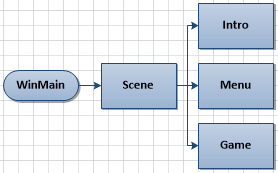
**포트폴리오 제작 기술서**

**1. 게임의 틀.**

 Winmain을 최대한 건들지 않기 위하여 Scene 클래스의 상속 기능을 이용한 상태패턴을 사용. Intro, Menu, Game 3개의 씬이 변동될 수 있도록 설정하였습니다.

**2. 맵의 제작.**

- 높이맵 제작 방법을 이용하여 버텍스를 1024 x 1024 사이즈로 중점에 0.0 이 올 수 있도록   
for ( z = (전체z축길이/2) ; z >= -(전체z축길이/2) ; z-- )  
 for ( x = -(전체x축길이/2) ; x　＜= (전체x축길이/2) ; x++ )

두 개의 for문을 이용하여 버텍스의 위치를 설정하고

흑백인 bmp 파일로부터 x, z위치의 색상 값을 얻어와 y축의 설정을 하였습니다.

또한 16개의 셀로 나누어서 LOD를 적용하였습니다.

캐릭터의 위치에 보가된 높이 값을 얻기 위해 보간법을 사용하였습니다.

캐릭터의 좌표 pos의 (pox.x + 전체x축길이/2) / (한 셀의 넓이-1) z도 같은 방식으로 x 와 z값을 얻은 다음 전체 셀을 나눈 개수를 곱해주어 몇 번째 셀에 있는지 찾아내었으며 다시 이 방법을 사용해 해당 셀의 어느 위치의 사각형에 있는지 추적하였고, 해당 사각형의 인덱스 연결 순서에 따른 상삼각형 혹은 하삼각형에 있는지 체크한 후 해당삼각형과 대응되는 가상의 사각형을 벡터 차나 합 연산을 통해 사각형 첫 점부터 pos.x 와 z 까지의 벡터를 구하고 이를 외적하여 법선을 구한 값을 보간 된 높이로 사용하였습니다.

**3. 캐릭터의 동작.**

- 상향벡터와 우향벡터를 외적하여 전방벡터를 만든 후 이 전방벡터에 속도를 곱해 더해주는

방식으로 캐릭터의 전후 움직임을 만들었으며 y축은 항상 고정인 게임이기에 D3DXMatrix

RotationY 함수를 이용하여 회전 이동을 넣었습니다. 플레이어의 상태를 각각 정하여 동작하는 프레임과 애니메이션 플레이타임을 다르게 설정하였으며, 동작 버퍼를 넣어 현재 연속공격이 이루어졌는지 혹은 단타로 끝났는지 유무를 체크 후 콤보데미지 증가 유무를 결정하였습니다. 지형을 좀 더 이동방해요소와 도움 요소로 사용하기 위하여 외적을 이용한 정규 직교화를 사용하였습니다



정규 직교화를 이용하여 그림에서 보이는 노란색 벡터를 구하여 캐릭터 이동에 곱해줌으로써 각이 완만한 곳은 정상이동 급한 곳은 느릿하게 이동되며 내리막길은 빠르게 이동 혹은 미끄러지는 것을 구연하였습니다.

**4. 외부 라이브러리 참조.**

- 3Ds Max로부터 캐릭터의 재질, 뼈대정보, 정점정보 등을 가져와 뼈대를 컨트롤하며 뭉개지지 않도록 트위스트 본 등을 활용한 방법은 저에겐 아직 무리라는 판단으로 포트폴리오를 만들기 위해 외부 라이브러리를 사용하여 캐릭터의 애니메이션 재생을 하였습니다.

**5. 쉐이더**

 - **버프**

흡혈, 스태미너 회복, 공격속도 증가, 공격력 증가 의 4가지 버프를 만들었으며 그림과 같이 해당 버프가 눈에 보이도록 카메라 위치벡터와 캐릭터의 각 정점에 대하여 내적이 0에 가까울수록 진하게 표현되도록 하는 쉐이더 코드를 작성하였습니다.

 **- 그림자**

그림자의 제작과정은 랜더타겟 변경하여 빛의 위치에서의 뷰행렬 그리고 D3DXMatrixOrthoLH를 이용한 직교투형 핼렬을 곱하면 -1~1까지의 값이 나오는데 이를 uv좌표처럼 0~1로 맞춰주기 위해 1을 더한 후 나누기 2를 해주어 화면상의 좌표와 맞추었습니다. 이렇게 만들어진 행렬을 쉐이더코드로 넣고, 쉐이더 내에서 해당하는 정점 부분을 검은색으로 덧칠하여 그림자를 표현했습니다. 캐릭터 쉐이더와 맵쉐이더 두 개의 파일에 모두 적용하여 적 그림자에 혹은 자신의 그림자에 자신의 몸이 그늘지는 셀프 그림자가 되었습니다.

- **안개**

사실 초기 맵은 평탄한 지형이었습니다. 이 지형에 안개를 구현하니 너무 티가 나질 않아 높이를 주게 되었습니다. 높이안개와 범위안개 2개를 적용시켰으며 높이안개는 쉐이더 코드로 출력될 높이 값에서 안개를 쌓이게 할 적정 높이 값을 뺀 후 saturate 함수를 이용해 0~1사이의 수로 만든 것을 1에서 뺌으로써 적정높이 보다 높은 곳은 0에 수렴하여 원래의 색을 낮은 곳은 회색에 가까워지도록 하였습니다.



**<- 적용 전**

**적용 후->**

범위안개는 카메라 위치를 기준으로 벡터크기 계산을 통해 위 공식과 마찬가지로 일정 거리 이하는 본래의 색을 띄고 멀어질수록 짙은 검은색으로 덧칠하였습니다.



**<- 적용 전**

**적용 후->**

- **타격지점 표시 및 타격 시 시각처리**

 몬스터가 타격 모션을 하면 타격판정을 할 위치의 끝 표면에 붉은색 원을 표시하였습니다. 초기엔 타격모션을 보고 피해야 했던 것을 유저친화인터페이스를 고민하며 범위를 보여주자는 생각으로 표시하였습니다.

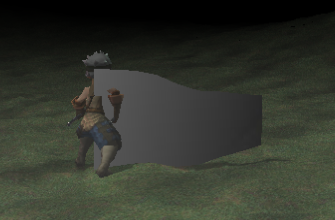
 타격 시에 좀 더 타격 감을 줄 방법이 없을까 하는

생각에 붉게 적을 물들이는 방법을 택했습니다.

몬스터가 맞는 판정이 되면 쉐이더에서 붉은색

으로 모든 색상을 바꾸는 방법을 사용 하였습니다.

불

 - **댓쉬**

본래 제작의도는 댓쉬 시에 그냥 갑자기 순간이동 되는 것이 부자연스러워 중간중간 모션의 잔상을 넣고 싶었으나 해결방법을 찾지 못 하여 이러한 연출을 생각하게 되었습니다. 캐릭터의 얼굴높이를 최대높이로 캐릭터위치 y축을 최하높이로 잡고 본래 위치점 ~ 댓쉬 후 이동점 까지 버텍스를 찍어 연결 후 버텍스가 끝으로 갈수록 색이 바뀌게 색 벡터 값에 변경을 넣어 그라데이션 효과를 냈습니다.

의도하지 않게 댓쉬 시 나온 효과에도 그림자가 생기는데 금방 사라지기 때문에 버그수정은 하지 않았습니다.

**6. 카메라**

- 카메라의 회전 및 이동은 캐릭터를 따라가도록 설정하여 항상 뒤를 바라보며 캐릭터의 회전 시 0.1%씩 따라가도록 하여 가속카메라 효과를 넣었고, 마우스 상하 값에 따라 카메라를 약간만 우향벡터를 기준으로 회전시킬 수 있도록 했습니다.

**7. 스카이박스**

 스카이박스는 3Ds맥스로 사각형을 만들어 해당 매쉬에 텍스쳐를 입힌 것을 스케일을 조정하여 띄웠습니다. 또한 사각형인 스카이박스의 각이 티 나지 않도록 원모양의 맵을 만들었습니다. 그 결과 착시현상으로 둥근 것처럼 보이는 효과를 주었습니다.

**8. 몬스터의 동작**

- 몬스터는 캐릭터를 따라가야 함으로 캐릭터의 좌표를 몬스터 클래스에서 받아오도록 하여 몬스터와 캐릭터의 좌표를 이용한 거리벡터 와 몬스터의 전방벡터를 외적 하면 법선을 구할 수 있는데 외적의 성질 중 “외적이 0 이면 두 벡터는 수평 하다“ 란 성질을 이용해 외적 한 벡터의 높이 값이 0에 가까울 때까지 감소 시켰습니다. 추가적으로 내적을 이용해 내적 값이 전방벡터에서 0.99 이하로 들어오는 조건을 먼저 두었습니다. 이러한 결과로 몬스터는 항상 캐릭터를 바라보고 그 방향으로 이동 하도록 하였습니다.

**(UI 와 캐릭터 스킬 조작, 파티클 부분은 다른 팀원이 작업하였습니다.)**