

C++ BattleShip Project 보고서

20132651 이성재

1. Ncurses 개발환경 구축 및 예제 실습
2. 설계도 분석 및 헤더파일 / CPP 파일 작성
3. Makefile 의 작성 및 시행
4. 프로젝트를 진행하며 느낀 점.

1. Ncurses 개발환경 구축 및 예제 실습

가장 먼저 BattleShip Project를 받고 시행한 것은 주어진 ncurses라는 환경의 구축이었습니다. 처음 접하는 개발 환경이었기 때문에 처음에는 주어진 대로 리눅스의 우분투 환경에서, 다음으로는 맥북의 OSX 환경에서 개발 가능하도록 노력하였습니다.

OSX 환경에서 우분투 OS를 설치하기 위해 Virtual Machine을 사용하고, 가상 환경에서 개발을 시작하였습니다. 일차적으로 우분투 환경에서 ncurses 개발을 경험해 본 다음, OSX 환경에서 ncurses 를 설치할 방법을 찾아보았습니다. 최종적으로 Brew를 이용해 ncurses를 설치할 수 있었고, 익숙한 OSX에서 개발을 시작하게 되었습니다.

```
Sungjaeui-MacBook-Pro:ex sungjae$ brew install ncurses
```

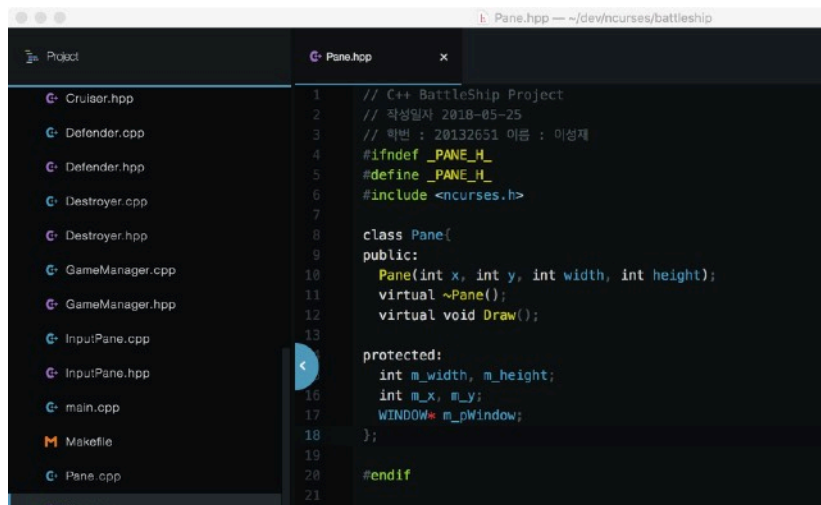
기본적인 개발 환경이 구축된 다음, ncurses에 대한 이해를 높이기 위해 주어진 예제를 수행하기 시작하였습니다. EX1 부터 EX4 까지 진행하며 어떻게 terminal 을 통해 Graphical 한 표현이 가능한지 이해할 수 있었습니다.

[illegible]

2. 설계도 분석 및 헤더파일 / CPP파일의 작성

ncurses에 대한 기초적인 이해가 끝난 다음, 주어진 설명에 따라 BattleShip 프로그램 개발을 본격적으로 시작하였습니다. 먼저 주어진 코드를 동일하게 작성해 보며 성공 여부를 테스트 했습니다. 이 과정에서 생성자가 작성되어 있지 않아 오류가 발생하는 등의 문제를 조금씩 해결하였습니다

이 과정에서 가장 많이 배운 것은 헤더파일의 중복이 일어나지 않도록 작성하는 법이었습니다. 이전까지는 대부분 하나의 CPP 파일에 코드를 작성하여 중복이 일어나지 않았지만, 다중 파일로 작업하다 보니 헤더파일에 중복이 일어나 컴파일이 되지 않았습니다. 실제로 ifndef 문을 이용해 헤더파일의 중복을 방지하는 법을 알게 되었고, 그 중요성도 깨달았습니다.

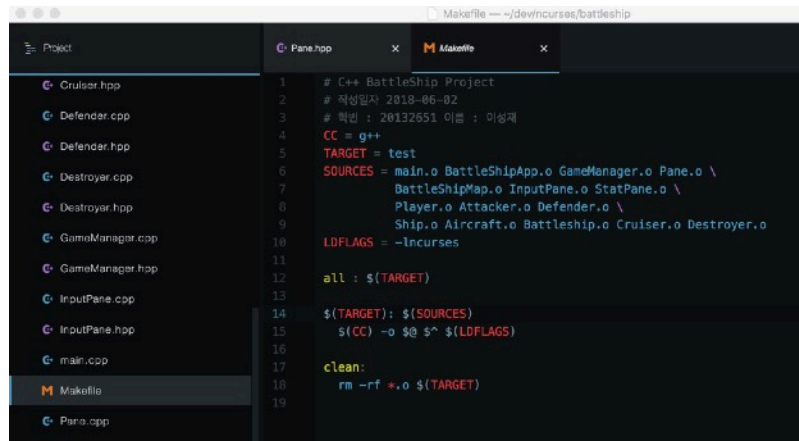


기본적으로 주어진 코드를 모두 작성하고 난 다음, 이 코드만으로는 BattleShip 프로젝트를 완수할 수 없다는 점을 알았습니다. 이 때 처음으로 UML 구조를 보고, 공부하기 시작하였습니다. UML 구조를 처음 접하여 생소했지만, 모두 게임을 구성하기 위한 핵심 요소들이란 점을 구현하는 과정에서 알게 되었습니다. 대부분 앞에서 구현되었던 구조를 다시 활용하여 유사하게 구현해 보았지만, 실력이 부족해 구체적인 내용을 구현할 수 없었습니다.

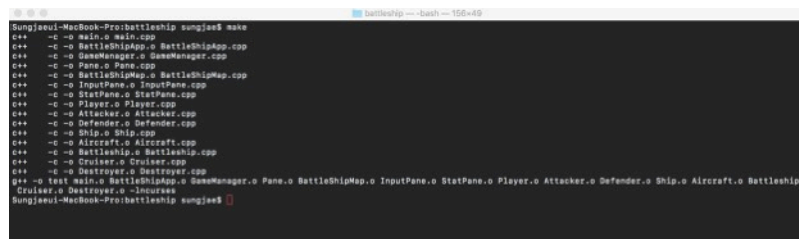
3. Makefile 의 작성 및 시행

항상 CPP 파일의 시행에서, Atom 의 Script 를 이용한 간단한 실행만 하다가 g++ 컴파일을 더 편리하게 해 주는 Makefile을 처음 사용하게 되었습니다. g++ 명령어를 이용한 컴파일 자체가 익숙하지 않았기 때문에, 이를 이해하는 데 시간이 많이 걸렸습니다. 관련 자료를 검색하여 공부하면서, Makefile이 크게 어렵지 않으며, 상당히 편리한 도구라는 것을 알았습니다. 나아가 자체적으로 매크로 기능이 존재하며, 이를 활용하면 적은 코드로 많은 컴파일이 가능하다는 것을 알게 되었습니다.

이렇게 배운 내용을 바탕으로 Makefile을 작성해 보았지만, 정상 작동까지는 꽤 험난한 길이 되었습니다. 이 때 새롭게 알게 된 사실은 헤더 파일에 대해 제가 잘못 이해하고 있었다는 점입니다. 지금까지는 특정 헤더파일을 include 할 때, 헤더파일을 포함한 CPP 파일을 include해야 한다고 생각했습니다. 이는 g++을 이용해 링크 컴파일 하지 않고, Script를 이용해 CPP파일만 실행시켰던 잘못된 습관 때문이었습니다.



잘못된 습관에서 비롯된 문제는 찾는데 상당한 시간을 소요하였고, 결국 프로젝트가 거의 끝나갈 때 즈음 알게 되었습니다. 결과적으로 프로젝트를 완수하지는 못하였지만, Makefile을 작성하는 법, 헤더파일의 활용법 등은 확실하게 배우게 되었습니다.



4. 프로젝트를 진행하며 느낀 점.

프로젝트를 처음 접했을 때, 부족한 지식이 많아 크게 어려움을 겪었습니다. ncurses 환경부터 g++ 컴파일, 헤더파일의 작성 등이 대부분 서툴렀고, 이를 기초부터 공부하며 많은 시간이 소요되었습니다. 하지만 이를 통해 어느 부분에서 C++ 실력이 부족한지 크게 알 수 있는 기회가 되었으며, 생각보다 모르는 분야가 많았다는 점을 느꼈습니다.

무엇보다 이 정도로 많은 CPP / HPP 파일이 있는 프로젝트는 처음 진행해 보는 것으로, 상당히 체계적인 구조가 필요하다는 점을 많이 느꼈습니다. 구조화된 프로젝트를 처음 경험해 보며 앞으로의 프로그래밍에 있어서 UML 작성 등이 중요하겠다는 생각을 했습니다. 비록 프로젝트를 시간 내에 완수하지는 못했지만 마무리 할 수 있도록 도전해 보겠습니다.

