Functionality

* Takes arguments from user for the chance of fire spreading and refresh rate.
* Creates an array of Tree objects that derive from the Tree parent class.
* Gets Weather attributes and applies them to the chance of fire spreading, the chance of a tree randomly combusting and tree regrowth values.
* Loops through the map array and checks each Tree status and neighbours’ status; setting their next status at the fire chance rate if the neighbour tree is on fire.
* Loops through the map array again and updates all statuses to next statuses and prints out the grid to console.
* Regrows and combusts trees according to a random number generator and weather conditions.

Evaluation

I started this project off by drawing out a general idea in my head about how I wanted the project to look like. From this I created a class diagram and continued to improve on it, eventually ending up with what I have now which is pretty much what the implementation looks like. I wish I had explored more design architectures during this step as I feel my final layout is a bit messy and I can already see many improvements to the structure such as using templates and lambdas. Once I had the class diagram I began working on the basic implementations such as creating Tree objects and the map array as well as simple burning logic. Again I wish I had spent more time with the basic structure of the program as it would have solved a few issues I ran into later on. I then continued to implement weather, regrowth and combustion effects which were simple enough but I ran into some issues with the random number generator and seeds but eventually fixed those with some coding broscience.

I wanted to add things like wet patches, wind direction and speed but due to the architecture design it was hard to implement in the current state. I could have done this by initialising the map as a 2d array on top of another 2d layer; the top layer being the Tree objects and the lower layer being a map of wet patches and then compare the x, y of both layers and apply respective calculations.

The strengths lay in the user being able to pick certain variables so they feel like they have an impact on the end result – the random selection of weather conditions also affects a lot of the program’s functionality.

Appendix

Instructions on running the simulation are as follows:

1. Open Release\ForestFire.exe or run from the IDE.
2. Enter in an integer between 1 and 100 to determine the raw chance that a neighbouring tree will catch fire.
3. Enter in an integer between 100 and 500 to determine the refresh rate of the simulation in milliseconds.
4. Watch the simulation progress; using ESC to exit out of the program.