Exercise: *Due by classtime Weds, April 10*

Exercise: (Exercise is due by classtime Wednesday, April 10th.

Exercise 17.0: Random number generation.
Journal file: `yourName_ex17.0.pro`
Exercise 17.1: See how `1/randomu` behaves
Journal file, `yourName_ex17.1.pro`
Exercise 17.2: Playing Dice
Journal file, `yourName_ex17.2.pro`

Whuduzidon't? No Whuduzitodo's for this assignment.

Turn in on github. `git add` these files:
`yourName_ex17.0.pro`
`yourName_ex17.1.pro`
`yourName_ex17.2.pro`
then `git commit` and `git push` the changes.

Homework Problems Due Monday, April 15 at 11:59 PM

Homework 17.0: Random initialization of stars

Extra credit (2 points each) will be given for keywords that set the animation length in years and/or the x & y display ranges in AU (one keyword used for both x & y ranges). Be sure your code sets reasonable defaults if the user doesn't use those keywords.

Extra credit (6 points) Define an array of 7 colors for the stars. Instead of using `thisStar` to choose a `psym`, use the asterisk for all stars but use `thisStar` to pick one of the 7 colors.

Homework 17.A: (not in the book) Sampling from a distribution.
In order to make a realistic star cluster, you will create stars randomly positioned where their random positions are selected from a realistic distribution. You'll use a simple power-law distribution:

$$p(r) = \begin{cases} Cr^{-\alpha} & \text{if } r > r_0 \\ Cr_0^{-\alpha} & \text{if } r \leq r_0 \end{cases} \tag{1}$$

where $p(r)$ is the probability that a star will be found at a given radius and r_0 is the "core radius" of the cluster.
$\int_0^\infty p(r) = 1$, so $C = \left(1 - \frac{1}{\alpha}\right) r_0^{\alpha-1}$.

This approach is demonstrated in lecture.

You can grab code that's been prepared for you from github: `cluster_making.pro`. It contains the functions you'll need to make your 'realistic' cluster.

You need to change your `initialize_allstars` function to use:
`random_xyz` to determine the stars' initial positions
`random_masses` to determine the stars' masses
`star_velocity` to determine the stars' initial velocities (which requires that you use `star_speeds` to determine the star speeds first)

Once these are implemented, play with a few different numbers of stars and seeds until you get a cluster you like. Note that it may be best to have your cluster radius scaled by 10 AU instead of 1 AU; try both.

Turn in on github:
`yourName_3D_Nbody_RandomAnimation_HW17.0.pro`
`yourName_3D_Nbody_RandomAnimation_HW17.A.pro`