# 좀비 감지/추격 시스템

## ■ 시스템 이름

좀비 감지/추격 시스템 (Zombie Detection & Chase System)

## ■ 목적

밤 시간 동안 플레이어가 자원을 파밍하는 도중, 좀비에 의해 감지될 경우 추격이 시작되어 위기 상황이 발생하도록 하여 스릴과 긴장감을 부여한다.

## ■ 기본 개념

- 좀비는 일정 범위 내에서 시야 또는 청각으로 플레이어를 감지할 수 있음  
- 감지 조건이 충족되면 좀비는 플레이어 방향으로 추격을 시작함  
- 플레이어가 일정 거리 이상 벗어나거나 시야에서 벗어나면 추격 중단  
- 감지/추격 중에는 경고 UI 또는 효과음을 통해 긴장감을 전달함

## ■ 데이터 구조

public class ZombieEnemy : MonoBehaviour {  
 public float visionRange;  
 public float chaseSpeed;  
 public bool isChasing;  
 public Transform target;  
 public void DetectPlayer(Transform player);  
 public void ChasePlayer();  
 public void StopChase();  
}

## ■ 동작 흐름

1. 좀비가 일정 시간 간격으로 주변에 플레이어 존재 여부 감지  
2. 플레이어가 감지 범위에 들어오면 DetectPlayer() 호출  
3. 감지되면 isChasing = true → ChasePlayer() 실행  
4. 추격 중 일정 거리 이상 벌어지거나 장애물에 의해 시야 차단 시 StopChase() 실행  
5. 추격 실패 또는 성공 결과에 따라 리셋 또는 게임 오버 처리

## ■ 조건/트리거

- 플레이어가 visionRange 내에 진입 + 장애물 없음 → 감지  
- 감지 후 일정 시간 추격 진행  
- 플레이어가 일정 거리 이상 도망 → 추격 중단  
- 플레이어가 잡힐 경우 → 패널티 발생 or 게임 오버

## ■ 예상 결과

- 감지 성공: 좀비가 추격 시작, 경고 UI/사운드 출력  
- 추격 성공: 플레이어 패널티 or 죽음  
- 추격 실패: 일정 시간 후 좀비 원래 위치로 복귀

## ■ UI 연동

- 감지 시 화면에 빨간 테두리 또는 경고 아이콘 표시  
- 추격 중 사운드/음악 변경 (긴장감 전달)  
- 추격 종료 시 안정화 효과

## ■ 기타 고려 사항

- 추후 좀비마다 다른 시야 범위, 속도, 청각 감지 능력 추가 가능  
- 플레이어가 소리를 내는 행위 (달리기, 파밍)도 감지 요인이 될 수 있음  
- 특정 도구 사용 시 일시적으로 시야 차단 가능 (연막, 횃불 등)

## ■ 프로그래밍 연동 예상 함수

public void DetectPlayer(Transform player); // 플레이어 감지  
public void ChasePlayer(); // 추격 로직 시작  
public void StopChase(); // 추격 종료