

CS 414 Homework 4

Due Oct. 5

Overview

The purpose of this homework is to build on the experience with C# and Visual Studio gained in Homework problem 1. We'll work by extending the solution you developed in solving that problem.

Skills we want you to learn

Hands-on experience with thread synchronization in C#.

The assignment

For this assignment, start with your homework 1 assignment but add 2 more colored boxes for a total of 4. Make two red and two green. Implement the following weird and rather arbitrary rules:

- When a box hits a wall it rebounds in the opposite direction, just like in homework 1
- When a box hits another box *of the same color* the two boxes bounce, swapping directions and speeds. Thus if both were initially moving, both are still moving after the impact.
- When a box hits another box *of the other color* whichever box “notices” the impact first comes to a halt. Its speed gets added to the speed of the other box. In effect, the other box “bounces”, but the first box stops.
- When a user clicks on a box, change its speed to a random value.

How to build it.

1. You'll need to modify the HasAThread class to keep a list of the instances – each instance owns one button, but you'll also want to have a data structure that will let you traverse the instances. We recommend a double-linked list but if you are unfamiliar with data structures you can create a small one-dimensional array.
2. You need to make sure certain class members are public so that they can be accessed by some other class member.
3. Think about the synchronization issue raised when traversing this list. Use the C# thread synchronization primitives to protect your solution against problems (hint: read about the “lock” statement).
4. *For extra credit*, use the MSDN (microsoft developer network) resources to find out how to replace your buttons with arbitrary shaped objects. Modify the application so that each of the bouncing boxes becomes some sort of randomly selected, bouncing, arbitrary shape. (The collision code gets trickier!) Part 4 is *not required* and we include it only so that people can learn about MSDN and the sort of stuff you can easily find online by searching the site.

General comment: Yes, you can change any aspect of the “given” solution to make it work!