Mattiece Oujua, Joseph Williams

11/26/18

Watkins

Data Structures & Algorithims

Analysis

To start, we decided to study Minimax heavily, and it can be seen in the final code that it was focused on the most. Alongside Minimax, we also watched numerous videos on design trees, but we ran into some issues during implementation.

Design

For the design aspect, we decided to have two main .cpp files and a .h file. While we decided to add more at a later stage, the initial thought process of division of functions was similar. This was a division between “Board Functions” and “Player Functions”. This design choice was simply made for easier readability, as well as it being a requirement.

Implementation

Overall, there were few insights other than learning how to implement Minimax, but there were a few challenges such as refactoring the original code as we were using a version that was not compatible with what we were planning to do. Resolution-wise, there weren’t many issues. In our main function, we call the game constructor, as well as initializing gameplay against a CPU.

Testing

During testing, we simply used a stub/driver method, which is highlighted in the first part of the project. This is quite simple, but the testing was effective.

Reflection

Overall, the project was slightly confusing, but was a good challenge overall. It seems as though it proves to be a good intro to group coding, and the main complaint would be based on the timeframe it was assigned. Otherwise, the project was great for learning and experience.