|  |  |  |
| --- | --- | --- |
| EmberFall 개발 일지 | | |
| **4주차** | **2025.01.19 ~ 2025.01.25** | 회의 내용 |
| 금주 회의는 컨벤션을 필수적인 부분을 제외하고 삭제하는 시간을 가졌다.  1.모듈 외부로 노출되는 인터페이스가 아니라면 모듈 내부 코드의 매개변수와 반환 값은 언제나 유효해야 합니다.  2.디버깅이 필요한 코드에는 특별한 Crash함수를 사용합니다.  3.Crash 함수는 Debug모드의 경우 assert 함수를, Release 모드일 경우 따로 작성된 Crash매크로를 호출합니다.  4.재사용하지 않는 코드일 경우 함수로 작성하는 것을 되도록 지양합니다  .  필요성을 느끼지 못해서 컨벤션의 간소화를 위해 삭제가 필요하다고 판단하였다.  5.포인터()나 레퍼런스 기호(&)는 자료형에 붙입니다.  모두가 이미 사용하는 방식이기때문에 삭제를 하였다.  6.생성자의 Initialize List는 한 줄에 하나씩 작성합니다. 를  생성자의 Initialize List는 다음 줄에 모두 작성합니다. 으로 수정하였다.  그 이유는 나눠서 쓸 정도로 길 필요가 없기 때문이다.  7.goto문을 사용하지 않습니다.  쓰는 사람이 없기 때문에 삭제하였다.  이번 달 목표로는  김승범 – 예제Scene 완성 후, 평면위에 정육면체 띄우기  김성준 – 테스트 클라이언트 완성 후, 위치 동기화, FlatBuffer적용, 동기화 문제 해결 중  정영기 – 그림자 완성 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| EmberFall 개발 일지 | | 작성자 : 김승범 | |
| **4주차** | **2025.01.19 ~ 2025.01.25** | | 이번 주 진행 사항 |
| **1월 19일**  컴포넌트를 사용하기 위해, Scene 을 세 가지 파트로 나누기로 가정하였다.  관련 있는 정보를 저장하는 Component  컴포넌트들의 집합인 Entity  컴포넌트의 조합으로 로직을 구성하는 System  생성한 Game 프로젝트에 가정한 내용을 기반으로 베이스 클래스들을 제작하였다.  **1월 20일**  Component 의 조합인 GameObject 와, 조합을 연속된 메모리 공간에 저장하는 ArcheType 에 대해 생각하였다.  ECS concepts | Entities | 0.2.0-preview.18  Unity 의 자료를 토대로 이를 C++ 로 옮기기 위한 아이디어의 토대를 만들었다.   1. ArcheType 에는 Container 의 집합이 있다. 2. Container 에는 ComponentBase 의 배열이 있으며, 이 장소에는 ComponentBase 로부터 파생된   컴포넌트의 배열이 올 수 있다.   1. 이를 사용하여 타입이 늦게 결정 될 수 있는 배열을 만들어, ArcheType 의 구현에 적용한다. 2. 타입은 bitset 을 사용하여 구별하며, 모든 컴포넌트 타입은 각 비트의 자리수를 표현하는 index 라는   속성을 가지고 있어야 한다.   1. Container 는 컴포넌트의 개수에 따라 가변적으로 길이가 변할 수 있는 자료구조이다. 2. GameObject 는 자신의 컴포넌트 조합이 있는 ArcheType 포인터와, ArcheType 에서의 index 를 가진다.   **1월 21일**  20일의 결과를 토대로 ArcheType 과 GameObject 를 구현하였다.  앞으로는 이를 테스트 할 수 있는 기본 System 과 Scene 을 만들어, 내가 생각한 구조가 정상적으로 작동하는지  확인해야 한다.  **1월 22일**  21 일에 진행한 결과에 약간의 실수가 있어 이를 보완하는 시간을 가졌다.  ArcheType 에서 원하는 컴포넌트를 찾는 부분에서 std::find 를 쓰는 대신 FlatMap 을 구현하여  자주 일어나는 컴포넌트 검색을 보다 빠르게 할 수 있게 함과 동시에 순차적 접근에도 유효하도록 만들었다. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| EmberFall 개발 일지 | | 작성자 : 김승범 | |
| **4주차** | **2025.01.19 ~ 2025.01.25** | | 이번 주 진행 사항 |
| **1월 23일**  제작한 FlatMap 과 std::map, std::unordered\_map 사이의 퍼포먼스 테스트를 진행하였다.  그 결과 FlatMap 이 오히려 unordered\_map 보다 전체적인 성능이 떨어지는 것으로 판단되어,  Unordered\_map 을 사용하기로 결정하였다.  ArcheType 에 FlatMap 으로 되어있는 파트를 모두 unordered\_map 으로 변경하는 작업을 하였다.  Scene 에도 같은 방법으로 ArcheType 들을 저장하도록 만들었다.  **1월 24일**  해당없음  **1월 25일**  현재 Framework 중, ArcheType 과 GameObject 는 완성되었다.  남은 System 과, 이것을 Directx12 와 연결하는 과정에 대해 고민하였다.  특히 Shader-Material 로 연결되는 부분에 있어서 배치 처리를 하는 것이 Set 의 횟수가 적어 효율적인데,  이것을 현재 게임 시스템과 어떻게 연결지을지에 대한 생각을 하는 시간을 가졌다.  1주일 간 남은 1월 동안의 목표를 Scene Framework 를 완성하고 예제 Scene 을 만들어 보는 것으로 조정하였다. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| EmberFall 개발 일지 | | 작성자 : 김성준 | |
| **4주차** | **2025.01.19 ~ 2025.01.25** | | 이번 주 진행 사항 |
| **01-20 ~ 01-21**  클라이언트에도 네트워크 기능을 넣기 위해 Server 라이브러리에 클라이언트 기능을 추가.  기존 구조 그대로 Session 클래스가 데이터 송/수신을 담당하게 하고 클라이언트에는 별도의 SessionManager, Listener 클래스가 존재하지 않도록 함.  이미 만들어둔 Session, PacketHandler, IOCPCore, WorkerThread들은 공통되는 부분이기에 원래 전역변수로 선언되었던 변수들을 하나로 묶어 인터페이스를 구성하고 그 인터페이스를 상속한 서버, 클라이언트 클래스를 만들어 사용하도록 만듦.  **01-22**  이전에 만든 서버 라이브러리를 적용시키고 접속테스트 까지 해보는데 성공함.  접속과정에서 Connect함수가 작동하지 않는 오류, 클라이언트의 작업자 쓰레드가 정상적으로 종료되지 않는 오류들을 수정함.  **01-23**  송수신 과정에서 사용할 프로토콜을 임시로 정의하고 Protocol 파일을 만듦.  데이터를 구조체로 묶어 보내는 과정에서 TCP프로토콜이기에 데이터가 잘려서 오는 경우를 대비할 수 있도록 패킷의 크기와 현재 수신한 크기가 같은 지 확인하는 코드를 작성하고 키입력을 서버로 보내고 클라이언트에 업데이트 결과를 텍스트로 보내는 과정까지 테스트를 완료함.  **01-24**  해당 없음  **01-25**  테스트해볼 클라이언트에서 OpenGL을 활용해 정육면체를 띄우고 키입력을 보내는 것까지 완성.  클라이언트가 데이터를 송/수신하기 전에 자신이 서버에서 어떻게 식별되는지를 알 필요가 있다고 판단하여 연결 설정 이후 바로 서버에서 클라이언트에게 자신의 ID를 알려주는 코드를 작성했으나 제대로 동작하지 않아 원인을 파악하고 있음. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| EmberFall 개발 일지 | | 작성자 : 정영기 | |
| **4주차** | **2025.01.19 ~ 2025.01.25** | | 이번 주 진행 사항 |
| **01-19**  종합설계 프로젝트에서 가장 먼저 그림자를 구현 할 계획이기 때문에 그림자매핑 학습을 시작하였다.  평면 투영 그림자를 학습하였다.  **01-20**  그림자매핑을 위한 투영 텍스쳐 매핑을 학습하였고, 스텐실 적용의 이론과 예제를 학습하였다.  **01-21**  기하셰이더의 이론을 학습하였고,이를 바탕으로 빌보드 적용하였다, 게임수학교재를 통한  그림자매핑을 학습하였다.  **01-22**  빌보드에 다양한 텍스쳐를 적용하기위해 텍스쳐 배열을 학습하였고 적용하였다,  그림자 매핑 학습, 그림자의 PCF를 학습하기위한 멀티샘플링을 복습하였다.  **01-23**  그림자매핑의 PCF를 학습하였다,  그림자매핑 학습에서 기존의 이론이 부족하다고 판단하여 다이렉트X의 행렬 변환 복습하였다.  **01-24**  개인 사정으로 인해 해당없음  **01-25**  그림자매핑의 방향성조명 그림자를 학습하였다, 다이렉트X의 행렬 변환 복습을 이어나갔다. | | | |

|  |  |  |
| --- | --- | --- |
| EmberFall 개발 일지 | | |
| **4주차** | **2025.01.19 ~ 2025.01.25** | 다음 주 진행 사항 |
| 1. 김승범 ( 클라이언트 )   Scene Framework 완성. 평면 위 정육면체를 띄우는 예제 Scene 제작.  3D 애니메이션 구현 시작.   1. 김성준 ( 서버 )   위치 전송 테스트, 실외지형 띄우기 Flatbuffer 적용   1. 정영기 ( 클라이언트 )   그림자 작업물 완성, 계산 셰이더, 테셀레이션, 절두체 컬링 | | |
| **특이사항** | | |
| 김승범 ( 1월 24일 ) – 학교에서 집으로 귀가하여, 잠시 쉬는 시간을 가짐.  김성준 ( 1월 24일 ) – 학교에서 집으로 귀가하여, 잠시 쉬는 시간을 가짐.  정영기 ( 1월 24일 ) – 개인사정으로 인한 휴식 | | |