More Structures

11/9/2007

Opening Discussion

- Let's look at solutions to the interclass problem.
- Do you have any questions about the assignment?

struct Assignment

- When you do an assignment from one struct to another, it copies all the contents over.
- It is like doing an assignment for each element in the structure one after another.

Nested Structures

- As we saw last class, we can nest structures inside of one another. This allows us to build more complex structures.
- Let's look at a possible usage of this where we build a scene with several geometric objects in it.

Arrays in Structures

- When we declare an array inside of a structure, that array adds to the size of the structure.
- For this reason, structures can become very large.

Pointers in Structures

- Pointers in structures are a bit trickier than arrays.
- Image a structure where we have a dynamic array that we store as a pointer. Now assign that structure to another variable. What does this produce?
- Structure assignment does a copy, but it is not a deep copy. So pointers have to be treated with extreme care.

Passing and Returning structs

- Structures are passed and returned by value.
- So when you pass a struct to a function it gets a full copy. When you return a struct a full copy comes out.
- Returning structs will effectively allow you to return multiple pieces of information. It isn't as efficient as passing by reference in most situations though.
- As we discussed last time, a structure will normally be passed by reference for efficiency. If it shouldn't be changed make it const.

Multiple Compilation Units

- Large programs don't exist in a single .c file. We need several of them and we will compile them each separately.
- There can only be one main, but this allows us to effectively write library methods and stick them in other .c files that are shared among many programs.
- The functions we want to call from the outside need to have declarations in a .h file.
- You can include your own .h files using #include “myfile.h”. The double quotes tell the compiler to search in the current directory.
- We compile the .c files separately and then link them.

Makefiles

- Compiling separately and linking is a pain. For this reason, we have make files.
- Also, compiling large projects takes a long time if you have to compile everything. The make file will check dependencies and only do what is needed when a few things have changed.
- The rules in makefiles don't have to compile things, but they can. Let's look at a sample makefile with some compile rules and some other rules.

Minute Essay

- If you have a struct with a variable length array that you have malloced, what do you need to do when you copy the struct?
- Interclass Problem – Do problem 30 on page 814.