____

Name

Recitation Assignment # 6

November 7, 2007

You may complete this in class. However, if you are unable to do so, it is expected that you complete this for recitation next time.

If you have any questions, please ask.

In today's recitation, you are going to play horseshoes. The idea is that the computer will lay down a target at random at some distance from you on the ground, and your goal will be to hit the target, by specifying an x- and y- component of velocity. Of course, the thing that makes this a game (albeit a fairly dull one) is the fact that the position of the target will change each time.

New Python Concepts: User Input, Random numbers

New Physics Concepts: Friction

1. Start your program in the normal way ("from visual..."), and begin by drawing a thin box in the y=0 plane. Make it green. This will be the ground. Draw a stick figure at x=0. Be creative. That will be you. (Okay, if you really want, you can make "you" a cube, like the old video games in the early 80's.).

   The ground should end 50m from "you", and afterwards, you should include the command:

   ```
   scene.autoscale=0
   ```

2. I would now like you to put the target at a random place. The target should be a small cylinder (say, height=0.1m). Where should it be? Anywhere from 0m to 50m along the x-axis. How do you generate a random number? At the top of your code, you should have:

   ```
   from random import *
   ```

   To generate a random position at any point in the code, enter:

   ```
   xpos=random()
   ```

   This will produce a random number from 0 to 1. How do we generate a random number from 0 to 50? Exactly!

3. You will have the user "throw" a 0.1kg ball, in an attempt to get it near the target. First, ask the user to enter the x and y components of velocity. For example:

   ```
   print ‘‘Please enter v_x (m/s)’’
   vx=input()
   ```

   will set vx to whatever the user specifies. Have them enter a value for vx and vy.

4. Now throw the ball and keep it running until it lands on the ground. You know that:

   $$\vec{F}_g = -mg\hat{\jmath}$$

   The ball can be displayed however you like, and should generate a path behind it. If the ball gets within, say, 0.2m of the target and has not yet het this ground, you should print that the person had a "hit", and regenerate a new position for the target. Otherwise, let them try again.

5. Now add air resistance:

   $$\vec{F}_r = -\frac{1}{2}C\rho A v^2 \hat{v}$$

   where C is a geometric factor (set it to 1), ρ is the density of the air, $\sim 1.3kg/m^3$, and A is the surface area of your object (say $0.002m^2$ for something like a baseball). This should be in addition to the effect of gravity.

   Now rerun the program. In addition to the true path of the ball, you must show the path that the ball *would* have had in the absence of air resistance.

6. That's it! That's all that's required. **However**, if you'd like to have some fun, you may try some of the following:

   - Change the randomization process so that the "target" appears in a random y-position as well as a random distance from the thrower.
   - Add a scoring procedure to make this a more interesting game.
   - Change the target to look like a person – (Use the power of VPython to make a cool object.)
   - Write the output in the VPython window rather than on the terminal.
   - Add a random "wind" force:

     $$\vec{F}_{wind} = \alpha\hat{r}$$

     where $\hat{r}$ is a random unit vector, and α is constant (random or not).