|  |  |  |  |
| --- | --- | --- | --- |
| [Cranding(크랜딩)] | | | |
| **12 주차** | **2024.03.10 ~ 2024.03.16** | **작성자** | **정경서** |
| **주간 회의** | | | |
| **회의 날짜** | | 2024.03.11 | |
| 1. 머지(merge)   맵, 애니메이션 블렌딩, 상하체 분리 등 머지를 완료하였습니다   1. 맵 관련   맵의 색을 그라데이션으로 구현하려면 C#을 조금 배워서 추출 코드를 수정해야 한다   * 단색으로 한 뒤 외곽선과 조명이 들어가면 괜찮을 것 같으니 단색으로 하자  1. 애니메이션   검 공격 애니메이션이 플레이어 모델의 머리를 뚫고 들어간다   * 애니메이션 변경을 통해 더 자연스럽게 하자 * 기존 검 공격 애니메이션의 수를 3개에서 2개로 변경   기절 애니메이션이 위치 이동을 해요   * 위치이동 안하도록 수정하여 파일 전달 완료  1. 창업   사업 계획서, 동업 계약서 작성 완료 | | | |
| **이번주 한 일** | | | |
| 1. 신동엽(서버)   **1. 애니메이션 동기화**  - > 서버 측에서 플레이어 이동 하는 입력값에 따라 패킷을 보낼 때 애니메이션 상태 값을 추가하여 전송,  ( 문제 ) 무브 패킷을 전해주고 다시 Idle 상태가 되었을 때 , 돌아오는 패킷이 없음 ,  Idle 패킷을 추가해야하는지..?  **2. 프레임워크 수정**  -> 동기화 하는 과정에서 회전과 이동이 동시에 들어가거나, 많은 회전량이 입력되면 클라에서 보내오는 패킷들을 서버측에서 제대로 소화시키지 못하여 프로그램이 죽거나 버벅거리는 상황이 계속해서 발생하였다. 네트워크 과정에서 문제가 생긴걸로 판단하고 서버와 클라이언트 네트워크 클래스의 수정을 조금 바꾸기로 하였다. 우선 클라이언트 네트워크를 수정하였다. 연결을 메인쓰레드에서 하고 recv용 스레드를 생성하여 별도의 스레드를 두어 처리하게 수정하였다.    연결 이후 새로운 스레드를 생성, 스레드 함수 호출하여 이 스레드에선 항상 recv 상태,  ..여기서도 재조립 문제가 발생한다. \*\*  결과를 확인해보기도 전에 클라와 서버를 연결하는 과정에서 사소한 문제 , 엔디언 문제였다. 잘확인하고 사용하자  현재 진행도, 로그인과 Add 하는 부분 까지 완료하였으나 , 기존의 이동,회전 방식과는 다르게 해야할것 같다.    **3. 개인공부**  -> Timer  NPC , 시간, 등 컨텐츠를 수행하기 위해 Timer , NPC – AI 에 대해 공부했다.  NPC들이 많으면 많을수록 시스템콜은 줄어들어야 한다. BusyWaitng 문제가 생긴다.  따라서 NPC들은 자율적으로 움직이는데 외부의 요청이 없어도, 독자적으로 실행되어야 한다.  이를 기존에 Heartbeat 함수,    이러한 방식으로 하였지만, 이는 남는 시간을 처리하지 못하는 문제가 발생,  이를 해결하기 위해서 , Timer 개념이 도입된다.   1. 정경서(클라이언트)   **1. 맵**  맵을 추출할 때 material이 그라데이션 되어있는건 그려지지 않고 단색일 경우에만 정상적으로 그려진다.  이후 맵을 조금 수정한 뒤 다시 추출하려 했는데 갑자기 맵이 나오지 않았다.  원인은 에셋 자체를 다시 import 하면서 각 prefab의 read/write 옵션을 키지 않고 추출을 하려고 해서 추출이 정상적으로 되지 않고 있었던 것이다.  다시 read/write 옵션을 키니 잘 나왔다!  스크린샷, PC 게임, 그래픽 소프트웨어, 비디오 게임 소프트웨어이(가) 표시된 사진  자동 생성된 설명  위에서 단색만 잘 나오는 이유도 알게되었다.  현재 맵을 추출할 때 교수님의 cs 코드를 사용하여 맵을 추출하고있다. 교수님 코드에 기록되는 값은 albedo, emissive, glossiness, metallic 등 standard shader에 있는 값들만 기록을 하기에 단색은 albedo 값으로 들어가서 잘 나오고  그라데이션이 되는 material은 top down이라는 모델러가 만든 shader라서 값이 기록이 안되기 때문에 그려지지 않는 것이었다…  해결하려면 추출 코드를 수정하면 될 문제이지만  졸작회의 결과 그냥 단색도 나쁘지 않아서 그냥 단색으로 하자는 결론이 났다!  **2. 프로젝트 변경**  지금 프로젝트는 교수님 labproject와 다른게 없어서 변경이 필요했다.  Cranding 프로젝트를 하나 만들어서 현재까지 진행된 내용을 모두 옮겨주었다.  애니메이션 블렌딩과 각 키를 누르면 애니메이션이 나오게 하고 맵을 띄웠다.  모델의 머리가 크기 때문에 생기는 여러 문제도 애니메이션 블렌딩 코드를 통해 해결했다. (예나언니 고마워)  텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진  자동 생성된 설명    조금이라도 보기 편하도록 회의를 통해 결정된 방법으로 정리를 해주었다.  **3. 디퍼드 렌더링**  먼저 멀티 렌더 타겟 코드를 통해서 후처리를 하고 gbuffer나 여러 pass 등 계산을 하고 싶은데 멀티렌터타겟 코드를 옮겨오는 과정을 일주일 내내 반복중이다.. 그래도 조금씩 진전이 있어서 기쁘다 ^\_^  4. 그 외에 잡다한 일  바운딩 박스 체크를 했다  스크린샷, 그래픽 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진  자동 생성된 설명  아무리봐도 플레이어 에셋을 잘 구한것같다..  애니메이션을 다시 구했다. 원래 사용하던 플레이어의 검 공격 모션은 팔이 머리를 뚫고 지나가서 회의를 통해 다른 애니메이션을 구하기로 했다.  유니티 에셋스토어와 믹사모를 뒤져서 다시 귀여운 애니메이션을 찾고 잘 동작하는 것을 확인했다.   1. 홍예나(클라이언트) 2. **애니메이션 상하체 분리 마무리**   : 게임이 시작되기 전, 먼저 사용할 무기를 지정하고 게임이 시작된다.  그러나 아직 무기 선택하는 기능을 구현하기 이전 단계라서 우선 광선검을 사용하는 플레이어라는 가정 하에 구현하였다.  광선 검을 이용한 공격 애니메이션은 8번, 9번 2가지이다.  8번, 9번 애니메이션이 반복되며 진행되어야 하기 때문에    사진과 같이 m\_nAttackAniNum이라는 int형 변수를 만들었고 해당 변수는 공격할 때에 실행되어야 할 상체 애니메이션의 애니메이션 번호를 저장하기 위해 생성하였다.  (8번, 9번 순서로 반복되기 때문에 8번으로 초기화 한 것)    사진과 같이 공격시 실행되어야 하는 상체 부분의 변환행렬을 m\_nAttackAniNum번의 애니메이션에서 가져오도록 변경하였다.  또한, 마우스 왼쪽 버튼이 눌린 후 떨어지고 난 다음에도 공격 애니메이션은 해당 상체 애니메이션의 마지막까지 실행되어야 하기 때문에(좌클릭이 떨어진다고 공격 애니메이션도 중간에서 끝나서는 안되기 때문에) fPosition2가 0이 되는 경우(상체 애니메이션이 마지막까지 실행된 후. 즉, 상체 애니메이션의 position이 다시 처음=0 으로 돌아가는 경우)에 m\_nAttackAniNum이 8이었다면 그 다음 공격은 9번 애니메이션으로, m\_nAttackAniNum이 9였다면 그 다음 공격은 8번 애니메이션으로 실행되도록 m\_nAttackAniNum 변수를 변경하였다. 그 후, 공격이 종료되도록 하였다.  이렇게 하여 좌클릭을 공격 애니메이션이 끝나기 전에 중단해도 공격 애니메이션이 끝까지 실행된 후 공격 애니메이션이 종료되도록 할 수 있었다.  또한, 8번, 9번 상체 애니메이션이 반복되어 공격 애니메이션이 실행되도록 할 수 있었다.  기존의 애니메이션 블렌딩 코드에서는 m\_nBlendingCnt라는 변수를 사용하였었는데 코드가 점점 추가 될수록 해당 변수로 인해 복잡해지는 것이 느껴졌다.  이러한 이유로 해당 변수를 대체할 변수인 m\_nMoveCnt 변수를 만들어주었다.  애초에 m\_nBlendingCnt 변수를 사용하던 이유는 애니메이션 블렌딩이 애니메이션이 실행되는 처음에만 실행되고, 그 이후부터는 블렌딩 되지 않도록 하기 위함이었다.  Player의 move 함수는 dwDirection이라는 변수의 유무로 내부 코드가 실행될지 말지가 결정되는 함수인데, dwDirection은 플레이어가 이동할 때 사용되는 상,하,좌,우 키가 눌리는 경우에 그 값이 결정된다.  상키를 눌러서 앞으로 가던 중, 좌키를 눌러 왼쪽으로 이동하는 경우가 있을 수도 있는데 이럴 때, 처음 상 키가 눌릴 때에 딱 한 번만 애니메이션 블렌딩이 실행되어야 한다.  이때, 상,하,좌,우 키를 누를 때(플레이어의 이동이 있을 때마다)마다 m\_bIsBlending(애니메이션 블렌딩 유무 변수)을 true로 해주면 상키 좌키를 연속으로 누르는 경우에 블렌딩이 중간에 한 번 더 실행되게 된다.  이러한 경우를 방지하기 위하여 비슷한 기능을 하고 있던 m\_nBlendingCnt를 m\_nMoveCnt로 바꾸어 0으로 생성하고, 방향키가 눌린 경우에만 m\_nMoveCnt를 ++해주도록 하였다.    이렇게 하여 아직 처음 블렌딩이 되지 않은 경우(m\_nMoveCnt가 0인 경우 == 플레이어가 움직임이 없다가 처음 이동한 경우)에만 m\_nIsBlending이 true가 되어 블렌딩이 실행되도록 하였다.   1. **기절, 부활, 상호작용 애니메이션 추가**   아직 서버와 애니메이션 동기화를 하지 않은 상태이기에 우선 임시로 f1키를 누르면 기절상태(hp==0), f2키를 누르면 부활상태가 되도록 구현해두었다.        M\_bIsDead라는 변수를 생성하여, f1키가 누르면 true가 되고, f2키를 누르면 false가 되도록 하였다.  이때, 기절상태가 될 때 처음 블렌딩(m\_bIsBlending) 변수를 true로, 부활 상태가 될 때, 마지막 블렌딩(m\_bIsLastBlending)변수를 true로 변경하여 애니메이션 블렌딩도 되도록 하였다.    기절상태에서는 공격, 상호작용 등등 다른 행위가 불가능하기 때문에 m\_bIsDead가 true인 상태에서는 m\_bIsAttack, m\_bIsHeal 모두 false이도록 하였다.  그리고 애니메이션에 사용될 m\_nAnimatinBefore(블렌딩 이전에 실행되던 애니메이션의 번호), m\_nAnimationAfter(블렌딩 이후에 실행되어야 할 애니메이션의 번호) 또한 변경해주었다.  M\_nAnimationBefore는 어떤 행동을 하다가 기절상태에 빠졌는지 알 수 없어서 상수로 박아놓을 수 없기 때문에, 이전에 실행된 마지막 블렌딩 이후에 실행되는 애니메이션의 번호로 설정하였다.  M\_nAnimationAfter는 기절 상태 애니메이션의 번호로 설정하였다.  1번 애니메이션(idle)과 3번 애니메이션(run)의 상태를 false로 하고, 5번 애니메이션(기절)만 enable 하도록 true로 변경해주었다.  부활 부분은 m\_bIsDead가 false이며, 동시에 m\_nAnimataionAfter가 5인 경우에만(이전에 기절 상태였던 경우만) 실행되도록 하였다.  왜냐하면 이전에 기절 상태가 아니었던 경우 또한 m\_bIsDead가 false이기 때문이다.    Spacebar를 누르면 상호작용 애니메이션이 실행되도록 하였다.  치료 행위 또한 상호작용 애니메이션으로 실행되기 때문에 상호작용의 유무를 m\_bIsHeal이라는 변수로 생성하였다.  상호작용 행위는 플레이어가 아무 움직임이 없을 때에만 가능하기 때문에 m\_bIsMove라는 변수를 생성하여 해당 변수가 false이며(플레이어에게 이동이 없으며) spacebar를 누른 상태에만 m\_bIsHeal이 true가 되도록 하였다.    상호작용을 하는 동안에는 다른 행위를 할 수 없기 때문에, m\_bIsHeal이 false이며 m\_nAnimationAfter가 10이 아닌 경우(상호작용 중이 아닌 경우, 10번 애니메이션 == 상호작용 애니)  + 현재 이동 거리가 0인 경우 = 즉, 플레이어가 idle 애니메이션을 실행해야 하는 경우에 m\_nMoveCnt를 0으로, idle 애니메이션이 활성화되도록 하였다.    스페이스바를 누른 후, m\_bIsHeal이 true가 되고, m\_nAnimationAfter가 10이 아닌 경우(상호작용 애니메이션이 처음 시작될 경우, 처음 블렌딩이 아직 실행되지 않은 경우)에 m\_nIsBlending이 true가 되어 처음 블렌딩이 실행되도록 하였다.  또한, 상호작용 중에는 다른 행위가 불가능하기 때문에 idle, run 애니메이션을 false로, 상호작용 애니메이션은 true로 변경하였다.    스페이스바에서 손을 떼게 되면 m\_bIsHeal이 fasle로 즉, 상호작용애니메이션이 종료되도록 하였다.  m\_bIsHeal이 false이고(스페이스바에서 손을 뗀 상태), m\_nAnimationAfter가 10인 상태(상호작용 애니메이션의 처음 블렌딩만 실행됐고, 마지막 블렌딩이 실행되기 전 상태)일 경우 마지막 블렌딩을 실행하도록 구현하였다.   1. **가우시안 블러 공부**   : 우선 큰 틀만 공부하였다.    가우시안 블러를 구현하기 위해서는 텍스처, 렌더타겟, 샘플링, 컴퓨트 쉐이더에 대한 이해가 필요하다. | | | |
| **다음주 할 일** | | | |
| 1. 신동엽(서버)  * Timer를 이용한 NPC 구현 * 프레임워크 마무리 * 이동/회전 동기화  1. 정경서(클라이언트) 2. 홍예나(클라이언트)  * 가우시안 블러 공부 * 애니메이션 동기화 | | | |
| **비고** | | | |
|  | | | |