|  |  |  |  |
| --- | --- | --- | --- |
| [Cranding(크랜딩)] | | | |
| **8주차** | **2024.02.04 ~ 2024.02.17** | **작성자** | **정경서** |
| **주간 회의** | | | |
| **회의 날짜** | | 2024.02.14 | |
| 1. 각자 개발 진척도 공유  1. 신동엽  1. 프레임워크는 어느정도 완성  2. 서버와 클라 하나 연결해서 이동하는 부분  2. 정경서  1. 애니메이션 문제 해결을 위해 여러 방법 시도중  3. 홍예나  1. 애니메이션 블렌딩 잘 되는중  2. 상하체 분리와 가우시안 블러 시작  2. 창업  1. 창업하는 법 알아오기  2. 개강 전 완료하기 | | | |
| **이번주 한 일** | | | |
| 1. 신동엽   1. 클라이언트 동기화 ( 클라이언트 네트워크 클래스 )  클라이언트에 네트워크 클래스 생성  네트워크를 담당하는 쓰레드를 따로 생성하여 한쪽 쓰레드에선 렌더링, 한쪽 쓰레드에선 네트워크 통신을 하도록 설계  텍스트, 스크린샷, 폰트이(가) 표시된 사진  자동 생성된 설명  이벤트 기법을 사용하여 네트워크가 연결이 된 후에 렌더링이 되도록 설정  게임서버 텀프로젝트에서 사용했던 패킷 재조립부분을 그대로 활용 ( 추후 내 방식 수정 필요 )  텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진  자동 생성된 설명  수신한 데이터와 이전에 남아있는 데이터의 크기가 활용가능한 패킷사이즈보다 작다면 수신하여 processpacket함수로 들어가 패킷 후처리를 진행  텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진  자동 생성된 설명  클라이언트의 로그인, 다른 클라이언트의 접속을 추가하는 Add, 이동을 담당하는 move, 연결 해제를 알리는 remove만 우선 처리  팀원들과 회의를 통해 이동하는 부분을 어떻게 처리할지 의논 해본 결과 클라이언트에서 이동계산을 마친 이후에 좌표를 서버에게 전송한 후 서버는 그 좌표를 실시간으로 모든 클라이언트에게 전송하여 동기화를 시도하였고 아직 렌더링에 오류가 발생하여 해결하지 못하였지만 데이터 상으로 이동된 좌표들이 제대로 들어오는 것을 확인  다른 클라이언트 들의 정보를 담는 array 또한 extern으로 전역  왜 array? 한 룸에 3명씩 정보를 담기 때문에 제한된 인원수에 array가 적합하다고 판단  텍스트, 스크린샷, 멀티미디어 소프트웨어, 소프트웨어이(가) 표시된 사진  자동 생성된 설명  다른 클라이언트의 이동정보가 실시간으로 들어오는 것을 확인 -> 다음주 렌더링  이동하는 패킷을 전송하기 위해 키입력을 받았을 때 데이터를 전송 할 수 있도록 기존에 네트워크 클래스에만 있던 clientsocket변수를 메인 쓰레드에서 생성하고 extern을 통해 전역으로 gamefrawork클래스에서 사용하여 데이터를 전송할 수 있었다.    텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진  자동 생성된 설명  키입력을 받은 후 키 값에 따라 데이터 계산을 하고 이동연산을 마친 이후에 이동된 패킷을 전송하도록 설정 ( 임시로 확인을 위해 그냥 생코딩.. )  그 외에 잡다한 작업들… 이전 서버 코드 분석, 클라 코드 분석, 넷겜플 텀프로젝트 코드 분석,  인프런 게임 서버 강의를 통해 후처리 복습  설날과 여행, 알바가 계속해서 겹치다 보니 실제 작업할 시간이 많이 적었던 것 같다. 시간을 조금 더 효율적으로 사용해야 할 것 같다. 작업 시간을 본격적으로 늘려야 할 듯  문제점  (1) 받은 데이터의 정보가 초기화 되지 않은 불안정한 데이터들이 넘어왔다.  클라이언트 측에서 recv를 받았을 때 제대로된 정보가 넘어오지 않았다 이유는 포인터의 주소를 넘겨줬기 때문에 제대로 작동하지 X  (2) 한 쓰레드에선 계속 recv 루프를 돈다. 그렇다면 키입력이 있을 땐 send를 어떻게?  처음에 생각했던 건 전역으로 키 입력 큐를 생성하여 큐에 키 입력 데이터들이 쌓이면 네트워크 쓰레드에서 전송 처리를 해주려 시도, 하지만 비동기 방식이 아니였기에 리시브 상태로 들어가면 새로운 정보가 들어올 때 까지 send 처리를 할 수 없었다. 따라서 전역으로 소켓을 두고 extern을 활용하여 다른 클래스 안에서도 연결된 소켓을 자유롭게 사용할 수 있었다. 그렇게 하여 키입력이 있을 때 마다 send를 호출하여 사용하였다.   1. 정경서   1. 동엽님 요청사항    처음에는 배열을 통해 해결하려고 했다.    Scene::BuildObjects에서 먼저 다른 플레이어 두명을 그릴 준비만 하고 Render에서 실제 접속 한 플레이어의 수를 받아와서 그리는 방법으로 해결하려 했지만 몇명의 플레이어가 접속했는지 받아오는 과정에서 문제가 발생했다.    서버 관련 코드를 건드릴 수 있을거라고 생각했는데 IOCP로 작성된 코드라 내가 만지기 어려웠고 코드 분석에도 너무 많은 시간이 소요됐다.    Network.cpp에서 사용하는 Clients 배열에 접근하는 방법을 좀 알면 바로 해결 가능할 것 같은데 첫 부분부터 잘못 접근하는 코드를 작성하면 패킷에 문제가 생겨 이후에 더 큰 문제로 돌아올 것 같아서 주의하며 코드를 작성하고 있다.    2. 애니메이션 관련 오류      다른 모델들은 큰 문제 없이 출력이 되는데 불 지형 보스만 늘어난 채로 출력이 된다.    지금까지 문제는 strcpm로 단어를 찾아 중복되는 부분이 있는 경우 한 뼈에만 적용되어야 할 포지션, 스케일, 로테이트 값이 중복으로 적용되는 문제로 알았지만 해당 모델의 뼈가 위와 같은 상태로 있어서 문제 해결에 애를 먹었다…    같은 모델을 사용하고 같은 문제를 해결한 선배의 조언을 다시 보고 생각을 정리해보았다.  한참을 생각하고 추론해 낸 결과 문제 해결 방법을 알아냈다.    예를들어 RigHair1 과 RigHair1Gizmo가 있을때 RigHair1에 값을 적용하기 위해 RigHair1을 검색하면 RigHair1Gizmo가 같이 검색되어 생기는 문제라는 뜻이었다.  이 문제를 해결하기 위해 이름이 겹치는 경우 뒷 부분에 숫자를 넣어 이름을 변경해 주었다.    그랬더니 하나하나 원래 위치로 돌아왔다. 위 사진은 제 위치를 찾은 보스의 발..  중간중간 적용하면서 애니메이션이 잘 되는지 확인도 하면서 진행했다.  또 문제가 발생했다        RigHair1과 같은 뼈가 있을 경우 뒤에 00을 붙이며 문제를 해결했는데 배 근처의 뼈에도 00을 붙였더니 갑자기 전체 애니메이션이 멈췄다..  해당 뼈가 계층구조 맨 위에 있는 뼈고, 모든 뼈가 해당 계층구조에 포함되어 있어서 그런 것 같다.  그래서 이번에만 RigPelvis가 아닌 RigPelvisGizmo 앞부분에 00을 붙였다    그랬더니 문제가 해결되었다 ㅠㅠㅠ 정말 너무 행복해서 팔짝팔짝 뛰었다      도움을 주신  조창근 선배님 최재준 선배님 두 분 다 정말 감사드립니다    문제를 모두 해결하니 좌표가 두배씩 이동하는 오류도 고쳐졌다.    원래 10,0,0에 위치시키면 20,0,0에 그려졌는데 해당 오류도 뼈 이름이 겹치기 때문에 생기는 오류인 것 같다.  이후 다른 모델들도 뼈 이름을 모두 수정하여 모두 문제 없이 애니메이션이 작동하게 되었다.   1. 홍예나 2. 애니메이션 블렌딩 구현   : 기존에 발생했던 문제점인 애니메이션 블렌딩 이후 애니메이션이 진행되지 않고 동일한 모습만 렌더링된다는 것의 원인을 찾았다. 내가 예상했던 m\_fBlendingTime의 문제가 아닌, 애니메이션 블렌딩이 한 번만 실행되어야 하는데 키가 눌리는 내내 계속해서 애니메이션 블렌딩이 실행되는 이유였다.  이를 해결하기 위해 애니메이션 블렌딩이 진행될 때마다 count를 +=1 해주는 변수 m\_nBlendingCnt 를 생성하여 블렌딩이 키를 누른 시점에 한 번만 진행되도록 구현하였다.  또한 키가 눌리다가 떨어진 순간, 애니메이션의 교체가 한 번 더 일어나기 때문에 그 때의 애니메이션 블렌딩 실행을 위한 변수인 m\_bIsLastBlending 도 생성하였다.        위의 코드 상에 빨간 줄이 보이는 것은 애니메이션 상하체 분리를 위해 코드를 건드렸기 때문이므로 무시하셔도 됩니당      위와 같이 구현하여 애니메이션 블렌딩을 마무리 하였다.  그러던 중 회의에서 애니메이션 상,하체 분리를 추가로 진행하자는 것이 결정되었기에 상하체 분리에 대해 공부하였다.    위와 같이 생각하고 구현을 진행하려 하였는데 한 가지 문제점이 있었다.  우리가 사용하는 에셋의 본프레임이 상체와 하체가 구분된 것이 아니라 모두 섞여있었던 것이다. 따라서 상체와 하체의 본프레임을 구분하기 위한 수단으로 상하체를 구분해주는 변수를 추가하여 애니메이션 정보가 처음에 입력되는 순간 상체인지 하체인지 구분하기로 하였다.  이런 생각으로 이어질 수 있었던 계기는 에셋의 본프레임들이 계층 구조를 이루고 있기 때문이다. 상체 본프레임 중 부모의 역할을 하는 것이 입력되는 순간, m\_bUpperBody 변수를 true로 바꿔준다. 그 다음부터는 해당 본프레임의 부모를 찾고, 그 부모의 m\_bUpperBody가 true라면 해당 본프레임의 m\_bUpperBody 또한 true가 되게 하였다. 이렇게 하면 상체와 하체의 구분이 가능해진다.        해당 코드는 CGameObject::LoadAnimationFromFile에서 진행하였다.  이렇게 구분이 상, 하체의 구분이 가능해졌으니 상, 하체 분리 코드를 구현하여야 하는데 생각만큼 잘 되질 않고있다. 우선, 내가 생각한 대략적인 구현 내용은 키가 눌릴 때,  애니메이션 분리를 진행 시작한다.  상체를 가져와야 할 애니메이션(애니메이션1)과 하체를 가져와야 할 애니메이션(애니메이션2)의 set을 생성하고, 애니메이션2의 true인 부분(애니메이션 중 상체인 본프레임)에 애니메이션1의 true인 부분을 넣어준 후, 기존의 애니메이션 진행 과정과 같이 보간을 진행한 후 실행한다.  간단한 아이디어인 것 같은데 잘 안되는 것을 보아 뭔가 문제가 있는 것 같다. | | | |
| **다음주 할 일** | | | |
| 1. 신동엽   - 유니티로 스테이지 맵 합치기,  - 프레임워크 다듬기  - 클라이언트 네트워크 클래스, (렌더링 관련 정보 처리)  - Lock, DeadLock, atomic 등 동기화 기법 공부   1. 정경서   - 디퍼드 렌더링 공부 및 코드 작성  - 클라이언트-서버 플레이어 그리기 완료   1. 홍예나   - 애니메이션 상, 하체 분리 마무리  - 가우시안 블러 공부 | | | |
| **비고** | | | |
|  | | | |