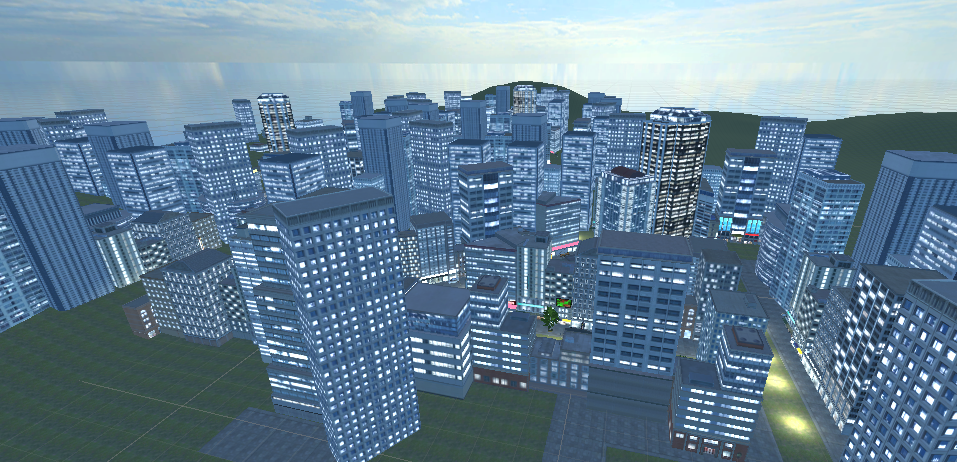
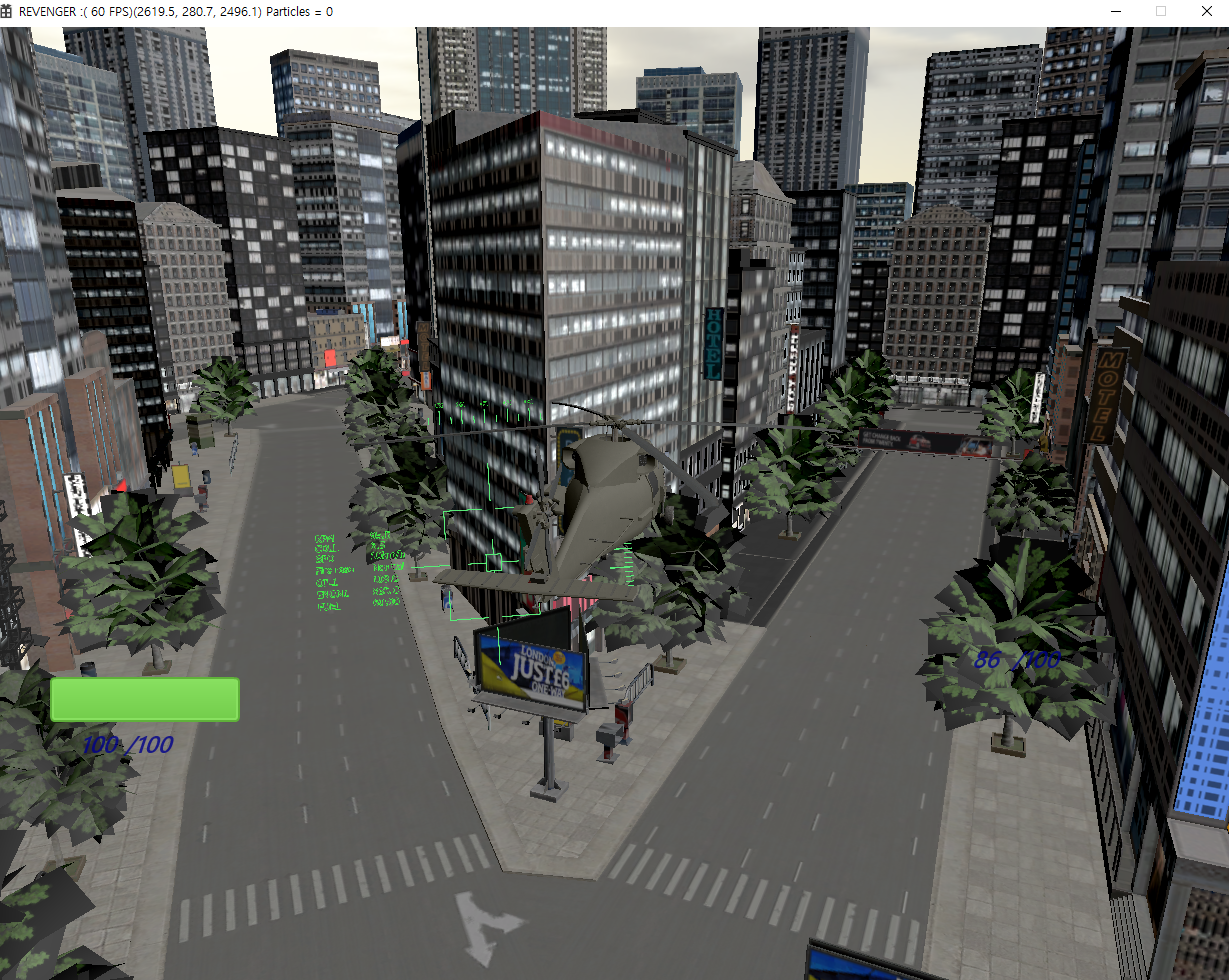
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | **작성자** | **2018180033 이세철 2018180046 허재성 2018182009 김승환** | **팀명** | XD |
| **계획 시트** | <https://docs.google.com/spreadsheets/d/1rUOOKpcVfkma18YsvTZXZz44z6UhbGfOsrimVgs-mzE/edit#gid=0> | | | | |
| **주차** | **4** | **기간** | **2023.02.27~2023.03.05** | **지도교수** | **정 내 훈**(서명) |
| **이번주 한일** | * **이세철:** * **허재성: Stage1 맵 로드, 그림자 기술요소 분석, 레이트레이싱 모델로드코드 작성** * **김승환:** | | | | |

**<상세 수행내용>**

1. **이세철**
2. **허재성**
3. 유니티 에셋 스토어에서 구매한 Stage1 맵 에셋을 로드 했음.



<유니티에서 임포트한 City View >



<Direct 12에서 로드한 City View>

조명, 환경매핑등 여러요소들을 배제하고, 컴포넌트를 설정하여 데모씬을 로드하였음.

일부 빌딩건물과 나무 등이 문제가 있지만, 나무는 다른 흔들거리는 애니메이션이 포함된 다른 모델을 배치할 것이고, 빌딩건물에는 일부가 Depth 가 너무 없어보이는 것이 있었기 때문에 그러한 빌딩건물들을 대체하고 내부에 조명을 넣을 생각임.

1. 레이트레이싱 모델 로드 코드작성

기존에 메쉬를 조립하는 개념을 잘 인지하고 있지 못한 부분에 대해 다시 코드를 수정하였음.



Vertex와 index를 vector로 선언하여서 정점정보와 인덱스 정보를 버퍼에 연결하였음.

빌드는 정상적으로 성공했으나, 아무런 오류도 뜨지않고 실행조차 되지않았음.

가령 실행이 되었을땐 메모리 이슈 문제가 있어서 컴퓨터적으로 성능이 받쳐주지 못한다고 판단중임.

후보로 두고 있는 중점연구과제인 파편에 따른 부분 손상을 위해 파티클이 퍼져나가는 로직을 구현중임.

1. 그림자 코드 분석

파티클 작업을 우선적으로 하려하였으나, 프레임워크를 안정화하기 위해서 그림자를 먼저 구현해놓고 시작해야겠다고 생각하여서 순서를 변경하였음.

이번주는DepthRender와 Shadowmap을 렌더하기 위해 기술적 요소를 분석하였음.

1. 쉐도우 맵은 빛이 있는 공간에서 투영되는 그림자를 텍스쳐로 표현함으로써, 렌더링 패스를 추가하여 생성 및 렌더링 한다는 것이다.
2. 쉐도우 맵을 렌더링할 때 깊이 버퍼가 사용되는데, 빛이 있는 공간을 렌더링할 때까지, 씬에 그려진 객체의 깊이를 깊이 버퍼에 기록하여 쉐도우 맵에 렌더링한다.
3. 쉐이더는 Direct 12에서 모델 5.0 이상을 지원하고, 쉐도우 맵 필터링을 사용하여 그림자를 부드럽게 만들 수 있는데, PCF 기술을 사용하기로 하였다.
4. 빛의 3종류인 점광원 , 방향성 광원, 원통형 광원등의 빛의 종류를 사용하여 그림자를 구현한다.
5. 렌더링 파이프라인은 쉐도우 맵과 깊이 버퍼를 생성하기 위해 헨더링 패스를 구성한다.
6. 깊이 버퍼를 사용하여 DepthRender을 하고 버퍼에 기록된 쉐도우 맵을 렌더링하는데, 필요한 객체들, 즉, 그림자가 적용되어야하는 객체들을 따로 모아두는 쉐이더 클래스를 생성하고 그곳에 모여진 객체들을 기반으로 렌더링하여 객체와 빛의 상호작용후에 그림자가 생성된다.
7. **김승환**

1. 서버 복구 관련 문제 일부 해결

기존의 코드에는 두 가지 문제점이 있었습니다.

첫번째는, Active서버를 먼저 켜지 않고 Stand-by서버를 먼저 켜게 되면 두 서버끼리 연결이 되지 않는다는 문제였습니다.

두번째는, Stand-by서버를 강제로 종료했을 때에는 Active서버가 Stand-by서버를 재실행하여 정상적으로 복구 작업이 이루어졌으나, Active서버를 종료하였을 때에는 Stand-by서버가 복구 역할을 수행하지 않았다는 문제였습니다.

첫번째 문제는, ConnectEx가 실패하였을 때 다시 ConnectEx를 불러주지 않았기 때문에 발생했던 문제로, 워커스레드에 있는 GetQueuedCompletionStatus의 리턴값이 0이 아닐 때 ConnectEx가 실패한 것이라면 ConnectEx를 다시 해주는 방법으로 해결하였습니다.

두번째 문제는, 서버가 실행될 때 서버번호가 작은 서버(Server[0])가 큰 서버(Server[1])에게 Connect 요청을 보내도록 설계하였기 때문에 번호가 작은 서버는 다운되었다가 재실행되어도 번호가 큰 서버에게 Connect 요청을 바로 보낼 수 있었지만, 수평 확장된 서버군의 마지막 구성원이 다운되었을 때에는 재실행되어도 자신은 Connect 요청을 보내지 않기 때문에 발생하던 문제였습니다. 그래서 저는 서버군의 마지막 구성원의 바로 이전 서버가 마지막 서버가 다운되었음을 감지하면, 마지막 서버를 재실행하고나서 Connect 요청을 보내도록 수정하여 문제를 해결하였습니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

지금은, 어떤 서버가 다운되어도 재실행되고 연결되는 것까지의 과정에는 문제가 없습니다.

2. NPC 동기화

텍스트, 실내이(가) 표시된 사진

자동 생성된 설명

(서로 다른 두 클라이언트에서 NPC가 동기화 되어있는 모습입니다.)

기존의 코드에도 NPC의 로직은 존재하였으나, 이상하게도 동작하지 않던 문제가 있어 그동안 NPC가 동작하는 모습을 클라이언트에서 볼 수가 없었습니다.

하지만 이번 주차에 NPC에서 생기는 문제를 해결하여 서버에서 계산된 NPC들의 좌표나 벡터 같은 정보들을 클라이언트로 전송하는 데에 성공하였습니다.

다만, NPC 동기화가 성공한 이후로 속도가 크게 저하되었고, 이는 아마도 NPC에 관련된 동작들이 서버에 부하를 크게 주고 있어서 생긴 문제로 의심되며 추후에 자세한 원인 파악 후 개선할 예정입니다.

|  |  |  |  |
| --- | --- | --- | --- |
| **문제점 정리** | 클라이언트 : 그림자를 코드로 구현, 중점연구과제 빠르게 결정 후 적용  서버: Active 서버가 다운되었을 때 Failover 작업이 필요함. | **해결 방안** |  |
| **다음 주차** | **5** | **다음 기간** | **2023.03.06~2023.03.13** |
| **다음주 할 일** | 이세철:  허재성: 그림자를 구현하고, 프레임워크를 안정화 시킨다.  안정화된 프레임워크를 기반으로 오프닝영상, 로딩창, 컷신을 구상하여 씬을 구상한다. 시간적 여유가 생긴다면 파티클 복구작업을 마저한다.  김승환: Active 서버가 다운되었을 때 Failover 작업 | | |
| **지도교수**  **Comment** |  | | |