About GravSim v0.2

by Adam Wight

This is the first program I've ever distributed (welll, it's the second version) in 13 years (my age), so smile! This program is email-ware, but if you don't have Internet access, plEasE mail me <u>sumthing</u> (to help pay for my e-mail address, etc.) at

1448 Fernside Blvd.,
Alameda, CA,
94501 (USA),

if you do have e-mail access, it would be really great if you could just write one message so I know at least **someone** got my program. Feel a lot more than free to distribute this! I would be **<u>delighted</u>** to trade **<u>source code</u>** with **<u>anyone</u>** else!!!!! (emphasized enough???)

If I get enough response so that I know the world would appreciate a new version, it will probably include more dimensions, throwing the ball with the mouse, wind, air resistance, friction while rolling, and more user "modificability" (easy to modify, I like making up words).

The reason I wrote this is because I was depressed at how few good or fun non-commercial scientific programs there are for the Mac.

I don't have my solver working yet, and I still can't display the values of the sliders, but E-mail me if you want to negotiate for the source code (hehehehehe).

If you know assembly, I think you should at least be able to use my code, because you probably spent years to learn it. The only thing you can't do is charge for it. My brand new only ever E-mail address is:

adamw@holonet.net

Now, the program:

<u>The Program</u>

This program is a real-time, but not real-gravity simulator. To make "real gravity", on a 72-dpi screen, the pixel/s/s acceleration would have to be ≈2834.64, so: not real gravity.

Anyway, you can drag the ball with the mouse and drop it my releasing the mouse button. If you hold down the shift key, you can "throw" the ball. the line that you see is what the ball's vector will be when you let go.

You can change it’s shape by editing PICT resource 128. My program can even handle a picture that is a different size!!! Just make sure that the PICT’s rectangle is as small against the picture as it can get.

The “gravity” slider controls the gravity’s force on the ball in units of pixel/s/s acceleration. It goes from 1 to 100.

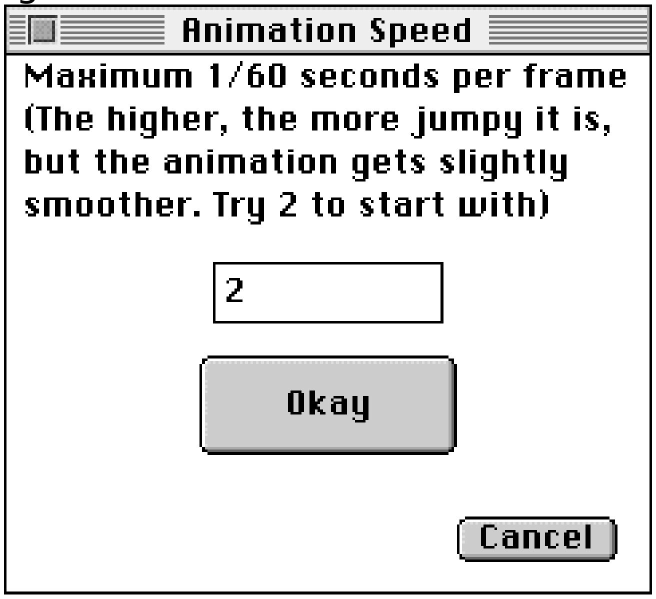
The “Coefficient of Restitution” (call it COR!!) slider controls the percent of the velocity that is conserved on a bounce. The velocity after bouncing is v_{before}*the coefficient of restitution. A good example is, if you hit a golf ball with the club, the COR is around 1. If you hit a wool sweater, it’s around 0.

The “ruler” on the right is so that you can see how high the ball is bouncing. They are increments if 5 pixels.

The “Here, good smiley” button homes the smiley when it gets lost.

The ball bounces in real-time, no matter what’s happening (unless you use one of my program’s dialog boxes, since they reset the timer on purpose), so it should bounce exactly the same on any Mac.

The menu item “Animation…” in the “Characteristics” menu brings up a dialog box that looks like this:



Animation Speed

Maximum 1/60 seconds per frame (The higher, the more jumpy it is, but the animation gets slightly smoother. Try 2 to start with)

2

What it means is that my program doesn’t update if it already updated a certain amount of time ago. This sets that variable in 1/60ths of a second. It’s most useful if you’re using a ‘slow’ machine, then you probably want to set it to 0, so it updates every chance it can get.

I would also REALLY appreciate it if anyone could help me with my animation, and turning pascal strings into C doubles!!! They are <u>really</u> bad.

<u>Known Bugs</u>

Since the program uses real time, if it doesn't get an update for long enough (like if you use the disk while it's in the background) it will gain huge velocity. To home it, press the "Here, good smiley" button.

If the COR is 1 and the gravity is high enough, the ball will actually gain velocity instead of losing it.

Due to some very strange rounding off, the ball can get stuck on the left side sometimes (rarely (Whew! (I love nesting parentheses)))

<u>Some things to do</u>

Drop things around the house on VERY hard surfaces (the harder the surface, the higher the COR, I think) and measure how high they bounce compared to the first time. You can use the formula I labored on for hours (don't be worried, I'm not) to calculate the COR. It is:

$$\frac{-\sqrt{(2gh_i)}}{\sqrt{(2gh_f)}}$$

Looks simple, doesn't it???!!!! Anyway, g is the gravitational constant (which, by the way, isn't actually a constant), which is 9.8 m/s/s acceleration, and h is the height. The subscript 'i' means where you drop it from, and 'f' means the top of the first bounce.

HAVE FUN!!! (and e-mail me)

adamw@holonet.net