

ZADÁNÍ 5. DOMÁCÍ ÚLOHY, IB111, SK. 11 A 12

Pokyny

- Máte za úkol naprogramovat simulátor jednoduché hry s náhodností podle zadání níže. Celkem lze získat až 40 bodů.
- Své řešení tvořte jako jeden soubor, pojmenujte jako "prijmeni_du5.py" a uložte do příslušné odevzdávnice.
- Nezapomeňte na **prohlášení** a **sebehodnocení** dle pokynů <https://www.fi.muni.cz/IB111/?p=dupokyny>

STYL

Pište co nejčistší kód, zejména:

- volte vhodná jména proměnných a vámi definovaných funkcí (anglicky)
- problém rozložte na dílčí funkce
- kód vhodně komentujte, cítíte-li, že to přispěje jeho čitelnosti (anglicky, česky, slovensky)
- omezte maximální délku řádku ideálně na 79 znaků, maximálně na 99 (podle PEP 8)
 - <https://www.python.org/dev/peps/pep-0008/#maximum-line-length>

DEADLINE:

- skupina 11: 2018-12-23 23:59
- skupina 12: 2018-12-25 23:59

Zadání

1. část - zpracování textu (13 bodů)

Dataset **student-mat.txt** obsahuje informace o studentech, o jejich podmínkách, chování a výsledcích. Vaším úkolem je napsat program, který načte a zpracuje data ze souboru. Jednotlivé záznamy odpovídají řádkům, atributy jsou odděleny středníky, první řádek je hlavička.

Napište funkci **analyzeDataset(filename = "student-mat.txt", filter, requestedColumn)**: s parametry:

- **filename**: jméno souboru s daty
- **filter**: string ve tvaru "sloupecOPERATORhodnota" udávající výraz - do statistik se pak započítají pouze záznamy, pro které je výraz pravdivý
 - **sloupec**: udává jméno atributu z hlavičky, podle kterého se data filtrují
 - **OPERATOR**: >, >=, <, <=, =, != (pro výčtové hodnoty dává smysl pouze = a !=)
 - **hodnota**: referenční hodnota filtru
 - Příklad: filter = "age>16"
 - "GP";"F";17;"U";"GT3";"T";1;1;"at_home";"other"; ... použitý:
 - "GP";"F";15;"U";"LE3";"T";1;1;"at_home";"other"; ... ignorovaný:
 - !! Bude-li filtr **nevalidní** (neexistující sloupec, operátor atd), vypište **varování** a oznamte, že se pokračuje **bez aplikace filtru**. To samé, pokud bude použit výčtový typ sloupce společně některým z operátorů >, >=, <, <=.
- **requestedColumn**: volba sloupce, pro který budeme ukládat statistiky

Funkce načte soubor, vyfiltruje řádky dle zadaného filtru a vypíše počty výskytů pro jednotlivé hodnoty **requestedColumn**.

Příklad užití: Mám balík dat (**filename = "student-mat.txt"**) a chci z něho vytáhnout počty chlapců a dívek (**requestedColumn = "sex"**) a chci se omezit jen na patnáctileté studenty/studentky (**filter = "age=15"**).

2. část - práce s bitmapami (13 bodů)

Připravte RGB obrázek s **bílým pozadím** 640 x 480 bodů

- 1) Napište funkci, která **vykreslí černou kružnici (obrys)** pro zadaný střed a poloměr. Doporučuji implementovat tzv. **Bresenhamův nebo Midpoint** algoritmus - wiki. S její pomocí napište funkci **generate()**, která do obrázku vykreslí 6 náhodně rozmístěných kružnic s poloměrem náhodně zvoleným mezi 10 a 240 (pohlídejte si, aby se vešly celé, poloměr > 240 se na plátno o výšce 480 nevejde nikdy).
- 2) Napište funkci **detect()**, která v obrázku detekuje křížení/dotyky kružnic a zvolená místa podbarví červeně. Nemusí jít přesně pouze o bod křížení, stačí přibližně, viz ukázka. (Použijte průchod celým obrázkem a sledujte počty černých bodů v matici 3x3)
- 3) Napište funkci **blur()**, která obrázek rozmaže (Použijte průchod celým obrázkem a intenzity RGB složek počítejte z matice 3x3)

Tyto funkce postupně volejte a po každé funkci uložte výsledný obrázek pod nějakým jménem, aby se nepřepisovaly (např. První funkce vytvoří „generated.bmp“, druhá ho načte, něco udělá a uloží jako „detected.bmp“ atd). Pokud se vám nepodaří nějakou funkci napsat a potřebujete vstup pro následující, můžete použít obrázky u Isu.

3. část - implemetace želvy (14 bodů)

Udělejte **objektovou** implementaci želví grafiky s vykreslováním do SVG (na základě kódů uvedených ve slidech k přednášce). Implementujte metody pro „otočení směrem k zadané želvě“ a „vykreslení spojnice se zadanou želvou“. Za využití těchto metod vytvořte zajímavé obrázky využívající více želv. Konkrétně např. následující dva obrázky:

- První obrázek je „želví honička“, kdy několik želv honí jednu (červenou).
- Druhý obrázek vznikne tak, že dvě želvy jdou po kružnici, jedna jde rychleji než druhá a pravidelně vykresluje jejich spojnice.

Barvy řešit můžete, ale nemusíte (uvedené obrázky využívají barevné palety z knihovny seaborn, https://seaborn.pydata.org/tutorial/color_palettes.html).

<https://www.fi.muni.cz/~xpelane/IB111/podklady/turtles.png>

