Administrivia

- Reminder: Homework 4 design due today, code Thursday.

Slide 1

Lists — Recap

- List ADT (review):
  - "Values" are lists of elements.
  - Many operations possible — add element, remove element, search for element, etc., etc.
- Something we often want to do with this and other "container classes" is do something to all elements — idea of "iterator". `java.util.Iterator` defines interface for this.

Slide 2

Lists, Continued

- A really simple unordered list might provide only methods to add an element (don't care where), remove an element (if found), and create an iterator that could be used to "walk through" the list.

- First let's define an interface for this version of lists, and then write some very simple code using `Vector`s. (No, no reasonable person would do it this way, but it's an example of how to write an iterator.)

Slide 3

Lists, Continued

- Now think about how to implement the same interface with a "linked list" as described last time. First, draw pictures . . .

- Then think about what you need to turn the pictures into code. Probably you'll need:

    – Variables (e.g., something to point to the first "node" (little box).

    – Classes-within-the-class (for nodes / little boxes, iterators).

    – Methods for interface.

Slide 4

Slide 5

Minute Essay

- In Homework 2 you wrote, as part of your screen class, a method called `createEntityIterator`. Using this method and what you now know about iterators, sketch code you could use elsewhere in your game to "walk through" the list of entities on a screen and do something with them (count them, for example). Assume that variable `firstScreen` points to an instance of your screen class.

- Can you think of something you want to do in your game where you would need similar code?

Slide 6

Minute Essay Answer

- You could write something like the following:

```
Iterator<MyGameEntity> iter =
            firstScreen.createEntityIterator();
int count = 0;
while (iter.hasNext()) {
    iter.next();
    ++count;
}
```

- It might be useful for detecting whether your player has collided with another entity.