

# 턴제 전술 게임 AI 개선을 위한 포괄적 연구

CompanionAI v3의 핵심 문제인 매직 넘버 하드코딩과 상황 인식 한계는 **Response Curves** 기반 Utility AI 재설계, **Influence Maps**를 활용한 전술적 공간 인식, 그리고 **Blackboard** 시스템 기반 멀티 에이전트 협력을 통해 해결할 수 있다. 이 세 가지 접근법은 XCOM, Dragon Age: Inquisition, Divinity: Original Sin 2 등 검증된 상용 게임에서 이미 실전 적용되었으며, 모드 환경의 100ms 제약 내에서 충분히 구현 가능하다. 특히 Dave Mark의 Infinite Axis Utility System(IAUS)과 [Wikipedia](#) Days Gone의 Squad Coordination 모델은 즉시 적용 가능한 아키텍처 패턴을 제공한다.

---

## Response Curves가 매직 넘버를 대체하는 방법

현재 시스템의 가장 큰 약점인 하드코딩된 매직 넘버는 **Response Curves**를 통해 체계적으로 제거할 수 있다. [Guild Wars 2: Heart of Thorns](#)에서 처음 대규모로 적용된 이 기법은  $\text{Raw Input} \rightarrow \text{Normalize } [0,1] \rightarrow \text{Response Curve} \rightarrow \text{Final Score}$ 의 파이프라인을 따른다. 예를 들어, 체력 기반 방어 행동 우선순위를 결정할 때 기존의 `if (health < 30) priority = HIGH`와 같은 임계값 분기 대신, **시그모이드 곡선**을 적용하면 체력 감소에 따라 방어 우선순위가 자연스럽게 상승한다.

핵심 곡선 유형은 네 가지로 구분된다. **선형 곡선**( $y = mx + b$ )은 거리-위협 같은 비례 관계에 적합하고, **다항 곡선**( $y = x^n$ )은 지수값에 따라 초반 가속( $n < 1$ ) 또는 후반 가속( $n > 1$ )을 제어한다. **로지스틱/시그모이드 곡선**은 체력 임계값처럼 특정 구간에서 급격한 전환이 필요할 때 사용하며, **구간별 선형 곡선**은 디자이너가 정밀 제어해야 할 때 활용한다. Unity의 AnimationCurve를 활용하면 에디터에서 시각적으로 조정 가능하며, JSON으로 내보내 버전 관리할 수 있다.

ArenaNet의 Dave Mark가 개발한 **IAUS(Infinite Axis Utility System)**\*\*는 이 개념을 확장한다.

[Wikipedia](#) 각 행동(Decision)은 여러 Consideration의 점수를 곱셈으로 결합하여 최종 점수를 산출한다. 중요한 점은 어떤 Consideration이라도 0점을 반환하면 해당 행동 전체가 즉시 탈락한다는 것이다. 이 특성을 활용해 저비용 Consideration을 먼저 평가하면 **조기 종료(early-out) 최적화**가 가능하다. [Dragon Age: Inquisition](#)의 Behavior Decision System도 이 패턴을 채택하여 평균 10-20개의 Behavior Snippet을 캐릭터당 관리한다. [Aiandgames](#)

---

## Influence Maps를 통한 전술적 공간 인식

현재 CompanionAI의 상황 인식 한계는 **Influence Maps** 도입으로 극복할 수 있다. Influence Map은 전장을 그리드로 분할하고 각 셀에 위협도, 아군 지원도, 엠페 품질 등의 값을 할당하는 기법이다. [gameapro](#) [Guild Wars 2](#)의 Centaur AI는 수백 단위 규모에서 이 시스템을 성공적으로 운영했으며, 핵심은 **템플릿 기반 전파**다. 영향력 감쇠 패턴을 미리 계산한  $21 \times 21$  템플릿을 준비하고, 유닛 위치에 "스탬프"처럼 찍어 맵을 갱신하면 계산 비용이 크게 줄어든다. [gameapro](#)

실전 활용 예시로, 탱크 포지셔닝을 위해  $\text{TacticalScore} = \text{CoverMap} \times 0.4 + (1 - \text{ThreatMap}) \times 0.6$ 과 같이 여러 맵을 조합할 수 있다. Days Gone은 여기에 **Frontline** 시스템을 추가하여 아군 무게중심에서 적 무게중심을 향한 방향 벡터를 계산하고, 이를 기준으로 레인을 분할해 각 대원에게 할당한다. [gameapro](#) 이 접근법은 자연스러운 측면 공격과 포위 기동을 가능케 한다.

위협 평가 알고리즘은 Influence Map과 결합하여 더 강력해진다. 기본 공식은  $\text{ThreatScore} = (\text{Damage} \times \text{DamageWeight}) + (1/\text{HP} \times \text{LethalityWeight}) + (1/\text{Distance} \times \text{ProximityWeight}) + \text{RoleBonus}$ 의 형태를 취한다. 적 힐러에게 +2.0, 딜러에게 +1.0, 탱크에게 +0.5의 역할 보너스를 부여하면 "힐러 우선 처치" 같은 전술적 판단이 자동으로 이루어진다. XCOM 시리즈는 여기에 **명중률**, **측면 공격 여부**, **킬 확정 가능성**을 추가로 고려하여 더 정교한 타겟팅을 구현한다. (PC Gamer)

---

## 멀티 에이전트 협력을 위한 Blackboard와 신뢰도 시스템

여러 동료가 함께 싸우는 상황에서 효과적인 협력을 위해 **Blackboard 시스템**이 핵심 인프라가 된다. Blackboard는 에이전트 간 공유되는 키-값 저장소로, 현재 교전 중인 적, 역할 선점 상태, 집중 공격 대상 등 의 정보를 교환한다. (GDQuest) Days Gone의 스쿼드 AI는 이를 한 단계 발전시켜 **신뢰도(Confidence)** 시스템을 도입했다.

신뢰도 공식은  $\text{Confidence} = (\text{자신의 전투력} + \sum(\text{아군 신뢰도} \times \text{아군 전투력}) + \text{킬 수}) / (\sum(\text{적 신뢰도} \times \text{적 전투력}) + \text{피해})$  형태다. (gameaiopro) 이 값에 따라 행동이 달라진다. **Heroic/Confident** 상태에서는 적극적 공격과 측면 기동을, **Neutral**에서는 현 위치 유지를, **Worried/Panicked** 상태에서는 후퇴와 엄폐를 우선한다. 자기 강화 효과로 인해 자신감 있는 아군은 주변 아군의 신뢰도를 높이고, 패닉 상태의 아군은 오히려 적의 위협으로 계산되어 팀 전체 사기에 영향을 준다. (gameaiopro)

**역할 기반 협력**에서 MMO의 "Holy Trinity" 시스템을 참조할 수 있다. 탱크는 어그로 관리에 집중하며, (Wikipedia)  $\text{Utility_Taunt} = (\text{적에게 타겟되지 않은 적 수} / \text{전체 적 수}) \times \text{아군 체력 압박도}$ 로 도발 우선순위를 계산 한다. 힐러는  $\text{Utility_Heal} = \sum(\text{아군 손실 체력} \times \text{아군 우선도} \times 1/\text{거리})$ 로 치유 대상을 결정하고, 딜러는  $\text{Utility_Attack} = \text{타겟 위협도} \times \text{킬 확정 보너스} \times (1 - \text{어그로 위험도})$ 로 공격 대상을 선정한다. Divinity: Original Sin 2는 여기에 **시뮬레이션 기반 점수 산정**을 추가하여, 스킬 사용 전 예상 결과를 시뮬레이션하고 가장 높은 점수의 행동을 선택한다. (larian)

---

## 상용 게임에서 검증된 AI 아키텍처

XCOM 시리즈의 AI는 Utility 기반 시스템의 교과서적 사례다. 각 유닛은 모든 가능한 행동에 대해 공격적, 방어적, 무형적 이점을 평가하고 가장 높은 유용성 점수의 행동을 선택한다. 포지셔닝 AI는 이동 가능한 모든 위치에 대해 엄폐 보너스, 측면 공격 기회, 노출된 적 수(1명이 최적)를 점수화한다. (PC Gamer) 17가지 이상의 유닛 타입별 AI 행동 세트가 있으며, Muton Blood Call은 아군 버프에 높은 가중치를, Sectopod는 다중 타겟 범위 공격에 높은 가중치를 둔다.

**Divinity: Original Sin 2**의 AI 2.0(Jane)은 **자동 추론 시스템**이 특징이다. 새 스킬이 추가되어도 별도 설정 없이 AI가 스스로 활용법을 파악한다. 스킬별로 데미지 점수, 체력 점수, 버프 점수를 시뮬레이션하고, 가장 낮은 방어력의 적을 우선 타겟으로 삼는다. (larian) 환경 상호작용(폭발물, 표면 효과)도 자동으로 고려되며, (PC Invasion) **Archetype 시스템**으로 같은 스킬셋이라도 다른 "성격"의 AI를 쉽게 생성할 수 있다. (larian)

**Dragon Age: Inquisition**의 BDS(Behavior Decision System)는 **Behavior Snippet**이라는 모듈형 단위를 사용한다. 각 스니펫은 스킬과 평가 트리의 조합이며, 캐릭터당 10-20개가 등록된다. 핵심 혁신은 같은 능력의 **다중 용도 평가**다. Charging Bull 스킬은 공격 진입용으로도, 긴급 탈출용으로도 점수 산정되어 상황에 맞게 선택된다. (Aiandgames) 점수 임계값 시스템으로 전투 중에는 Idle 행동이 선택되지 않도록 보장한다.

---

## 모드 환경을 위한 실용적 구현 전략

**100ms 제약** 내에서 구현하려면 철저한 최적화가 필수다. 핵심 기법은 세 가지다. 첫째, 조기 종료 최적화로 저비용 Consideration을 먼저 평가하고 0점 발생 시 즉시 다음 행동으로 넘어간다. 둘째, 턴 캐싱으로 경로 탐색, 시야 판정 등 비용이 큰 계산을 턴 시작 시 한 번만 수행하고 재사용한다. 셋째, 타겟 프루닝으로 전체 점수 산정 전에 빠른 휴리스틱으로 후보를 5-10개로 축소한다.

Consideration 평가 순서도 중요하다. 가장 저렴하고 실패 가능성이 높은 검사(사거리, 쿨다운)를 앞에 배치하고, 비용이 큰 계산(경로 탐색, AoE 시뮬레이션)은 뒤로 미룬다. 이 원칙을 따르면 순수 Utility AI는 **1-5ms**, Influence Maps 추가 시 **5-15ms**, 경량 MCTS 추가 시 **30-80ms** 수준의 성능을 달성할 수 있다.

오픈소스 프로젝트 중 **CrystalAI**(GitHub, 471 stars, MIT 라이선스)가 가장 직접적으로 활용 가능하다. 순수 C# 라이브러리로 Unity .NET 3.5+와 완전 호환되며, 확장 가능한 Utility AI 패턴을 구현한다. XCOM 2 모드 중 **A Better ADVENT**는 모드 환경에서 AI 튜닝의 좋은 사례로, **.ini** 설정 파일을 통한 데이터 주도 조정과 유닛 타입별 특화 로직을 보여준다.

---

## MCTS와 HTN: 고급 옵션

\*\*MCTS(Monte Carlo Tree Search)\*\*는 멀티스텝 계획이 필요할 때 고려할 수 있다. 100ms 제약 하에서 순수 MCTS는 어렵지만, **하이브리드 접근**이 가능하다. Utility AI로 행동 후보를 10-15개로 축소한 후, 남은 시간(50-80ms)에 MCTS를 적용한다. 시뮬레이션 단계에서 완전 랜덤 대신 휴리스틱 정책을 사용하면 적은 반복으로도 유의미한 결과를 얻는다. μRTS 연구에 따르면 파라미터화된 휴리스틱 기반 프리셀렉션이 분기 계수를 10-100배 줄인다.

\*\*HTN(Hierarchical Task Network)\*\*은 복합 전술을 자연스럽게 표현할 수 있다. "전투 턴 실행"이라는 복합 태스크가 상황에 따라 "공격적 접근"(체력 50% 이상, 적 근접), "방어적 플레이"(체력 30% 미만), "아군 지원"(아군 체력 50% 미만) 메소드로 분해된다. 각 메소드는 다시 원시 태스크(이동, 스킬 사용)로 분해되어 실행된다. F.E.A.R., Killzone, Horizon Zero Dawn이 이 방식을 채택했으며, 계획 시간은 **1-10ms**로 매우 빠르다.

---

## 결론: 단계적 개선 로드맵

CompanionAI v3의 아키텍처 개선은 **3단계 로드맵**으로 접근하는 것이 현실적이다.

**1단계(즉시 적용):** 기존 Utility AI의 매직 넘버를 Response Curves로 교체하고, 조기 종료 최적화와 턴 캐싱을 도입한다. 모든 가중치와 곡선 파라미터를 JSON 설정 파일로 외부화하여 코드 수정 없이 튜닝 가능하게 만든다. 예상 성능 개선은 30-50%다.

**2단계(중기):** Influence Maps를 도입하여 위협 평가와 포지셔닝 AI를 강화한다. Blackboard 시스템으로 동료 간 기본 협력(역할 선점, 집중 공격 대상 공유)을 구현한다. Tank/DPS/Support 역할별 Utility 가중치 프로파일을 분리하여 각 역할의 특성을 강화한다.

**3단계(장기):** 신뢰도 시스템과 Frontline 계산으로 동적 전술 조정을 구현한다. 필요시 경량 MCTS를 특정 상황(다수 타겟 AoE 최적화, 킬 확정 시퀀스 탐색)에 한정 적용한다. HTN을 도입하여 복합 전술 패턴(측면 공격, 집중 사격, 철수)을 표현한다.

이 접근법의 강점은 각 단계가 독립적으로 가치를 제공하면서도 다음 단계의 기반이 된다는 점이다. 1단계만으로도 현재 시스템의 핵심 문제가 해결되며, 2-3단계를 통해 상용 게임 수준의 AI 품질에 도달할 수 있다. 모든 기법은 강화학습이나 딥러닝 없이 순수 휴리스틱과 알고리즘으로 구현 가능하며, 100ms 성능 제약을 충족한다.