|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **주차** | 7 주차 | **기간** | 8.8 ~ 8.14 | **지도교수** | (서명) |
| 이번주 한일 요약 | 유데미 Unreal Engine 5 C++ 멀티플레이어 강의 수강(섹션 1) | | | | |

<상세 수행내용>

금주의 시작이었던 6주차 발표 수강 후 한 학생분의 리슨 서버 발표 이후 여러 부분에서 배울 점이 많아 급히 6주차에 했던 비디오 게임 제작하기에서 멀티플레이어 서버 강의를 듣고 공부하는 것으로 전환했습니다. 원래 기존 iocp를 통해 졸업 작품을 하려 했지만 짧은 식견이었던 것 같습니다. 6주차 발표 이후 1~2일 동안 언리얼 서버에 대해 여러 부분 공부하는 시간을 보냈습니다.

결론적으로 언리얼 엔진 소스코드에 네트워크 관련 코드가 있어 자체적으로 서버 코드가 있는 것을 알게 되었습니다. Select방식이어 동접자의 한계가 있지만 FPS 장르에 적합합니다. 대표적으로 리슨 서버와 데디케이티드 서버로 리슨 서버는 1개의 클라가 서버 역할까지 하는 것이며, 데디케이티드 서버는 서버가 서버 역할만 하는 것입니다. 졸업작품은 데디케이티드 서버로 진행할 예정입니다. 장점이 클라와 서버를 한 번에 구현하는 것이 가능하며 HasAuthority()를 사용해 게임 서버와 클라이언트 간의 역할을 구분할 수 있을 것 같습니다.

HasAuthority는 서버로부터 허용되지 않는 작업을 클라이언트가 차단되도록 할 수 있는 기능입니다. 결론적으로 클라이언트에서 액터를 업데이트 해도 서버로 다시 값이 업데이트 되는 것이 아닌 것입니다. 이러한 멀티플레이어 리플리케이션 핵심 함수를 통해 클라와 서버를 한 번에 구현하 되, 서버에서만 처리되어야 하는 기능들을 구현할 수 있을 것 같습니다.

if (HasAuthority())

{

FVector Location = GetActorLocation(); Location += FVector(5 \* DeltaTime, 0, 0);

SetActorLocation(Location);

}

바로 데디케이티드 서버 공부로 들어가면 좋지만 자료가 마땅치 않고 언리얼 서버 코드를 이해하는데 좋다고 판단해 리슨 서버 기반의 강의인 유데미 Unreal Engine 5 C++ 멀티플레이어 강의를 수강했습니다. 클라이언트-서버 모델에 대해 공부했으며, CMD를 통해 에디터를 사용하지 않고 바로 게임에 접근하는 법 및 서버를 열어 클라이언트 접속시키는 것을 해봤습니다. 마지막으로, 콘솔창을 사용해 클라이언트가 서버에 접속을 해봤습니다.

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명 구름, 하늘, 스크린샷, 야외이(가) 표시된 사진

자동 생성된 설명

하늘, 구름, 스크린샷, 3D 모델링이(가) 표시된 사진

자동 생성된 설명

|  |  |  |  |
| --- | --- | --- | --- |
| **문제점 정리** | X | | |
| **해결방안** | X | | |
| **다음주차** | 8 | **다음기간** | 8.16 ~ 8.22 |
| **다음주 할일** | 언리얼 c++ 유데미 강의 수강(2,3) | | |
| **지도 교수**  **Comment** |  | | |

교수님 노트북의 수리가 오래 걸려 31일 발표까지는 휴대폰을 통해 참여하도록 하겠습니다. 죄송합니다.

깃허브 커밋은 김준현 학생에게 작업일지를 보내 제 파일에 커밋을 부탁해 올리도록 하겠습니다.

부모님 노트북으로 차주부터 사용해 6주차 부터는 발표에 참여하도록 하겠습니다.