Top down from input to end use of function

**Main**

First step is to check if we have any modifiers (a different file name, if it’s text only, the start level and seed (implemented later).

There are stack variables set to the defaults for each of these which are:

* Filename = sequence.txt
* textMode = false
* startLevel = 0

Next we create the board. This doesn’t require any input, it should just work.

Next we create the textDisplay. Again this doesn’t require any input, it should just work. (right now you have (int r, int c) as the arguments…. But since the display should be always the same size… we can probably get rid of these and just have textDisplay();)

Next we create an xWindow based on the value of textMode. This variable also needs to extend down through other functions to activate proper updating of the graphics… work on this later.

Next we create the nextBlock class. This class is going to need to know the current level and the filename (if the level is 0 it will use this… if not, it still may if the level gets decremented during play to level 0).

**nextBlock**

If the level is 0, nextBlock will read from the filename, else it will generate blocks based off the proper ratios described in the requirements. Really, the only method we need to care about is getblocktype() that will generate the character of the block we need to make.

Next we create two new blocks, the first two blocks of the game.

We finally create an interpreter object.

Now we have everything we need to start the name… the next step is commands.

Interpreter has two methods we care about, effectuating a command and effectuating multiple commands at once, based on right or 3right (both possible inputs). We have an if/else that separates single calls from numerical first calls.

The method takes in the current block, the next block, the board, the display and the xWindow if needed. The interpreter does **ALL** the calling on the blocks.

**Interpreter**

If the command is right for example, the interpreter will decipher this whether or not it is inputted as “rig” or “right” excreta… and call the respective function in board that will return either true or false, depending if it worked or not. We use that Boolean output to decide whether to update the display and draw to the xWindow (I think).

At the end of every loop for the cin, we reoutput the board with the new changes and set current block to the next block and call nextBlock() to get our new next block. We keep doing this until we a) get a ‘a’ back from the newBlock generator (this means the input stream from the file has need exhausted), we lose (work on triggering this somehow) or an EOF is given.

For tomorrow (Wednesday) we need to have at least the ability to take in a sequence.txt file and see and drop the blocks into an updating array.