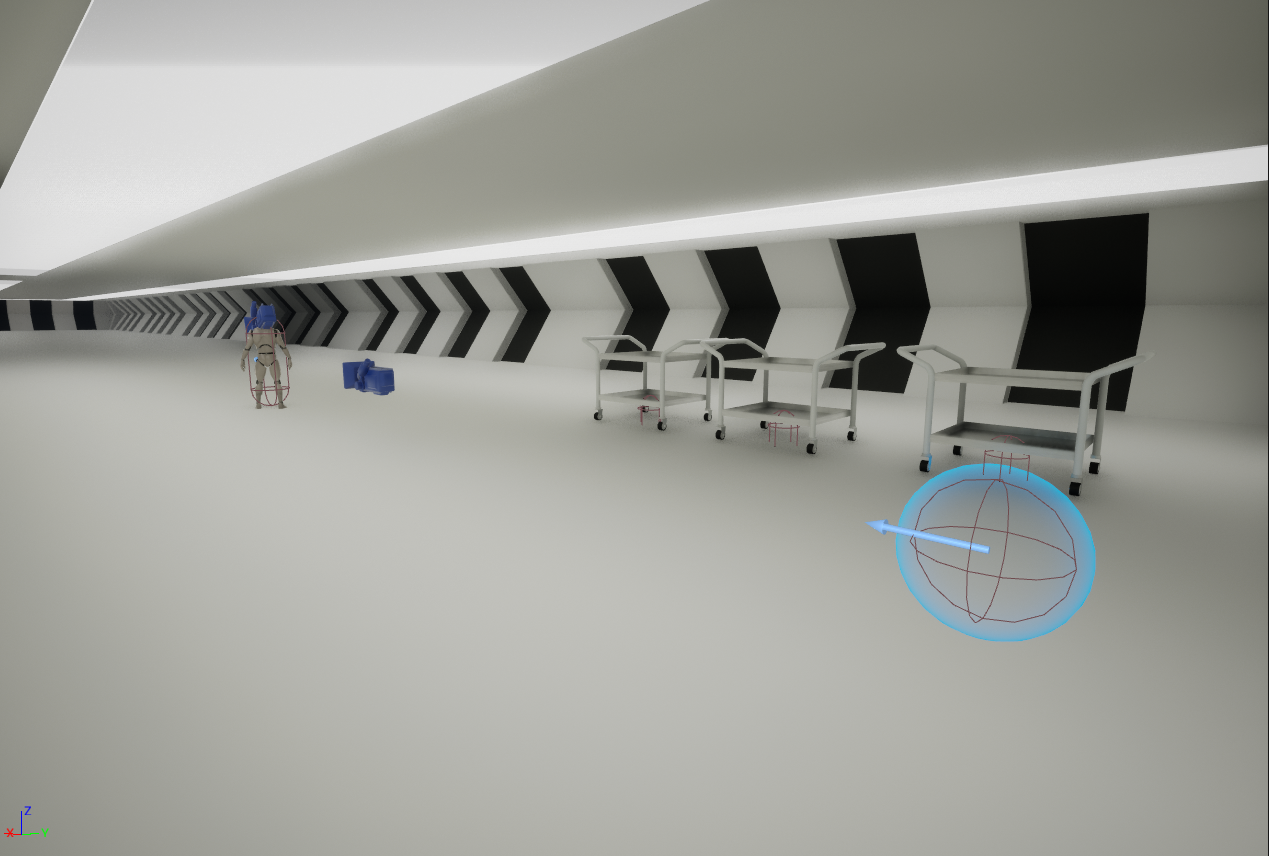
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **주차** | 16주차 | **기간** | 2023.10.24~ 2023.10.30 | **지도교수** | (서명) |
| 이번주 한일 요약 | 구현된 무브먼트 C++로 변경 계속, 공동 프로젝트로 만든 내용 이전 | | | | |

<상세 수행내용>

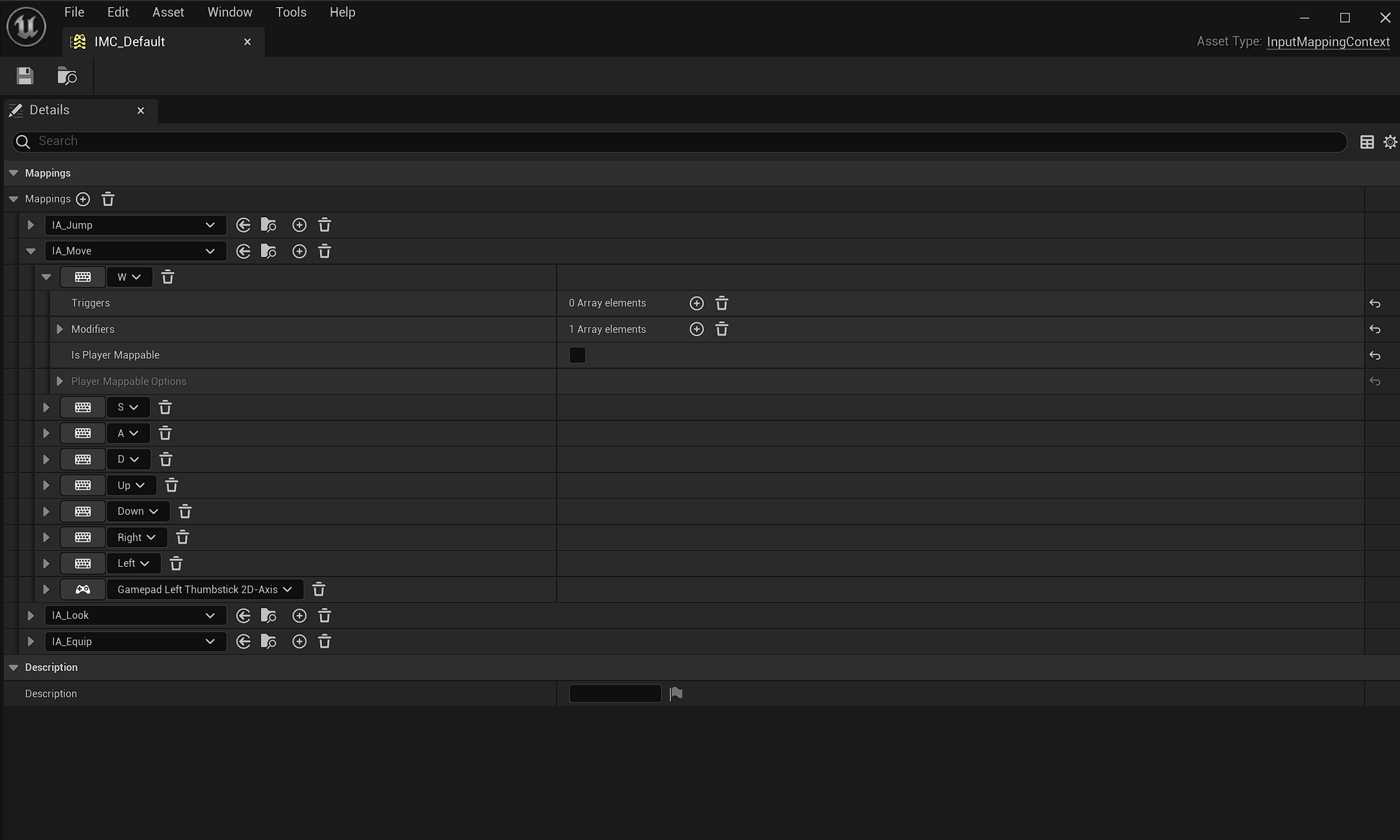
기존에 구현하던 내용들을 새로 만들어진 'EatUpAll' 프로젝트로 옮겼다. 앞으로 구현한 내용은 해당 프로젝트에 추가할 예정이다.



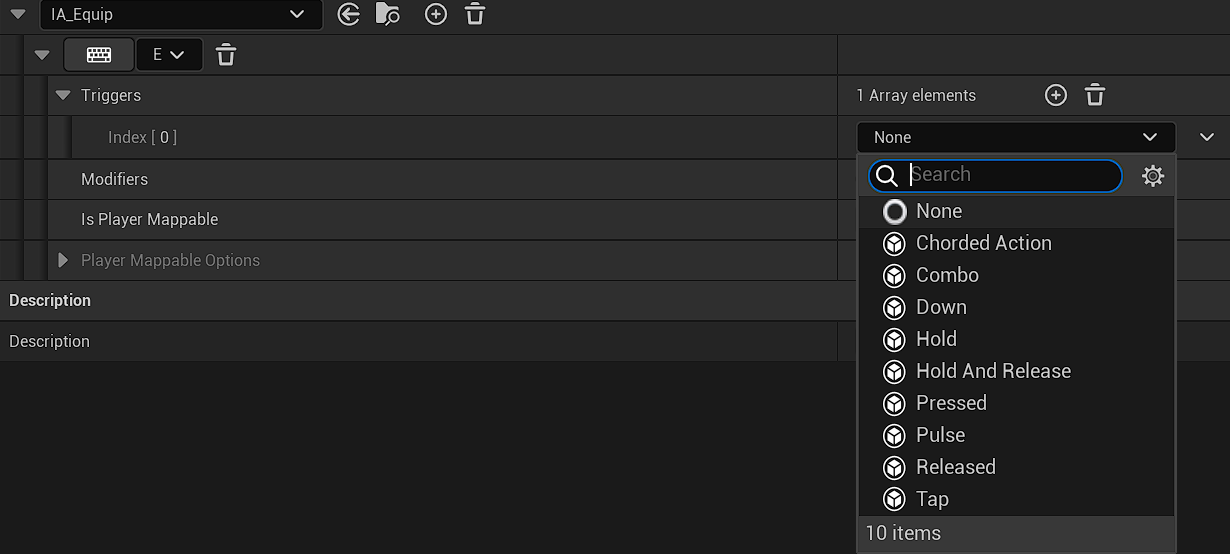
이번 주차는 저번 주차에 이어 만들어진 블루프린트 코드를 C++ 코드로 하나씩 변경하는 작업을 이어서 했다.

클래스 작업에 앞서 inhanced\_input의 기능에 대해 살펴보았다.

inhanced\_input은 사용자가 원하는 행동을 먼저 만들고, input mapping context를 통해 행동과 매칭시키는 방식을 쓴다. 여기까지는 기존의 입력과 크게 다를 것이 없는데, inhanced input 의 input mapping context는 여러 개의 행동을 우선 순위를 두고 사용이 가능하다는 점이 있다. 예를 들면 인벤토리가 열렸을 때 아이템을 둘러보는 키가 기본 상태에서의 이동 키와 같다고 했을 때, 기존 방식으로는 인벤토리가 열려 있을 때를 확인하는 변수를 만들고 인벤토리를 움직일 수 있는 지 여부를 판단했다면, input mapping context를 사용하면 인벤토리를 움직이는 걸 우선순위를 위로 해서 인벤토리 조작이 안되는 경우에만 움직이도록 하여 변수를 만들 필요가 없게 된다.



또 하나로 trigger 기능이 있는데 해당 기능을 쓰면 하나의 키 입력을 누르거나 떼었는지의 여부, 시간, 연속 입력 등을 판단해 키 하나의 입력으로 5개 이상의 행동을 만들 수 있다.



모든 캐릭터가 같이 사용할 Base\_Char 클래스를 만들고, 다른 캐릭터들이 상속받아 사용할 수 있도록 공통된 기능을 넣도록 했다.

이후 F\_Slime 클래스를 만들고 해당 클래스에서는 슬라임 캐릭터에 추가했던 기능인 오브젝트 접촉 시 체력 변경과 체력 변경 시 함께 이루어지는 스케일 값 변경 함수를 추가했고, 포스트 프로세스 기능을 사용하는 함수를 현재 구현 중에 있다.

프로젝트 이전 중에 프로젝트를 플러그인의 형태로 패키징해서 이동시키는 방법을 사용했는데, 블루프린트로 만들어졌던 클래스는 제대로 옮겨졌지만, C++ 코드로 짜인 클래스가 같은 코드를 그대로 옮겼음에도 빌드가 되지 않는 문제가 발생해 현재 해결 중에 있다.

|  |  |  |  |
| --- | --- | --- | --- |
| **문제점 정리** | 플러그인 방식으로 코드가 옮겨졌을때 빌드가 되지 않는 문제  11월에 늦게 시험 기간을 가진다. | | |
| **해결방안** | 언리얼 공식 문서 등을 이용해 플러그인 패키징에 대해 다시 한번 알아본다.  본격적인 시험기간이 오기 전에 할 수 있는 작업을 완수 | | |
| **다음주차** | 15주차 | **다음기간** | 2023.10.31 ~ 2023.11.6 |
| **다음주 할일** | 이동 작업 완료,  팀원과 작업에 대한 논의 | | |
| **지도 교수**  **Comment** |  | | |