

[Cache Cache (캐시 캐시)]			
1 주	2019. 12. 30 ~ 2020. 1. 5	작성자	***
이번 주 한 일	<p>[0] 공동</p> <p>12. 30 제안서 발표</p> <p>01. 03 주간 회의 (1주차)</p> <ul style="list-style-type: none"> - 팀원 간 주간 목표와 진척도, 다음 주 목표 공유 - 맵 툴 저장 방식에 대하여 회의 진행 - 클라우드 서버 자원에 관한 회의 진행 <p>무료로 사용가능한 자원 CPU: 2개, 메모리: 4GB</p> <p>AWS계정 4개 이용 로비 서버 2, 매치 메이킹 서버 1, 배틀 서버 1 운용, 배틀서버는 다른 서버에 비해 자원이 더 필요하다 판단하여 AWS에 문의하여 연구 지원 받을 수 있는지 확인하기로 함.</p>		
	<p>[1] *** (서버)</p> <p>주간목표:</p> <ul style="list-style-type: none"> - 배틀서버 하이 레벨 디자인 100% 완료 - 로비서버 하이 레벨 디자인 50% 완료 <p>진척도: 배틀서버 50%, 로비서버 0%</p> <p>- 배틀서버 프레임워크 설계</p> <p>어떻게 배틀서버가 돌아갈 지 큰 흐름에 대한 설계는 완료, 클래스를 나누고 IOCP API를 다루는 부분 코드 일부 작성 완료.</p> <ul style="list-style-type: none"> - Thread, ThreadHandler 부분을 작성하면서 싱글톤 패턴을 공부함. <p>[2] ***(클라이언트)</p> <p>주간 목표 : Core 구현완료(70%)</p> <p>< Core 초기화 및 파이프라인 프레임워크 구현 ></p> <ul style="list-style-type: none"> - CommandList (Graphics / Compute) 구현하기 		

- App이 Core에게 전달할 CommandContext 구현
- PipelineStateObject(Graphics/Compute) 구현
- RootSignature 제작

[3] ***

주간 목표 : 유니티를 사용한 맵틀 구현(100%)

12. 30 제안서 발표

12. 31 맵틀 유니티로 구현 시작

구매한 에셋말고 일단 그냥 네모로 기본적인 기능이 돌아가게 하는 것이 목표이다. 현재 AddBox버튼을 누르면 커서에 박스가 생기고 다른 박스와 겹치면 빨강게, 겹치는 박스가 없으면 초록색으로 뜬다. 리지드 바디를 추가하고 트리거를 true로 해서 원래 설치된 오브젝트에는 변화가 없도록 하였다.

1. 1 맵틀 만들기 2일차.

스크롤 박스에 버튼 동적으로 등록되게 만들고 그거 누르면 커서에 오브젝트가 추가되도록 했다. 에셋 추가해도 자동으로 버튼 생성된다!

1. 2 맵틀 만들기 3일차. 70퍼센트 정도 진행된 것 같다.

스폰 지역도 추가 할 수 있게 만들었고. 스폰 지역 추가할 때도 스승 마법사와 학생 마법사 크기에 맞게 표시된다. 겹쳤는지 안 겹쳤는지 표시하는 부분도 버그 수정하고 오브젝트 크기에 맞춰서 반투명하게 나오도록 구현하였다

1. 3 팀 주간 회의 진행 (회의 내용은 트렐로에!)

모든 오브젝트 프리팹에 스크립트를 추가해서 fix, changeable, collideable, type 값을 가질 수 있게 했다.

type값도 오브젝트 크기별로

변신 가능

- s : 0~

- m : 100~

- l : 200~

변신 불가능

- collide(x) : 300~

- collide(o) : 400~

부터로 다 각각 넣어주었다

또한 type_def.cpp를 만들어서 모든 타입 값을 define해놓았다.
노가다의 날이었다.

1. 4 쉬었다..

1. 5 맵툴 파일 입출력까지 완료했다

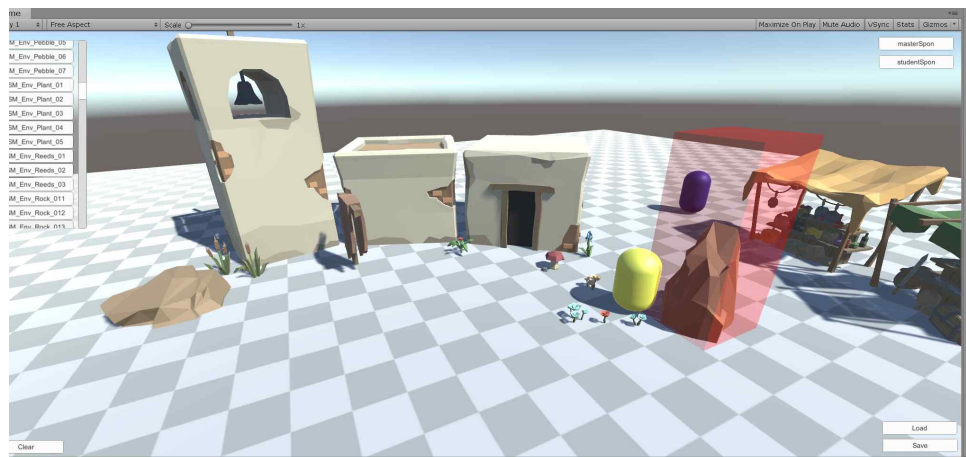
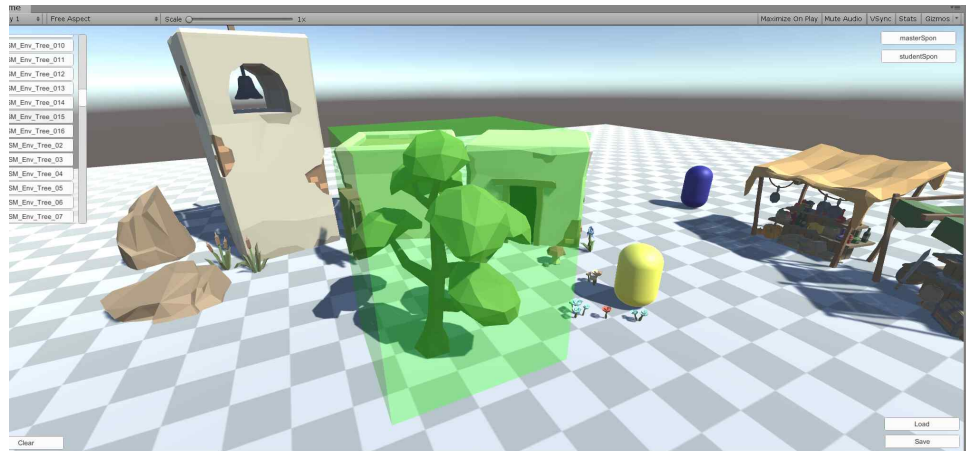
퀵터니언 때문에 삽질하느라 조금 시간이 걸렸다

맵에 배치된 모든 오브젝트를 삭제하는 버튼도 추가했다

저장하면 map.txt에

name pos rot bb.size change_able fix collide_able object_type
object_id 순서로 저장된다

파일을 읽어 로드하는 부분도 모두 구현하였다



나름 완성된 맵툴 모습

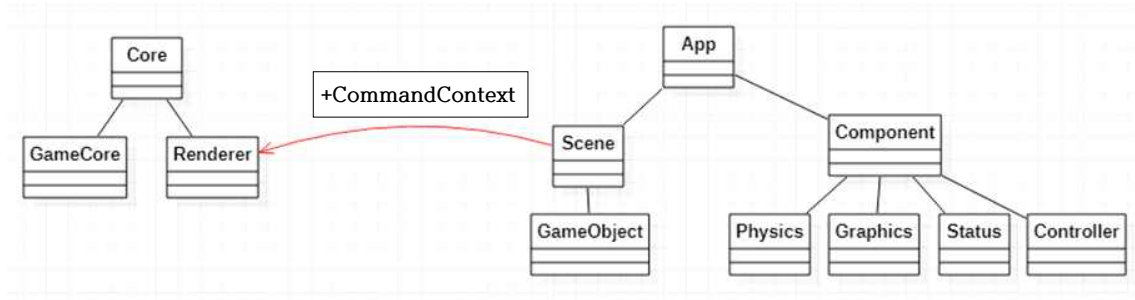
	<p>[4] *** (클라이언트) 주간 목표 : OBJ파일과 FBX파일을 읽고 화면에 띄우기 (50%)</p> <ul style="list-style-type: none"> - ***의 프레임워크 구조에 대해 설명을 받고 예전에 비슷한 프레임워크 프로젝트를 받아서 흐름을 파악하였다. 현재 엔진이 초기화부분까지만 완성한 상태 이므로 예전 엔진 버전에서 작업을 하고 있다. - OBJ 파일 속성들에 대해 공부하고 .obj 파일을 불러올 수 있는 OBJLoader를 작성하였다. - OBJ에서 재질 정보와 텍스처 정보를 가지고 있는 mtl 파일 또한 속성들에 대해 공부를 하고 .mtl 파일을 불러올 수 있는 MTLLoader를 만들었다. - FBX 파일을 로드하기 위한 FBX SDK 문서를 보고 FBX SDK란 무엇인지 파악을 하고 이것만으로는 이해하는데 한계가 있어 google을 이용하여 여러 자료들을 찾아보았다. - FBX SDK를 어떤 식으로 사용을 하는지에 대한 이해를 한 후에 예제 코드를 보면서 다시 한번 익숙해지려고 하였다. - FBX SDK를 이용하여 FBX Mesh를 불러오는 코드를 작성하였다.
<p>다음 주 할 일</p>	<p>[0] 공동 01.10 주간 회의 (2주차) 안전 - 맵 레벨 디자인(구획, 오브젝트 배치 장소)</p> <p>[1] *** (서버) 주간 목표</p> <ul style="list-style-type: none"> - 배틀서버, 로비서버 하이, 로우 레벨 디자인 완료. - AWS에 연구 목적 사용관련 문의하기. <p>[2] *** (클라이언트)</p> <ul style="list-style-type: none"> - 맵툴이 완성되면 FBX 구현된것과 연동하여 씬픽업보기 - App 파트 제작 (Scene, GameObject, Component) - 애니메이션 프레임워크 설계 <p>[3] *** (서버)</p> <ul style="list-style-type: none"> - 로비 서버, 매치 메이킹 서버 하이레벨 디자인 (주요)

	<ul style="list-style-type: none"> - 게임 서버 강의에서 배운 내용 복습 - 매치 메이킹 알고리즘 찾아보기 - 프레임워크 구현을 위한 c++공부(effective c++읽을 예정) - 그 다음 주부터 바로 서버 구현 들어갈 수 있도록 준비 <p>[4] *** (클라이언트)</p> <ul style="list-style-type: none"> - 애니메이션 특강(월)에 가서 이용희 교수님께 FBX SDK관련 질문하기 - FBX SDK를 이용하여 FBX Mesh 띄우기 [우선순위 1] - 재질과 uv정보도 불러오기 [우선순위 2] - FBXParsar를 좀 더 범용적으로 사용할 수 있는 형태로 제작 (현재는 하나의 fbx파일만 불러올 수 있는 형태) [우선순위 3]
문제점	<p>[1] ***</p> <ul style="list-style-type: none"> - 종강, 발표가 끝나서 조금 마음이 풀어졌음. 그 결과로 주간 목표를 달성하지 못하였음. 설 만큼 쉬었다고 생각되어 노트북의 모든 게임을 삭제하였음. - AWS를 사용하기로 했는데 가용한 자원문제도 그렇고, 사용법도 아직 모름. 프레임 워크 설계 이외 시간에 AWS 자습서를 통해 공부해야할 필요성을 느낌. <p>[2] *** (클라이언트)</p> <p>파이프라인 과정이 잘 진행되는지 확인하기 위해서는 App의 일부분(GameObject와 Mesh/Material Component)을 구현 후 연동하여 테스트가 필요함. Core 제작기간이 예정 계획일지보다 지연될 수 있음</p> <p>[3] *** (서버)</p> <ul style="list-style-type: none"> - 부족한 기본적인 c++ 실력으로 서버 프레임워크를 구현할 수 있을지 불안하다. - 클라우드 서버를 리눅스로 사용하게 되면 생기는 문제점 <p>[4] *** (클라이언트)</p> <ul style="list-style-type: none"> - FBX SDK를 너무 쉽게 본것인지 생각보다 어려운 친구였다! 생각보다 mesh를 띄우는 것이 오래 걸리고 있다. - 또한 ***의 프레임워크에 대한 완벽한 이해를 하지 못해서 띄우는데

어려움을 겪고 있다! 이 부분은 ***에게 계속 질문을 하면서 해결해야 할 거 같다.

[추가]

<프레임워크 설계 그림설명>



Core

- Renderer

CommandContext를 씬에게 받아 그려주는 역할

파이프라인 상태객체, 루트시그니처, 디스크립터 힙 등 렌더링에 필요한 객체들을 소유함

App

- Scene

어떤 오브젝트를 그릴지, 어디에 배치할지에 대한 정보를 서술하는 클래스

- Component

간단한 이동, 회전에 관한 간단한 물리연산(Physics), 그릴 때 필요한 정보들(Graphics)을 디커플링 하기 위한 구조