

## Re: Haven't done anything real with OOP yet.

**Source:** <http://coding.derkeiler.com/Archive/General/comp.object/2004-10/1398.html>

---

**From:** H. S. Lahman ([h.lahman\\_at\\_verizon.net](mailto:h.lahman_at_verizon.net))

**Date:** 10/27/04

Date: Wed, 27 Oct 2004 20:05:23 GMT

Responding to Hansen...

> *I was hoping my little project would be a vehicle for OO study through  
> practical application. When I look for software design books, I mostly  
> find handbooks for specific applications, the web, My First Book on  
> Programming type of books, and then there's the set that seem like  
> they're intended for people with a lot more experience than I have and are  
> priced at standard textbook levels. If it were a Dover book for \$12.95 I  
> could just buy it and take a chance, but I'm not as free with an \$88  
> textbook.*

I agree the selecting a book is rather subjective, so my blog provides some criteria to use when browsing the local bookstore or library.

>>...*the real problem here is a graphic engine. That is an entirely  
>>different problem space from clams running along beaches. That is, the  
>>semantics here is independent of the game (i.e., a Sprite can be a clam  
>>or a B2 bomber). Game software is very state-of-the-art algorithmic  
>>software, which is generally not a good application of OO development.  
>>That's because the benefits of OO development lie in software  
>>maintainability when requirements are volatile over time. Graphics  
>>algorithms, though, are mathematically defined (i.e., they are not  
>>unique to the specific customer problem (game)) so they are invariant  
>>over time. About the only OO benefits one can obtain using OO  
>>principles is large-scale modularity and reuse (i.e., at the subsystem  
>>level).*  
>  
>  
> *That's not really what I expected, since OOP seemed well suited to  
> graphical interfaces where you might have a window with a bunch of  
> controls to manipulate. I thought, in the case of a game, each object  
> running around (a monster or whatever) would be ideally represented as an*