**Retreat Protocol**

**적 AI 기술 문서**

본 문서에서는 게임에 구현된 적대적 NPC의 AI의 구현 과정에 대해 기술한다.

**이동**

NPC는 플레이어와 일정 거리를 유지한 채 플레이어를 따라다닌다. NPC도 플레이어와 마찬가지로 직진만 가능하며, 함선을 기울여 직진 방향을 변경함으로써 이동 경로를 수정한다.

NPC는 생성 시 플레이어와 유지할 최대 거리와 최소 거리를 구한다. 최소 거리는 30 ~ 50 사이의 무작위 값이며, 최대 거리는 최소 거리에 30 ~ 100 사이의 무작위 값을 더한 값이다.

이후 NPC는 매 프레임마다 NPC와 플레이어 사이의 거리를 계산하면서, 다음과 같은 조건에 맞춰 목표 방향을 설정한다:

* NPC와 플레이어 사이의 거리가 최소 거리보다 짧다면, NPC는 플레이어를 등지는 방향을 목표 방향으로 설정한다.
* NPC와 플레이어 사이의 거리가 최대 거리보다 멀다면, NPC는 플레이어를 바라보는 방향을 목표 방향으로 설정한다.
* 그 외의 경우 현재 이동 방향을 유지한다.

목표 방향은 플레이어와 NPC의 위치 벡터의 뺄셈으로 구한다. 이 때 목표 방향은 Vector3 형으로 계산되는데, 실제로 이 값을 사용하려면 Quaternion형이 필요하기 때문에 Quaternion.LoockAt()을 호출하여 Quaternion형으로 변환한다.

이렇게 목표 방향을 설정하였다면, Quaternion.Slerp 함수로 계산한 보간 값을 이용하여 목표 방향에 도달할 때까지 서서히 rotation 값을 조절하여 NPC의 이동 경로를 수정한다.

**이동 (의사 코드)**

|  |
| --- |
| void Update()  {       바라보는 방향(Vector3.foward)으로 이동한다.       목표\_방향\_계산\_함수 호출       목표 방향을 바라보도록 회전(Quaternion.Slerp로 계산한 보간 값 활용)  }  void 목표\_방향\_계산\_함수()  {  Vector3.Distance 함수를 활용하여 플레이어와 자기 자신 사이의 거리를 계산       if( 최소 거리보다 짧은 경우 ) 플레이어를 등지는 방향으로 목표 방향을 설정       else if( 최대 거리보다 긴 경우 ) 플레이어를 바라보는 방향으로 목표 방향을 설정  else 아무것도 하지 않는다. (현재 이동 방향을 유지)  } |

**공격**

NPC는 공격이 준비될 때마다 플레이어의 예상 위치를 계산하고, 이 위치를 목표로 직선 방향으로 날아가는 공격 투사체를 발사한다. 플레이어의 속도와 방향이 일정할 것을 전제하기 때문에, 플레이어는 이동 경로를 수정하여 NPC의 공격을 회피할 수 있다.

현재 플레이어의 위치와 이동 방향, 이동 속도, 현재 NPC의 위치와 NPC 포탄의 이동 속도를 기반으로, 다음과 같은 절차를 따라 플레이어의 미래의 예상 위치를 계산하여 목표 위치로 삼는다:

1. 플레이어가 이동 중인 방향과 이동 속도 정보를 담은 벡터 v1을 구한다.
2. NPC의 위치에서 플레이어의 위치로 향하는 벡터 v2를 구한다.
3. NPC가 발사하는 포탄의 이동속도와 벡터 v2의 크기를 기반으로, NPC의 위치에서 벡터 v2 방향으로 발사한 포탄이 플레이어의 위치에 도달하는 데 걸리는 시간 t1을 계산한다.
4. 시간 t1 이후, 플레이어가 벡터 v1을 따라 이동한 결과 위치 p1을 계산한다.
5. NPC의 위치에서 위치 p1으로 향하는 벡터 v3를 구한다.
6. NPC가 발사하는 포탄의 이동속도와 벡터 v3의 크기를 기반으로, NPC의 위치에서 벡터 v3 방향으로 발사한 포탄이 위치 p1에 도달하는 데 걸리는 시간 t2를 계산한다.
7. 시간 t2 이후, 플레이어가 벡터 v1을 따라 이동한 결과 위치 p2를 계산한다.
8. 이렇게 계산한 p2를 조준하여 포탄을 발사한다.

NPC의 명중률이 너무 높아 게임 난이도 조절에 어려움이 있을 경우 5~7번 단계를 생략하고, 4번 단계에서 계산한 p1을 조준한다. 명중률이 너무 낮은 경우, 5~7번 단계를 반복하여 더욱 정밀한 위치를 계산한다.

**공격 (의사 코드)**

|  |
| --- |
| void 공격\_실행  {  공격 속도만큼 시간이 지날 때가지 대기  If( 시간이 충분히 지나면 )  {  공격\_위치\_계산 함수를 호출하여 공격 목표 위치를 계산  목표 위치를 향해 날아가는 투사체 생성  공격 속도 초기화  }  }  Vector3 공격\_위치\_계산  {  현재 발사할 포탄의 이동속도 참조  플레이어의 현재 위치와 이동 방향, 속도 정보를 담은 벡터 v1 생성  현재 위치에서 플레이어로 향하는 방향과 포탄의 이동속도 정보를 담은 벡터 v2 생성         벡터 v2 방향으로 날아가는 포탄이 벡터 v1 위치에 도달할 시간 t 계산  시간 t 이후 플레이어의 예상 위치 정보를 담은 위치 p1 생성  위치 p1을 반환  // 명중률이 낮을 경우, 플레이어의 현재 위치에 p1을 대입하여 이 과정을 반복.  } |