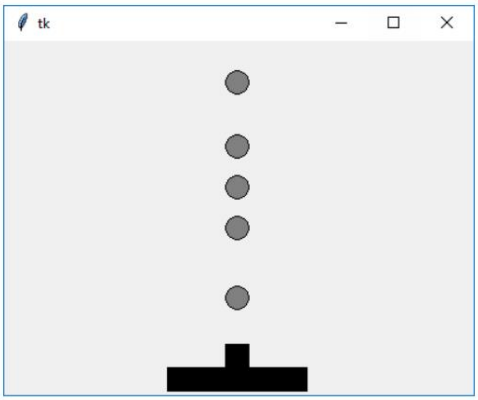
Слайд № 8

Мабуть, ви помітили, що з ядром ще не все гаразд. Коли вилітає наступне ядро, попереднє зупиняється та залишається на екрані.

Отже, під час запуску ядра попереднє ядро потрібно видаляти.

Для вилучення об'єкта з полотна використовують команду

```
canvas.delete(об'єкт)
```



Об'єкт не видаляється зовсім, однак зникає його зображення на полотні.

Вправа

Вправа № 5	**Вправа 5. Додайте до програми команду вилучення об'єкта b.**
	Запустіть програму та натисніть клавішу **пробіл** декілька разів, спостерігаючи за рухом ядра.
	`b=0`
	`def ball(event):`
	`  global b`
	`  canvas.delete(b)`
	`  b=canvas.create_oval(240,460,260,480,fill='gray')`
	`canvas.bind_all('<space>', ball)`
	`for y in range (200):`
	`  canvas.move(s,-3,0)`
	`  canvas.move(b,0,-5)`
	`  tk.update()`
	 
	Тепер програма має працювати, як треба.