|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **주차** | 2 주차 | **기간** | 23.07.03~ 23.07.09 | **지도교수** | (서명) |
| 이번주 한일 요약 | 1. 전문가를 위한 C++책(C++ 공부1) 이어서 공부  2. 게임 프로그래밍 패턴 공부(책) 이어서 공부  3. 언리얼 개발문서 정독,이해,분석 [ X ]  4. 언리얼4 C++책(저자 이득우) 마무리 [ 90% - > 0 % ] | | | | |

작성자(학번 이름): 2018182037 조상준

팀명:

<상세 수행내용>

1번에서는

Chapter2 스트링과 스트링 뷰 다루기

- 스트링에 대한 C와 C++의 차이, C++의 std::string, std::string\_view의 용도, 로 스트링 리터럴 등을 공부하였다.

Chapter 3 코딩 스타일

- 코드 문서화의 중요성과 다양한 주석 스타일, 코드 분해, 명명 규칙등을 공부하였다.

Chapter4 전문가답게 C++프로그램 설계하기

- 프로그램 설계의 중요성, C++ 고유의 설계 고려사항, 설계를 위한 핵심 개념인 추상화와 재사용성등에 대해 공부하였다. -> 어떤 프로젝트를 진행하더라도 가장 먼저 소프트웨어 설계 작업부터 해야 한다.

2번에서는

Chapter2 명령 – 명령 패턴은 메서드 호출을 실체화한 것이며, 다시 말하면 함수 호출을 객체가 감싼 것이다. 예를 들어 입력키 변경에서 쓰인다.

Chapter3 경량 – 경량 패턴은 어떤 객체의 수가 너무 많아서 좀 더 가볍게 만들고 싶을 때 사용한다. 객체 데이터를 모든 객체의 데이터 값이 같아 공유할 수 있는 데이터를 모은 데이터와 나머지 데이터 두 종류로 나눈다.

Chapter4 관찰자 – 모델-뷰-컨트롤러 패턴에 기반이 된다. 관찰자 패턴은 객체 사이에 일 대 다의 의존 관계를 정의해두어, 어떤 객체의 상태가 변할 때 그 객체에 의존성을 가진 다른 객체들이 그 변화를 통지 받고 자동으로 업데이트 될 수 있게 한다.

Chapter5 프로토타입 – 프로토타입 패턴은 객체를 생성하는 데 비용이 많이 들고, 비슷한 객체가 이미 있는 경우에 사용하는 패턴인데 코드가 두루뭉술해서 이해하기 어렵다는 큰 단점 때문에 거의 찾아보기 어렵다.

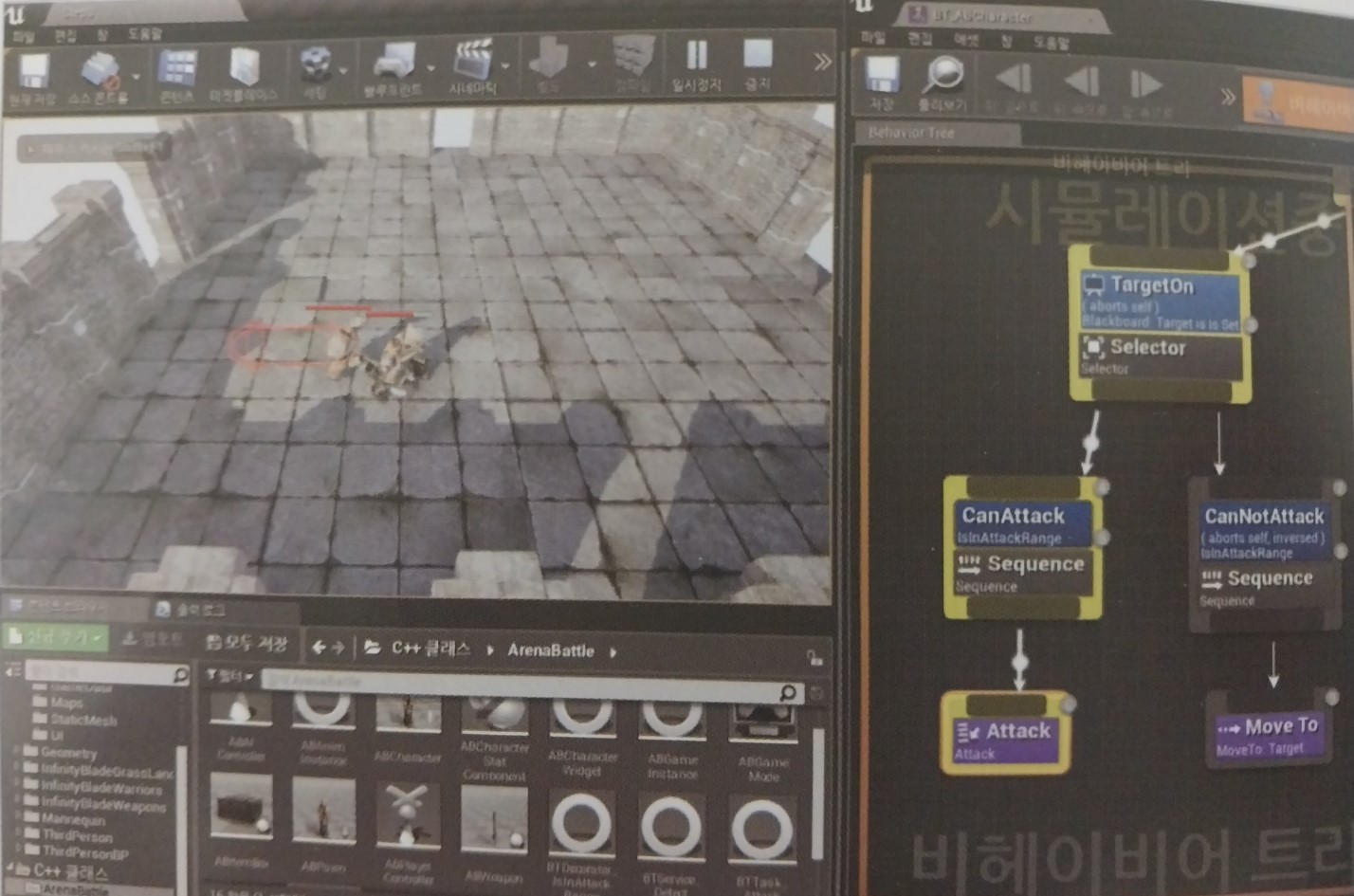
4번에서는

Chapter11 게임데이터와 UI위젯에서 Excel에 들어있는 HP, Damage 등에 데이터들을 캐릭터들에게 적용하고 체력이 0이되면 적이 쓰러지는 애니매이션을 하도록 하였으며,



UI에셋을 생성하여 HP바를 캐릭터 위에 나타나게 하였다.

Chapter12 AI컨트롤러와 비헤이비어 트리에서는 적 캐릭터가 일정 범위안에 Player 캐릭터가 있으면 쫓아오게 하였으며, 일정 거리에 다다르면 공격하도록 하였다. 이것을 behaivor tree를 이용하여 적용하였다.



Chapter13을 공부하던 도중 잘못된 클래스 생성으로 인한 빌드 오류로 인하여 수정해도 계속 빌드 오류가 나타나 Chapter13,14,15는 하지 못하게 되었다.

* Github과 같이 사용하거나 Chapter가 끝날 때 마다 압축해 백업을 꼭 해야 한다는 것을 깨달은 1주였습니다.

-Chapter12까지 진행한 코드들도 github에 올려놓았습니다.

|  |  |  |  |
| --- | --- | --- | --- |
| **문제점 정리** | . | | |
| **해결방안** |  | | |
| **다음주차** | 3주차 | **다음기간** | 2023.7.10 ~ 2023.7.16 |
| **다음주 할일** | 1. 전문가를 위한 C++책(C++ 공부1)  2. 언리얼 소스코드 분석(Actor)  3. 게임 프로그래밍 패턴 공부(책) | | |
| **지도 교수**  **Comment** |  | | |