Slide 1

Administrivia

- Reminder: Homework 2 design due ASAP, code today. No penalty if in by 5pm tomorrow. Remember that for the design phase you just have to send me e-mail saying ”it’s ready”, but for the code phase you need to actually send me all your .java files (plus any other files that would be needed but that I wouldn’t have — but you probably don’t have any of those yet).

  (A bit more about this assignment in the next slide.)
- Homework 3 should be on the Web late today or early tomorrow. Probable due date is next Thursday, or the following Tuesday. This is the assignment where you really start writing your game.

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, so long as it’s different for each class.
- 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, I recommend solid colors).
  - 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.

Slide 3

Arrays — Review

- Arrays in Java are objects, with `length` field recording actual length (can't change it, but can use it in loops, etc.).
- Access is like in C, but safer.
- Multidimensional arrays are arrays of arrays, but with a little "syntactic sugar" to make it easier to define rectangular ones. If we declare

  `int[][] ints = new int[4][6]`

  we can get back the dimensions with `int.length` (4) and `int[0].length` (6). (Can't remember which is which? Write a little test program.)

Slide 4

Sorting and Searching Arrays

- A common thing to do with arrays is sort them. (Remember this from PAD I?)
- 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 5

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 6

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 7

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 . . .

Slide 8

Other Sorts

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

Slide 9

Searches

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

Slide 10

Sorting and Searching Arrays in Java

- `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`.

Slide 11

Minute Essay

- None — quiz.