

Slide 1

### Administrivia

- Reminder: Homework 3 design due Tuesday.
- Quiz 3 moved to after midterm. (Okay?) Review sheet for midterm will be on Web soon.
- Revised version of binary search example from last time in sample programs.

Slide 2

### A Little About Homework 3

- In this homework you start writing code for your player, to replace the stick figure in the starter game.
- Key parts of this assignment are making the player
  - interact with different kinds of blocks.
  - move in response to keyboard or mouse input from human player.(If these don't apply to your game, talk to me about whether there are reasonable substitutes.)  
For design phase, you just need to describe this interaction.

### Homework 3, Continued

Slide 3

- `Player` defines some constants you should use.
- You will implement `KeyListener` or one/both of the mouse-listener interfaces. When you do this, the framework will deliver key and/or mouse “events” to you.
- Most logic will go in `update`, `getUpdateTime`, and the listener methods.
- A general comment: If you find yourself looking up something like the ASCII value of a character, or the value of one of the game framework’s constants — *stop*. There is probably an easier and more Javaesque way to do what you want.

### Abstract Data Types

Slide 4

- “Abstract data type” (ADT) is defined as
  - A set of values.
  - A set of operations on those values.
- In other words — something that stores data (in an unknown form) and provides a standard interface for dealing with it.

### Stack ADT

Slide 5

- Value — list of elements.
- Operations — push, pop, “empty?”
- Implementing this? might be a good example of
  - Defining a (generic) interface.
  - Writing a class to implement it (using arrays — for, um, fun? practice?).
  - Working with exceptions.

### Queue ADT

Slide 6

- Value — list of elements.
- Operations — enqueue, dequeue, “empty?”
- We could implement similarly to what we did for stacks . . .

## Minute Essay

- TBA

Slide 7