|  |  |  |
| --- | --- | --- |
| EmberFall 개발 일지 | | |
| **11주차** | **2025.03.09 ~ 2025.03.15** | 회의 내용 |
| 서버와 클라이언트간의 패킷 규격을 정하는 시간을 갖는다.  컨텐츠의 추가로 인한 패킷의 추가는 추후에 진행 할 예정이다.  모든 패킷은 패킷의 사이즈, 타입, 송신자의 ID를 멤버로 갖는 패킷 헤더 구조체를 상속한다.  성수, 십자가와 같은 아이템은 프레임마다 상태여부를 보내면 버퍼가 포화될 가능성이 있으므로 이벤트로 처리한다.  오브젝트 패킷은 오브젝트의 ID, 회전 각도, 위치를 멤버로 가진다.  서버는 클라이언트에게 오브젝트의 등장과 퇴장을 알리는 패킷을 보내며 해당 패킷들은 오브젝트ID, 엔티티 타입, 체력, 회전 각도, 위치를 갖는다.  클라이언트는 서버에게 카메라의 위치와 방향을 구조체로 선언하여 보내고, 입력한 키와 키의 Down여부를 구조체로 선언하여 보낸다.  공격방향의 패킷과, 선택한 무기의 타입, 플레이어의 역할을 나타내는 패킷 구조체를 선언한다.  매주 대면 회의 날짜를 토요일에서 모두가 강의가 없는 수요일로 변경하였다.  애니메이션 관련 패킷은 다음주 수요일 회의에 논의하기로 합의하였다. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| EmberFall 개발 일지 | | 작성자 : 김승범 | |
| **11주차** | **2025.03.09 ~ 2025.03.15** | | 이번 주 진행 사항 |
| **03-09**  UI 등에 활용될 화면 위에 글자를 그리는 기능을 추가하였다.  DirectWrite 를 사용하여 구현하였으며, 외부의 폰트 파일을 읽어, 한글, 영어 모두 그릴 수 있도록 구현하였다.  화면에 그릴 글자의 내용을 결정하는 부분에서 TextBox 라는 객체를 받아, 이 객체를 할당받았다면, 언제나  자신이 할당받은 영역의 글자 내용이나, 영역 크기, 폰트 등을 변경 할 수 있도록 제작하였다.  **03-10**  맵 배치 이전에 가장 기본적으로 사용될 오브젝트들을 지형 위로 정확히 올리는 기능을 개발하기 시작하였다.  기존 Height map 만을 읽어 지형 모델을 생성하여 그렸다면, 단순히 Height map 만을 읽어 겹선형 보간으로  정확한 위치를 결정 할 수 있으나, 현재 우리 게임의 지형은 Tessellation 을 통해 높이를 결정하므로,  CPU 에서도 Tessellation 으로 정점 생성을 시뮬레이션 하여 원하는 x,z 위치의 높이를 겹선형 보간 하여야지만  정확한 높이를 구할 수 있다.  하지만 이것을 단순히 매 프레임마다 오브젝트의 개수 만큼 시뮬레이션을 반복한다면, Frame rate 가 낮아지는  것은 자명하므로, 미리 지형의 모든 점을 Max Tessellation Factor 만큼 시뮬레이션을 하여, 각 점의 높이를  파일에 기록 한 후, 이를 미리 모두 읽어 런타임에서는 겹선형 보간만을 진행하는 것이 합당할 것이다.  따라서 오늘은 Tessellation 을 시뮬레이션 하여 파일에 기록하는 것을 시도하였으나, 아직은 제대로 된 결과를  얻지 못했다  **03-11**  어제 전체 Tessellation 을 시뮬레이션 하여 파일에 기록하는 것 대신, 런타임에 그 부분의 패치에 해당하여  Tessellation 을 시뮬레이션 하여 정확한 높이를 구하는 것을 시도하였다.  이는 정확한 높이를 구하는데 성공하였으나, 예상한 대로 Frame rate 가 10 이하로 내려가는 현상이 발생하였다.  따라서 현재 만든 패치에서만 Tessellation 을 시뮬레이션 하는 코드를 토대로, 전체 Tessellation 을 시뮬레이션  하여 파일에 기록하는 것에 성공하였다. 이를 읽어 런타임에 정확한 높이를 구하는 데 성공하였으며,  오늘 오브젝트들을 지형 위로 올리는 GetHeight 함수를 구현하였다. | | | |
| EmberFall 개발 일지 | | 작성자 : 김승범 | |
| **11주차** | **2025.03.09 ~ 2025.03.15** | | 이번 주 진행 사항 |
| **03-12**  오늘은 애니메이션 클립 사이의 전환을 부드럽게 하는 애니메이션 블렌딩을 시작하였다.  애니메이션 블렌딩은 현재 Animation 카운터 ( 시간 ) 을 기준으로 현재 재생 중인 애니메이션 키 프레임과  전환 되어야 할 애니메이션의 첫 번째 키프레임 을 일정한 시간동안 보간 하여 재생함으로써, 애니메이션 전환  사이의 각 관절의 순간이동을 하지 않도록 만드는 것이다.  이를 위해, 기존에 있던 Animator 가 여러 개의 애니메이션 클립 을 저장 할 수 있도록 만들고,  ReadNodeHeierarchyTransition 함수를 추가하여, Bone Node 들을 순회 하면서, 두 개의 애니메이션 클립의  키 프레임들을 보간 할 수 있도록 ( 일종의 겹선형 보간 처럼 작동한다 ) 구현하였다.  이를 통해 오늘 애니메이션 블렌딩 자체는 구현하는데 성공하였으나,  결과물이 아직 마음에 들 만큼 자연스럽다는 느낌이 들지 않았다.  **03-13**  어제 구현했던 애니메이션 블렌딩이 보다 자연스럽게 작동할 수 있는 방법을 고민하였다.  첫 번째로, 각 애니메이션에 root node 에 position key 에 내용이 있는 경우, 모델 자체가 이동하면서,  블렌딩이 부자연스럽게 일어난다는 것을 알았다. 이를 위해, 우리가 인간 캐릭터로 사용할 기사 모델에  애니메이션을 새로 입히고, 이 과정에서 모든 애니메이션의 root node 에 position key frame 을 제거하여,  모든 동작이 제자리에서 일어나도록 변경하였다.  두 번째로, 두 애니메이션 클립 사이의 보간 간격 ( 시간 ) 이 너무 짧아, 실제로는 블렌딩이 일어나고 있으나,  눈으로 보기에 너무 짧아, 마치 순간이동 처럼 느껴진다는 것을 확인하였다. 이를 위해 수치를 조정해가며  블렌딩이 눈으로 확인되면서도, 자연스럽게 넘어 갈 수 있는 시간은 0.9 초를 적용하여 자연스럽게 전환되는 것  처럼 보이도록 수정하였다.  이 과정을 통해 애니메이션 블렌딩이 만족스러울 정도로 자연스럽게 전환 되는 것을 확인하였으며, 그 결과를  팀원들과 공유하였다. | | | |
| EmberFall 개발 일지 | | 작성자 : 김승범 | |
| **11주차** | **2025.03.09 ~ 2025.03.15** | | 이번 주 진행 사항 |
| **03-14**  오늘은 교과 수업과 어머니의 생신 축하를 위해 집으로 이동 하는 과정에서 추가적인 작업을 진행 할 수 없었다.  **03-15**  나의 마지막 중점 연구 과제인 애니메이션 상-하체 분리를 시작하였다. 이를 위해 Bone-Mask 를 도입하여,  원하는 Bone Group 를 둘로 나누어, 각 Group 에 적용 할 애니메이션을 선택 할 수 있도록 만들고,  각 Group 에 서로 다른 Animation 이 Set 되었을 때에는 각각 Bone Transform 을 계산하도록  구현할 예정이다.  아직은 설계 단계이나, 월요일 까지 상-하체 분리를 완전히 구현하는 것이 현재 목표이다. | | | |
| EmberFall 개발 일지 | | 작성자 : 김성준 | |
| **11주차** | **2025.03.09 ~ 2025.03.15** | | 이번 주 진행 사항 |
| **03-09 – 03-10**  감기몸살로 진행 X  **03-11**  아이템 효과들 구현 완료  **03-12**  클라이언트- 서버간 통신을 위한 패킷을 정의하고 패킷을 보내는 순서에 대해 정의함.  1. 클라이언트-서버 연결간에 서버는 클라이언트에게 그 클라이언트의 ID를 제일 먼저 보냄  2. 그 후에 서버는 클라이언트에게 자신의 패킷 프로토콜 버전을 보냄 프로토콜 버전이 다르면 패킷타입이 이상하거나 구조체 형식이 다를 수 있으므로 실행중에 오류가 발생할 수 있으므로 이를 방지하기 위한 장치로 설계함  **03-13~03-14**  클라이언트에서 패킷을 서버 코드를 몰라도 패킷 구조체만 알면 구성해서 보내고, 서버에서 온 패킷도 동일하게 처리할 수 있도록 패킷 처리함수를 등록하면 패킷 처리를 자동으로 할 수 있도록 해주는 클래스를 완성함/  PacketProcessor는 클라이언트에서 입력을 처리하는 것 처럼 패킷 타입에 대한 처리 함수만 등록해주면 현재까지 받아온 패킷에 대해 등록된 함수를 호출하는 역할만 하도록함.  처음에는 어떤 파라미터가 있든 상관없이 등록할 수 있도록 만들고자 했으나, 람다 타입추론, C++의 템플릿 타입 추론 방식 때문에 가능은 하지만 당초의 목적과는 조금씩 달라지는 부분이 있어 포기함..  **03-15**  캐릭터나 몬스터의 공격과 보스의 스킬을 구현 중.. | | | |
| EmberFall 개발 일지 | | 작성자 : 정영기 | |
| **11주차** | **2025.03.09 ~ 2025.03.15** | | 이번 주 진행 사항 |
| 03-09  그림자의 리소스를 가지는 그림자 클래스를 작성하였고, 그림자 리소스의 DSV, SRV를 만들었으며,  리소스의 상태전환 기능, 핸들 게터등등 그림자관련 기능을 갖추었다.  추후 추가할 기능이 있으면 추가 할 예정이다.  조명은 후에 조명 클래스를 구현 예정이지만, 일단 태양 역할을 하는 하나의 조명만 그림자맵을 가질 예정이기 때문에, 임시로 조명 정보와, 상수버퍼를 클래스내에 가진다  03-10  그림자맵을 렌더링 하기위한 그림자 셰이더를 작성한다. 조명 상수버퍼, 인스턴싱 월드 변환 행렬을 가진다.  해당 셰이더를 컴파일 하여 바이너리 파일에 저장할 수 있도록 하였으며, 베이스 셰이더 클래스를 상속받아 그림자맵 PSO를 제작중이다.  03-11  수업일정과 수시평가준비로 인한 해당없음  03-12  10일차 진행내용을 이어 진행했으며, PSO가 완성 되었고 렌더링 파트를 제작할 준비가 되었다.  03-13  그림자패스를 작성하였으며, G버퍼패스의 다음인 두번째 패스로 렌더링을 진행하였다.  03-14  .그림자패스를 렌더링 코드에 추가하였을 때, 제대로 돌아가지 않는 문제점이 발견되었고, 하루동안 해결하는데 시간을 썼다.  03-15  해결의 실마리로 메쉬의 정보를 담고있는 Context를 지워버린 것이 문제였으며, 해결을 하는 중이다. | | | |

|  |  |  |
| --- | --- | --- |
| EmberFall 개발 일지 | | |
| **11주차** | **2025.03.09 ~ 2025.03.15** | 다음 주 목표 |
| 1. 김승범 ( 클라이언트 )   애니메이션 상-하체 분리, 맵 배치 시작   1. 김성준 ( 서버 )   공격 구현, 보스 스킬 구현   1. 정영기 ( 클라이언트 )   계산셰이더를 이용한 블러링 작업 | | |
| **특이사항** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| EmberFall 개발 일지 | | |
| **11주차** | **2025.03.09 ~ 2025.03.15** | [ 별첨 1 ] 1차 패킷 구조 |
|  | | |