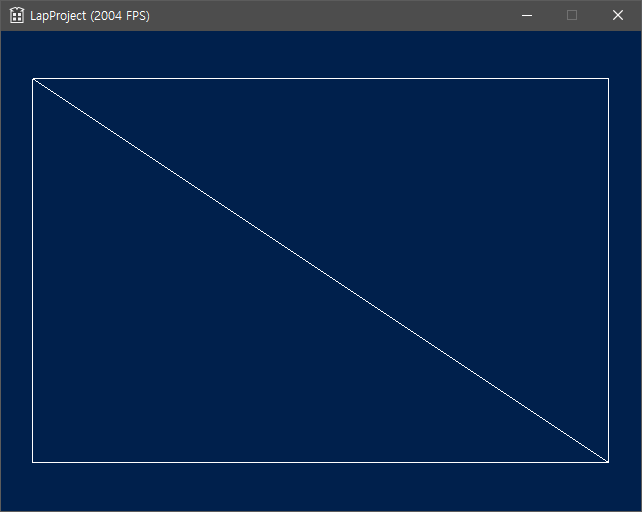
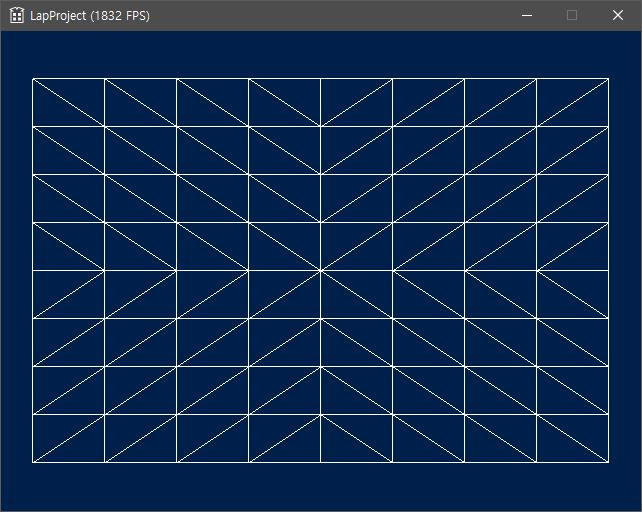
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | **작성자** | 2013180003  김나단 | **팀명** | 낙서의 전설  Legend Of Doodles |
| **주차** | 21주차 | **기간** | 2018.5.27 ~ 2018.6.2 | **지도교수** | 이 형 구 (서명) |
| **이번주 한일** | 테셀레이션 연구 및 구현  Terrain에 테셀레이션 적용 및 라이트 적용 | | | | |

**<상세 수행내용>**

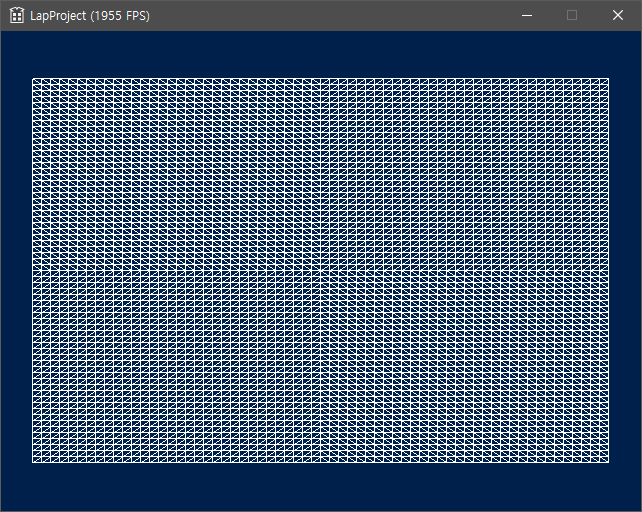
* 테셀레이션 구현 성공



기존 메쉬: 삼각형 두개로 이루어진 사각형 메쉬

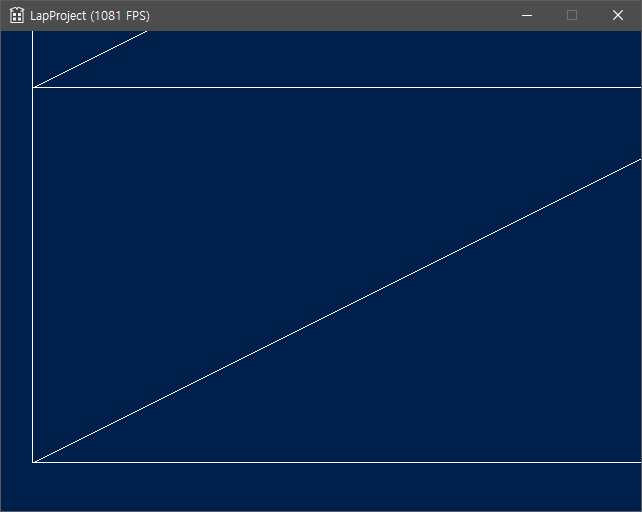
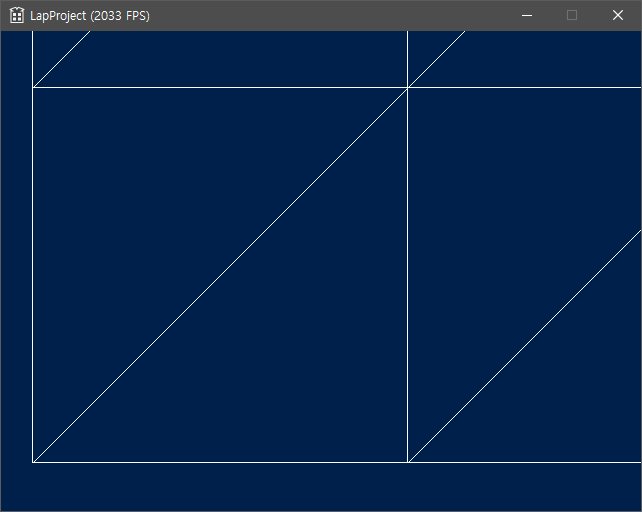


각 컨트롤 포인트 기준 8개씩 균일하게 테셀레이션 적용한 결과



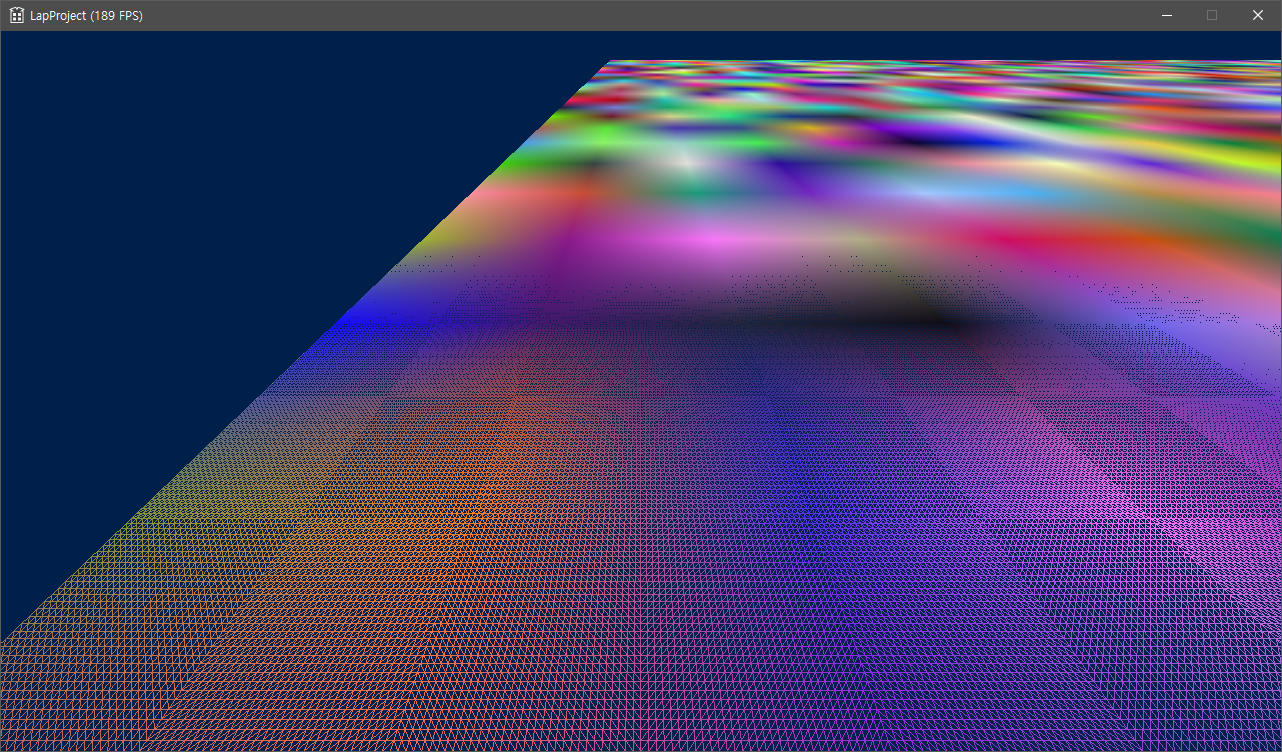
각 컨트롤 포인트 기준 64개씩 균일하게 테셀레이션 한 결과

* 테셀레이션을 터레인에 적용시 문제점
  + 최대64개씩으로 밖에 나눌 수 없기 때문에 사각형 하나로는 표현에 제한이 있음
  + 아 게임의 지형의 크기는 10000 X 5000이므로 그에 맞게 매쉬를 준비할 필요가 있을 것으로 보임

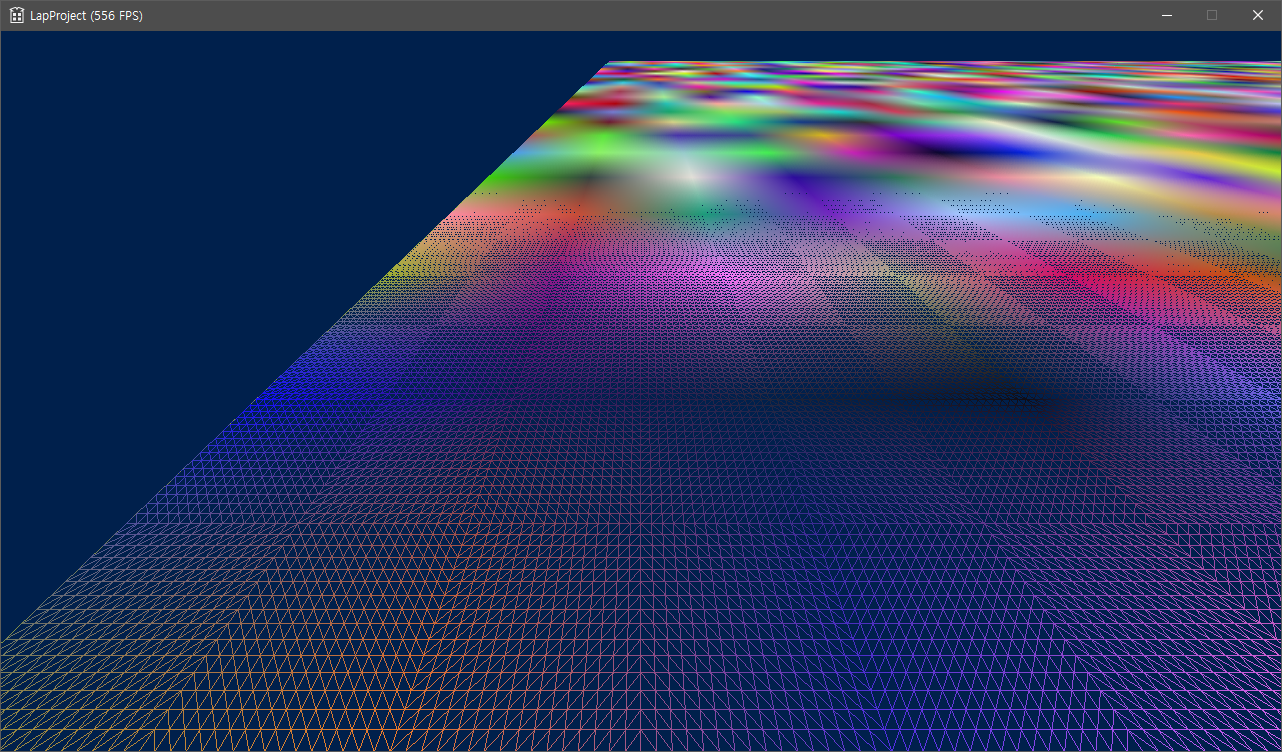
 

10000 X 5000 + 64개 테셀레이션 적용 가로로 두배 긴 메쉬에 동일하게 64개 테셀레이션 적용

* 테셀레이션 개선 작업 진행
  + 기존: 포지션의 변환을 테셀레이션 이후 도메인 쉐이더에서 진행 -> 변환되어 증가된 정점의 개수 만큼 행렬 곱 계산
  + 개선: 포지션 변환을 버텍스 쉐이더에서 진행 -> 기존에 가지고 있는 정점의 개수 만큼만 행렬 곱 진행
  + 현재의 예제 에서는 변환이후 64 \* 64 번 이상의 행렬 곱 -> 4번의 행렬 곱
* 기존 터레인 베이스 메쉬를 테셀레이션을 적용할 수 있도록 새로 메쉬 제작
  + 기존 방식: 버텍스 개수 125,000개, 인덱스 개수 248,752개
  + 변환 한 방식: 버텍스 개수 1,250개, 인덱스 개수 4796 개
  + 결론
    - 단순 정점 개수만 가지고 본다면 약 100배 정도의 메모리상 여유를 얻을 수 있음
    - 프레임 레이트에서 실험 상에서 봤을 때 64번씩의 테셀레이션(최대 수치) 적용 시 약 10배, 32번씩의 테셀레이션 적용 시 약 20배 정도의 향상을 확인(디버그 모드 기준)
    - 렌더링 결과로 볼 경우 32번씩의 테셀레이션은 기존보다 4배, 64번씩의 테셀레이션은 기존보다 8배 많은 정점을 표현하고 컨트롤 가능 -> 더 세밀한 표현이 가능



터레인 베이스에 64번씩 테셀레이션 적용한 결과 -> 약 230프레임



터레인 베이스에 32번씩 테셀레이션 결과 -> 약 550 프레임

* 인 게임 터레인에 적용 완료
  + 디버그 기준 약 3프레임 정도 빨라진 것 확인
* 굴곡을 잘 나타낼 수 있는 텍스처로 변경하는 것을 고려중



테셀레이션 인게임 적용 결과

* You tube: <https://youtu.be/o3zl5y-j5-M>

|  |  |  |  |
| --- | --- | --- | --- |
| **문제점 정리** |  | **해결 방안** |  |
| **다음 주차** | 22주차 | **다음 기간** | 2018.6.3 ~ 2018.6.9 |
| **다음주 할 일** | 그림자 연구 및 구현 | | |
| **지도교수**  **Comment** |  | | |