|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **주차** | 15 주차 | **기간** | 2020.04.06 ~ 2020.04.12 | **지도교수 이용희** | (서명) |
| 이번주 한일 요약 | -오브젝트 움직임을 따라가는 락 온 UI 제작  -미니맵 제작  -유도미사일 성능 개선  -텍스쳐 애니메이션 구현  -바다 텍스쳐 개선  -충돌 매니저 제작 | | | | |

<상세 수행내용>

**김령운 :**

**조준 UI의 문제점들을 개선하고 플레이어와 적의 위치를 알려주는 미니맵을 제작**

1. **조준 UI**

직교 투영한 조준 UI가 적 물체를 제대로 따라가지 못하고 보는 각도에 따라서 달라지는 문제를 해결하기 위해서 조준 UI에 빌보드 처리를 하여 카메라를 바라보게끔 해봤지만 여전히 적 물체를 따라가는 것이 부자연스러운 문제점 발생.

**[여전히 발생되는 문제점]**

해결법으로 적 오브젝트의 월드 좌표를 가지고 화면 좌표를 얻어내서 게임 세상 어딘가 미리 만들어 놓은 조준 UI가 적 오브젝트의 화면 좌표에 그려져서 따라다니는 것처럼 보이게 만들었다.

**[화면 좌표를 얻어내는 계산식]**



화면 상에서 조준 UI의 위치는 x, y 축 동일하게 -1 ~ 1로 표현되고 있다. 적 오브젝트를 조준 UI가 따라다니게 하기 위해서는 적 오브젝트의 화면 좌표를 -1 ~ 1 사이의 숫자로 표현하는 연산이 필요했다.

**[적 오브젝트의 화면 좌표를 -1 과 1 사이의 숫자로 변환하는 식]**



화면 크기의 절반 좌표인 (960, 540)이 넘어올 경우 조준 UI를 (0, 0)에 그리고 (1920, 1080)이 넘어올 경우 (1, 1)에 그려야 했다. 이를 위해서 (오브젝트 화면 좌표 \* 1/화면 절반 크기) - 1 연산을 통해 얻어진 값을 조준 UI가 그려질 좌표로 설정했다.



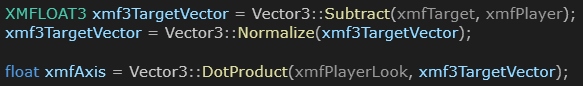
조준 UI가 적 오브젝트의 화면 좌표를 받아와서 잘 따라다니는 것을 확인

기존 UI들과 동일하게 조준 UI도 변환 과정에서 카메라 행렬과 투영 행렬을 생략하여 플레이어가 어떻게 움직이든 항상 화면에 동일하게 나오도록 하였다.

**[문제점 발생]**

플레이어가 적 오브젝트를 등지고 있을 때도 화면 상에 조준 UI가 그려지는 문제 발생

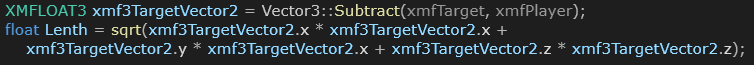


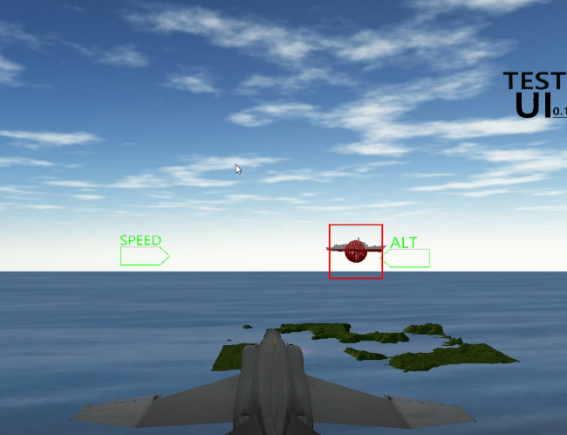
**[해결방안]**

적 오브젝트의 위치 벡터에서 플레이어의 벡터를 빼서 타겟 벡터를 만들고 플레이어의 룩 벡터와 타겟 벡터를 내적해서 플레이어가 적 오브젝트를 바라보고 있을 경우에만 조준 UI 그려지도록 했다.

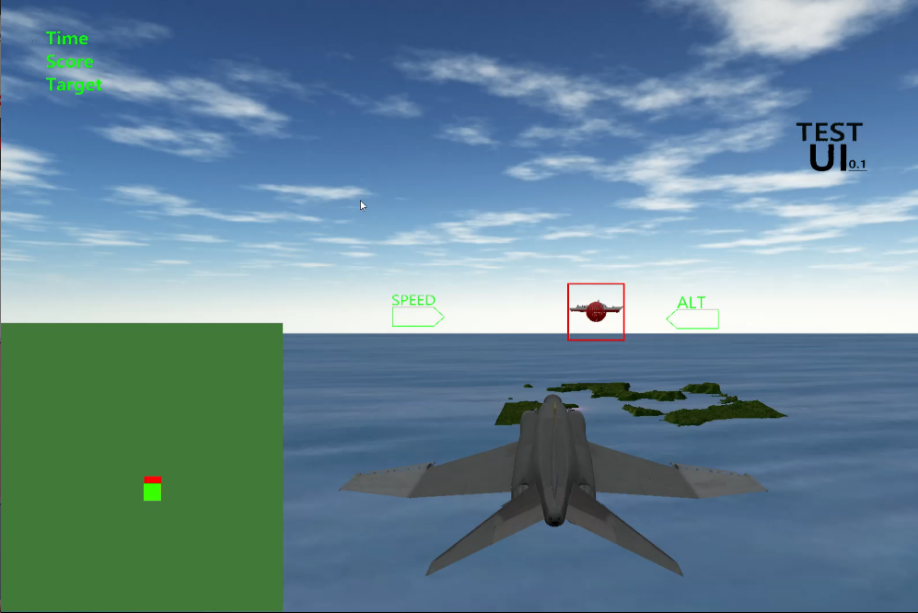
조준 가능 상태를 알려주기 위해서 플레이어가 조준 가능한 거리에서는 조준 UI의 색깔이 빨간 색으로 바뀌도록 했다.

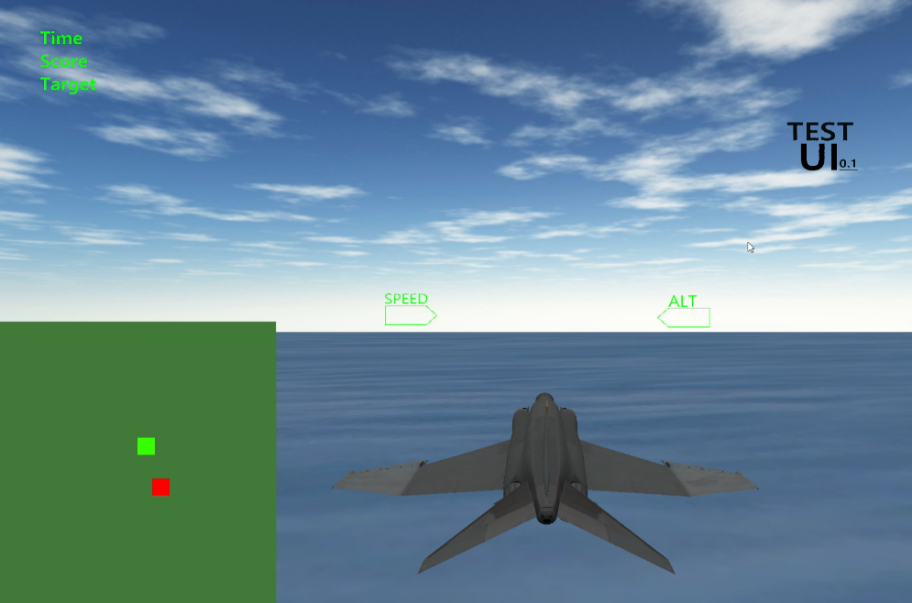
**[플레이어와 적 오브젝트 거리 구하는 식]**





1. **미니맵**





조준 UI를 적 오브젝트 화면 좌표에 그리기 위해서 사용했던 공식과 비슷한 개념으로 위치 정보가 필요한 오브젝트들의 월드 좌표를 미니맵 텍스쳐 좌표로 투영하여 점으로 오브젝트들의 위치를 표현하도록 했다.

**[-30000 ~ 30000 까지의 월드 좌표를 미니맵에 투영하기 위해서 사용한 계산식]**



미니맵을 화면 비율이 아닌 정사각형으로 비율을 맞추기 위해서 x, y 값에 0.54, 0.96을 곱했다.

**[문제점]**

1. 그려지는 순서가 불분명하여 미니맵이 점을 가리는 현상 발생, 미니맵에 그려질 객체가 많을 수록 문제가 발생할 가능성이 크다.
2. 객체마다 함수 인자 설정을 따로 하고 있어 비효율적이다.

**[해결방안]**

미니맵 매니저 클래스를 만들어서 오브젝트 매니저에서 오브젝트들의 태그 별로(적, 아군, 플레이어) 정보를 받아와 한번에 그려지고 관리할 수 있도록 해야 한다.

**이재원 :**

**1. 유도미사일 성능 개선**

충분한 근거리 에서 발사 하여도 미사일이 적중하지 않으며 미사일 모델의 머리 방향이 날아가는 방향의 Look을 향하지 않는 것을 해결

m\_xmfLunchPosition = xmfLunchPosition

기존엔 위의 코드로 플레이어의 미사일 발사 위치를 받아와서 타겟 오브젝트와의 방향을 알아 냈지만 이것이 오류의 시발점이 되었다는 것을 알았음.

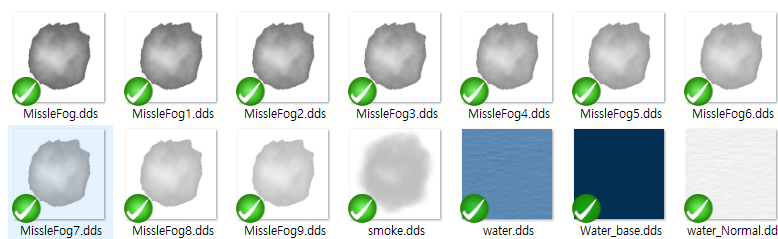
미사일이 날라가는 동안 새로 갱신된 미사일의 위치와 갱신된 타겟 오브젝트의 위치를 통하여 방향을 얻어내야 정상적인 유도 성능이 나온다는 것을 알고 m\_xmf3Position 으로 고쳐 오류를 해결함

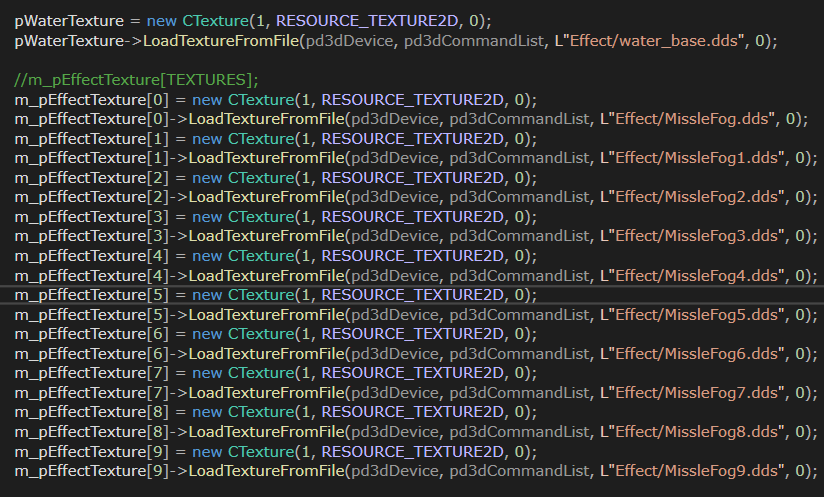
또한 이전 코드에서 세타만큼 미사일 모델 자체를 틀어주는 함수를 제거하여 미사일의 방향을 Look 방향으로 맞춰줌

**2. 텍스쳐 애니메이션 구현**

빌보드 등에 사용될 텍스쳐 애니메이션 기능을 구현함.

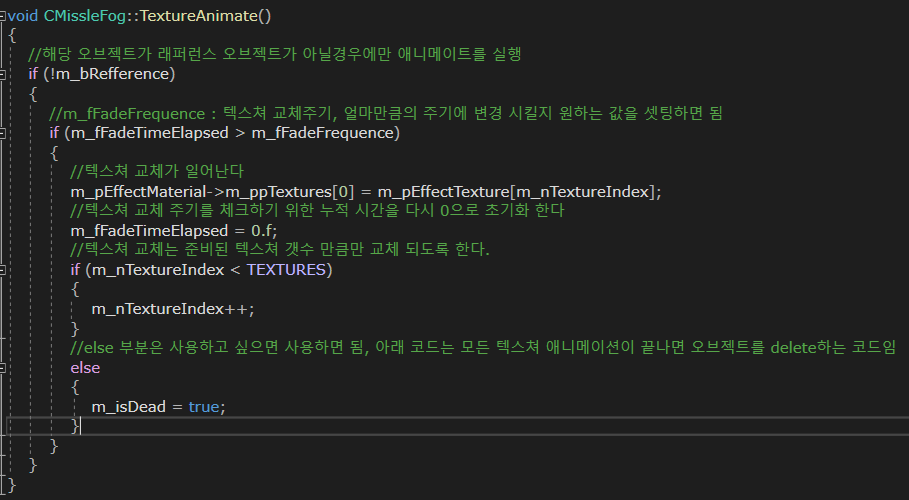
우선 저번주차에 만든 MissleFog를 알파값을 점진적으로 적용한 텍스쳐를 활용하여 시간이 지날수록 연기가 흐려지게 만드는 효과를 주어 테스트를 하기로 한다.





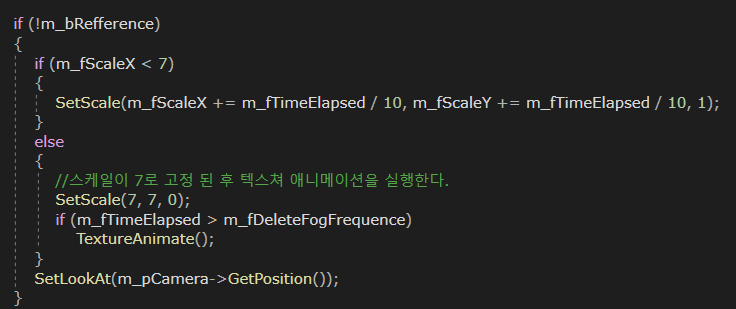
포토샵을 통해 알파값을 점진적으로 적용한 dds 파일을 생성하여 MissleFog 생성자에 로드한다.

그 후 Animate 함수에서 호출할 TextureAnimate 함수를 구현한다.



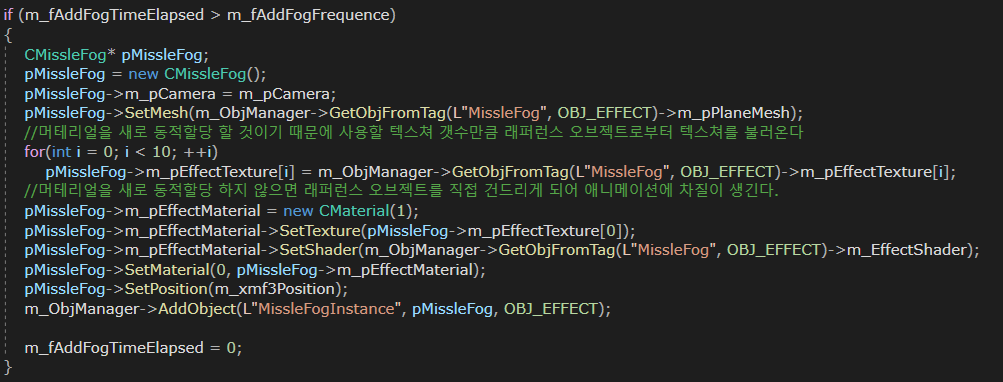
평면 오브젝트들은 머테리얼의 0번인덱스의 텍스쳐만 사용함으로 0번 텍스처에

생성자에서 로드하였던 텍스쳐의 인덱스로 일정 시간 간격으로 교체를 해주고 모든 텍스쳐 까지 교체가 끝나면 m\_isDaed를 true로 하여 오브젝트 매니저에서 삭제되게 만든다.



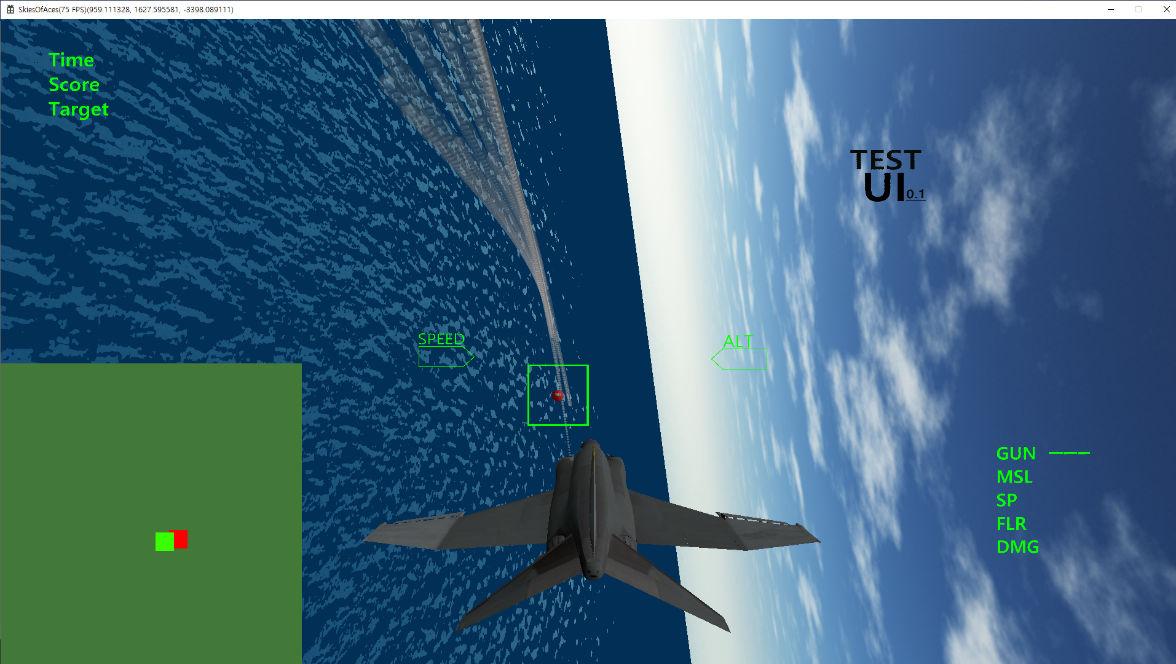
Animate 함수에서 TextureAnimate가 호출되는 위치는 오브젝트의 스케일이 7로 고정되고 일정 시간이 흐른 후 호출된다.

MissleFog는 TestScene에서 래퍼런스 오브젝트로 생성한 것의 내용을 공유하여 생성하기 때문에 Material을 동적생성하지 않으면 MissleFog의 삭제 주기가 동기화 되는 버그가 있음으로 생성할때 동적 생성을 한다.



Missile 클래스의 Animate의 MissleFog 생성 부분의 이전 주차 코드에서 for문을 통해 로드한 모든 텍스쳐 정보를 불러오는것과 머테리얼을 새로 동적할당 하여 생성한 부분이 추가되었다.

이로써 미사일 생성과 애니메이션은 되었으나 알파값이 0.5 이하로는 클리핑 되어 텍스쳐가 안보이는 현상이 발생하여 블랜드 설정에서 AlphaToCoverageEnable 을 False로 바꾼 후 쉐이더 코드에서 직접 클리핑을 하여 해결하였다.  

시간이 흐름에 따라 점차 흐려지는 MissleFog 오브젝트들

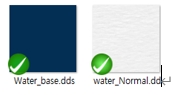
**3. 바다 텍스쳐 개선**

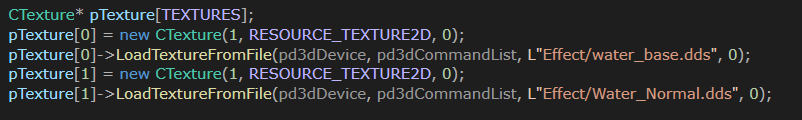
기존의 텍스처에서 타일링 + 알파 블랜딩 + 2중 텍스처를 활용하여 물결과 유사한 효과를 만듦

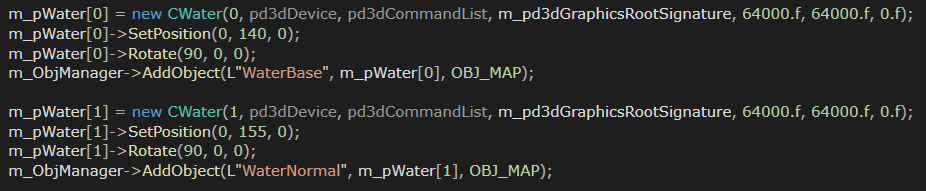
MissleFog의 쉐이더 코드와 블랜드 스테이트를 활용하여 바다 텍스처를 개선한다.

일단 짙은 파란색의 텍스처를 배치하고 상단에 이전에 사용하던 바다 텍스처에 노말 텍스처를

타일링 하여 배치한다.

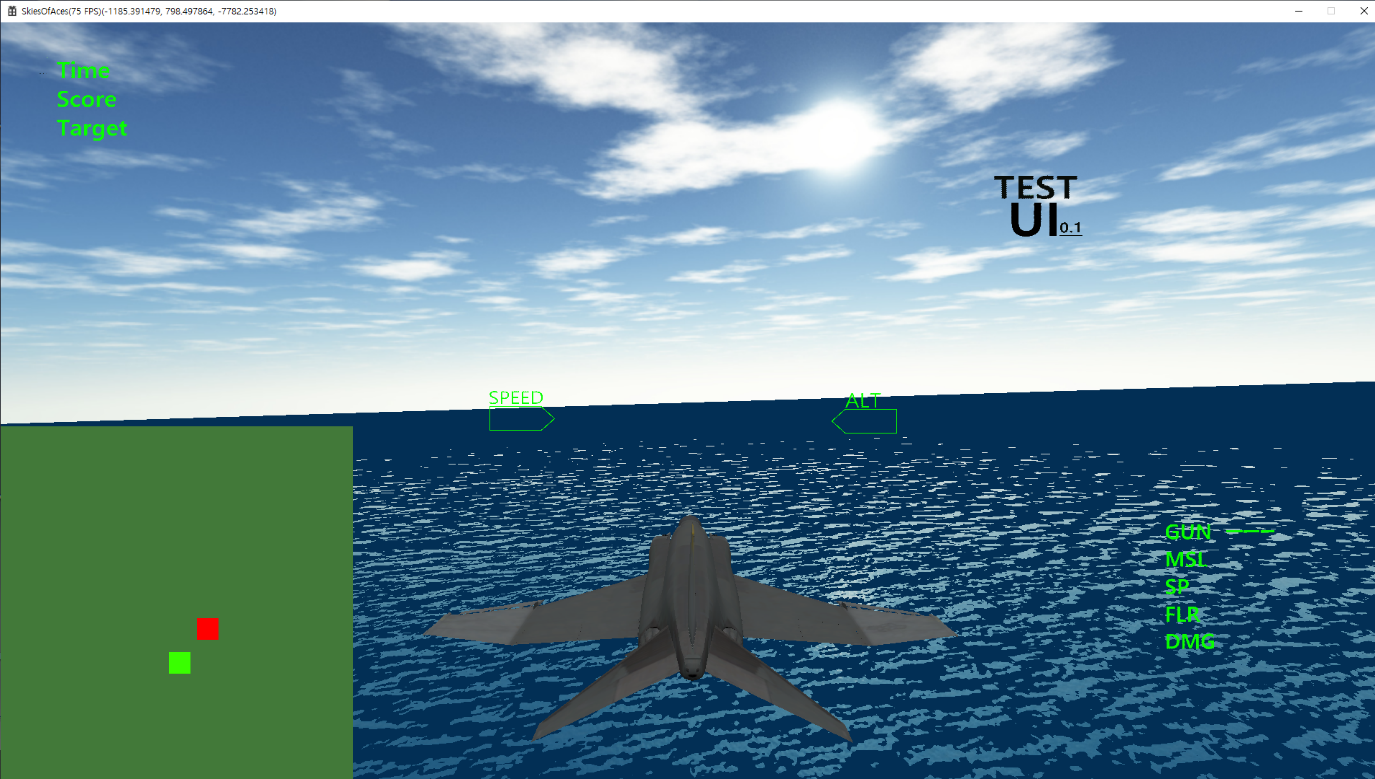






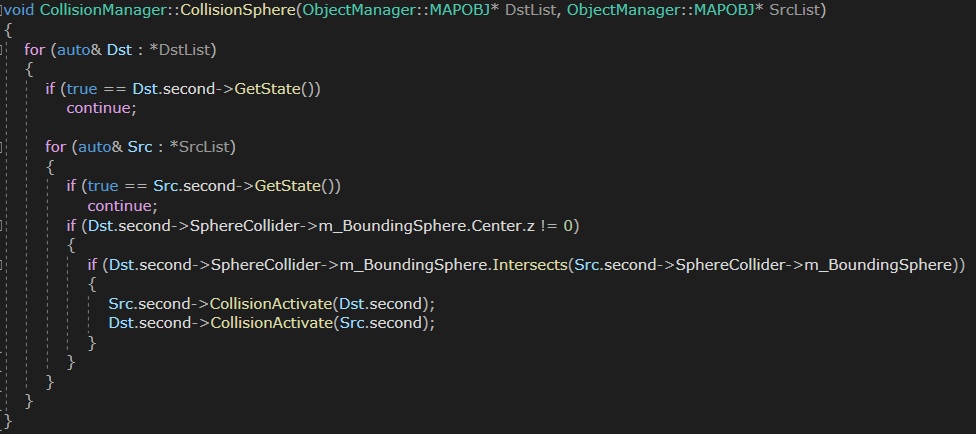
두 텍스쳐 모두 알파 블랜딩을 활성화 하였고 쉐이더 코드의 clip으로 인하여 노말 텍스쳐가 일정부분 클리핑 되어 밑이 뚫려 보이게 된다.

스카이박스의 하단부분은 하늘에 비춰진 바다임으로 물결과 유사한 효과를 본다.



**4. 충돌 매니저 제작**

오브젝트 매니저의 업데이트에서 오브젝트 타입별로 충돌 검사를 하기 위해 충돌 매니저 클래스를 작성함



충돌 매니저의 구 충돌 검사 함수의 내용

오브젝트 매니저에서 오브젝트 타입(enum으로 정의함)으로 오브젝트 간의 충돌을 검사하게 만들었다.

충돌이 되면 오브젝트에 가상함수로 만들어진 CollisionActivate라는 함수를 통해 충돌후의 행위를 정의할 수 있다. 지금은 미사일과 적 오브젝트가 충돌했을 경우 미사일이 삭제되게끔 해놓았다.

앞으로 구충돌 이외에 박스 충돌, 높이값 충돌 등등을 추가할 예정이다.



충돌 매니저는 오브젝트 매니저의 업데이트에서 호출된다.

오브젝트 매니저의 이터레이터가 순회하면서 충돌 매니저의 구충돌을 검사하게 된다.

**[문제점]**

텍스처 애니메이션 중 인덱스를 넘어가는 버그가 발생

그로인해 물 텍스쳐가 마지막 포그 오브젝트에 씌워지게 됨

**[해결방안]**

물 텍스처일 경우 렌더하지 않는 방향으로 임시조치함

|  |  |  |  |
| --- | --- | --- | --- |
| **문제점 정리** | 미니맵에 그려질 객체들을 관리하는 클래스의 부재, 그려지는 순서 불분명  물 텍스쳐가 마지막 포그 오브젝트에 씌워지게 됨 | | |
| **해결방안** | 미니맵 매니저 클래스 작성하여 태그 별로 관리  물 텍스처일 경우 렌더하지 않는 방향으로 임시조치 | | |
| **다음주차** | 16 주차 | **다음기간** | 2020.04.13 ~ 2020.04.19 |
| **다음주 할일** | 김령운 : 미니맵 매니저 클래스 작성, Behavior 툴 제작  이재원 : Behavior 툴 제작, 기타 자잘한 작업 해결 | | |
| **지도 교수**  **Comment** |  | | |