
4주차 베타

분류 3인칭 아케이드

제목 미정

작성자 김호곤

○ 주요 이슈

1. 일부 요소는 카메라가 근처에 있을 때만 보인다.

적은 멀리 있을 때 메쉬를 숨겼다가 카메라가 가까이 있으면 나타난다. 월드 안에 다수의 적이 생성될 것이고, 현재 모델은 단순한 캡슐로 표현하고 있으나, 나중에 정점이 많은 모델로 대체하면 이러한 최적화가 필요할 것이다. 또한 순찰용 적의 경우 카메라가 안 보고 있을 때 굳이 순찰 돌고 있을 필요 없으므로 카메라 근처에서만 순찰하는 모습을 보이게 했다.

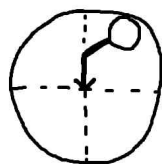
2. 미니맵에서 아이콘 표시, 화면에 표시되는 미니맵 크기 조정

미니맵은 지도 그림을 그린 것이 아니라 원근법 없는 카메라를 따로 만들어 위에서 아래로 수직으로 내려다 본 것을 RenderTexture로 만들어낸 것이다. RenderTexture는 안드로이드에서도 쓸 수 있게 하기 위해 DepthStencil 포맷은 D16_UNORM으로 했고, 스프라이트로 못 쓰기 때문에 씬에서는 RawImage 컴포넌트로 가져와야 되는데, RawImage는 그냥 Image와는 다르게 원본의 가로, 세로 비율을 유지하는 기능이 없어서 미니맵 크기 조정하는 스크립트를 만들었다. 세로 길이를 기준으로 가로 길이를 계산하고, 윈도우 플랫폼에서 컴파일하는 경우에 한해 화면 크기가 바뀌면 미니맵 크기를 다시 계산하게 했다.

3. UI 조이스틱을 만들어서 키보드의 방향키 기능을 대체했다.

안드로이드로 빌드할 경우 키보드가 없기 때문에 화면에 방향키를 대신할 것을 만들어줘야 된다. 조이스틱을 만들어서 터치했을 때 처음 터치한 위치에서부터 얼마만큼 이동했는지 알아내고 그것을 Vector2의 정규화된 값으로 나타낸다. 조이스틱에서 손을 놓으면 자동으로 조이스틱이 되돌아가도록 하는데, 원위치로 순간이동 하는 것보다 서서히 이동하는 것이 더 자연스러울 것이다. 되돌아가는 것도 정확히 원위치 방향으로 돌아가도록 했다.

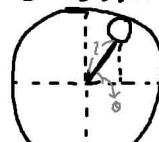
$$\Delta x = \Delta y$$



(X)

$$\Delta x = l \cos \theta$$

$$\Delta y = l \sin \theta$$



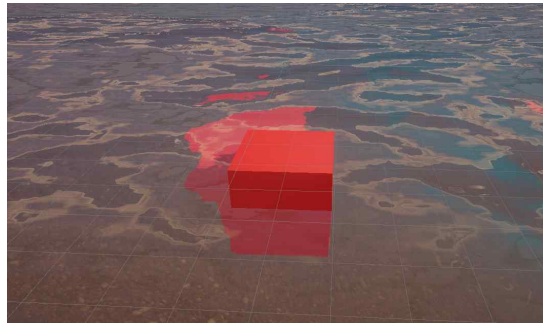
(O)

4. 적에 Stencil Mask 적용했다.

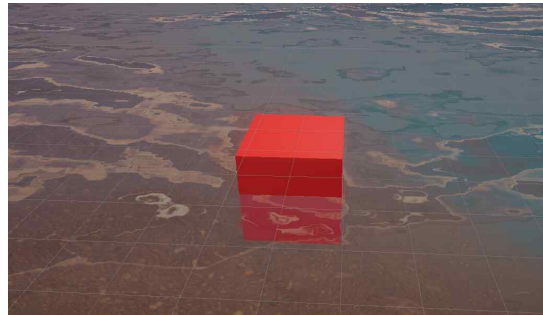
벽 뒤에 있어서 안 보이는 적 때문에 Actor가 적한테 맞는 불편함을 발견해서 Stencil Mask를 적용하는 것으로 해결했다. RenderFeature에 Stencil 값 대입, 대상에 메타리얼 대입을 추가했다. Stencil에 값을 대입할 대상은 지형을 제외한 건물이고 메타리얼 대입 대상은 적이다. 이때 벽 뒤를 본다는 설정이므로 홀로그램 같은 느낌을 주기 위해 줄무늬 텍스처를 사용해서 UV 애니메이션을 만들었다.

5. 이중으로 depth 차를 계산해서 물 셰이더를 만들었다.

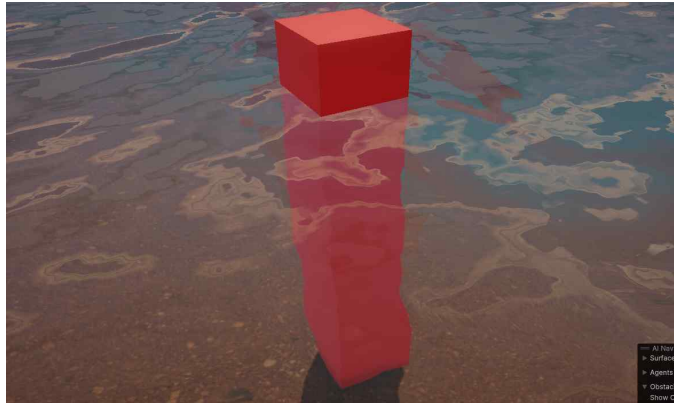
측면에서 볼 때 하늘이 반사되는 것, 수직으로 볼수록 투명하게 보이는 것을 구현하고, 투명하게 보일 때 굴절되는 것을 구현했다. 굴절은 화면을 캡처해서 외곡 시키는 것으로 흉내냈다. 따라서 화면에 보이는 전체가 굴절되는 것처럼 보인다.



Depth 차이를 이용해서 물체가 수면 밖에 있는지 확인하면 될 것이나, 여기서는 얕은 수심에서 굴절이 적게 보이고 깊은 수심에서 굴절이 더 보이도록 할 것이다. 따라서 depth 차이만큼 굴절 정도가 변하는 것인데, depth 차이는 saturate로 0에서 1사이로 잘라냈으므로 수면 밖 굴절은 자연스럽게 상쇄된다.



그러나 이 방식에서 부자연스러운 부분은 굴절이 이중으로 보이는 것이다. 물체 부분의 depth 차이는 적고, 강바닥 부분의 depth 차이는 크다. 물체의 모양을 따라서는 굴절이 적게 보이나, 강바닥 부분은 굴절이 크게 보여서 왜곡되는 정도가 큰데, 왜곡이 크다 보니 물체의 색깔까지 끌어당겨서 렌더링 되기 때문에 굴절이 이중으로 보이는 현상이 발생한다. 이를 해결하기 위해 depth 차이 계산을 이중으로 한다. 1차로 depth 차이를 계산해서 굴절에 의한 왜곡을 계산했을 때, 그 왜곡을 바로 렌더링하지 않고 그 왜곡된 상태에서 다시 depth 차이를 계산한다. 그러면 1차 굴절에 의해 물체 색깔이 보이는 부분은 다시 적은 굴절로 계산되고 강바닥 색깔로 보이는 부분은 다시 큰 굴절로 계산된다. 이렇게 해서 얕은 수심에서 굴절이 적고 깊은 수심에서 굴절이 많이 보이게 된다. 즉, 1차 depth 계산은 보이는 굴절을 위한 것이고 2차 depth 계산은 이중굴절을 상쇄하기 위함이다.



이 방식은 당연히 현실 굴절과 다르나, 어차피 이 굴절 효과 자체가 흉내 내기에 불과하므로 허용하기로 했다.

6. Actor의 수영 애니메이션을 추가했다.

AnimationController에서 해당 Actor가 물에 있는지 bool 값으로 확인한다. 참이면 수영용 BlendTree가 동작하고 거짓이면 걷기용 BlendTree가 동작한다. C_Actor 스크립트에서 물을 감지하는데, 물에서 육지로 이동하는 것을 감지하기 위해 물 감지뿐만 아니라 육지 감지도 한다. 일단 물에 닿으면 수영 상태가 되고, 수영 상태에서는 육지 감지를 시작해서 육지에 닿으면 걷기 상태가 된다.

7. 에디터 확장기능으로 스크린샷 찍는다.

Actor의 프로필 사진을 찍기 위해 스크린샷 기능을 만들었는데, 프로필 사진 만들기는 주로 디자이너가 할 것으로 판단해서 상단 메뉴 버튼으로 스크린샷이 저장되게 했다.

○ 문제점

1. 물, 육지 경계를 명확하게 구분할 수 없다.

물, 육지 경계에 바다 거품 텍스처를 입히려 했으나, 경계를 확인하는 방법으로 depth 차이를 썼을 때, 실제 수심에 의한 경계가 아니라 보는 방향을 기준으로 하는 depth 차이를 계산하기 때문에 카메라 각도에 따라서 바다 거품의 정도가 달라진다. 실제 수심을 가져올 만한 방법이 필요하다.

2. 안드로이드 빌드 시 Terrain이 안 보인다.

현재 프로젝트를 URP로 만들었는데, 윈도우로 빌드 시 정상적이거나, 안드로이드 빌드 시 Terrain이 달라진다. 첫 번째로 Terrain 재질에서 갑자기 Smoothness가 높아지는데, 인터넷 검색을 통해서 알아낸 것에 따르면 텍스처의 알파값이 Smoothness 값으로 적용되는 유니티 오류인 것 같다. 텍스처 애셋 설정에서 AlphaSource를 None으로 뒀는데, 이는 소용 없었고, 텍스처 압축 포맷을 ETC2로 하니까 해결됐다. 그러나 두 번째는 훨씬 심각한 문제인데, 안드로이드로 빌드해서 전화기로 실행했을 때, 가까운 terrain은 아예 안 보인다. 아직까지 해결이 안 되고 있다.