|  |  |  |
| --- | --- | --- |
| EmberFall 개발 일지 | | |
| **12주차** | **2025.03.16 ~ 2025.03.22** | 회의 내용 |
| 금주 회의는 애니메이션 관련 패킷 규격을 정하였다.  회전정보를 전달할때 yaw를 쿼터니언에서 추출을 해야하는데 이 방법에 대해 논의하였다.  현재까지의 작업물을 Merge하는 시간을 가졌고,  남은 기간까지의 개인일정을 공유하는 시간을 가졌다. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| EmberFall 개발 일지 | | 작성자 : 김승범 | |
| **12주차** | **2025.03.16 ~ 2025.03.22** | | 이번 주 진행 사항 |
| **03-16**  애니메이션 상-하체 분리 구현을 시작하였다. 오늘 시도한 방법은 기존에 존재하던 Animator 을 두 개 사용하여  해당 Animator 가 원하는 애니메이션을 입히고 싶은 부분 ( Bone ) 의 배열을 Bone Mask 로 받아와서,  업데이트가 일어 날 때, 각 Animator 에서 계산한 행렬 배열들을 Masking 하여 결합하여 Bone Transform 을  계산하도록 구현하였다.  **03-17**  전날 구현한 결과, 상체와 하체의 애니메이션이 완전히 따로 놀아 새로운 방법으로 구현을 시작하였다.  전날의 결과가 이상했던 이유는 상체와 하체의 연결 부위 ( 허리, 허벅지 ) 에 해당하는 정점은 양 쪽의 Bone 을  모두 참조하기도 하는데, 이를 전혀 고려하지 않아, 연결 부위에서 비정상 적으로 접히거나, 늘어나는 현상이  발생한 것이다.  이를 해결하기 위해, 각 Animator 에서 계산하던 Bone Transform 을 하나의 Animator 에서 계산하도록 변경하고,  마지막 Boen Transform 버퍼에 입력 할 때, Bone Mask 를 참조하여 입력하도록 수정하였다.  **03-18**  상-하체 분리를 하는 과정에서, 상체가 하체 자체를 기울게 하거나, 반대의 경우, 상체가 하체를 이끌게 된다는  문제가 발생하였다. 이를 해결하고자 노력하였으나, 만족할 만한 결과를 얻지 못하고, 상-하체 분리를 구현하기  위해, 상체와 하체에 적용하고자 하는 애니메이션 둘 다 되도록 정 자세를 유지하는 활 애니메이션을 사용하여  애니메이션 상-하체 분리를 적용하기로 결정하였다.  이를 통해 애니메이션 상-하체 분리를 완성하였다.  **03-19**  이제 구현된 애니메이션을 게임에서 활용 할 수 있도록 만들기 위해 Animation Controller 를 만들기 시작하였다.  Animation Controller 는 지정된 Parameter 를 매 프레임 확인하여, 애니메이션의 전환이 이루어 져야 하는지,  검사하고, 애니메이션의 전환을 관리하는 객체이다. | | | |
| EmberFall 개발 일지 | | 작성자 : 김승범 | |
| **12주차** | **2025.03.16 ~ 2025.03.22** | | 이번 주 진행 사항 |
| 이는 Unity 엔진에 존재하는 개념으로 이에서 착안해 제작하였다.  일반적인 ( 상 – 하체를 분리하여 관리하지 않는 ) 애니메이션 컨트롤러와,  상-하체 애니메이션을 분리하여 관리하는 애니메이션 컨트롤러로 구분하여 구현하였다.  이를 사용하여 키 입력을 통해 애니메이션의 전환을 확인하였다.  **03-20**  서버와 클라이언트의 연결 작업을 시작하였다. 성준이의 Test Client 를 참고하여 연결하였으며,  오늘은 연결 작업을 마무리하여, 앞으로는 서버의 로직을 따라 프로그램이 동작하도록 구현하였다.  3인칭 카메라를 수정하였다. 기존에는 캐릭터의 움직임에 종속되어 카메라가 답답하다는 피드백이 있어  마우스를 가로로 움직여 캐릭터를 회전시킬 수 있도록 만들었으며, 세로 방향으로는 20도 반경으로  위, 아래를 볼 수 이도록 수정하였다.  **03-21**  로비 씬 ( 무기를 선택하기 전 단계 ) 에서 사용될 사람 에니메이션 컨트롤러를 구현하였다.  기존에 구현했었던 애니메이션 컨트롤러에 내가 원하는 내용을 입력하여 상-하-좌-우 4 방향과,  상+좌 , 상+우 와 같은 4방향도 45도 회전하도록 하여 자연스럽게 방향 전환이 가능하도록 수정하였다.  **03-22**  대검을 사용하는 캐릭터의 애니메이션 컨트롤러를 구현하였다. 이는 기존에 존재하던 상,하,좌,우 이동과 더불어  이동 중 점프, 공격 모션을 추가하여 구현하였다.  기존 지형 파일 쓰기, 읽기 부분에서 반복문이 실행되는 것이 비효율적이라는 문제가 제기되어, 이를 수정하였다.  기존 테셀레이션 패치를 중심으로 저장하여 읽고, 높이를 구할 때에도 이를 단위로 읽도록 하였으나,  이를 수정하여 정점을 중심으로 저장하고 읽도록 변경하였다. | | | |
| EmberFall 개발 일지 | | 작성자 : 김성준 | |
| **12주차** | **2025.03.16 ~ 2025.03.22** | | 이번 주 진행 사항 |
| **03-16~03-18**  몬스터와 캐릭터 간의 공격 상호작용을 완성했다.  **03-19**  클라이언트와 서버 간 현재 진행상황을 확인하고 통합했다.  통합한 후 현재 처리 방식이 다른 오브젝트의 ID에 대해 처리 방식을 통일화하기로 약속했다.  **03-20~03~21**  플레이어가 무기를 선택하면 무기를 바꾸고, 무기에 따라 다양한 공격효과를 줄 수 있도록 만들었다.  추가로 활/스태프를 선택했을 때 투사체를 발사하는 기능 또한 완성했다.  **03-22**  테셀레이션 된 실외지형에 대한 높이 처리를 서버에서도 정확하게 할 수 있도록 변경하였다.  현재 정의된 모든 패킷 데이터를 보낼 수 있도록 서버 구조를 조금 수정하였다. | | | |
| EmberFall 개발 일지 | | 작성자 : 정영기 | |
| **12주차** | **2025.03.16 ~ 2025.03.22** | | 이번 주 진행 사항 |
| **03-16**  전체적인 3패스 렌더링을 완성하였다. 하지만, 애니메이션 행렬이 그림자, G버퍼의 2번 렌더링의 비용이 상당히 비싸 프레임이 상상 이상으로 낮아졌다. 이 문제를 해결하고자. 그림자와 G버퍼를 한 패스에 진행하는 방법에 대해 탐구하였다.  **03-17**  고안해낸 첫번째 방법은, G버퍼에 그림자맵을 추가하는것이다. 서로의 계산 과정에서 카메라 상수버퍼와 조명 상수버퍼를 다르게 줘서 서로 다른 시점에서의 렌더링을 기대하였다. 하지만 그림자맵 렌더링 도중 조명버퍼를 제대로 연결했지만, 조명의 위치가 아닌 카메라의 위치에서의 그림자만 기록되었다. 그이유를 탐색해보니. G버퍼에 서로다른 시점의 정보를 기록하는 것이 불가능하다고 하였다.  **03-18**  두번째 방법은 G버퍼에 그림자맵이 아닌 카메라 시점에서의 각 픽셀의 그림자 여부와 그림자 정도를 나타내는 그림자 계수를 계산하여 렌더링하는 것이다. 하지만 생각대로 그려지지않았고, 원인을 찾아보았지만 해결을 하지 못했다.  **03-19**  위의 문제를 해결하고자 이유를 찾아보았지만 짐작만 했을 뿐 전혀 찾지 못하였고, 짐작해본 원인 마저도 실패로 돌아갔다. 결국 3패스로 돌아가게 되었다.  **03-20**  3패스의 문제점은 그림자의 품질이다. 그림자는 잘 그려지지만 넓은맵을 덮을정도의 큰 그림자맵을 그렸지만, 메모리 사용량문제로 크기의 제한을 걸어야 했다. 제한을 가진채로 그림자품질 향상을 위해 다양한 방법을 찾아보았지만, 그림자맵의 작은 크기로는 한계가 있었다.  **03-21**  다음 방법으로는 그림자맵을 위한 조명의 위치를 고정시키는 것이 아닌, 카메라의 위치와 방향에 따라 달라지게 하하는 것이다. 다시 말해 그림자맵이 프레임마다 다른 맵이 그려지게 되는 것이다. 해당 작업을 위한 그림자 관련 정보를 공부하고 작업을 시작하였다.  **03-22**  위의 작업을 위해 카메라 절두체의 크기를 이용해 그림자맵의 크기를 구하는 과정에서 잘못된 연산이 있었다. 해당 문제를 위해 수정중이며, 기존의 프레임워크의 약간의 수정이 있었다. 예를들어 그림자맵 생성에 카메라의 위치가 필요하니 생성 시점을 카메라 객체 생성 이후로 미루는 등의 수정이 있었다. | | | |

|  |  |  |
| --- | --- | --- |
| EmberFall 개발 일지 | | |
| **12주차** | **2025.03.16 ~ 2025.03.22** | 다음 주 목표 |
| 1. 김승범 ( 클라이언트 )   캐릭터가 무기를 들 수 있도록 하는 기능 구현  각 무기의 바운딩 박스를 파일에 출력하는 기능 구현  서버용 애니메이션 컨트롤러 구현   1. 김성준 ( 서버 )   애니메이션 동기화 상호작용 방식 수정 현재 발생된 서버 버그들 수정   1. 정영기 ( 클라이언트 )   그림자 작업 완료, 계산 셰이더 학습 | | |
| **특이사항** | | |
|  | | |