C++ file i/o, model_t, polymorphic hits function

Friday, October 22, 2010
8:54 AM

main.cpp :

```
#include <iostream>
#include <fstream> — for C++ file I/O
        ⤷ provides
            std::ifstream : like cin
int main(int argc,        std::ofstream : like cout
         char *argv[])                (like fprintf)
{
    std::ifstream    model_ifs;  (model input
                                  file stream)
    std::ifstream    hits_ifs;  \ (hits file)

    // assume model filename is in argv[1]
    "  "     hits     "   "  " argv[2]
              ⤷ was just a sequence of ray directions
                        (vec_t's)

    // check for argc == 3
    model_ifs.open(argv[1], std::ifstream::in);
    // should check for 'good' status
    model_ifs >> model; // model reads itself
              ⤷ ifstream.operator>>(model_t& model)
                              ↑
                   your model_t object
                   has to have this operator
                         overloaded
    model_ifs.close(); //(don't forget to close file)

    // debugging
std::cout << model;   // get model to
                         print itself


// same for hits file
// open hits file, then loop for
// each dir vector found therein
                              a vec_t                vec_t function
                                                     that returns vec_t
while ((hits_ifs >> dir).good()) {
              ↑                                                   find_closest
         returns ifstream&      ifstream hits_ifs                 can alter contents
         (return s)                                                      ↑
     ;                                                                double&
                           vec_t
   obj = model.find_closest(pos, dir.norm(), (object_t *)NULL, dist);

   if (dist > 0) {                                   has to return vec_t
      // should probably check for obj != NULL        that find_closest
                                                      would have set
      std::cerr << " Loc = " << obj -> getlast_hit() << std::endl;

   }

   ;
   ;
  } //end while

  hits_ifs.close();

} // main
```

eventually, model_t
also has lights list, camera

- model_t :
  - private data members: mats, objs lists

- std::istream& operator>>(std::istream& s, model_t& rhs)

works for cin
or an ifstream
because istream $\xrightarrow{isa}$ ifstream

```
{
    :
    std::string
    while (!s.eof()) {
        if ((s >> token).good()) {        std::cin >> model;
            if (token == "material") {
                s >> (mat = new material_t());
                // add mat to mats list
            }
            if (token == "plane") {
                s >> (obj = new plane_t(token));
                // add obj to objs list
            }
            if (token == "sphere") {
```

mat: pointer to material_t

new: allocates mem for new

material_t(): material_t constructor

obj: object_t *

plane_t(token): plane_t constructor

new plane_t(token): result of new

// a plane_t *

```
}
if (token == "camera") {

}
if (token == "light") {
}
}//while
return s;
}
```

Friday, October 22, 2010
9:44 AM

return type (ptr to object_t)

```
object_t*   model_t :: find_closest(vec_t & pos,
                                    vec_t & dir,
                                    object_t * last_hit,
                                    double & retdist)
{

    objs.reset();
    // basic list iteration

    while( objs.not_end()) {
        obj = (object_t *) objs.get_data();
          :
          :
        if (( dist = obj -> hits(pos,dir)) < 0)
                   :
                   :
          :
          :
          :
    return(closest);
}
```

invoke polymorphic hits fn

plane_t must have one
sphere_t " " "

[illegible] virtual one

}

sphere_t

object_t now have virtual one