

MCS 260 Project Four : drawing random walks due Wednesday 7 November at 1PM

The goal of this project is to draw a random walk, using a module that exports the writing of POSTSCRIPT files. POSTSCRIPT describes how to print a page. Ghostscript, Ghostview and GSview are free (for noncommercial use) programs to render POSTSCRIPT documents to screen.

Imagine a frog sitting at the center of a tiled pavement. At each time step, the frog can remain seated, or jump to another tile located north, northeast, east, southeast, south, southwest, west, or northwest with respect to the current tile. First we will write a module to write coordinate lists to POSTSCRIPT files, and then we will use this module to draw a random walk.

On the course web site is a program `randgray.py` to generate a POSTSCRIPT file with random grayscales. Use the code in `randgray.py` to write a module which exports three functions:

1. Create a new file and write the beginning of a POSTSCRIPT file.
Input parameters are the file name, `#rows`, and `#columns`.
On return is a file object. The last word this function writes to the file is `image`.
2. Write the path of the random walk to file.
Input parameters are the file object, `#rows`, `#columns`, a list of tuples with x and y coordinates for each point on the path, and finally a fading parameter. This fading parameter is an integer added to the grayscale after printing each pixel. Pixels on the border must be black. This function does not return anything.
3. Write the final word and close the file.
The input parameter is a file object. The final word written to file is `showpage`.
This function does not return anything.

The main program first prompts the user to enter the name of the output file, then asks for the `#rows`, `#columns`, `#steps` in the walk, the step size, and the fading speed. If the step size is one, our frog can jump only to adjacent tiles. If the step size is two, then the frog may skip a tile in between the current tile and the destination tile. The fading speed is the increment to the grayscale of the pixels. For each step, a random integer will determine the direction of the destination in the jump. The walk starts in the middle, i.e.: at $(\#rows/2, \#columns/2)$ and x and y coordinates must respectively remain in $[0, \#rows-1]$ and $[0, \#columns-1]$. New random numbers must be generated if a jump would lead to coordinates off the pavement.

Some important points:

1. Correct programs without modules or functions can only get half of the points.
2. Appropriate documentation means that every function must have a documentation string.
3. Handing in an incomplete but working program is better than handing in a program that crashes or does not run at all.
4. The first line of your Python program must be
`# MCS 260 Project Four by <Author>`
where you replace the `<Author>` by your name.
5. Email your solution to the project to `jan@math.uic.edu` before 1PM on Wednesday 7 November so the date of the email is proof of an on time submission. As a backup, bring also a printed version of your solution to class.

If you have questions or difficulties with the project, feel free to come to my office for help.

If the program is in the file `randwalk.py`, then a session could go as

```
$ python randwalk.py
a picture of a random walk
Give file name : sample
#rows : 100
#columns : 100
#steps : 255
step size : 4
fading speed : 1
do ghostview on sample.ps
```

The file `sample.ps` when printed looks like:

