Documentation for 2+1 dimension `globals.lisp`

November 2, 2011

The following sets the random state for seeding the random number generator. This guarantees that we get a different sequence of random numbers each time we load the file. If you want the same sequence each time, which you would during testing to verify if a bug has been fixed, save the `*random-state*` to a file and load the state from that file.

```
(setf *random-state* (make-random-state t))
```

The next four lines are counters for 3-simplicies, points and space-like 2-simplices. We recycle the ids for 3-simplices

```
(defparameter *LAST-USED-3SXID* 0)
(defparameter *RECYCLED-3SX-IDS* '())
(defparameter *LAST-USED-POINT* 0)
(defparameter *LAST-USED-S2SXID* 0)
```

and the following functions access these counters. `next-3simplex-id` returns a recycled id,

```
(defmacro next-pt ()
  `(incf *LAST-USED-POINT*))
(defmacro set-last-used-pt (pt)
  `(setf *LAST-USED-POINT* ,pt))
(defmacro next-s2simplex-id ()
  `(incf *LAST-USED-S2SXID*))
(defmacro next-3simplex-id ()
  `(if (null *RECYCLED-3SX-IDS*)
       (incf *LAST-USED-3SXID*)
       (pop *RECYCLED-3SX-IDS*)))
(defmacro recycle-3simplex-id (sxid)
  `(push ,sxid *RECYCLED-3SX-IDS*))
```

if possible, else increments the 3-simplex counter.