Path Finding

11/6/2009

Opening Discussion

- Minute Essays
  - Word games
  - Hangman
  - Images like the one on my web page.
- When you leave your room to walk to your first class on the first day of school, what goes on in your head? What are you thinking in the process of planning how to get to class?

Objective

- We have been using the walkToward method for a long time now. This takes us to a destination, but it doesn't respect boundaries.
- I want us to be able to make an actor walk to a destination, but do it without passing through obstacles.
- As it happens, we will also make it find the shortest path as well.

Testing Multiple Options

- The biggest difference between what we want to do now and what we have been doing is that before we only really had to consider one option. Technically, we were greedy, always moving in a direction that took us closer.
- That doesn't work now. We have to consider multiple options and taking a step in the “right” direction now doesn't always help.

Recursion

- To do this, we are going to use a technique called recursion.
- A recursive method is a method that calls itself.
- Requires that the recursive call be conditional. Otherwise you have infinite recursion.
- When a method returns, the computer remembers where it was called from and goes back. This information is stored on the call stack.

Recursion for Iteration

- The simplest form of recursion is a recursive method that just calls itself once.
- These give us iteration. You could do this in place of loops, but it is less than ideal in Java.
- Let's see how we could make a recursive method to add a specified number of actors.

Recursing Multiple Options

- The real power of recursion comes from having a method call itself more than once.
- The call stack remembers where we are so we can return and try a different option.
- This behavior is difficult to mimic with loops.
- Have to be careful because these can take a long time to compute if not done intelligently.

Our Approach

- Our path finding algorithm will be recursive. It will call itself four times, once for each direction, and return how many steps it takes to get where we are going if we step in that direction.
- To prevent the algorithm from running in circles, we need to keep a 2D array that tells us where we have been. For optimization, we will store how many steps it took us to get there.

Minute Essay

- What questions do you have about today's topic?
- Remember that projects are due Monday.