



Lecția 8 – Python Games

Trainer: Hrișcă Miruna

Aplicații folosite + alternative ▾

01

PyCharm

[Link - windows](#)

[Link - Apple](#)

02

One Compiler

[Link](#)

03

VS Code

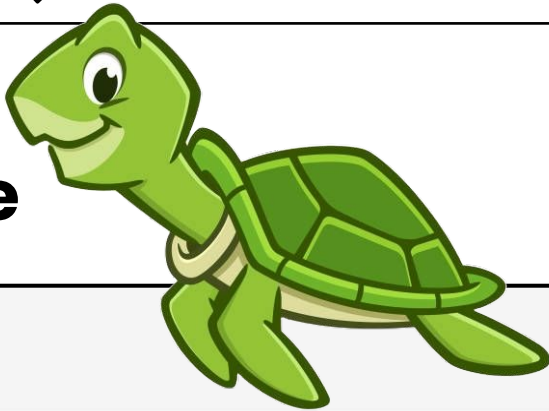
[Link](#)

04

Online GDB

[Link](#)

Funcții Turtle



main.py



```
1 clear() #șterge tot de pe ecran
2 goto(x, y) #merge la coordonatele specificate
3 dot(mărime, culoare) #desenează un punct colorat
4 update() #actualizează ecranul
5 onclick(funcție) #detectează clicurile pe ecran
6 ontimer(funcție, milisecunde) #apelează o funcție după un timp
```

Biblioteca Random



❖ **Random** ne ajută să generăm numere întâmplătoare în jocuri.

❖ **Funcții importante:**

- `randrange(start, stop)` - generează un număr întreg aleatoriu într-un interval
- Exemplu: `randrange(1, 10)` poate da: 1, 2, 3, 4, 5, 6, 7, 8 sau 9



Biblioteca FreeGames – Clasa Vector



Ce este un vector?



- ❖ O pereche de numere (x, y) care spune poziția pe ecran.
- ❖ Exemplu: `vector(0, 0)` înseamnă punctul din mijlocul ecranului.
- ❖ **Vector** este ca un "punct inteligent" care știe să se miște pe ecran.



main.py



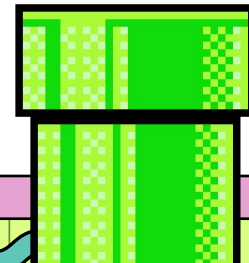
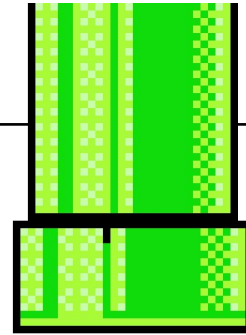
```
1 from freegames import vector
2
3 # Creez un punct la coordonatele (50, 100)
4 punct = vector(50, 100)
5
6 # Pot accesa coordonatele
7 print(punct.x) # afișează 50
8 print(punct.y) # afișează 100
9
10 # Pot muta punctul
11 miscare = vector(10, 20)
12 punct.move(miscare) # punctul se mută cu 10 pe x și 20 pe y
```



Despre jocul Flappy

Flappy este inspirat din celebrul joc Flappy Bird:

- ❖ **Pasărea verde** cade constant din cauza gravitației
- ❖ **Click pe ecran** = pasărea sare în sus
- ❖ **Bilele negre** vin din dreapta - trebuie evitate!
- ❖ **Obiectiv:** Supraviețuiește cât mai mult timp!



Programarea jocului

Pasul 1: Importurile

- ❖ Importăm toate funcțiile din biblioteca Turtle și Random
- ❖ Importăm funcția `vector` din FreeGames

```
from random import *  
from turtle import *  
from freegames import vector
```

Pasul 2: Variabilele jocului

- ❖ `bird` → pasărea noastră, la început în mijlocul ecranului (0, 0).
- ❖ `balls` → lista care va ține minte corbii negri (la început nu există niciunul).

```
bird = vector(0, 0)  
balls = []
```

Programarea jocului

Pasul 3: Funcția `tap(x, y)`

- ❖ Când faci clic pe ecran, această funcție se apelează.
- ❖ Creează o mișcare de 30 pixeli în sus (0 pe x, 30 pe y).
- ❖ Pasărea "sare" în sus cu 30 pixeli.



```
def tap(x, y):  
    up = vector(0, 30)  
    bird.move(up)
```

Pasul 4: Funcția `inside(point)`

Ce verifică:

- ❖ Dacă un punct (pasăre sau bilă) este încă pe ecran.
- ❖ Ecranul nostru merge de la -200 la +200 pe ambele axe.
- ❖ Returnează `True` dacă punctul e pe ecran, `False` dacă a ieșit.

```
def inside(point):  
    return -200 < point.x < 200 and -200 <  
    point.y < 200
```



Programarea jocului

Pasul 5: Funcția draw() - Desenarea jocului



```
❖ def draw(alive):
❖     clear()
❖     goto(bird.x, bird.y)
❖     ➤ Șterge ecranul (clear).
❖     ➤ Mută „creionul” la poziția păsării.
❖     if alive:
❖         dot(10, 'green')
❖     else:
❖         dot(10, 'red')
❖     ➤ Desenează pasărea:
❖         ○ verde dacă e vie,
❖         ○ roșie dacă a murit.
❖     for ball in balls:
❖         goto(ball.x, ball.y)
❖         dot(20, 'black')
❖     ➤ Desenează toți corbii negri (din lista balls).
❖     update() → Actualizează ecranul.
```

```
def draw(alive):
    """Draw screen objects."""
    clear()

    goto(bird.x, bird.y)

    if alive:
        dot(10, 'green')
    else:
        dot(10, 'red')

    for ball in balls:
        goto(ball.x, ball.y)
        dot(20, 'black')

    update()
```

Programarea jocului

Pasul 6: Funcția move() - Motorul jocului

Partea 1: Gravitația

`bird.y -= 5` # Pasărea cade cu 5 pixeli în jos



Partea 2: Mișcarea bilelor

`for ball in balls:`

`ball.x -= 3` # Fiecare bilă se mută cu 3 pixeli spre stânga



Partea 3: Generarea bilelor noi

`if randrange(10) == 0:` # 1 din 10 șanse să apară o bilă



`y = randrange(-199, 199)` # Alege o poziție aleatoare pe y

`ball = vector(199, y)` # Creează bila la dreapta ecranului

`balls.append(ball)` # Adaugă bila în listă

```
def move():
    """Update object positions."""
    bird.y -= 5
    for ball in balls:
        ball.x -= 3
    if randrange(10) == 0:
        y = randrange(-199, 199)
        ball = vector(199, y)
        balls.append(ball)
    while len(balls) > 0 and not
inside(balls[0]):
        balls.pop(0)
    if not inside(bird):
        draw(False)
        return
    for ball in balls:
        if abs(ball - bird) < 15:
            draw(False)
            return

    draw(True)
    ontimer(move, 50)
```

Programarea jocului

Pasul 6: Funcția move() - Motorul jocului

Partea 4: Curățenia - elimină bilele ieșite

```
while len(balls) > 0 and not inside(balls[0]):  
    balls.pop(0) # Elimină prima bilă dacă a ieșit din ecran
```



Partea 5: Verificări finale

```
# Verifică dacă pasărea a ieșit din ecran  
if not inside(bird):  
    draw(False) # Desenează pasărea roșie (moartă)  
    return # Oprește jocul  
  
# Verifică coliziunile cu bilele  
for ball in balls:  
    if abs(ball - bird) < 15: # Dacă distanța e mai mică de 15 pixeli  
        draw(False) # Pasărea moare  
        return # Oprește jocul  
  
# Dacă pasărea supraviețuiește  
draw(True) # Desenează totul (pasărea verde)  
ontimer(move, 50) # Apelează din nou move() după 50ms
```

```
def move():  
    """Update object positions."""  
    bird.y -= 5  
    for ball in balls:  
        ball.x -= 3  
    if randrange(10) == 0:  
        y = randrange(-199, 199)  
        ball = vector(199, y)  
        balls.append(ball)  
    while len(balls) > 0 and not  
inside(balls[0]):  
        balls.pop(0)  
    if not inside(bird):  
        draw(False)  
        return  
    for ball in balls:  
        if abs(ball - bird) < 15:  
            draw(False)  
            return  
  
draw(True)  
ontimer(move, 50)
```

Programarea jocului

Pasul 7: Inițializarea jocului

```
setup(420, 420, 370, 0) # creează fereastra
hideturtle()             # ascunde săgeata "îestoasă"
up()                     # ridică stiloul
tracer(False)            # face ca desenul să apară instant
onscreenclick(tap)       # când facem click → chemăm tap()
move()                   # pornește jocul
done()                   # menține fereastra deschisă
```