2020182031 이서연 졸업작품 16주차 진행

1. Terrain Map 과 Deferred Rendering과의 병합

저번 주의 상황에서 Normal값 계산은 올바르게 되고 있다는 것을 디버깅으로 확인하였다.

Normal값이 검정색, 즉 0,0,0으로 계속 들어가는 것은 Input에 전달하면서 생기는 일임을 알아냈고, 이 현상을 고쳐야 Normal이 잘 계산되는 지 확인할 수 있으므로 일단 Normal에 빨간색 (1,0,0)을 넣어두고 검정색이 아니라 빨간색이 잘 출력되는지 확인해보기로 하였다.

1. Mouse Click 위치에 따른 UI 선택

UI가 가장 많이 쓰일 부분은 당연히 버튼 부분일 것이다.

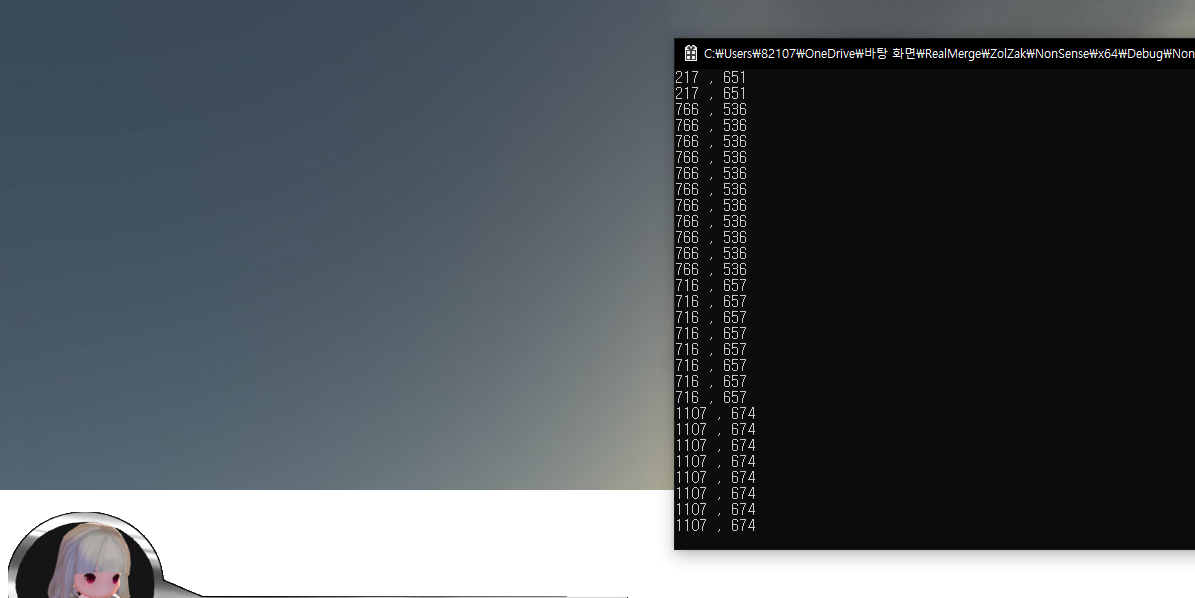
사각형이(가) 표시된 사진

자동 생성된 설명로그인, 로비 등에서 버튼을 UI로 제작할 것인데 이 때 플레이어가 마우스로 클릭한 부분이 어떤 UI를 가리키는지 알아내야 하고, 이것은 AABB방식으로 검사한다고 했을 때 다음과 같은 위치로 알 수 있다.

마우스의 클릭 지점과 다음 값을 AABB식으로 넘겨주어 판단하면 되지만 일단 마우스로 클릭을 하면 어떤 값이 나오는지 알아야 한다.

왜냐하면 저 X, Y, W, H는 0~1의 값으로 들어가기 때문에 범위 값을 맞춰줘야 하기 때문이다.

그래서 마우스가 클릭한 지점을 일단 출력을 해보기로 하였다.



그 결과 다음과 같이 0~1의 범위가 아닌 0 ~ Screen Width, 0 ~ Screen Height의 범위로 출력되는 것을 알았고 이를 0 ~ 1의 범위로 바꾸어 줄 필요가 있다.

텍스트, 전자제품, 디스플레이, 스크린샷이(가) 표시된 사진

자동 생성된 설명

그래서 마우스 클릭 좌표의 x에는 Screen Width를 나누어 주고, y에는 Screen Height를 나누어 주어 내가 사용하고 싶은 범위와 맞춰주었다.

이제 클릭한 위치에 어떤 버튼이 있는 지 알아내는 함수가 필요하고 다음주에 만들어보려고 한다.

1. Depth Shadow Mapping (DSM) 공부

저번주에 Cascade Shadow Mapping을 구현하겠다고 당당히 외쳤지만 아무래도 일에는 순서가 있는 법이라고 생각한다.

기본적인 DSM을 먼저 구현하고 문제점을 확인 후 해결 방안들을 생각 하고 어떤 Shadow Mapping을 사용할 지 정하는 것이 올바른 순서라고 생각하여 DSM을 먼저 구현해보기로 결정했다.

1. 카메라를 빛의 위치에 둔다
2. 그 상태에서 Object들의 Depth값을 저장하는 Depth Map을 생성한다.
3. 플레이어시점의 Depth Map과 새로 생성된 Depth Map을 비교하여 그림자가 드리울 곳을 판단
4. 그림자가 드리워야 할 곳에 그림자를 그려준다.

이것이 가장 간단하게 설명한 Shadow Mapping의 기본기이다.

Shadow를 공부한 결과, 이론들은 어느정도 깨우쳤으나, 이것들을 코드로 옮기는 작업이 쉽지 않다고 느꼈고, 시도를 하기가 두려워져서 이용희 교수님에게 Shadow Mapping 예제가 있는지 여쭈어 보는 메일을 보냈다.

**나:**

안녕하세요 교수님! 현재 한국공학대학교 4학년 재학중인 2020182031 이서연입니다!

DirectX12 졸업작품 진행중에 있는데 한가지 여쭈어 볼 게 있어서 연락 드립니다!

다름이 아니라 현재 Deferred Rendering, UI, Lighting을 어느정도 손봐 놓은 상태라서 이제 그림자를 구현해보려고 하는데 공부를 한 결과 이론은 잘 알겠지만 코드로 구현하기가 참 어렵다는 생각이 들었습니다!

원형 그림자가 아니라 DSM이나 CSM같이 캐릭터의 형태가 나오는 그림자를 구현하고 싶은데 강의자료는 있지만 실습을 하지 않았기 때문에 감이 잘 오지 않습니다

혹시 Lab Project중에 Shadow관련 예제가 있을까요?

**교수님:**

이론을 이해했으면 코드로 구현하는 걸 시도해보세요.

어떤 부분이 참 어렵다고 생각하나요?

**나:**

(위에서 설명한 DSM의 기본적인 순서를 설명해드리고)

큰 틀로 이런 방식을 사용한다고 이해하였는데 1번에서 카메라를 빛에 위치에 두지 않고 빛의 position과 direction을 가지고 계산을 할 수 있을 것 같은데 플레이어의 시점이 아닌 빛의 시점에서 Object의 깊이 값을 저장할 때 계산해야 하는 행렬식들을 어떻게 만들어야 할 지부터 막힙니다

어떤 글에서는 Light의 View와 Projection행렬이 필요하다고도 하는데 현재 저희 Light는 여러 빛 관련 수치들과 위치, 방향만을 가지고 있기 때문에 이런 행렬들을 어떻게 유도해야 하는지도 모르겠어 서 정말 이런 정보가 위의 과정에서 필요한 건지 교수님의 예제를 보면 참고가 될 거라 생각해서 메일 드렸었습니다!

**교수님:**

"1번에서 카메라를 빛에 위치에 두지 않고 빛의 position과 direction을 가지고 계산을 할 수 있을 것 같은데..."

무엇을 계산할 수 있다는 건가요? 그리고 어떻게 계산할 수 있다는 건가요?

" 빛의 시점에서 Object의 깊이 값을 저장할 때 계산해야 하는 행렬식들을 어떻게 만들어야 할지..."

깊이 값은 렌더링의 결과로 계산되는 것입니다. 그러니까 행렬식(?)이나 행렬이 별도로 필요한 게 아닙니다.

카메라를 조명의 위치와 방향에 두고 Scene을 렌더링을 하면 됩니다.

**나:**

교수님의 말씀을 듣고 곰곰이 생각해보니 제가 본 Light View, Projection행렬이라는 것들이 결국 카메라를 Light의 위치로 이동시키고 나서의 View, Projection행렬을 뜻하는 것을 알게 되었습니다.

 " 카메라를 빛에 위치에 두지 않고 빛의 position과 direction을 가지고 계산을 할 수 있을 것 같은데..."라고 했던 것은 결국 카메라도 Position과 Direction을 가지고 계산을 할 텐데 이미 Light의 Position과 Direction을 알고 있는 상태에서 어차피 Light의 View행렬과 Projection행렬을 따로 구해야 한다면 굳이 카메라를 옮겨서 계산을 해야 할까? 생각을 했던 것입니다.

(제가 본 블로그들의 Light View, Projection행렬이 Camera View, Projection행렬과 다른 것이라고 생각했으나 알고 보니 같았다는 것이죠)

이론을 보고 그대로 코딩하면 되었던 것인데 괜히 다른 블로그 예제들을 보고 더 좋은 방법을 참고하려 다가 더 헷갈려져 버렸었네요!

교수님이 답변해주신 말에 확신이 생겨 이제 영감이 떠오릅니다! 감사합니다

이렇게 메일을 주고받았는데 결국 Lab Project는 받지 못했지만 스스로 생각한 이론을 구현해보라는 교수님의 뜻을 받았고 내가 여러 블로그들을 보며 헷갈려 했던 부분에 대한 대답을 받았다.

일단 Render Target이 되어줄 Texture을 또 하나 생성해서 directional Light의 위치에 Camera를 두고 깊이 값을 그려주는 작업을 먼저 해보아야 할 것 같다.

1. 그 외

4/5 (수)에 게임 엔진 실습시험을 열심히 준비해서 높은 성과를 얻었다!!

4/7 (금) 학교 축제 겸 친구들과 밤 늦게까지 놀 예정이다.

시험 준비와 친구들과 놀게 되어서 이번주는 성과를 내기보단 공부를 하는 방향에 힘을 실었다.