CS343 Homework 2 - Flood-It! Reloaded

Due 11:59pm, September 6, 2013

In the previous homework assignment you completed the Flood-It! game by implementing the `flood` function, deciding which tiles should be added to the `flooded_list`. You also looked at your code an made some predictions regarding the execution time.

In this assignment you will measure the execution time of your `flood` function on boards of varying sizes to see whether your algorithm scales up nicely (linearly) or whether the execution time grows at a faster rate.

Set Up We have updated the `floodit.py` file to include support for running the game in non-interactive "batch" modes that either test your `flood` function for correctness or that time the execution time. Copy all of the files from github@IU in the following directory:

```
C343-Fall2013/jsiek-c343/hw2
```

into the `hw2` directory in your account on github@IU. Also copy your `flood.py` from the last assignment into your `hw2` directory.

You will need to make the following change to your `flood` function because the utility function `in_bounds` has changed. Add a third parameter named `screen_size` to your `flood` function and pass `screen_size` as the second argument to `in_bounds`.

```
def flood(color_of_tile, flooded_list, screen_size):
    ...
    in_bounds(..., screen_size)
    ...
```

Measure and Improve Your Flood Algorithm You're now ready to measure the execution time of your `flood` function. Run `floodit.py` with the command-line argument `time`:

```
python floodit.py time
```

This will create a file named `times.csv`. Each rows lists the number of tiles in the board and the execution time for `flood`. You can import this data into a spreadsheet such as Microsoft Excel or Google Docs and then graph it, or can you use the `plotter.py` script to create a graph. To use `plotter.py` you'll need to install `matplotlib`. For Mac users, there's a good set of instructions for this at:

`http://fonnesbeck.github.io/ScipySuperpack/`

Once you've created the graph, compare it to your prediction in assignment 1. Does the graph look like you expected? What is the rate of growth of the execution time? Include the graph that you've created in what you turn in. What parts of your algorithm do you think are contributing the most to the execution time?

Try to improve the scalability of your `flood` function. After changing the code for `flood`, time it and graph it again as above. Describe what changes you made inside `flood` and describe the rate of growth in the new graph. Did you improve the growth rate? Include the new graph in what you turn in

The Drought You're friends have been playing Flood-It! so much that it's getting too easy for them! We need to make it harder. In the `floodit.py` file, change

```
drought_enabled = False
```

to

```
drought_enabled = True
```

This will cause there to be a drought every seven plays. During a drought, 1/3 of the flooded tiles change their color and are no longer flooded. The file `drought.py` contains the implementation of the new drought feature.

Your first task is to measure the execution time the `create_drought` function instead of the `flood` function. You'll need to make the appropriate changes in `floodit.py`. Create a graph for the execution time of `create_drought` and include it in your turn-in.

What parts of the `create_drought` code do you think are contributing the most to the execution time? Make a change to `create_drought` by changing the algorithm or data-structures that it uses in an effort to improve the execution time. For example, you might try replacing the use of arrays (Python lists) with linked lists.

Turn-in Your `hw2` directory should include both versions of your `flood` function: put them in new files `flood1.py` and `flood2.py`. Similarly, put your two version of `create_drought` in `drought1.py` and `drought2.py`. Your `hw2` directory should also contain four graphs: the before and after execution times for flooding and for the drought. Finally, you should answer all of the questions in this document in the README.md for `hw2`. Once your `hw2` is complete, make a pull request.