PHY307, Science and Computers I
Monte Carlo Lab #1, November 5, 2002
(due by end of lab Today)

Making randomness

Summary of what you will do:

People sometimes make decisions using randomness. This is often a good strategy in competitive games, for example, as your opponent can not rely on you making a particular decision.

You will investigate why the result of a coin flip can be near impossible to predict. We will start lab today with a discussion about choosing Halloween candy and flipping objects.

LAB REPORT

1. Get set up. Start a lab report entitled "*MyLastName*CoinFlipLab.doc", where *MyLastName* is your actual last name. That way you can e-mail it as an attachment.

MODULE FOR COINS

2. A module has been written for you and is available as **CoinFlipModule.py** on the Codes page. Load this module and read it. You do not need to review all of the physics, but include in your report a listing of the functions defined and what they claim to do, written in your own words (don't just copy the comments directly.) Which ones return values? Familiarize yourself with the functions that are available to you.
3. Load and run the single coin flip experiment **SingleCoin.py**. Try a couple of different parameters and report on your results.
4. Starting from **SingleCoin.**py, add a second coin, with a different starting angle and a different x-position, saving your code as **TwoCoin.py**. Update both coins in the main loop. Have the loop terminate when *both* coins become still. Have the program report on the status of both coins at the end of the simulation. Include this code in your report. Briefly report what you see.
5. Finally, load **CoinSimulationList.py**. This module simulates a row of coins, each with different starting angles. The coin on the far right is the "reference coin" – the others differ slightly. These coins are in a list and updated all together. What you want to look for in this simulation is the *sensitivity of the motion of the coins to their initial angles*. Coins which start with nearly the same angle have similar motion, but what does "nearly the same" mean? How small of a difference in initial angle is small enough so that the motion is the same? Use this simulation to get one answer to this question. Summarize the results of the simulation and conclude, for these coin parameters, what initial angle difference separates motion similar to the reference coin from distinct motion. If you have time, try another value or two for **ref_angle**.