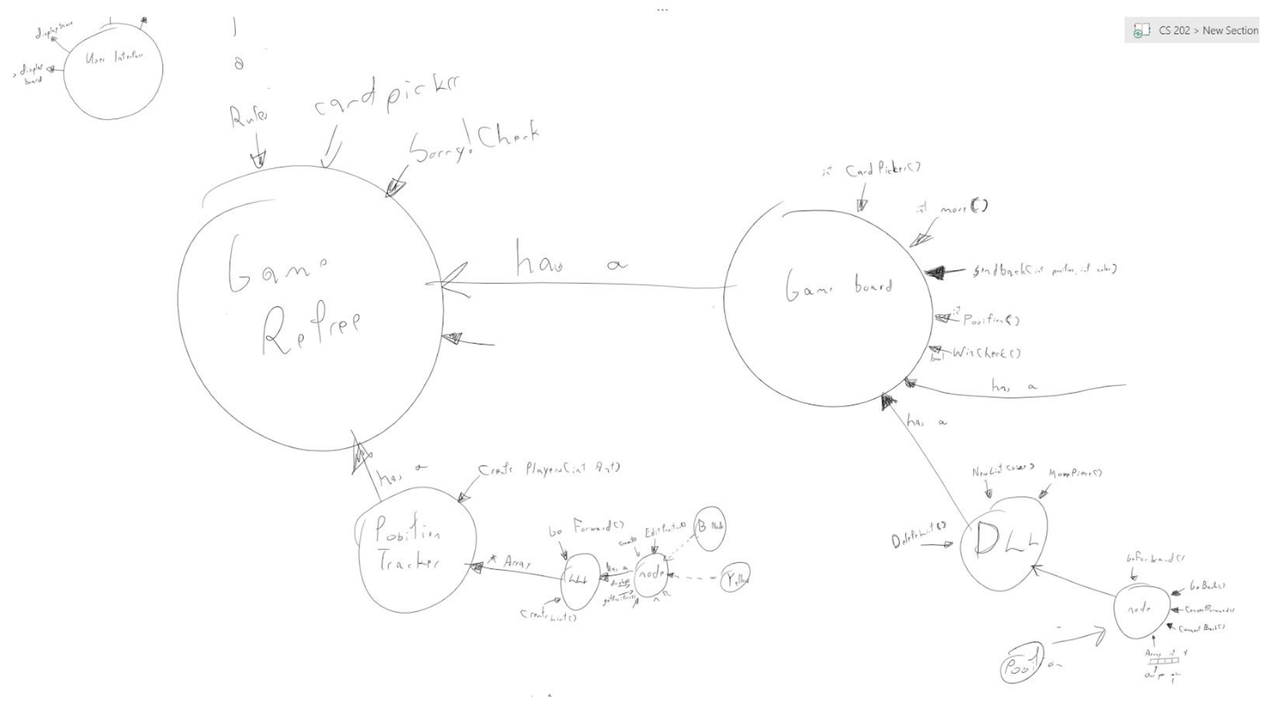
Design Write up

I think my design is pretty good within the requirements of the program. If I was doing this for fun I would not put half the stuff in here since it is so overkill. One thing I'm really happen with how it came out was my game board, originally it was just going to be a doubly linked list, and I tried to figure out the best way to have all four players on it at once, I considered making four list but I felt like that would increase how complex I would need it to be and be memory inefficient. So I made it circular to help solve some problems. Mainly now its super easy to move a node past the last node, or behind the first node. You just make it move like normal! That means it can handle negative numbers very easily. I did have to get rid of safe zones, without making a new list I would be able to handle them well. Instead I added in the option to roll the dice and skip your turn. The tree came out pretty well too! In that it worked. For some reason I could NOT find the source of the the memory error, its saying the nodes are no initialized but they are! This is really the biggest complaint about the tree(oh and how long it took to make). The tree works, no memory leaks. Adjusting the height factor could be made more efficient. Right now it goes down every node in the tree until it gets to the bottom and recalculates. This means each layer of the tree recalculates the height the height over and over again instead of using the values from the lower nodes. This makes it accurate(which was a problem I had with a older version of the height management system) but it makes it slower. From that sections its really the only thing I want to rework. Otherwise it works wells, gets the value from the root, deletes the root, and then creates a new node to insert. The last big independent section of the project was the array of linked list, I'm ok with how this turned out. The issue really is that for this one instead of using 1-4 for colors and 1-3 for pawn selection I used a character array and a actual name to organize the list. This meant when I was putting everything together I had to convert from char * to ints a couple times. Otherwise its ok, it serves its purpose. It did have a small issue with negative characters that was throwing everything off but it's fixed now. If I were making this as a product I would use this as the actual game board, nothing has to be moved, everything is quick to each node, and its simple. Not my favorite part, but its also not my least favorite. Moving onto the last section, the controller. This guy was fun to make, it brings everything together forcing them to work side by side, and it does so really well! Because I decided to make everything as stand alone as I could(to help with its OOness) everything clicked together like lego, with the controller just acting as the medium between them. The big functions are there only because some of the logic required to move required a lot of steps to make sure that it was good, nothing was there , to draw cards, ect. I wish I could shrink those down more but since they work, I had to to leave them. I did not make any massive changes to the project when I was making it, I spent a lot of time designed it and deciding how things were going to work so i think that's why nothing major needed to be changed.The logic in the tree was overhauled three times, but that's the only thing that was really changed, but its inputs and outputs were not touched. Overall I think this program came out great!(except for that annoying memory error), even if the requirements forced it to be inefficient.

GDB

As usual GDB saved my life a bunch of times. I would not have finished the tree without being able to step around stuff and see the exact value. Both times when I reworked my AVL tree logic I slowly went through the tree to see where the issues were. The first time I made my rotate function I ended up making it overly complicated which resulted in its doom, GDB pointed out a variety of issues and I ended up nuked the entire tree and starting over. The second time it was not complicated, but my fundamental logic was wrong, so the rotate functions had to go. In Version 2.5 the delete functions were working. I was having no luck figuring it out until I used GDB to trace the program carefully and determine what was wrong. Ultimately it was that I needed to pass in a extra node into the rotate functions so I could return it. This one took me a good half day or so and would have been three times more without GDB. It is ultimately responsible for why I have two inputs for my left and right rotate functions, and why on left right and right left I have to change the the temp node is. Without it I doubt I would have figured out what was going on in that silly tree. It helped a little bit here and there with other issues like the linked list and the circular doubly linked list too, although nothing super big there. Just some small glitches that turned out to most be due to how the negative cards were being handled(not well at first). At one point there was a glitch where I originally had the chance cards be every ten slots, but that meant all the homes were on chance cards. The pawns would roll, have the cards added, and end up with -1 and instant win since the win condition is one behind home. I think eventually I would have figured it out but GDB helped me find out the issue alot faster. The stepback feature was a little helpful, it didn't seem to work sometimes sadly but when it did it was good. Valgrind also helped a little bit, I was having issues with double free and deletes with my AVL tree and valgrind helped me pinpoint those down along with various memory issues and leaks. (But no the stupid one with the avl tree!!!!). One I had that turned to be very silly was a memory leak of 32 bytes, after I enabled some options valgrind pointed it out exactly to me, I was deleting things wrong in my array of linked list, I was saying delete array instead of delete [] array! I felt a little bit dumb after that one. Anyways, these two programs helped me out a bunch(thank you creators of GDB I own yo0u one).

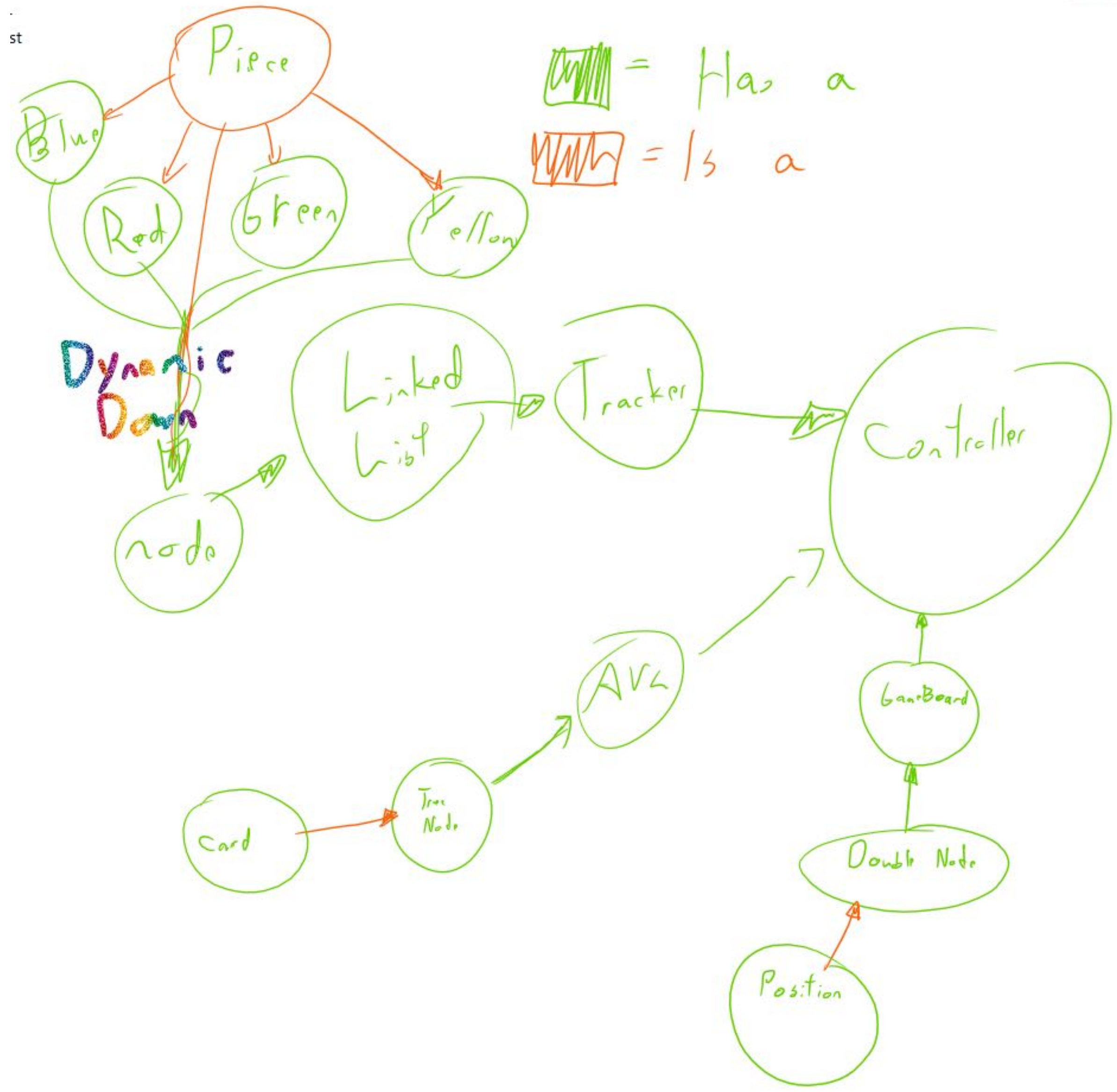
Original Design C++



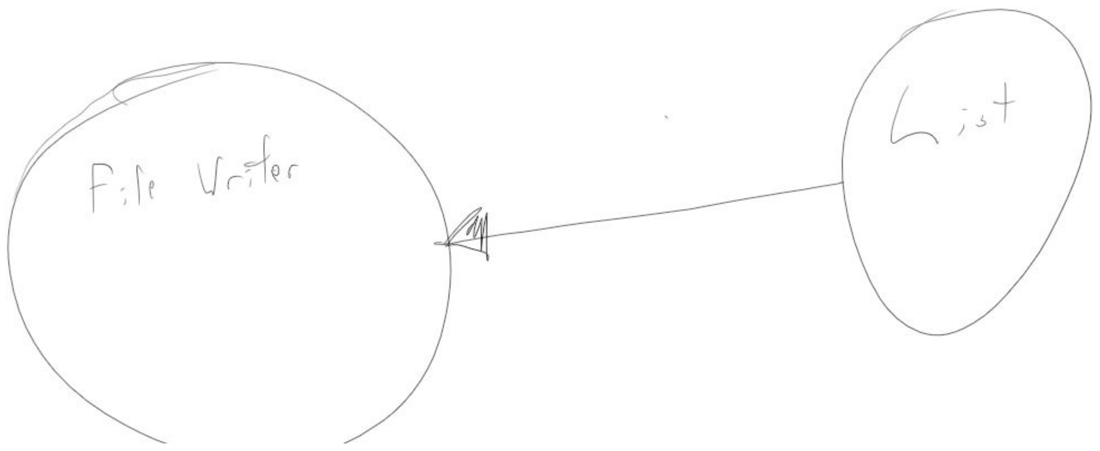
Final Design C++

.

st



Java(File Writer has a list)


File Writer
List