

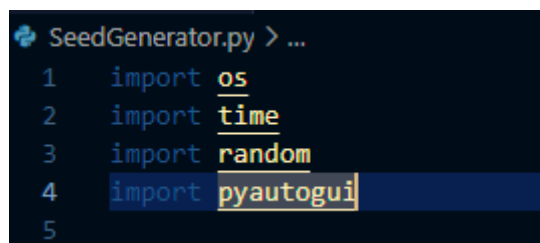
Generování náhodných čísel a testování generátorů

Zadání

Tento úkol bude poněkud kreativnějšího charakteru. Vaším úkolem je vytvořit vlastní generátor semínka do pseudonáhodných algoritmů. Jazyk Python umí sbírat přes ovladače hardwarových zařízení různá fyzická a fyzikální data. Můžete i sbírat data z historie prohlížeče, snímání pohybu myši, vyzvání uživatele zadat náhodné úhozy do klávesnice a jiná unikátní data uživatelů.

Řešení

Pro vytvoření vlastního generátoru semínka jsem si nejdřív naimportoval různé knihovny, které slouží k získání různých druhů dat. Knihovna **os** slouží k práci s operačním systémem a jeho daty. V mém kódu jej používám pro vygenerování náhodných bytů pomocí metody **urandom()**. Pomocí **time** knihovny zase získám aktuální čas. Knihovna **random** slouží k samotnému generování náhodných čísel. **Pyautogui** nám zase umožňuje skriptování myši a klávesnice a získávat o nich informace. Tuto knihovnu používám ve svém kódu pro získání aktuálních souřadnic x a y z aktuální polohy kurzoru myši.



```
SeedGenerator.py > ...
1  import os
2  import time
3  import random
4  import pyautogui
5
```

Při spuštění programu nejdříve získám informace pro semínko generátoru. Nejprve získám aktuální čas pomocí metody **time()**, ta nám vrátí tuto informaci v podobě uplynutých vteřin od 1. ledna 1970 00:00 UTC. Dále pomocí **pyautogui.position()** si uložím do proměnných souřadnice x a y polohy myši. Poté provedu bitový posun doleva u proměnné **mouse_x** a k nově vzniklé hodnotě přičtu hodnotu **mouse_y**, to uložím do proměnné **mouse_seed**. Na závěr této funkce ještě vyzvu uživatele k zadání nějakého náhodného vstupu do konzole. Pro tento vstup vygeneruji **hash()**. Poté si vrátím tyto části semínka v další funkci **generate_seed()**

```
def collect_seed_data():
    # Získání současného času
    current_time_seed = int(time.time())

    # Získání pozice myši
    mouse_x, mouse_y = pyautogui.position()
    mouse_seed = (mouse_x << 16) + mouse_y

    # Výzva pro uživatele k zadání náhodného vstupu
    user_input_seed = input("Enter something random and press Enter: ")

    return current_time_seed, mouse_seed, hash(user_input_seed)
```

Jak jsem již zmínil výš, na začátku funkce **generate_seed** si připravím pomocí funkce **collect_seed_data** části semínka z různých zdrojů a uložím do proměnných. Poté vygeneruji pomocí metody `urandom(16)` 16 náhodných bytů do proměnné **random_bytes_seed**. Na závěr této funkce zkombinuji všechny části semínka pomocí operace **XOR** do proměnné **combined_seed**.

```
def generate_seed():
    # Získání dat z různých zdrojů pomocí funkce
    current_time_seed, mouse_seed, user_input_seed = collect_seed_data()

    # Vygenerování náhodných bytů semínka
    random_bytes_seed = os.urandom(16)

    # Zkombinování dat pro vytvoření finálního semínka
    combined_seed = (
        current_time_seed ^ int.from_bytes(
            random_bytes_seed, byteorder='big') ^ mouse_seed ^ user_input_seed
    )

    return combined_seed
```

Takto vygenerované semínko nastavím generátoru pomocí metody **seed()**. Na závěr už jen nechám uživatele zadat dolní a horní hranici počtu generovaných čísel. Z tohoto rozmezí vygeneruji pomocí **randint()** náhodný počet čísel, která se mají vygenerovat. Pak už jen v loopu vygeneruji a vypíšu do konzole pseudonáhodná čísla v daném rozmezí.

```

36
37 # Inicializace random generátorů pomocí vytvořeného semínka
38 seed = generate_seed()
39 random.seed(seed)
40
41 #Vygenerování náhodného čísla iterací dle zadaného rozsahu od uživatele
42 min_iterations = int(input("Enter number and press Enter: "))
43 max_iterations = int(input(f"Enter second number bigger than {min_iterations} and press Enter: "))
44 num_of_iterations = random.randint(min_iterations, max_iterations)
45
46 #Generování pseudonáhodných čísel
47 for i in range(num_of_iterations):
48     random_number = random.randint(0, 5000)
49     print(f"Pseudorandom number:", random_number)
50

```

```

Enter something random and press Enter: asdf55h5h5h5
Enter number and press Enter: 10
Enter second number bigger than 10 and press Enter: 20
Pseudorandom number: 2768
Pseudorandom number: 2536
Pseudorandom number: 4807
Pseudorandom number: 881
Pseudorandom number: 4099
Pseudorandom number: 2932
Pseudorandom number: 3487
Pseudorandom number: 2525
Pseudorandom number: 3001
Pseudorandom number: 4372
Pseudorandom number: 4414
Pseudorandom number: 392

```

Závěr

Když inicializujeme generátor pseudonáhodných čísel s konkrétním semínkem pomocí **random.seed(seed)**, zajistíme, že každé volání funkcí pro generování náhodných čísel (např. **random.randint()**, **random.uniform()**, atd.) bude produkovat stejnou posloupnost čísel, pokud bude použito stejné semínko.

Tímto způsobem můžeme dosáhnout reprodukovatelnosti generovaných náhodných dat. Pokud potřebujeme vytvořit stejnou posloupnost náhodných čísel vícekrát (například pro testování nebo ladění), můžeme nastavit stejný seed a zajistit, že generátor vyprodukuje stejná čísla.