**COMP10050-Software Engineering Project 1-Assignment 2**

**Aidan Mac Neill (16349586) & Dalen Gumpal (16367713)**

**GIT Repo Link:** <https://github.com/AidanMacNeill/COMP10050>

For this assignment, we were expected to create the next part of the game CrossFire as part of a group, focusing on the slot, player and attack representation. We decided that working together was the best option, as this would limit communication differences and issues, as well as limit the issues between developing the project across two different operating systems. We then were able to solve issues together, including choosing struct statements over enum statements for the players due to the greater ease of use, how to represent the slots and their types, and implementing the attack logic.

We decided on certain times to meet up when we could to perform the task as a group, which I feel definitely helped us to complete the project easier than otherwise would have been possible. Closer to the deadline, I then took the task of debugging the code for minor issues, fixing formatting and implementing checks for edge cases with regards to the movement and attacks of players due to my increased use and knowledge of the C language. All in all, I feel we worked well as a group and we were able to get things done to a high standard and on time, and I enjoyed working on this project.

**Design Choices**

Throughout the course of the assignment, we had to make multiple choices for the design of the project and how it would run. One such choice was between using struct and enums for player representation. We decided that a struct statement would have been better as we had more knowledge of them, as well as the fact it would keep the slots and players under the same data type, which was very helpful later on in the project. We also chose to use Boolean values for the slots being occupied or not, as this helped to simplify the movement, attacking and limits of the players and slots, including player deaths. One difficulty we encountered was that the compiler we used with Eclipse, Cygwin, recognised \_Bool as the declaration for a Boolean variable, unlike bool, which we were used to.

Another place where we had to use different design choices was when a player decided to attack, but they couldn’t. We had to think of a way to allow the for loop to allow them to choose again, so we implemented a decrement counter to do this if the attack function returned that they couldn’t attack any players nearby.

The last major design choice we made was how to implement the slot types being assigned to each slot. At first, we used strcpy with each case and type, but instead we thought that, due to the length of the program, it would have been easier to represent the slot types in a character array, and refer to each value in it with its index. This also made the stat changes between slot types easier as we could use strcmp and refer to that element in the array, rather than type it out for each case.