

Regular Polyhedra.

This is yet another BackSpace module. It is somewhat different from any other module I've seen though.

It attempts to model the three-d behaviour of the regular polyhedra (tetrahedron, cube, octahedron, dodecahedron, and icosahedron), in the following manner:

At each vertex of the polyhedron, place a unit mass. Between each pair of vertices string a massless, rigid (i.e. non-bending), velocity damped spring. Fill in a selected few of the faces, to make it look good. Put in a large room. Give a random push in a random direction. Make sure that the room doesn't burn in...

Now, this gave me some real headaches. Be careful about changing the parameters as they are set, especially the use of small masses, large spring constants, and small damping factors: because with any or all of these, the polyhedron can become **very** unstable - and very lacking in anything that could be termed beauty. (Unstable behaviour could also cause random features to manifest themselves; I think I've stopped the nastiest, but you never know). As set though, the polyhedra are fairly stable, and will not behave in an inappropriate fashion.

So, as usual, copy PolyhedronView.o into BackSpace.app, or use make install.

Possible bugs:

Not so much a bug, but could someone tell me what the fastest way to do the drawing I'm doing? It would seem to me that a user path would be slower than a wrap, since the path would never be re-used...

I suspect that the algorithm I have for drawing opaque faces in the appropriate order isn't quite the most optimal.

Despite my best efforts, the dodecahedron is unstable (it loses its natural shape after only a

few collisions with the walls). I have suspicions about this one.

The compiler complains a bit. Don't worry. I think everything's ok. I'll fix this sometime.

Possible future extensions:

Notice that if a given solid polyhedron has sufficient symmetry (currently, each face must have the same number of vertices; each vertex in the same number of faces), it can also be handled by the simple expedient of adding the appropriate information to PolyhedronPartView.m (I got the information there mainly from Mathematica). (It had just occurred to me that this is a bit useless - I think that only the regular polyhedra have this property. So, then it becomes the more difficult task of handling polyhedron with different numbers of vertices per face, etc. Oh well).

This is strictly a black, grey and white production. When someone lends me a colour Station for a few days, I'll do something about it. There are some interesting tricks that one could play with the colour of an edge being related to its length. Amongst other things.

Certainly sometime in the future, you'll be able to select the polyhedron, and it's size, and characteristics.

Having more than one at once on the screen (and even bouncing off one another - or at least the 'solid' faces).

If you have suggestions, comments, or bugs, mail me. I have nothing better to do for a month or so.

Simon Marchant
simon@math.berkeley.edu