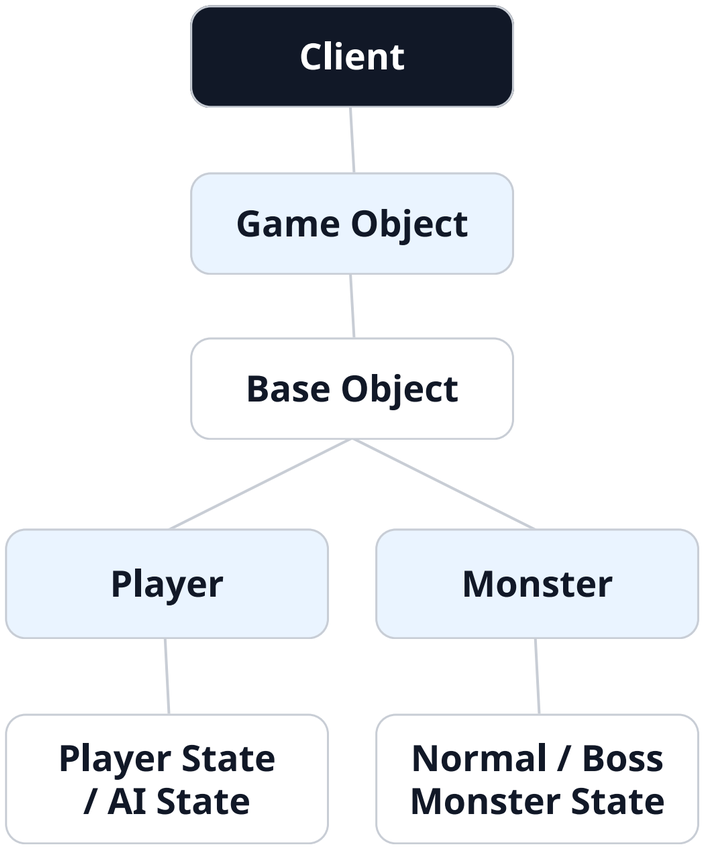
AI를 직접 설계하고,
상태와 조건으로 데이터화해
게임 안에서 구현했습니다.

Tales of Arise 모작 프로젝트 · FSM 기반 플레이어 AI / 몬스터 AI

공통 구조 위에 플레이어 AI와 몬스터 AI를 설계했습니다

구조



직접 담당한 설계 범위

PLAYER AI

플레이어 조작과 AI 모드를 자유롭게 전환가능

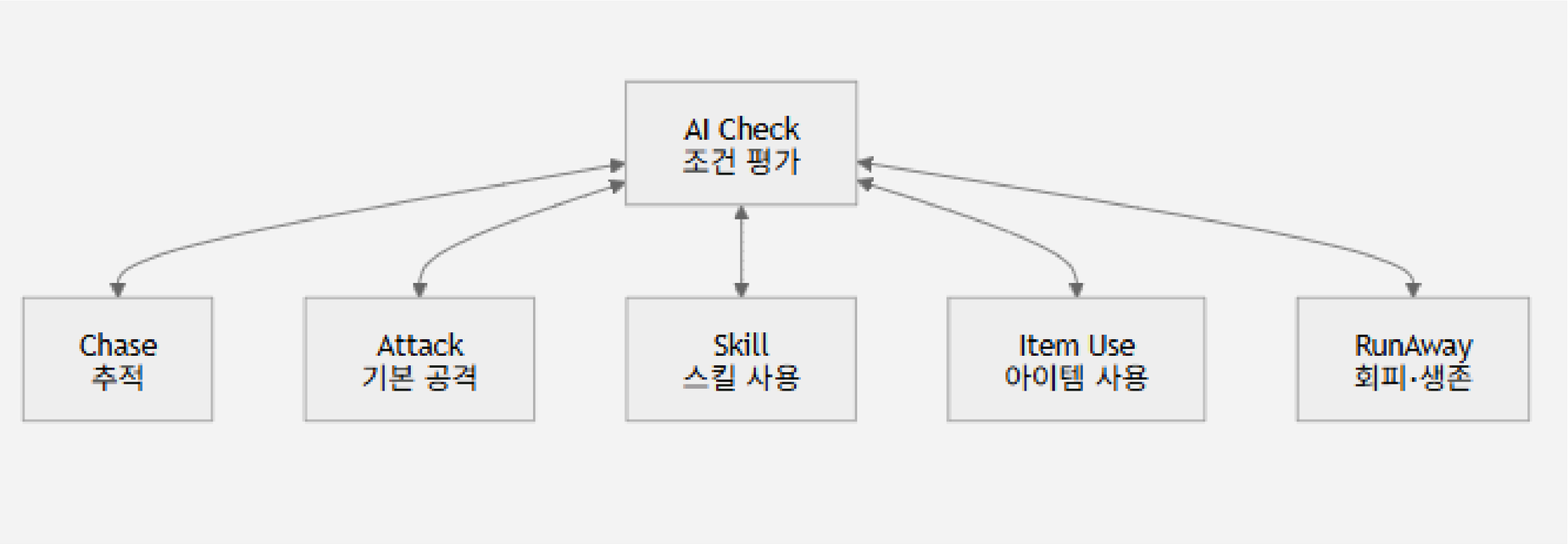
MONSTER AI

근거리·원거리 특성에 맞춰 추적, 회피, 피격, 공격, 스킬 상태를 독립적으로 구성

설계 의도

새로운 행동은 상태 클래스로 추가하고, 공통 전이 규칙은 재사용해 캐릭터와 몬스터가 늘어나도 확장 가능한 구조를 목표로 했습니다.

행동을 상태로 분리해 전이와 확장을 단순화했습니다



핵심

상태 추가만으로 행동 확장

상태 클래스 기본 구조.

Enter
상태 진입 시 초기화

Tick
매 프레임 행동 갱신

Exit
전환 전 필요한 작업

ChangeState
새 상태 반환·교체

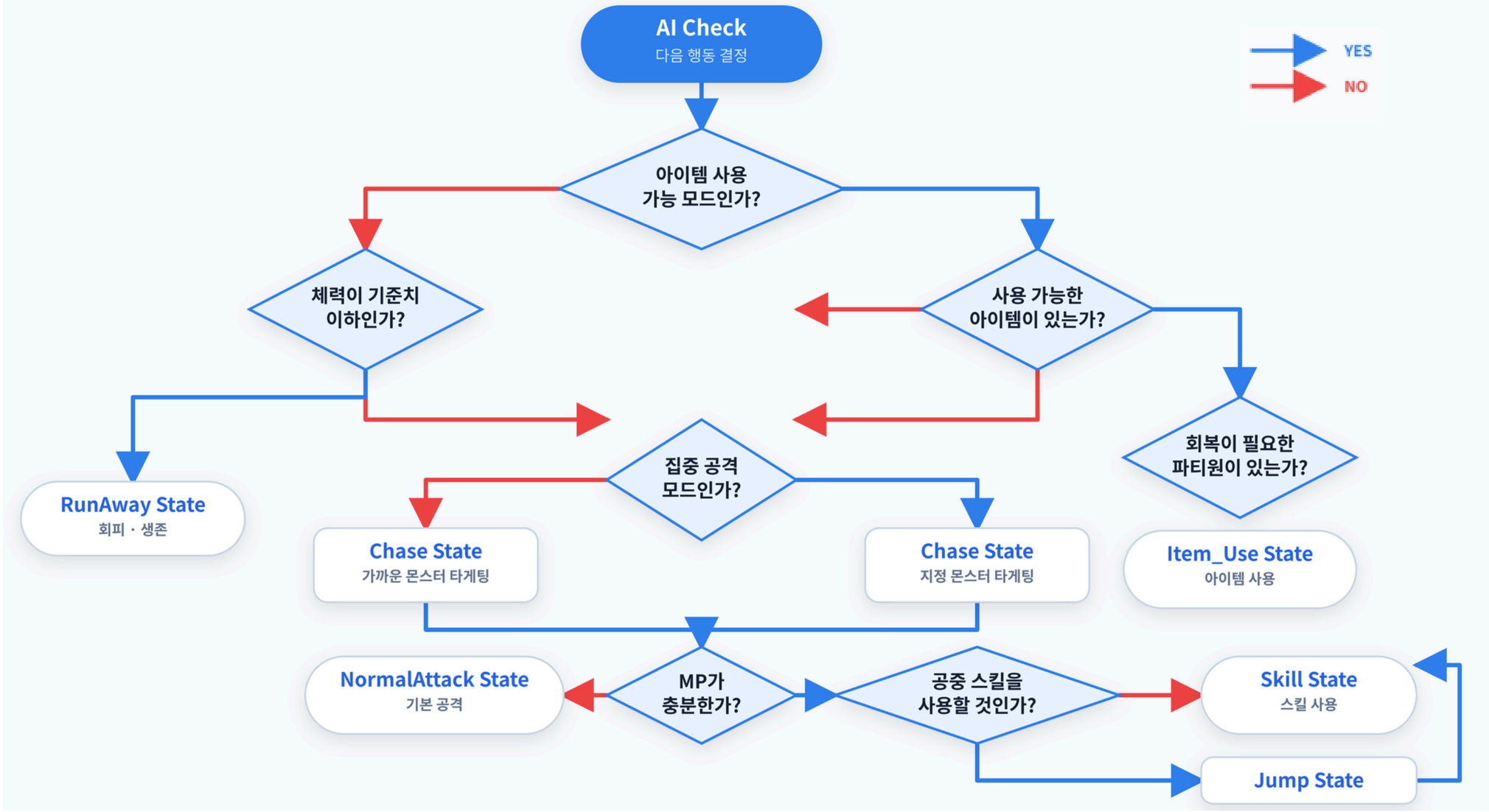
AI 판단 기준을 조건·우선순위·결과 상태로 데이터화했습니다

AI Check에서 조건을 순서대로 평가하고, 충족된 결과 상태로 전이합니다.

순위	판단 조건	AI 해석	전이 상태
1	아이템 사용 허용 + 사용 가능 아이템 보유	체력 감소 파티원 탐색	Item_Use State
2	현재 체력 ≤ 기준치	생존 행동 우선	RunAway State
3	집중 공격 모드	지정 몬스터를 타깃	Chase State
3	자유 공격 모드	가까운 몬스터 탐색	Chase State
4	MP 충분	공중 스킬 가능 여부 판단	Jump / Skill State
5	MP 부족 또는 스킬 사용 불가	기본 공격 선택	NormalAttack State

AI 알고리즘 순서도

AI Check에서 생존·타깃·공격 조건을 순서대로 평가해 다음 State를 결정합니다.



모드 선택을 통한 명령이 가능합니다.

ATTACK_LOCKON MODE

지정된 몬스터가 있으면 해당 타깃을 유지
지정 대상이 없으면 가까운 몬스터를 탐색

ATTACK_FREE MODE

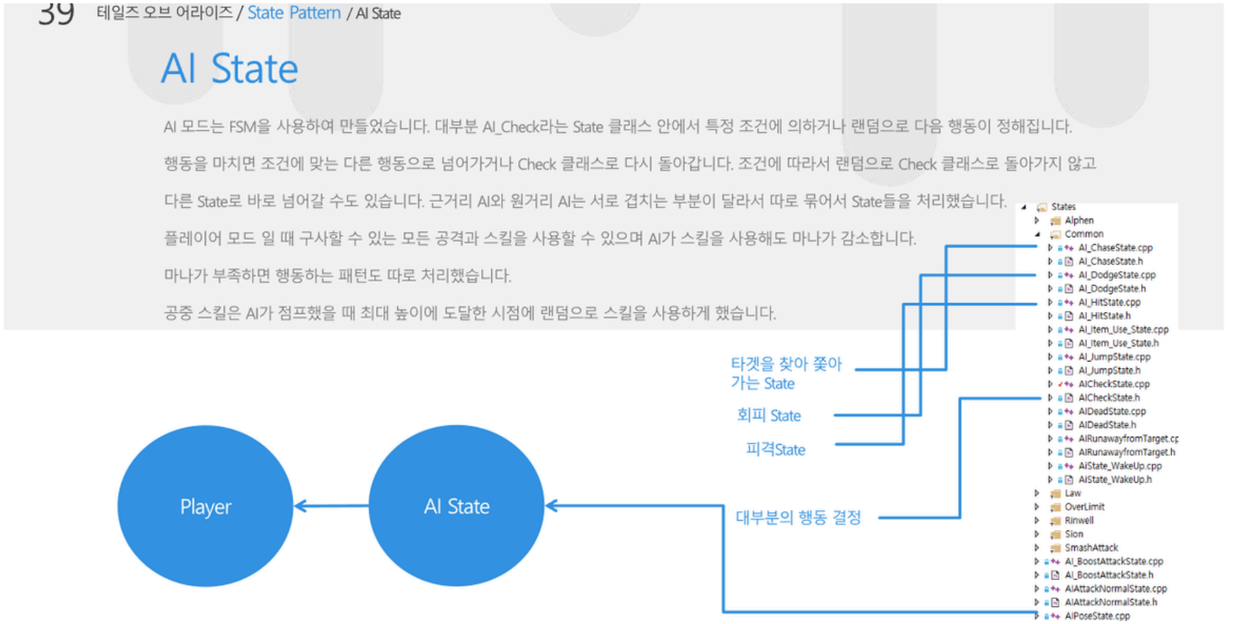
플레이어 위치를 기준으로 생존 중인 몬스터를 순회해
최단 거리 대상을 선택



기존 구현 코드와 인게임 AI 모드 선택 화면

C++·DirectX 11로 FSM 기반 AI를 구현했습니다





상태 전환 구현

Tick → 상태 반환 → Exit / Enter → 상태 교체

행동 단위 구현

추적 · 회피 · 피격 · 일반 공격 · 스킬 · 아이템 사용