Slide 1

Administrivia

- Reminder: Quiz 2 Thursday. Likely topics are arrays and `String`s.
- Reminder: Homework 2 code due today. I have office hours this afternoon.
- Homework 3 due dates on Web (design a week from today, code next Thursday).

Slide 2

Homework 2 Code

- Eclipse will suggest adding a variable called `serialVersionUID` to some of your classes. Do that. (Notice there's one of these in some of the provided code.) Value can be anything. We will talk later about what this means and how to make use of it.
- Notice that x/y coordinates of framework are opposite of row/column. `getSize()` in screen class should return width by height.
- To confirm that your code works:
  - Start the game, and verify that the playing field is what you defined (dimensions, plus appearance of blocks — for now, solid colors are okay).
  - Try running the screen editor (directions in "project description" document). If it comes up, and shows all the kinds of blocks you defined, all is well. (Actually it doesn't have to do that if you don't plan to use it — it just has to not crash.)

Slide 3

Sorting and Searching Arrays

- A common thing to do with arrays is sort them. (Remember this from PAD I or equivalent?)
- Various algorithms for sorting and searching. Some fast, some slow; some simple, some complex. Decide which to use based on considerations of simplicity versus speed.
- “Speed”? Yes, but expressed as order of magnitude (“big-oh notation”).

Slide 4

Order of Magnitude of Algorithms

- Idea is to estimate how work (execution time) for algorithm varies as a function of “problem size” (e.g., for sorting, size of array). (Similar idea can be applied to how much memory is required.)
- Usually do this by counting something that represents most of the “work” in the algorithm and varies with problem size (e.g., for sorting, how many comparisons).

Slide 5

Order of Magnitude of Algorithms, Continued

- Informally, $O(N)$ means work/time is proportional to N (problem size). $O(N^2)$ means ...?

  (Compare aN and bN^2 as N increases, for different values of a and b. bN^2 larger for larger enough N.)
- Formal definition (from CSCI 1323): $g(n)$ is $O(f(n))$ if there are positive constants n_0 and c such that for $n \geq n_0$,

$$g(n) \leq cf(n)$$

Slide 6

Simple (but Slow) Sorts

- Bubble sort. (First pass goes through the whole array, swapping consecutive elements if out of order, so largest element bubbles to the end. Next pass goes through all elements but last. And so forth.)
- Selection sort. (First pass finds largest element and puts it at end. Next pass finds next-to-largest element and puts it at next-to-end. And so forth.)
- Insertion sort. (First pass inserts second element into list of first element. Next pass inserts third element into list of first two elements. And so forth.)
- All of these are $O(N^2)$. And there are others ...

Other Sorts

- Quicksort (to be discussed later). $O(N \log N)$.
- Mergesort (to be discussed later). $O(N \log N)$.
- Many others ...

Slide 7

Searches

- Sequential search. $O(N)$.
- Binary search. $O(\log N)$.

Slide 8

Sorting and Searching — Example Code

- See “Sample programs” page (here)) for code performing an instrumented sort (count number of comparisons), and other examples.

Sorting and Searching Arrays in Java

- Writing your own sorting routines is pedagogically useful, but in practice you would probably use something from Java library. `Arrays` class has some useful methods.
- One thing that’s nice about Java is “polymorphic sorting”; can sort objects of any class that implements `Comparable`. Can also provide, when you call `Arrays.sort`, a `Comparator` that defines the ordering you want. Example: case-insensitive sorting of strings.

Minute Essay

- What are you finding interesting, annoying, educational, noteworthy, etc., about doing the homeworks?
- Look at the `String` example from last time (linked from the sample programs page). Do one of the following:
  - Fix the `countWords` method so it counts “words” separated by any (non-zero) number of blanks.
  - Write code for the `reverse` method.

Minute Essay Answer

- See “Sample programs” page (here)) for some possible solutions.