Slide 1

Administrivia

- Reminder: Homework 2 due next week (design Tuesday, code Thursday).

Slide 2

More Administrivia

- Some of you may want to do some or all development using your own machines. You can do that, but you may want to also have a copy of your code on our file servers. Two options for transferring files between machines:
  - Simple, not particularly smart: Just copy `.java` files between machines. Several options.
  - More complicated to get started, more professional: Use CVS (version-control software, Eclipse has built-in support).
- Some instructions in project description.

Slide 3

Homework 2 — General Comments

- Design phase is meant to be about defining classes and interfaces. For every class (or interface) and every method, I want comments (can be be brief). For classes, these should describe (to the best of your understanding) how they fit into your game (e.g., "class for wall blocks").
- In order to generate the HTML documentation ("javadoc"), you probably have to have something minimally compilable. As suggested in assignment — you can create skeleton/stub versions of methods, and fill in real code in code phase. (For classes where you get code, though, might be simpler just to copy it in right away, if there are comments in the code. Or copy comments from game framework API.)
- Be sure to get the updated JAR file (should have name `PAD2F09Assn2.jar`). With every assignment there will be a new JAR file, as you replace various parts of the starter code with your code.

Slide 4

Homework 2 Design

- Interfaces `YourBlock`, `YourEntity`: In project API, referred to as "general block type" and "general entity type". You will use these as replacements for `BasicBlock` and `BasicEntity`, and everywhere else you use one of the framework's generic classes.
- Player and game setup classes. Copy code from `BasicPlayer` and `BasicGameSetup` and edit (change `package` line, block and entity types). May want to change game setup more during code phase. Also edit your main class from the first assignment.

  Don't worry about player for now — you will start writing your own in the next assignment.

Slide 5

Homework 2 Design Continued

- Block class(es). These are blocks that make the playing field for your game. Should have one class for each kind of block (floor, walls, ladders, anything that doesn't move). Try to define as many as you can. Copy code from `BasicBlock`.
- Screen class (class implementing `Screen` interface). This is the most work in this assignment. Eclipse can make stub methods for you. Copy and paste comments from API.

Slide 6

Arrays in Java — Review

- Declaring and creating arrays in Java is different from in C — examples:

```
int[] x = new int[10];
String[] s = new String[n];
```

- Once created, though, some things are familiar — syntax for finding elements, range of indices.

  (Notice, though, that the second example above creates not `String` objects, but *references* to `String` objects.)

- Under the hood, more differences — in C, arrays are almost indistinguishable from pointers, but in Java, they're objects, with a `length` field you can use (but not change), and built-in bounds checking.
- Arrays as parameters to methods — what is passed is a *reference* to the array, so the method can change its elements.

Slide 7

Sorting and Searching Arrays

- A common thing to do with arrays is sort them. (In theory this is covered in PAD I, but in practice, not always, so we will spend time on it.)
- 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 8

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.)

And there are others . . .

Slide 9

Other Sorts

- Other comparison-based sorts (to be discussed later) include quicksort and mergesort.
- Other methods include bucket sort and radix sort.

Slide 10

Searches

- Sequential search — start with the first element, examine elements one after another until a match is found or there are no more to examine.
- Binary search (for sorted data only) — examine the middle element and either stop if a match is found or recursively search the left or right half of the array.

Slide 11

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 12

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 13

Order of Magnitude of Sorts and Searches

- Usually we count comparison (and sometimes also swaps).
- For bubble sort, how many comparisons? For N elements, first pass through the array makes $N-1$ comparisons, next pass makes $N-2$, etc. Total is $(N-1)(N-2)/2$ — which in order-of-magnitude terms is $O(N^2)$.
- Selection sort and insertion sort are also $O(N^2)$.
- Quicksort and mergesort are $O(N \log N)$. (More about this later.)
- Sequential search is — ? ($O(N)$) Binary search? ($O(\log N)$)

Slide 14

Minute Essay

- None — quiz.