avl Library – Development Journal

# 10-31-11

I finally have a break in schoolwork – it’s been just over a month since I’ve had time to work on the library. Not surprisingly, I don’t remember where I was when I last worked on it. As such, I’m beginning this development journal to keep track of what I’ve been doing and what I need to do. School isn’t going to let up for the next two years, so I need a system to make these cold starts quicker and easier.

The last thing that I did was add processing for newline characters in the utility::TextBox component – it now properly formats for newline characters. I don’t believe that I tested this functionality very thoroughly at all, but I’ve decided to move on to something else instead. There are some major bugs which were identified in the Arkanoid Clone prototype:

* Exception thrown in debugger when program exits after focus has switched to a different window and then back to the application.
* Pressing ctrl+alt+delete makes the application go black, and it doesn’t recover.
* In fullscreen mode, the application is just a black screen. Input still works.

These are serious bugs – it’s lazy to just keep adding features instead of dealing with them. As such, that’s the current task. Once these bugs are dealt with, then I need to go back and clean up the code a bit. I know that the view::BasicRenderer::RenderSprites() code is jumbled and confusing; it definitely needs to be refactored into multiple functions.

* Identify the major areas where refactoring is necessary. First is view::BasicRenderer::RenderSprites().

## Step 1:

I need easier control over the creation of a window. I’m adding a LeastSquaredProfile() method to view::Direct3DRendererBase which will find the best-fitting display profile for the requested parameters.

# 11-3-11

So…I finished the LeastSquaredProfile() method mentioned in the previous entry, except that I haven’t decided what exception to throw when it fails. So I started designing a better, more comprehensive hierarchy of exceptions to use in the library. Partway through, I realized that I need a better documentation solution, and set up Doxygen. So now I’m:

1. Switching over to Doxygen style comments.
2. Finishing an exception hierarchy.
3. Finishing the LeastSquaredProfile() method.
4. Fixing bugs!!!

# 12-7-11

Fall quarter is over! I’ve continued converting all header-file comments over to Doxygen style comments, and I’ve been making use of Doxygen’s built-in To-Do list functionality to document areas that need refactoring and/or bug fixes. I’ve been refraining from making any significant changes to the code because I don’t want to introduce any new bugs.