[CS 11 25.1] Lab 7a – Pixel Art

Cheatsheet is available here: https://oj.dcs.upd.edu.ph/cs11cheatsheet/

Submit solution [CS 11 25.1] Lab Exercise 7

My submissions

Points: 215 (partial)
Time limit: 6.0s
Memory limit: 2G

Problem type

Allowed languages
py3

Problem Statement

Inspired by all the games you played as a kid (you did play games when you were young, right?), you decide to make some pixel art!

You have an $r \times c$ canvas. The rows are numbered 1 to r from top to bottom, and the columns are numbered 1 to c from left to right. The cell at row i and column j is denoted as (i, j).

You will draw n rectangles; each rectangle is defined by two of its diagonally opposite corners (a, b) and (c, d), and you will fill in all cells inside this rectangle.

After drawing the n rectangles, show the final state of the canvas. Note that the following values must be customizable:

- the character used for cells that are filled,
- the character used for cells that are *not* filled, and
- the amount of spacing (in # of columns) between consecutive columns.

Task Details

Your task is to implement a function named `draw`. It should have three positional arguments, as well as some keyword arguments.

The positional arguments are r, c, and `rects`:

- r and c are integers denoting the size of your canvas.
- `rects` is a sequence, each element of which is a triple representing a rectangle:
  - the first two elements are pairs of integers (`int`s) denoting the top-left and bottom-right corners of the rectangle,
  - and the third is a character to fill the cells of that rectangle with.

The keyword arguments are `empty`, and `sep`.

- `empty` is a keyword argument whose value is a length-1 string (`str`) denoting the character to use for unfilled cells. By default, it must be `.`.
- `sep` is a keyword argument whose value is an integer denoting the amount of spacing between consecutive columns. By default, it must be 1.

The function must return a string denoting the state of the canvas after drawing the n rectangles. The rows of the canvas must be separated with newline (`\n`) characters.

Restrictions

- The following symbols are now allowed: `map`, `filter`
- The following imports are now allowed:
  - `count`, `islice`, `chain`, `takewhile`, `starmap and zip_longest` from `itertools`.
  - `cache`, `lru_cache`, `total_ordering`, `partial`, `reduce and wraps` from `functools`.
  - `randint`, `randrange and choice` from `random`.
  - `Fraction` from `fractions`.
  - `dataclass` from `dataclasses`.
  - `contextmanager` from `contextlib`.
  - `Enum`, `auto` from `enum`.
- Anonymous functions are now allowed.
- Inner functions are allowed.
- Classes, dataclasses, and enums are allowed.
- Recursion is allowed.
- Loops are allowed.
- Generators and comprehensions are allowed.
- Your source code must have at most 800 bytes.

Examples

Example 1 Function Call

```
draw(5, 5, [((1, 2), (5, 2), 'X'), ((1, 4), (5, 4), '#')])
```

Copy

Example 1 Return Value

```
"""\
..X...#..
..X...#..
..X...#..
..X...#..
..X...#.."""
```

Copy

Example 1 Explanation

The `\` here is Python syntax signifying that the first newline character is not part of the string.

Example 2 Function Call

```
draw(7, 8, [((1, 2), (7, 4), 'X'), ((3, 1), (5, 6), '#')],
empty='*', sep=0)
```

Copy

Example 2 Return Value

```
"""\
*XXX****
*XXX****
######**
######**
######**
*XXX****
*XXX****"""
```

Copy

Example 3 Function Call

```
draw(5, 5, [
    ((1, 1), (5, 1), 'o'),
    ((1, 3), (1, 5), 'o'),
    ((1, 5), (3, 5), 'o'),
    ((3, 3), (3, 5), 'o'),
    ((3, 3), (5, 3), 'o'),
    ((5, 3), (5, 5), 'o'),
], empty='.', sep=1)
```

Copy

Example 3 Return Value

```
"""\
o...o.o.o
o.......o
o...o.o.o
o...o....
o...o.o.o"""
```

Copy

Constraints

- The function `draw` will be called at most 100 times.
- $1 \le r, c, n \le 400$
- The sum of rc across all calls to the `draw` function is at most $160{,}000$.
- The sum of n across all calls to the `draw` function is at most 800.
- For each corner (a, b) of a rectangle, $1 \le a \le r$ and $1 \le b \le c$.
- `full` and `empty` are ASCII printable characters.
- `sep` is a nonnegative integer at most 5.

Scoring

Note: New tests may be added and all submissions may be rejudged at a later time. (All future tests will satisfy the constraints.)

- You get 75 ❤️ points if you solve all test cases where:
  - $r, c \le 50$
  - $n \le 50$
  - The sum of rc across all calls to the `draw` function is at most $2{,}500$.
- You get 140 🔴 points if you solve all test cases.

Clarifications

Report an issue

No clarifications have been made at this time.