|  |  |  |
| --- | --- | --- |
| EmberFall 개발 일지 | | |
| **9주차** | **2025.02.23 ~ 2025.03.01** | 회의 내용 |
| 금주 회의는 기존에 작성했던 종합설계기획의 기획서를 검토하는 시간을 가졌다.  보석과 아이템은 서버에서 구현이 완료되었다.  AI는 현재 구현중이며, 다시 설계해야 할 수도 있다. .  스킬은 좀더 디테일한 설정이 필요하다. 예를들어 안개가 퍼져나가는가, 고정되었는가?  시야흔들림은 패킷으로 처리할것이다.  몬스터의 소환은 땅에서 나오는 애니메이션을 찾아보아야 한다.  클라이언트는 이펙트를 신경쓰고, 서버는 디테일한 설명이 필요하다.  성수는 1회성 공격으로 확정지었다.  아이템의 습득방식은 습득 애니메이션의 존재 여부로 정한다.  아이템 습득과 보석 파괴의 상호작용키는 F로 통일한다.  아이템과 보석이 겹쳐지는 경우 우선 순위는 후에 정하도록 한다.  아이템은 바닥에 있는 상태는 3D 인벤토리에 습득상태일때는 2D텍스쳐로 표현  달리기는 아직 적용하지 않는 것으로 하였다.  달리기는 후에 논의를 해봐야 할 것 같다.  십자가 효과는 플레이어 주변의 원의 형태로 금빛색상으로 빛나는 형태  예시로는 리그오브레전드의 귀환이펙트가 있다.  스테이지의 길 표현을 위해 길을 정해야한다. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| EmberFall 개발 일지 | | 작성자 : 김승범 | |
| **9주차** | **2025.02.23 ~ 2025.03.01** | | 이번 주 진행 사항 |
| **02-23**  전날 만든 AssetLoader 로 불러온 에셋 파일을 렌더링하였다.  과정 중, PlainMesh 를 Bind 하는 과정에서 잘못된 view 들을 연결하는 문제를 확인하여 수정하였다.  **02-24**  1, 2 스테이지에서 사용되는 실외 지형을 만들기 시작하였다.  기존의 지형에서 카메라의 위치에 따라 지형을 테셀레이션하여, 지형을 보다 자세하게 그리되,  지형을 그리는데 걸리는 부담은 줄어들도록 변경하기로 결정하였다.  이를 위해 기존 삼각형 스트립으로 그리던 실외 지형을 5x5 패치의 형태로 기본 메시를 만든 후,  패치 테셀레이션을 하기로 결정하였다.  오늘은 지형의 5x5 패치 데이터를 만들고, 지형 테셀레이션 셰이더를 만들었다.  **02-25**  전날 만들었던 메시 데이터가 잘못된 부분이 있어 수정하였다.  오늘은 지형을 실제로 렌더링 하여 지형이 원하던 대로 LOD 가 잘 구현된 것을 확인하였다.  하지만 wireframe 으로 볼 때와 달리, solid 도 보았을 때, 패치 사이에 틈이 발생하여 보기 좋지 않다는 문제가  발생하였다.  이에 대한 정보를 탐색하여 이 현상이 T-Juntion 으로 불린다는 것을 확인하고, 해결 방법에 대해 고민하였다.  테셀레이션 팩터를 linear 하게 조정하거나, 패치 사이의 틈을 domain shader 에서 임의로 채우는 방법을  시도하였으나, 이것으로는 잘 해결되지 않았다.  **02-26**  지형의 문제를 해결하는 것을 잠시 뒤로 미뤘다.  오늘은 기존에 사용하던 127 \* 127 크기의 height map 에서 잠시 벗어나, 더 큰 height map 을 렌더링 해 보았다. | | | |
| EmberFall 개발 일지 | | 작성자 : 김승범 | |
| **9주차** | **2025.02.23 ~ 2025.03.01** | | 이번 주 진행 사항 |
| **02-26**  지형의 문제를 해결하는 것을 잠시 뒤로 미뤘다.  오늘은 기존에 사용하던 127 \* 127 크기의 height map 에서 잠시 벗어나, 더 큰 height map 을 렌더링 해 보았다.  인터넷에서는 height map 을 주로 raw 파일로 관리하지 않고, png 파일로 관리하여, 이를 원활히 사용하기 위해  Png 파일을 읽어와, 이를 그레이 스케일 값으로 변경하고, 0 – 255 사이의 값으로 정규화 하여, raw 파일로 만드는  파이썬 코드를 만들어 png height map 을 raw 파일로 변경하였다.  이를 사용하여 LOD 실외 지형을 렌더링 해 본 결과, 기존의 방식으로는 할 수 없었던 대형 지형 렌더링이 가능함을  확인하였다.  **02-27**  오늘 부터는 애니메이션 파싱과, 렌더링을 시작하였다.  기존에 애니메이션 로딩과 렌더링을 구현한 코드가 있어 이를 참고하였다.  애니메이션 파일을 파싱한 결과는 크게 3가지로 나눠 저장하였다.  BoneAnimation – 하나의 Bone 에 대한 키프레임들을 저장, Bone 하나당 하나의 객체를 사용하며,  위치, 회전, 스케일 에 대응하여 각각 < 시간 , 값 > pair 배열을 가지고 있다.  BoneNode – 스켈레톤 애니메이션은 계층 구조로 이루어져 있으므로, 이러한 계층을 저장하는 객체  안에는 노드 이름, Local Transform, 자식 노드들 로 구성되어 있다.  AnimationClip – 하나의 애니메이션 사이클을 저장하는 객체  애니메이션 시간, 1초당 tick 횟수, Bone offset 배열, Root Bone Node, Bone Index 맵,  BoneAnimation 맵 으로 구성되어 있다.  오늘은 애니메이션 파싱을 완성하였다. | | | |
| EmberFall 개발 일지 | | | 작성자 : 김승범 | |
| **9주차** | | **2025.02.23 ~ 2025.03.01** | | 이번 주 진행 사항 |
| **02-28**  기존 Asset Loader 에서 여러 개로 분할된 에셋 파일을 제대로 읽어오지 못하는 문제가 있었다.  Asset Scene 안에 있는 mesh 배열을 순회하며 데이터들을 저장하는데, 이 때, 인덱스를 초기화 시키지 않고  저장하였기 때문이였다.  따라서 여러 개로 분할된 mesh 도 올바르게 읽을 수 있도록 수정하고, 렌더링 되는 것을 확인하였다.  오늘은 파싱된 애니메이션을 활용하는 Animator 를 만들기 시작하였다.  Animator 는 BoneNode 를 따라 계층 구조를 순회하며, 키프레임 데이터에서 Animaton time 에 맞는 Bone 별  변환 행렬을 계산하고, 이것을 부모의 행렬에 곱하여, 최종적으로 해당 bone 의 월드 변환 행렬을 만드는 일을  한다.  오늘은 Node 하나에서 하는 일을 만들었다.  **03-01**  오늘은 전날 진행한 결과에서, Bone 별 변환 행렬을 만드는 부분에서 Animation Time 에 맞는 키프레임 데이터를  구하는 부분을 만들고, Node 를 순회하여 Animation Time 을 받아, 올바른 변환 행렬 배열을 만드는 함수를  만들었다. | | | | |
| EmberFall 개발 일지 | | | 작성자 : 김성준 | |
| **9주차** | | **2025.02.23 ~ 2025.03.01** | | 이번 주 진행 사항 |
| **02-23 ~ 02-24** 기획서 상의 역병 보석을 구현했다. 단순하게 F키를 누르면 플레이어의 ViewList 안에서 가장 가까운 오브젝트를 찾고 그 오브젝트가 역병 보석 오브젝트이면 색상을 바꾸는 것으로 상호작용을 구현했다.  **02-25**  어제 보석 파괴 상호작용을 조금 구현했었다. 오늘은 그 구조를 좀더 다듬기로 했다.  이미 GameEvent 처리는 만들어 놨으니 이 GameEvent를 조금 더 이용하면 될거 같아 확장해봤다. 일단 클라이언  에서 더 만들건 없으니, 서버에서 색상으로만 상호작용을 확인하기로 헸다.  새로운 게임 이벤트를 작성하고 상호작용 대상을 구별할 ID와 입력을 얼마나 유지했는지에 대한 정보를 플레이어  클래스에서 생성해 역병보석 오브젝트에게 전달하는 것으로 구현했다.  상호작용의 대상이 되는 오브젝트는 이 이벤트를 받아서 적절히 처리하면 되는데, 역병 보석은 색상을 바꾸고 5초가  지나면 파괴되도록 처리했다.  **02-26~02-27** 몬스터의 AI를 위한 BehaviorTree를 만들었다. 기본이 되는 BehaviorTree의 각 노드들은 다음과 같이 만들었다.  노드의 종류는 총 4개 Selector, Sequence, Condition, Action 노드가 반환하는 상태는 총 3가지 상태 FAIL, SUCCESS, RUNNING FAIL, SUCCESS는 말그대로 성공 실패 여부이고 RUNNING 상태는 아직 성공, 실패 여부가 결정되지 않았으며, 실행중인 상태이다.  Selector와 Sequence 노드는 컨트롤 노드이다. 내부적으로 1개 이상의 자식노드를 가진다.  Selector노드는 자신의 자식중 하나라도 SUCCESS를 반환하면 자신도 SUCCESS를 반환한다. 만약 하나의 자식 노드가 실패하면(FAIL을 반환하면) 다음 자식의 Update를 실행한다. 모든 노드가 실패하면 그제서야 Selector 노드도 FAIL을 반환한다.  Sequence노드는 자신의 자식 중 하나라도 FAIL을 반환하면 자신도 FAIL을 반환한다. 반대로 모든 자식이 SUCCESS를 반환하면 자신도 SUCCESS를 반환한다. 만약 하나의 자식 노드가 성공하면(SUCCESS를 반환하면) 다음 자식의 Update를 실행한다. 모든 노드가 성공하면 SUCCESS를 반환한다. 컨트롤 노드는 특정한 행동을 수행하지는 않는다. 자신의 자식들을 Update만 한다. 특정 행동을 수행하는 노드는 Condition과 Action노드이다. | | | | |
| EmberFall 개발 일지 | | | 작성자 : 김성준 | |
| **9주차** | | **2025.02.23 ~ 2025.03.01** | | 이번 주 진행 사항 |
| **02-26~02-27** Condition노드는 Update를 호출하면 두가지 상태 FAIL, SUCCESS만을 반환한다. 즉, Condition노드의 Update는 반드시 한 프레임 내에 끝마쳐져야 한다. RUNNING 상태는 Condition노드에서 사용하지 않는다.  Action노드는 실질적인 행동을 수행한다. (EX: 움직이기, 배회, 공격, 길찾기 등등…)  Condition노드와 Action노드는 Leaf Node이며 더이상 자식을 가지지 않는다.  BehaviorTree 클래스는 단순하게 mRoot를 Set하고 Update하는 코드만 존재한다. 각자의 BehaviorTree를 만들때 BehaviorTree 클래스를 상속하고 생성자에서 트리를 생성하도록 했다.  오늘은 랜덤 위치로 이동하는 것만 테스트 했으며 내일부터 본격적으로 다른 행동들을 만들어보려고 한다.  **02-28~03-01** BehaviorTree를 만들면서 몇가지 간과한 문제점이 있어 그것부터 해결했다. 다음과 같은 문제점이 있었고 2일 동안 이것들을 모두 해결했다,  1. 몬스터 행동을 만들면서 shared\_from\_this()문제가 생긴 점.  생성자에서 shared\_from\_this() 호출은 불가능하니, 따로 Init함수를 두어 초기화하게 했다.  2. 몬스터가 플레이어를 추격하기 위해서 어떻게 플레이어를 담고있는 오브젝트 리스트에 접근할 것인지,  여러가지 생각은 해보았으나 결국 문제를 해결하기 위해서 조금 더 복잡하게 돌아가는게 아닌가 싶어  일단 오브젝트가 자신이 속한 게임 월드를 참조 할 수 있도록 했다.  3. 특정 행동만 반복하게 되는 문제.  이부분은 Sequence노드와 Selector 노드를 설계하면서 모든 행동이 끝나 거나 조건에 맞지 않는 행동을 해서 노드의 업데이트가 실패하는 경우 리셋하지 않고 그대로 실행하던 것이 문제였다. | | | | |
| EmberFall 개발 일지 | | | 작성자 : 정영기 | |
| **9주차** | | **2025.02.23 ~ 2025.03.01** | | 이번 주 진행 사항 |
| **02-23**  디퍼드 렌더링 작성 도중 첫번째 패스에서 G버퍼에 데이터가 저장되었지만, 비정상적으로 저장되었다.  예를들어 하나의 큐브모형을 여러 변환 행렬을 통해 여러 개의 큐브를 렌더링을 하였지만 원형이되는 하나의 큐브만 렌더링되었고, 실외지형도 렌더링되지 않았다.  **02-24**  전날의 문제점을 해결하고자 하였지만, 기존에 포워드렌더링을 위해 코드를 작성했기 때문에 디퍼드 렌더링으로의 전환과정에서 수정해야 할 부분이 필요이상으로 많았다. 그래서 기존의 코드를 전부 삭제하고 디퍼드렌더링 기반으로 새롭게 작성하였다.  **02-25**  새롭게 작성하였지만, 원하는 결과로 이어지지 않았다. 백버퍼의 렌더타겟이 초기화된 상태로 진행이 안되었으며,  저장이 되었는지, 저장은 되었지만 2번째 패스에서 발생한 문제인지 찾았다. g버퍼에 기록이 안되었다.  **02-26**  위의 문제를 해결하고자 다시 디퍼드 렌더링을 공부하였으며, 해결하고자 했지만 해결하지 못했다.  **02-27**  개인일정으로 인한 휴무  **02-28**  교수님의 샘플과 강의의 샘플을 활용하여 수정하였고, 잘 저장이 되지 않았다.  **03-01**  기숙사 입주준비와 입주로 인한 휴무 | | | | |

|  |  |  |
| --- | --- | --- |
| EmberFall 개발 일지 | | |
| **9주차** | **2025.02.23 ~ 2025.03.01** | 다음 주 목표 |
| 1. 김승범 ( 클라이언트 )   Animation 렌더링   1. 김성준 ( 서버 )   몬스터 AI 행동 추가. 2. 서버/클라이언트 연결(애니메이션 작업이 끝나는대로) 3. 길 찾기 구현   1. 정영기 ( 클라이언트 )   디퍼드 렌더링 해결, 그림자 모듈화, 계산셰이더 학습 | | |
| **특이사항** | | |
|  | | |