|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **주차** | 8주차 | **기간** | 2023.8.14~ 2023.8.20 | **지도교수** | (서명) |
| 이번주 한일 요약 | - 게임 서버 공부(패킷 직렬화)  - 언리얼 멀티플레이 공부(유데미 강의) | | | | |

<상세 수행내용>

**패킷 직렬화**: 데이터를 네트워크를 통해 전송하기 위해서는 데이터를 연속된 바이트 스트림으로 변환해야 하며, 이 과정을 "직렬화"라고 한다. 원래대로 되돌리는건 “역직렬화”

**<패킷 직렬화의 주요 고려 사항>**

**데이터 형식**: 데이터의 구조와 타입을 정확히 알고 있어야 함. 예를 들어, 정수, 부동소수점, 문자열 등의 형식이 어떻게 구성되어 있는지 알아야 한다.

**엔디언**: 컴퓨터 아키텍처에 따라 바이트 순서가 달라질 수 있으므로, 네트워크 전송 시에는 일반적으로 빅 엔디언(Big-Endian) 방식을 사용한다. (메모리의 가장 낮은 주소에 상위 바이트를 넣음)

**프로토콜 정의**: 전송하려는 데이터의 형식과 순서를 정확히 정의해야 하며, 수신 측에서도 동일한 프로토콜을 사용해야 한다.

**오버헤드 최소화**: 네트워크를 통한 전송은 리소스를 많이 사용하므로, 직렬화 과정에서 불필요한 오버헤드를 최소화해야 한다.

언리얼 멀티플레이 공부

명령 프롬프트를 이용해서 간단한 전용 서버를 만들고 아이피를 이용해서 서버에 접속

"C:\UE\_5.2\Engine\Binaries\Win64\UnrealEditor.exe" "C:\Users\rndrn\Documents\Unreal Projects\PuzzlePlatforms\PuzzlePlatforms.uproject" /Game/ThirdPerson/Maps/ThirdPersonMap -server -log

"C:\UE\_5.2\Engine\Binaries\Win64\UnrealEditor.exe" "C:\Users\rndrn\Documents\Unreal Projects\PuzzlePlatforms\PuzzlePlatforms.uproject" IP주소:포트번호 -game -log

텍스트, 스크린샷, 폰트, 디스플레이이(가) 표시된 사진

자동 생성된 설명

HasAuthority() 함수를 이용해서 서버와 클라이언트를 구분하는 방법에 대해서 알아봄. 움직이는 엑터를 만들어서, 서버일 때만 움직이는 모습이 보이도록 했다. 특정 객체나 액터가 서버에서만 변경되어야 하는 경우에 이를 확인하는 데 사용된다.

|  |  |  |  |
| --- | --- | --- | --- |
| **문제점 정리** |  | | |
| **해결방안** |  | | |
| **다음주차** | 주차 | **다음기간** | 2023.8.21 ~ 2023.8.27 |
| **다음주 할일** | - 게임 서버 공부(JobQueue)  - 언리얼 멀티플레이 공부(유데미 강의) | | |
| **지도 교수**  **Comment** |  | | |