
3D 포트폴리오 계획서

제목 미정

분류 3인칭

작성자 김호곤

목차

가. 개요

나. 개발환경

다. 계획

라. 주차별 목표

가. 개요

Unity를 기반으로 제작하는 3인칭 게임이다. 플레이어는 적의 거점을 점령하면서 아군을 지원한다.

나. 개발환경

Unity 2022.3.5.f1

비주얼스튜디오

다. 계획

1. 자체 셰이더 제작

전체적으로 Toon 렌더링 느낌이 나도록 셰이더를 제작한다. 기본적으로 외곽선을 그린다. RimLight 또한 적용하는데, 사실적인 효과를 위해서 광원을 등진 방향에서만 효과가 나타나도록 한다.

2. 은신 효과 셰이더 제작

은신(은폐, 클로킹) 같은 기능을 사용하면 효과가 나타나도록 셰이더를 만든다. 노이즈 텍스처를 이용해서 부분에서 전체로 서서히 투명화된다. 완전히 투명화된 상태에서는 RimLight를 이용해서 경계 부분만 약간 보이도록 한다. 이때 RimLight에 텍스처를 적용해서 밋밋해 보이지 않게 한다. C# 스크립트에서 메타리얼을 다룰 때는 Asset을 직접 참조하는 것이 아니라 메타리얼을 메모리에 복사하고 복사본을 다루도록 한다.

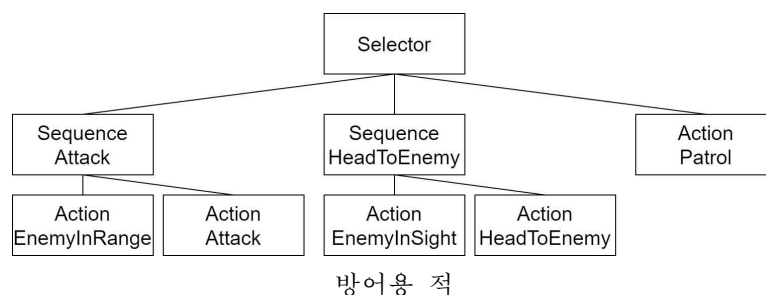
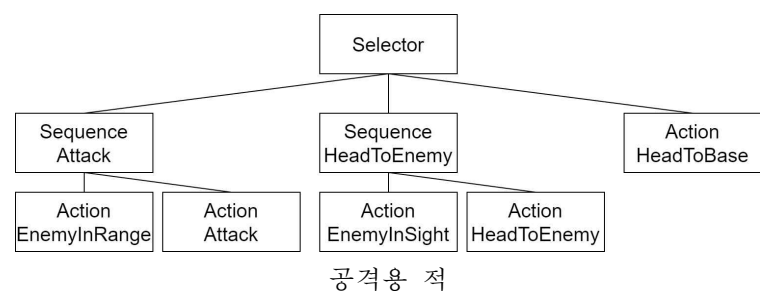
3. AnimationController 사용

플레이어 및 적 등의 동작을 Animation으로 만들고 AnimationController에서 동작시킨다.

4. ScriptableObject 사용

플레이어 정보, 적 정보, 지도 정보 등은 Unity 에디터를 쓰면 누구라도 수정이 편리하도록 Asset 파일로 다룬다.

5. 행동트리 적용

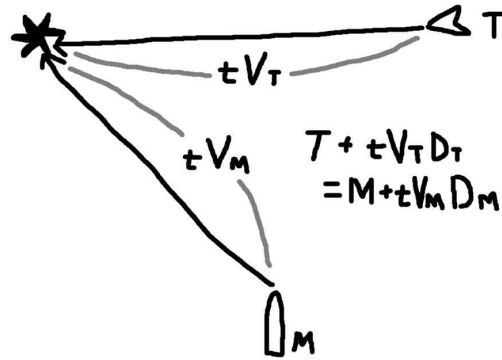


6. 상태패턴 적용

그 외 간단한 장비 움직임은 상태패턴을 적용한다.

7. 유도미사일 적용

유도미사일은 단순히 적만 보고 날아가는 것이 아니라 적의 예상 위치를 예측해서 적의 앞 방향으로 날아간다.



미사일 위치를 벡터 M , 목표물 위치를 벡터 T 로 표현하고, 미사일의 속력을 스칼라 V_m , 목표물의 속력을 스칼라 V_t 로 표현하고, 미사일의 방향을 벡터 D_m , 목표물의 방향을 벡터 D_t 라고 한다. 여기서 목표물의 위치, 방향, 속력과 미사일의 위치는 플레이 중 존재할 값이고, 미사일 속력은 게임 제작 중에 결정하게 될 것이다. 따라서 미사일의 방향만 계산하면 미사일을 유도할 수 있다. 어떤 시간 t 에 대해서, t 만큼의 시간이 지난 후의 목표물의 위치는 $T + tV_t D_t$ 가 될 것이고, 미사일의 위치는 $M + tV_m D_m$ 이 될 것이다. t 만큼의 시간이 지난 후에 두 물체의 위치가 같도록 하는 것이 목적이므로 $T + tV_t D_t = M + tV_m D_m$ 의 식을 성립시키는 D_m 을 찾아내야된다.

문제는 여기서 시간 t 가 아직 모르는 변수라는 것인데, t 를 구하기 위해 또 다른 식이 필요하다. 목표 지점을 표시할 벡터 $T + tV_t D_t$ 를 각 성분으로 나타내면 아래와 같다.

$$(T_x * tV_t D_t, T_y * tV_t D_t, T_z * tV_t D_t)$$

또한 미사일의 위치를 (M_x, M_y, M_z) 로 표현하면 목표 지점과 미사일까지의 거리를 아래와 같이 표현할 수 있다.

$$\sqrt{(T_x * tV_t D_t - M_x)^2 + (T_y * tV_t D_t - M_y)^2 + (T_z * tV_t D_t - M_z)^2}$$

이것이 tV_m 과 같은 것이다.

$$(T_x * tV_t D_t - M_x)^2 + (T_y * tV_t D_t - M_y)^2 + (T_z * tV_t D_t - M_z)^2 = (tV_m)^2$$

그러나 위 식은 워낙 복잡해서 $t = f(t)$ 꼴의 식으로 나타내기가 어렵다. 따라서 해를 구하기 위해 근의 공식 사용을 고려한다. 그렇게 나온 t 를 이용해 최종적으로 D_m 을 구한다.

8. 수직이착륙기 제작

기본적으로 플레이어의 장비는 수직이착륙기가 될 것이다. 플레이어는 원할 때 비행 기능, 수직이착륙 기능을 전환할 수 있다.

라. 주차별 목표

1주차 프로토타입

필요한 셰이더를 제작하고 기본적인 움직임 동작을 구현한다.

2주차 알파

플레이어의 장비의 움직임, 기능을 구현한다.

3주차 베타

전체 지도를 만들고 적의 행동을 구현한다.

4주차 최종

게임이 진행될수록 플레이어가 성장하는 것을 구현하고 승리 조건, 패배 조건을 만든다.