|  |  |  |  |
| --- | --- | --- | --- |
| [Cranding(크랜딩)] | | | |
| **22주차** | **2024.5.19 ~ 2024.5.25** | **작성자** | **정경서** |
| **주간 회의** | | | |
| **회의 날짜** | | 5/21 | |
| 1. 창업 현장실습 계획서 및 신청서 작성하기 2. 종강 전까지 할 일   동엽: 낮/밤 구현, 몬스터 관련 수정, 원소 관련 서버 동기화  경서: 외곽선, 게임 컨텐츠 위주 개발(원소, ui 중점적으로,,)  예나: 가우시안 블러, 충돌체크 등 컨텐츠 개발 | | | |
| **이번주 한 일** | | | |
| 1. 신동엽(서버)   직선거리로만 이동하던 NPC들의 움직임을 사정거리 안에 Client가 들어왔을 떄 client 방향으로 이동할 수 있게 추적 이동을 구현했다.  다이렉트 함수들을 이용하니 쉽게 구현 할 수 있었다. 아직까진 속도값 조정과 회전이 적용되지 않아 부자연스럽지만 추후에 팀원과 조율해볼 예정  한번 불릴 때마다 3번의 루프를 도는 것이 부담인지 테스트도 해봐야 할 것 같다.  텍스트, 스크린샷, 소프트웨어, 디스플레이이(가) 표시된 사진  자동 생성된 설명  씬전환 멀티쓰레드 오류 해결 ,  이전에는 메인 스레드에서 키입력을 받으면 서버에게 Ready 패킷을 전송함으로 준비를 알리고 서버 측에선 룸안에 모든 인원이 준비를 했으면 Gamestart패킷을 보내서 전체에게 시작을 알렸다 이때 클라이언트는 준비를 누르고 WaitforSingleObject를 사용해서 스레드를 대기시켰으며 게임시작 패킷을 받으면 Setevent를 사용해서 동시에 씬전환이 될수 있도록 설정해주었다. 그런데 네트워크 환경이 달라지면서 서버가 불안정할 때 이런식으로 아예 막아버리는건 위험하다고 판단하여 변경하였다.  접속하고 준비를 했을 떄 똑같이 패킷을 보내고 받는 것 까지는 동일하지만 방법을 바꾸었다. 메인문에 IngameStart라는 bool값을 두고 패킷을 받았을 때 true로 하는 방식으로 씬전환을 하고 다시 false로 바꿔 1회만 부르도록 하여 씬전환을 유도하였다.  결과 준비했을 때 움직일 수 있고 중간에 게임이 멈추는 등 오류가 발생하지 않았다.  텍스트, 스크린샷, 폰트이(가) 표시된 사진  자동 생성된 설명   1. 정경서(클라이언트) 2. 모델 애니메이션 변경 및 맵 충돌 박스 수정     상하체 분리 코드에서 서버를 연동할 때 딜레이 때문에 상하체 애니메이션 속도가 달라지는 문제가 생겨 새로운 애니메이션을 찾아 추출하고 언니에게 넘겨주었다  언니가 맵에서 충돌 박스 범위에 대한 이야기를 해주어서 해당 부분도 수정해주고 있다   1. 몬스터 look, right, up벡터 설정   해당 함수를 동엽님이 쓰시는 브랜치에 만들어두었다.   1. 릴리즈 해결하기   디버깅을 하면서 메모리 릭을 잡고있다...  com객체들도 문제가 있는 것 같아서 comptr로 변경하고  원시 포인터들도 스마트 포인터로 바꾸고 있다  시간이 오래 걸리지만 이걸 해결하면 릴리즈에서나 어디서 터지는 걱정을 덜 해도 될 것 같아서 열심히 고치는 중이다.   1. 홍예나(클라이언트)   애니메이션 갈아엎기  공격만 상하체 분리 제외하도록 바꾸려고 했는데 move, update함수가 너무 복잡하고 잘 읽히지가 않아서 그냥 전부 다 갈아 엎어야겠다  CTerrainPlayer의 move와 update가 지금 보니 너무 복잡해서 간단하게 바꾸려고 한다  move 함수에서는 이동 관련 애니메이션 번호만 보내주려고 하고,  update에서는 그 밖의 다른 애니메이션 관련 번호 + 블렌딩 여부만 보내줄 것이다  이전 방식도 비슷하긴 했는데 너무 복잡해서 코드가 눈에 잘 들어오질 않았다  좀 더 직관적으로 바꿀 예정이다    경서가 달릴 때의 공격 애니 추가해준 것 기준으로 구현해보자  일단 기존에 달릴 때 처음 블렌딩을 딱 한 번만 될 수 있게 하기 위해 사용했던 변수인 m\_nMoveCnt를 없애보자  얘 때문에 뭔가 복잡해보임  그러기 위해서는 애니메이션 블렌딩이 있을 경우, 없을 경우 등등 여러 경우에서 m\_nAnimationBefore과 m\_nAnimationAfter를 업데이트 해줘야 했다  아 블렌딩 변수도 걍 하나로 통일하고 싶은데  직관적으로 코드 짜는걸 최우선으로 하고 move, update의 전체적인 구조를 바꿔보았다  일단 불필요한 변수들은 최대한 제거했다  m\_nMoveCnt를 없앴고, m\_bIsBlending과 m\_bIsLastBlending을 그냥 하나의 블렌딩 여부를 나타내는 변수로 변경하였다(m\_bIsBlending으로 통일)  그렇게 하기 위해서는 앞서 말한 것처럼 m\_nAnimationBefore, m\_nAnimationAfter의 값을 블렌딩 여부에 따라 업데이트 해주어야 했다  자세히 설명하면,  m\_nAnimationBefore는 원래 실행되고 있던 애니메이션의 번호, m\_nAnimationAfter는 이제 실행되어야 할 애니메이션의 번호이다  예를 들면, idle(1번) → run(3번)으로 애니메이션이 변경되어야 하는 상황일 때  우선, 다른 애니메이션 말고, idle과 run만 구현하였다    ilde → run으로 애니가 바뀌는 조건은  상호작용 하는 중이 아니거나(m\_bIsHeal == false), m\_bIsMove가 true이어야 한다  그 중에서도 블렌딩이 일어나야 하는 지점은 딱 처음에 애니메이션이 idle → run으로 변경되는 지점  즉, m\_nAnimaionBefore(기존에 실행중이던 애니메이션의 번호)이 run이 아닌, 다른 애니메이션의 번호일 때이다  그 때에만 블렌딩이 되어야 하고, 그 이후부터는 run 애니메이션만 실행되도록 해야한다  그렇게 하기 위해 478번째 줄과 같이 m\_nAnimationBefore가 3(run 애니메이션 번호)이 아닌 경우를 조건에 포함하도록 하고, 그 조건문에 해당하는 경우에는  앞으로 실행되어야 할 애니메이션 번호인 m\_nAnimationAfter를 run의 번호인 3번으로 변경하고,  블렌딩 여부를 true로 하여, 블렌딩이 진행되도록 하였다.  또한, run 애니메이션을 활성화하고, 기존에 진행되던 애니메이션을 비활성화 시켰다.    update 함수도 move 함수와 마찬가지로 다 없애버리고 다시 구현중인데  우선, run → idle 먼저 구현하였다  idle이 되는 경우는 이전과 같다  플레이어의 이동거리가 0이 되면 그 때 idle 애니메이션으로 변경되는 것이다  run → idle이 될 때에도 블렌딩이 일어나야 하는 시점이 있다  move함수에서와 마찬가지로 기존에 실행되던 애니메이션의 번호가 idle의 번호가 아닌 경우이다  그래서 543번째 줄과 같은 조건을 추가해주었다  나머지는 move 함수 구현과 동일하다    그리고 블렌딩이 끝난 후,  기존에 실행되던 애니메이션의 번호를 앞으로 실행될 애니메이션의 번호로 변경해주어야 한다  그래야 블렌딩이 애니메이션이 바뀌는 시점에만 일어나게 되기 때문이다  사실 갈아엎게 된 계기가 공격할 때 상하체 분리 말고, 그냥 달리면서 공격하는 애니메이션을 사용하려는 의도였는데 달리면서 칼을 휘두르는 애니메이션을 찾지를 못해서 다시 상하체 분리로 하기로 했다…..  공격도 다시 바꿔서 구현했었는데 좀 아깝다  일 벌이기 전에 그거 할 자원이 되는지 먼저 좀 알아보고 시작했어야 됐는데  다음부터는 미리 알아보고 하든 말든 해야겠다  어차피 언젠가는 구조 갈아엎었어야 했으니 잘 한거라고 생각하자 ^ㅁ^   1. 가우시안 블러(컴퓨트 쉐이더 안쓰는 코드 버전)   용교수님 코드는 컴퓨트 쉐이더를 쓰는데 안쓰고 싶어서 현석오빠네 코드를 분석해봄    우선 이렇게 블러 처리 할 텍스처를 생성해주고, → 맞나?  블러 쉐이더를 생성 후, build 해준다    근데 보니까 이렇게 디퍼드로 인해 달라지는 부분들이 있다  저게 먼 뜻이지  렌더타겟을 디퍼드가 쓰는 거에서 하나 더 추가해서 블러 처리한거 저장하는 듯 → 맞나?  디퍼드 수보다 하나 많은 수만큼의 렌더타겟뷰의 포멧을 지정해주는데 그럼 맨 마지막 렌더타겟 포멧은 대체 어케 정해지는거임?  알아냈다  경서한테 디퍼드 렌더링 설명을 듣고 알게 되었는데    위에 표시된 부분이 pdxgiResourceForamts로 렌더타겟 뷰가 어떤 형식의 정보를 받을지를 설정해주는 부분인 것 같은데 저걸 create 할 때 미리 받을 값의 형식을 명시해준다    저 부분은 shaders.hlsl의    이 부분과 연결된다  블러를 할 때 디퍼드에서 쓰던 렌더타겟 다음에 하나를 더 만들고, 그걸 이용하게 되는데 그 때에 저 부분에 하나를 추가해서 사용하면 될 것 같다!!!!    cbv는 필요없어서 안만들어주고 srv만 위에서 생성해준 블러쉐이더의 m\_nTexture만큼 만들어줌 | | | |
| **다음주 할 일** | | | |
| 1. 신동엽(서버)  * 애니메이션 동기화 수정 * 낮 밤 타이머 적용 * NPC 로직 수정  1. 정경서(클라이언트)  * 스카이박스 수정 * 외곽선 그리기 시작 * 원소 관련 컨텐츠 구현  1. 홍예나(클라이언트)  * 애니메이션 동기화 + 전체적인 구조 재정비 * 가우시안 블러 | | | |
| **비고** | | | |
|  | | | |