|  |  |  |  |
| --- | --- | --- | --- |
| Magician War | | | |
| 17주 | 2021. 4. 18 ~ 2021. 4. 24 | 작성자 | 홍범도 |
| 주간 목표 | 홍범도   1. **서버 소켓 공부**  * 블로킹 소켓 & 논 블로킹 소켓 * Overlapped I/O   김덕규   1. **스킬(불길) 제작** | | |
| 이번주 한 일 | 홍범도   1. **서버 소켓 공부**   서버에서 모든 클라이언트에서의 연산들을 담당하는 Work스레드 1개를 만들고 싶어서 서버 소켓을 공부하게 되었다.  TCP 통신의 경우 서버는 클라이언트의 개수만큼의 소켓을 보유하고 있는데,  CPU 코어의 개수보다 스레드의 개수가 더 많아진다면 컨텍스트 스위칭 현상이 발생한다.  컨텍스트 스위치  각 스레드를 실행하다 말고 다른 스레드를 마저 실행하는 과정  컨텍스트 스위칭은 CPU연산에 큰 부담을 주게 되므로 최대한 피하는 게 좋다고 하는데, 이를 생각하며 서버를 짜는게 중요하다고 생각된다.   * **블로킹 소켓 & 논 블로킹 소켓**   보통의 블로킹 소켓의 경우 클라이언트의 개수만큼 스레드의 개수를 정해주고 클라이언트의 개수가 늘어날수록 데이터 송수신 처리를 할 때 스레드 간의 컨텍스트 스위칭이 발생해서 CPU가 낭비된다.  그리고 수신하는 쪽의 수신 버퍼에서 송신 속도를 따라가지 못하면 송신 쪽 버퍼가 가득 차게 되고 이 때 블로킹이 발생한다.  그래서 논 블로킹 소켓을 활용하는 데  논 블로킹 소켓을 활용하면 블로킹이 일어나지 않게되고  블로킹 소켓의 경우 위와 같이 블로킹이 발생하면 송신하는 쪽은 송신 버퍼가 빌때까지 CPU가 놀게되는데, 이를 논 블로킹으로 처리하면 CPU를 최대한으로 활용할 수 있게 된다.  그리고 논 블로킹 소켓은 하나의 스레드에서 여러 소켓을 한꺼번에 다룰 수가 있게 됩니다.   * **Overlapped I/O**   논 블로킹의 경우 소켓 I/O 함수가 would block(블로킹 확률이 있다)일 경우  재시도 호출 낭비가 발생하고  소켓의 송수신에서 데이터 블록을 성공적으로 보낼 수 있다 판단이 되면 메모리 복사 연산이 발생하는데, 메모리 복사 연산도 또한 부담이 됩니다.  그래서 Overlapped I/O를 통해  위의 논 블로킹 소켓을 사용했을 때 발생할 수 있는 문제들을 해결할 수 있습니다.  그래서 성능의 우위를 점하기 위해서 Overlapped I/O를 사용하는 IOCP를 구현해야 다수의 클라이언트가 있는 팀 데스매치를 구현할 때 유리할 것 같다.  김덕규   1. **스킬(불길) 제작**      * 텍스쳐 두장을 통해 불길 이펙트를 제작했다. 노이즈 텍스쳐를 1배, 2배, 3배로 스케일링 하여 각 이미지를 다른 속도로 uv를 이동시켜 깊이감을 주는 방식을 통해 불을 제작했다. 각 이미지에 왜곡 값을 더해 자연스럽게 불이 흔들리는 듯한 효과도 주었다. * 그리고 스킬들의 통일성을 위해 스킬 클래스와 스킬이펙트 클래스를 제작했다. 각 클래스를 상속해 순수가상 함수를 정의하면 바로 화면에 띄울 수 있도록 제작했다. | | |
| 다음주 할 일 | 김덕규   1. **충돌 체크 삽입**   홍범도   1. **WorkThread 구현** 2. **스킬(불기둥) 제작** | | |
| 문제점 | 홍범도  어쩌다보니 구현 순서가 바뀐 감이 없지 않아 있다.  그리고 시험기간 중에 졸작을 하려니 생각보다 진행이 안된 것 같다. | | |