|  |  |  |
| --- | --- | --- |
| EmberFall 개발 일지 | | |
| **7주차** | **2025.02.09 ~ 2025.02.15** | 회의 내용 |
| 금주 회의는 남은 2주간의 계획에 대해 의논하는 시간을 가졌다.  김승범 - 평면위에 정육면체 그리기를 이번주 내에 완료할 예정이며,  02-22까지 애니메이션 파싱을 제작한 이후, 개강 직전까지 애니메이션 띄우기를 진행할 예정이다.  김성준 – 1주일의 기간동안 공간 분할을 작성할 예정이며, 개강 직전까지 공격 구현이 목표이며  여유가 있다면 몬스터 AI의 일부도 할 예정이다.  정영기 – 실외지형 그림자 마무리 예정이며, 실제 프로젝트에 쓰일 그림자 모듈화, 디퍼드 렌더링으로 그림자 구현을 완료한 이후,  컴퓨트 셰이더로 블러링 구현, 헐 셰이더, 도메인 셰이더로 테셀레이션 작업, 큐브매핑, 노말매핑 연습을 진행할 예정이다. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| EmberFall 개발 일지 | | 작성자 : 김승범 | |
| **7주차** | **2025.02.09 ~ 2025.02.15** | | 이번 주 진행 사항 |
| **2월 9일**  텍스쳐와 텍스쳐를 동적 인덱싱 방법으로 관리하는 Material 을 관리하는 TextureManager / MaterialManager  를 만들었다. TextureManager 는 지정한 폴더 안의 모든 이미지 파일을 읽어오도록 구현하였다.  **2월 10일 – 2월 12일**  독감 – 장염으로 정상적으로 작업을 진행할 수 없었다.  **2월 13일**  현재 구현하고 있는 컴포넌트-시스템 방법의 구현이 너무 길어지고, 앞으로도 남은 부분이 너무 많다.  하나하나의 문제를 해결하는데 시간이 너무 오래 걸려, 현재 구현하고 있는 내용의 일부를 포기하고,  기존 ( 3D 게임 프로그래밍 1/2 ) 에서 활용하기 위해 만들었던 구조를 활용하기로 노선을 변경하였다.  오늘은 필요없는 파일을 정리하고, GameObject 를 만들었다  **2월 14일**  16일까지 평면 위에 정육면체를 띄우는 샘플을 만들기로 결정하였다. 기존 구조에 있던 Transform 에서  필요 없는 부분을 제거하고, 현재 구조에 맞게 인터페이스를 변경하였다.  프러스텀 컬링과, 디테일한 충돌 판정을 위한 Collider 를 만들었다.  Collider 는 GameObject 에 쓰일 일반 Collider 와, 카메라 ( FrustumBox ) 를 위한 FrustumCollider 를  만들었다.  **2월 15일**  Collider 가 단지 Box 만을 가지지 않고, 하나의 Callable 을 페어로 가지게 하여, 충돌했을 때 콜백을  설정 할 수 있도록 변경하였다. Scene 을 만들고, 예제 Mesh 와 텍스쳐를 만들어 샘플을 만들 준비를 하였다. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| EmberFall 개발 일지 | | 작성자 : 김성준 | |
| **7주차** | **2025.02.09 ~ 2025.02.15** | | 이번 주 진행 사항 |
| **02-09**  공유변수 문제 때문에 서버가 죽는 현상을 수정했다. Player를 추가/삭제하는 과정은 멀티쓰레드 에서 돌아가는데 이에 대한 적절한 조치들은 취해주지 않아 생기는  문제였다.  **02-10**  중력을 적용 하는데에 오류가 있어서 이를 수정했다.  플레이어에 대한 스크립트를 작성하는데 Input 클래스가 게임 오브젝트에 붙어있는 구조를 수정하고 플레이어의 생성/삭제시에 Input객체도 같이 생성/삭제되도록 관리하고 PlayerScript가 기본적으로 Input객체를 참조하도록 수정했다.  **02-11**  여러 오류들을 수정하는 데에 로그를 관리하기가 힘들어져서 로그를 출력하는 클래스를 따로 만들기로 했다. 다만 로그 출력하는 cout, file I/O 는 꽤나 비싼 연산이기에 게임루프가 돌아가는 와중에는 바로바로 출력되지 않도록 하고 따로 이를 전담하는 쓰레드를 하나 생성해서 관리하기로 하였다. 로그를 출력해야 하면 이 클래스에 Push를 하고 로그가 큐에 쌓여 있다면 자동적으로 전담 쓰레드가 로그를 출력하도록 구성했다.  오늘 오브젝트에 대한 송/수신도 진행하게 되었는데 빈번하게 서버가 죽어버리는 현상이 발생하여 해결하고 있다. 현재 패킷을 너무 많이 보내기도 하고 네트워크 송/수신 버퍼에 쌓인 내용을 무조건 1024바이트씩만 복사해서 처리하고 있기 때문에 문제가 생겼다. 일단 버퍼의 크기를 늘리고 한번에 처리하는 양을 늘려서 문제를 해결했다.  또, 현재 Overlapped 구조체에서 자신이 네트워크 작업이 끝나고 후처리를 어디서 해야 할지 자신의 관리 주체를 shared\_ptr로 참조하도록 하고 있는데, GQCS 함수에서 에러가 발생했을 때 이 shared\_ptr 참조를 적절히 해제해주지 않아서 데드락 문제와 객체가 소멸하지 않는 현상이 생기고 있어 오류처리 부분을 다시 수정했다. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| EmberFall 개발 일지 | | 작성자 : 김성준 | |
| **7주차** | **2025.02.09 ~ 2025.02.15** | | 이번 주 진행 사항 |
| **02-12**  ViewList작성  어제의 문제에서 지금 패킷을 너무 많이 보낸다는 사실을 깨달았다.  ViewList를 적용하면 좋다는 말을 들었던 기억이 있어서 이를 적용해보기로 했다 먼저 ViewList는 사용자의 시야 범위에 있을만한 오브젝트만을 골라서 패킷을 보내는 것이다.  일단은 구형 범위를 계산해서 적용시켰다. PlayerScript에서 자동으로 생성을 하도록 하고 송신을 위한 함수는 Session과 SessionManager에 정의 되어 있으므로 SessionManager를 참조하도록 했다. PlayerScript에서 이미 현재 플레이어가 있는 게임 씬을 참조하도록 했으므로 ViewList도 이에 맞게 현재 게임씬을 참조하고 Update 함수에서 자신의 시야범위에 들어오는지 검사하도록 했다. 범위에 있는 오브젝트들은 Set에 저장하기로 했다.  std::set에 저장한 이유:  추가 삭제가 너무 빈번하게 일어난다. 범위가 충분히 크다면 (맵 전체정도?) 추가/삭제는 거의 일어나지 않겠지만 이러한 상황은 그리 바람직한 상황은 아니다.  범위를 너무 크게 잡으면 ViewList를 쓸 이유가 없다. 또, 중복되어 있는지에 대한 여부를 계속해서 검사해주어야 한다. 이 두가지 이유 때문에 std::set을 썼다.  물론 std::vector보다는 순회속도가 현저히 느리겠지만 추가/삭제가 매우 빈번하게 일어나는 상황에서 std::vector를 쓰는 것 보다는 나을 것이라는 생각에 set을 사용했다.   게임 단위 설정 게임 단위에 대한 부분을 std::chrono::literals 처럼 하자는 승범이의 제안에 대해서 생각해보고 만들어보게 되었다.  아직 단위간 연산은 어떻게 구현해야할지 더 생각해봐야하는 문제라고 생각하여 넣지 않았다. 대부분은 std::chrono를 참고해서 작성했고, 단위들은 float으로 표현하도록 작성했다.  **02-13**  게임 단위 작성 단위 간에 곱셈,덧셈, 나눗셈 까지 작성을 해놔서 잘 작동하는거 같기는 한데… 실제 물리연산이나 게임 단위를 적을 때 어떻게 적용해야할 지는 조금더 생각해봐야겠다. 아마도 속력→속도, 힘→속도 이런식으로 변환하려면 별도로 함수를 더 만들어야 할거 같다.  ViewList에 Player 추가, Scene에서 SendUpdateResults 삭제 ViewList에서 Send를 매번 하고 있으므로 더이상 SendUpdateResults라는 함수는 필요 없다고 판단되어 삭제하게 되었다. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| EmberFall 개발 일지 | | 작성자 : 김성준 | |
| **7주차** | **2025.02.09 ~ 2025.02.15** | | 이번 주 진행 사항 |
| **02-14**  충돌이 이상하게 일어나거나 일어나지 않는 문제를 해결했다. 충돌상태를 오브젝트의 ID를 이용해 구별하고 있는데 플레이어가 아닌 오브젝트에게 별도로 ID를 부여하지 않아서 생기는 문제 였다.  네트워크로 전송할 때도 어짜피 필요하기에 오브젝트에 ID를 기억할 수 있도록 했다.  **02-15**  충돌처리에서 시간을 너무 많이 쓰고 있다는 생각이 들어서 공간 분할에 대해 생각하고 이를 구현하려고 하고 있다. 일단은 Grid 방식으로 구현해보기로 하고 있다. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| EmberFall 개발 일지 | | 작성자 : 정영기 | |
| **7주차** | **2025.02.09 ~ 2025.02.15** | | 이번 주 진행 사항 |
| **02-09**  조명에서의 문제점을 해결하였다. 셰이더에서 잘못된 행렬을 사용하였던 것이 원인이었다.  **02-10**  태양빛과 유사한 방향성 조명을 이용해 평지위에 그림자를 만들었다. 하지만 그림자 영역과 비그림자 영역의 경계가 뚜렷하여 현실감이 조금 떨어지므로 PCF를 이용해 그림자 모서리에 흐릿한 그림자를 구현해야한다.  **02-11**  2x2 PCF와 3x3PCF를 구현하여 흐릿한 그림자 모서리를 구현하였다. 오브젝트에서 미세한 그림자여드름이 발견되었고, 그림자맵의 해상도가 낮아 계단현상이 있는 상태이다.  **02-12**  bias값을 이용해 그림자여드름을 없애고, 해상도를 높이고, 그림자맵의 해상도를 높혀 계단현상을 없앴다.  실외지형 그림자를 위한 실외지형 이론 학습을 열린 강의, PPT 예제 샘플로 진행하였다.  **02-13**  컨디션 난조로 인한 해당 없음  **02-14**  실외지형에 그림자를 표현하기 위해 실외지형을 띄우는 작업을 하였다. 파일을 읽어온 후, 정점버퍼와 인덱스버퍼를 만들어 셰이더에 전달하였다.  **02-15**  실외지형 그림자 마무리 이후, 컴퓨트 셰이더 이론 학습을 하였다. | | | |

|  |  |  |
| --- | --- | --- |
| EmberFall 개발 일지 | | |
| **7주차** | **2025.02.09 ~ 2025.02.15** | 다음 주 목표 |
| 1. 김승범 ( 클라이언트 )   GameScene 샘플 완성하기  애니메이션 파싱 시작하기.     1. 김성준 ( 서버 )   1. 공간분할 2. 게임 이벤트 처리 (몬스터/플레이어 공격, 안개 생성 등등…) 3. 길 찾기 알고리즘 적용   1. 정영기 ( 클라이언트 )   프로젝트에 쓰일 그림자 모듈화, 디퍼드 렌더링으로 그림자 구현  컴퓨트 셰이더로 블러링 구현, 헐 셰이더, 도메인 셰이더로 테셀레이션 작업 | | |
| **특이사항** | | |
|  | | |