

Slide 1

Administrivia

- Reminder: Homework 4 code due today. Homework 5 design due Thursday.

Slide 2

More Homework Tips

- In Homework 3 you start implementing interaction between entities and blocks. Finding which block an entity is on can be done with methods of `Location` and your screen class. (We talked about this in an earlier class.)
- In Homework 4 you start implementing interaction between entities. Finding other entities can be done using methods of your screen class (and the iterator returned by one of them).
- In both cases you may want to differentiate among different kinds of blocks/entities (e.g., is this block a wall? is this entity an enemy?). How to do that?

Slide 3

More Homework Tips, Continued

- One way to differentiate among types of blocks/entities is to use the `instanceof` operator and check for specific classes.
- A maybe-nicer way, though, is to add methods to your block and entity interfaces — e.g., an `isPassable` for blocks that says whether entities can pass through this kind of block (false for walls, true for other types of blocks).

(This, by the way, is one of the advantages of making the game-framework classes generics — you write your code in terms of your block/entity interfaces, which may include methods not present in `Block` and `GameEntity`.)

Slide 4

Stacks and Queues, Another Way

- We talked about stacks and queues as ADTs. We showed an array-based implementation.
- Could we do a different implementation — with linked lists? (Of course. Let's sketch some code.)

Slide 5

One More ADT — Priority Queue

- Value — list of elements, of some type we can put in order.
- Operations:
 - Add element.
 - Remove element with lowest (or highest) value.
 - “Is empty?”

(Look at game framework `PriorityQueue` interface for a slightly different, but equivalent, list. You will write one of these for Homework 5.)

- How to implement? (Unordered list? List kept in order by value? Compare order of magnitude of “add” and “remove” operations.)
- (Write code.)

Slide 6

Minute Essay

- None — sign in.