MC39 Homework 6

Due March 29, 1996; Max Hailperin

You have a new job at a game company. Your boss would like you to take over where your predecessor left off when he quit his job. He was at work on a video game called Photon. In particular, you have been asked to finish implementing the class hierarchy descended from `Obstacle`, i.e., things in the game that can be collided with.

Luckily, this isn't an original game, and you've been able to get your hands on an executable version written by a competitor, so you can try out first hand what your finished product is expected to do. Of course, your boss has hopes that you'll add features rather than just doing an exact clone. The executable version is in `~max/MC39/Photon`. You can run it from a terminal window on one of the SGIs, but you need to first give the command:

```
setenv LD_LIBRARY_PATH /usr/lib:/lib:/usr/share/local/lib
```

(You'll need that to run your own version too.) Playing with the game enough to understand it is your first order of business.

Next, you should probably look at the four files of which I've attached printouts. Of these, `Obstacle.h` and `Obstacle.C` are the module you'll be finishing work on, while the other two (`Photon.h` and `Photon.C`) represent the immediate client of your module within the program structure.

Although the attached files should be your primary focus, you can look at the rest of the files in the program as well. They are all available from the course web page.

To help you understand the `Obstacle` class hierarchy, draw a diagram showing the class derivations. Also, indicate which classes are used primarily in order to achieve commonality of interface and which are used primarilyto achieve commonality of implementation.

Once you think you are ready to resume work on `Obstacle` where your predecessor left off, you should make yourself a private copy of the directory `~max/www-docs/MC39/hw6`. This contains not only the four files that are attached, but also the rest of the program. Since there is a `Makefile`, once you have finished fleshing out `Obstacle.h` and `Obstacle.C` you should be able to just type `make` and have a working version in your directory.

Hopefully you'll have time and enthusiasm to spare beyond what it takes to make an exact clone of the competitor's program. Luckily, this is a perfect program to try extending. There is no limit to what kinds of things you can create that respond in interesting ways to being hit. For example, you can have bonus point targets or or mirrors that switch direction when the ball hits them or traps that hold the ball for some number of time steps before releasing it.

Show some creativity and do what you can to elevate Photon above a mere clone. If you need help, for example on the graphics side, ask! (I *will* be leaving the book on TCL/Tk in the computer lab's monitor room—this is the language and graphical interface package that I used to create Photon's user interface. However, I don't necessarily expect you to read this.)