| 주차/일자 | 10주차 / 2.25~3.2 | 작성자 | 김진선 |
| --- | --- | --- | --- |
| 개요 | 개강 전 마지막 회의 | | |
| 회의 내용 | 2024.03.01 주간회의   * 이동&서버 관련 논의   + 기존의 move() 함수 사용에 어려움이 있어 수정 방향에 대하여 논의 함 * 저번 주 목표   + 맵 유니티로 띄워서(바운딩까지) 로드 -> 진선     - 모델 load 구조를 새로 만들어야 해서 진행 중, 개강후 모델 구매 예정.   + 씬 전환 ->진선     - 이전 버전에 제작한 구조를 수정하여 적용. Scene전환을 위한 기반 제작 완료 * MOVE 개선 -> 상민   + 애니메이션 삽입 -> 상민   + 맵 제작(1/4)->상민(취소 : 3월되면 지원금이 들어오기 때문)   + 애니메이션 통신 -> 정훈     - Move 처리로 인한     - 패킷 { 애니메이션 타입, 애니메이션 타입, float 블렌딩 타임} 변경 필요 * 개강 후   + 회의&병합 -> 금 or 토     - 큰 변경사안, 중요사안 있을 시 추가 회의 진행 | | |
| [다음 주 회의 안건] 2024.03.09 (17:00)   * 개발 진척도 확인 | | |
| 주간  수행 내용 | [김정훈 - 서버]   * gameframwork에서 네트워크 서버와 통신하던 코드를 Scene에서 패킷을 받고 보내는 방식으로 변경 * 다른 클라이언트에서 이동 시 자신의 화면에서 이동을 볼 수 있음 * 목표로 했던 이동 시 애니메이션을 적용하는건 실패해서 차주까지 연장 | | |
| [김진선 - 클라이언트]   * Scene 변환 기반   + 8주차에 작업한 내용을 기반으로 이후 추가된 사안도 문제 없이 작동하도록 제작     - 변수를 사용하지 않고 하드 코딩한 코드들 일괄 수정   + 키 입력시 Scene이 변환되며, Scene 변환 입력 시 기존의 Scene 내용은 삭제 후 새로운 Scene을 그리는 형식으로 제작 * 오브젝트 load (진행중)   + 개강 후 구매한 에셋의 경우에만 지원비가 나오기 때문에 무료 에셋으로 우선 진행   + 시행착오     - 기존에 사용하던 모델 load 방식은 애니메이션이 없을 경우 출력 되지 않으며, 유니티에서 바운딩 박스를 적용하는 방식이 다르기 때문에 기존 방식 사용에 어려움이 있음   -> 지형지물 오브젝트 load를 위한 구조 제작 필요   * 오브젝트 풀 기술 구현을 위하여 여러 객체가 한번에 있는 구조가 아닌, 각각 오브젝트 객체를 이용할 수 있는 형식으로 제작해야 함 * 유니티에서 모델을 추출하기 위한 c# 스크립트와 모델 Load 구조 제작 중 * 자잘한 수정   + 작업을 병합하는 과정에서 사라진 자잘한 변경사안 업데이트   + 터레인 수정, 스카이 박스 변경 | | |
| [이상민 - 클라이언트]   * 애니메이션 삽입   + 애니메이션은 각 캐릭터마다의 11개를 보유하고 있다. 그 애니메이션을 삽입할 때의 오류가 있었는데 그 오류는 캐릭터 오브젝트가 로드가 안되는 문제였다. 하지만 그런 큰 오류가 떴던 것치고는 다른 오류가 없어 로드자체는 잘 작동했다고 판단해서 다른 문제점을 생각해 보았다.   + 오류는 정말 간단한 문제였다. 유니티에서 모델좌표가 0이 아니어서 월드좌표로 변환하니 캐릭터의 월드좌표는 0이었지만 다른 곳에 로드돼서 보이지 않는 문제였다. 이 후에 다시 모델좌표계를 맞추니 정상적으로 로드가 되었다.   + 그 후에 필요한 애니메이션을 정리했다.      * 애니메이션들을 전체적으로 관리가 용이하도록 맥스값과 함께 각 애니메이션을 define으로 선언해 두었다. * 애니메이션들은 기존의 모델에 있던 애니메이션을 대부분 사용했다. 그리고 추가적으로 필요했던 애니메이션은 CRAWL, USE, EAT, DUMMY가 필요해 보였기에 Mixamo에서 가져와서 사용했다. 하지만 CRAWL애니메이션 같은 경우에 애니메이션 자체가 앞으로 기어갔기 때문에 실제 사용시 더 앞으로 기어가는 문제가 있었다. 그래서 in place모드로 제자리에서 미끄러지듯 보여지는 애니메이션으로 다시 다운받아 사용했다. * 그리고 본래 가지고 있던 애니메이션에 비해서 Mixamo의 애니메이션은 호환이 매끄럽지 않았기 때문에 원래는 DUMMY 애니메이션은 춤추는 애니메이션을 넣으려다 기존에 가지고 있던 ATTACK 애니메이션으로 대체했다. * 블렌딩 함수 제작   + 블렌딩은 기존에 사용하던 방식은 블렌딩타임을 기준으로 사용했다. 블렌딩타임은 키보드의 입력값이 다를 때마다 0.0f로 변경되며 ElapsedTime에 특정한 가중치를 곱한 값이 블렌딩타임에 중첩되며 그 값을 가중치로 한 블렌딩 애니메이션이 보여지게 된다. 블렌딩타임이 1.0f가 되면 다음 애니메이션이 활성화되며 블렌딩작업을 중단하는 방식이 이뤄지는데 이 과정을 기존에는 각 부분마다 실행되었다. 하지만 코드가 길어져 가독성이 떨어져서 함수로 대체하게 되었다. 함수는 CTerrainPlayer의 매서드이며 아래와 같은 코드로 구성되었다.      * 이 함수를 실제로 필요한 부분에서 사용할 때에는 다음과 같은 방식으로 사용된다.      * 확연히 가독성이 줄어 들었으며 이 후에 패킷을 통해 애니메이션의 타입 두 개와 지정한 클라이언트의 블렌딩타임만 받는다면 클라이언트 부분에서 쉽게 처리할 수 있다는 장점이 있다. 이 후 플레이어 데이터의 패킷에 추가할 예정이다. * 블렌딩 오류 수정   + 기존의 블렌딩의 오류는 키보드 입력이 들어오는 순간에 캐릭터의 월드좌표나 혹은 애니메이션의 무게 설정의 오류로 인해서 잠시 지정한 거리와 다른 곳에 렌더가 되는 현상이 일어나서 모델이 깜빡이는 것 같은 오류가 있었다. 처음에는 무조건 깜빡이는 오류라고 생각했지만 자세히 오류를 반복해서 보니 모델이 z축을 기준으로 먼 거리에 렌더되는 것을 확인할 수 있었다.   + 이 부분에 대해서는 키보드 입력 부분 한 가지만 신경쓰면 된다고 생각했다. 그래서 그 부분을 확인해 봤다.   + 기존의 방식은 (현재 키보드 입력값 저장) -> (블렌딩타임이 0이 아니라면) -> (블렌딩 애니메이션사용) -> (이전 키보드 입력과 현재 키보드 입력 비교) -> (다르다면 블렌딩 타임을 0으로 초기화) -> (현재 키보드 입력을 이전 키보드 입력으로 저장) 이 방식을 루트로 반복하는 과정이었다.   + 하지만 저 방식대로라면 전과 현재 키보드의 입력값을 비교하는 과정이 애니메이션 블렌딩을 사용하기 전이어야 정상적으로 동작할 것으로 보인다. 그래서 그 부분을 블렌딩을 사용하기 전 작업으로 올려주는 것만으로 해결할 수 있었다. * move 함수 개선(일부)   + 다른 게임들은 대부분 각자의 움직이는 방식이 있었다.  1. 마우스의 방향으로 캐릭터의 LookVector + 키보드는 캐릭터의 Position 2. 마우스의 방향은 화면의 방향 + 키보드의 방향이 캐릭터의 LookVector + 캐릭터의 Position  * 대부분은 이 두 가지 방식을 사용하기 때문에 이 두 가지 방식만 놓고 비교해 보았다. 게임의 특성상 우리의 게임은 뒤를 돌아보는 일이 잦기 때문에 1번을 사용한다면 멀미의 유발이 클 것이라 생각했다. 그래서 2번의 방식으로 바꾸려고 하였다. * 우선 기존 방식은 1번 방식이었기 때문에 2번 방식으로 전환하기 위해서 마우스의 움직임이 카메라의 회전만 할 수 있도록 했다. 그 과정을 위해서 기존 카메라는 Player의 회전값을 바탕으로 움직였기 때문에 Player의 회전값을 담당하던 기존 변수들에 ‘C’를 붙혀서 다른 변수들을 제작하고 마우스는 가상의 C회전값만을 움직일 수 있도록 하였다. * 또한 move()함수 자체도 Player의 LookVector를 기준으로 이동했다. 우리가 원하던 방식은      * 화면에 보인 두 가지 상황에 대해서 키보드로 뒤로 이동하면 화면 기준으로 뒤로 이동하는 것(왼쪽캐릭터는 Look방향, 오른쪽캐릭터는 Look반대 방향)을 원하지만 현재까지의 과정으로는 둘 다 Look 반대 방향으로 이동할 뿐이었다. * 하지만 아까 제작해둔 CLookVector를 이 캐릭터의 move()함수의 기준으로 삼는다면 문제점이 해결되었다. 이제 추가적으로 이 캐릭터의 LookVector를 어떤 방식으로 바라보게 할지에 대한 작업만 한다면 마무리가 될 것으로 보인다. * 키보드의 방향으로 캐릭터의 LookVector를 바로 바꿀 것인지 그 방향으로 회전을 할 것인지 회전을 한다면 무엇을 기준으로 어느 방향으로 회전할 것인지를 정하지 못해서 다음주 중으로 제작할 예정이다. | | |
| 특이사항 |  | | |
| 다음 주 수행 계획 | [공통]   * 코드 정리하기   + 필요 없는 구조, 코드 삭제   [김정훈 - 서버]   * 네트워크 사용 시 이동 애니메이션 적용 * 회전 값 통신   [김진선 - 클라이언트]   * 오브젝트 모델 load 마무리   [이상민 - 클라이언트]   * Player 컨테이너 정리 * update 정리 * Player 이동관련 처리 | | |