How do I make for loops run side by side?

Asked 16 years, 1 month ago Modified 15 years, 4 months ago

Viewed 2k times

4

I have been working on a childish little program: there are a bunch of little circles on the screen, of different colors and sizes. When a larger circle encounters a smaller circle it eats the smaller circle, and when a circle has eaten enough other circles it reproduces. It's kind of neat!

However, the way I have it implemented, the process of detecting nearby circles and checking them for edibility is done with a for loop that cycles through the entire living population of circles... which takes longer and longer as the population tends to spike into the 3000 before it starts to drop. The process doesn't slow my computer down, I can go off and play Dawn of War or whatever and there isn't any slow down: it's just the process of checking every circle to see if it has collided with every other circle...

So what occurred to me, is that I could try to separate the application window into four quadrants, and have the circles in the quadrants do their checks simultaneously, since they would have almost no chance of interfering with each other: or something to that effect!

My question, then, is: how does one make for loops that run side by side? In Java, say.

java loops

Share

Improve this question

Follow

edited Mar 3, 2009 at 22:47

user42092

asked Nov 8, 2008 at 6:45

Ziggy

22.3k ● 28 ● 78 ● 109

1 Bear in mind that if you change your circles program so it uses all the cores on your CPU, it might start to affect Dawn of War. Be careful what you wish for :-) – Steve Jessop Nov 8, 2008 at 14:02

are these circles moving ? randomly? i think you need to say a little about that before I'd offer my suggestions. – Scott Evernden Nov 8, 2008 at 20:25

7 Answers

Sorted by: Highest score (default)

9

the problem you have here can actually be solved without threads.

What you need is a spatial data structure. a quad tree would be best, or if the field in which the spheres move is fixed (i assume it is) you could use a simple grid. Heres the idea.

Divide the display area into a square grid where each cell is at least as big as your biggest circle. for each cell keep a list (linked list is best) of all the circles whose center is in that cell. Then during the collision detection step go through each cell and check each circle in that cell against all the other circles in that cell and the surrounding cells.

technically you don't have to check all the cells around each one as some of them might have already been checked.

you can combine this technique with multithreading techniques to get even better performance.

Share Improve this answer Follow

edited Nov 8, 2008 at 19:50
Jonathan Leffler
752k ● 145 ● 946 ● 1.3k

answered Nov 8, 2008 at 7:16
luke
14.8k ● 5 ● 49 ● 57

8

Computers are usually single tasked, this means they can usually execute one instruction at a time per CPU or core.

However, as you have noticed, your operation system (and other programs) appear to run many tasks at the same time.

This is accomplished by splitting the work into processes, and each process can further implement concurrency by spawning threads. The operation system then switches between each process and thread very quickly to give the illusion of multitasking.

In your situation,your java program is a single process, and you would need to create 4 threads each running their own loop. It can get tricky, because threads need to synchronize their access to local variables, to prevent one thread editing a variable while another thread is trying to access it.

Because threading is a complex subject it would take far more explaining than I can do here.

However, you can read Suns excellent tutorial on Concurrency, which covers everything you need to know:

http://java.sun.com/docs/books/tutorial/essential/concurrency/

Share Improve this answer Follow

edited Nov 8, 2008 at 6:59

answered Nov 8, 2008 at 6:51

FlySwat
175k ● 75 ● 248 ● 314

Just know what this is called, this "threading", is enough information for me to be able to start learning more about it: thanks! Often just not knowing the word for something is the barrier between being able to learn about it and not. – Ziggy Nov 9, 2008 at 10:30

Wow! Threads are out of my league: I read that tutorial and will try to write something of my own with threads, but I think it'll be a while before I can use them for speeding up collision detection. Thanks! – Ziggy Nov 13, 2008 at 8:22

5

What you're looking for is not a way to have these run simultaneously (as people have noted, this depends on how many cores you have, and can only offer a 2x or maybe 4x speedup), but instead to somehow cut down on the number of collisions you have to detect.

You should look into using a quadtree. In brief, you recursively break down your 2D region into four quadrants (as needed), and then only need to detect collisions between objects in nearby components. In good cases, it can effectively reduce your collision detection time from N^2 to N * log N.

Share Improve this answer Follow

edited Nov 8, 2008 at 18:16

answered Nov 8, 2008 at 7:00

Jesse Beder
34k ● 22 ● 110 ● 146

Instead of trying to do parallel-processing, you may want to look for collision detection optimization. Because in

1

many situations, performing less calculations in one thread is better than distributing the calculations among multiple threads, plus it's easy to shoot yourself on the foot in this multi-threading business. Try googling "collision detection algorithm" and see where it gets you ;)

Share Improve this answer Follow

answered Nov 8, 2008 at 18:23

Lukman

19.1k ● 7 ● 58 ● 68

Heh, definitely. At first I had each ball check each other ball, which meant that some balls were checking balls that had already been eaten! I fixed that, and it helped a bunch. I'll certainly google these "collision detection algorithms" you speak of! – Ziggy Nov 9, 2008 at 10:33

0

IF your computer has multiple processors or multiple cores, then you could easily run multiple threads and run smaller parts of the loops in each thread. Many PCs these days do have multiple cores -- so have it so that each thread gets 1/nth of the loop count and then create n threads.

Share Improve this answer Follow

answered Nov 8, 2008 at 6:50

anjanb

13.8k ● 19 ● 80 ● 106

If you really want to get into concurrent programming, you need to learn how to use threads.

0

Sun has a tutorial for programming Java threads here: http://java.sun.com/docs/books/tutorial/essential/concurrency/

Share Improve this answer Follow

answered Nov 8, 2008 at 6:52

Bill Karwin
561k ● 87 ● 698 ● 854

0

This sounds quite similar to an experiment of mine - check it out...

http://tinyurl.com/3fn8w8

I'm also interested in quadtrees (which is why I'm here)... hope you figured it all out.

Share Improve this answer Follow

answered Aug 25, 2009 at 8:04

Michael