Finite-state machine (FSM)을 활용한 Non-player character (NPC) 인공지능 구현 성능 비교

이학주(*), 이아현(**)
(*)순천향대학교 메타버스&게임학과, hakju.lee@ramicstudio.com
(**)순천향대학교 메타버스&게임학과, ahyun.lee@sch.ac.kr

1. 연구 배경

게임 산업이 발전하면서 NPC(Non-Player Character) 인공지능의 중요성이 커지고 있다. 현대 게임에서는 다수의 NPC가 플레이어와 동시에 상호작용하므로, 이들의 역할이 매우 중요하다. 좋은 게임 경험을 위해서는 NPC의 자연스러운 행동, 예측 가능한 패턴, 그리고 유지보수의 효율성이 필수적이다. 특히 오픈월드나 모바일처럼 자원이 한정된 환경에서는 수많은 NPC를 성능 저하 없이 구동하는 것이 핵심 과제이다 [1].

이러한 문제를 해결할 효과적인 방법으로 유한 상태 기계(Finite State Machine, FSM)가 주목받고 있다. FSM은 NPC의 행동을 '상태(State)'와 '전이(Transition)'로 명확히 구조화하여 예측과 관리를 용이하게 만든다. 또한 유니티(Unity) 엔진의 애니메이터 시스템과 쉽게 연동되어, 각 상태에 맞는 애니메이션을 자연스럽게 제어할 수 있다 [2].

본 연구는 FSM 기반 NPC의 성능을 정량적으로 분석하는 것을 목표로 한다. NPC 수를 1개, 10개, 100개, 1000개로 점진적으로 늘려가며 FPS 성능을 측정했다. 이를 통해 FSM 방식이 다수의 NPC를 제어하는 환경에서 얼마나 효율적이고 안정적인지 입증하고자 한다.

2. 연구 내용

본 연구는 FSM(유한 상태 기계)을 기반으로 한 NPC 인공지능 구조를 제안한다. 개발 환경은 Unity 2022.3.25f1이며, C#을 사용해 상태별 행동과 전이 조건을 모듈화했다. NPC는 'Chase Fast', 'Chase Slow', 'Attack'의 세 가지 상태를 가지며, 각 상태는 지정된 애니메이션과 행동 로직을 수행한다. 상태가 변화하는 핵심 기준은 NPC와 플레이어 간의 거리이다. 이 상태 변화를 위한 세부 알고리즘은 표 1과 같다.

표 1. FSM 상태 전이 의사코드

```
procedure Update_State(E, P):
    prevState = E.state
    dist = Distance(E.pos, P.pos)

    if dist > E.sightRange then
        E.state = CHASE_FAST
    else if dist > E.attackRange then
        E.state = CHASE_SLOW
    else
        E.state = ATTACK
    end if

    if E.state != prevState then
        Set_Color(E, E.state)
    end if
end procedure
```

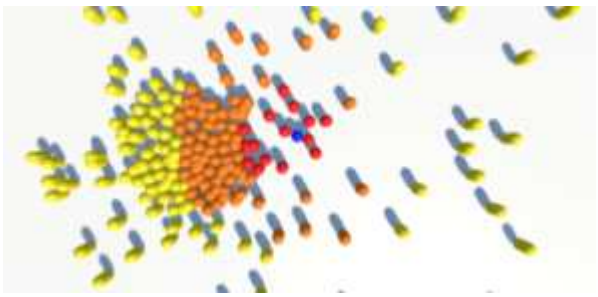

그림 1. FSM 기반 NPC 군집 제어 실험의 렌더링 결과: 파란색 객체는 플레이어를 나타내며, 노란색은 Chase Fast, 주황색은 Chase Slow, 빨간색은 Attack 상태.

그림 1은 FSM 기반 NPC 군집 제어 실험의 렌더링 결

과를 보인다. NPC의 상태는 색상으로 시각화하여, 'Chase Fast'는 노란색, 'Chase Slow'는 주황색, 'Attack'은 빨간색으로 표시된다. 실험 조건으로 플레이어는 x축 양의 방향으로 초당 5 unit의 일정한 속도로 이동하며, 이 과정에서 NPC 군집의 행동 변화 및 시스템 성능을 측정한다.

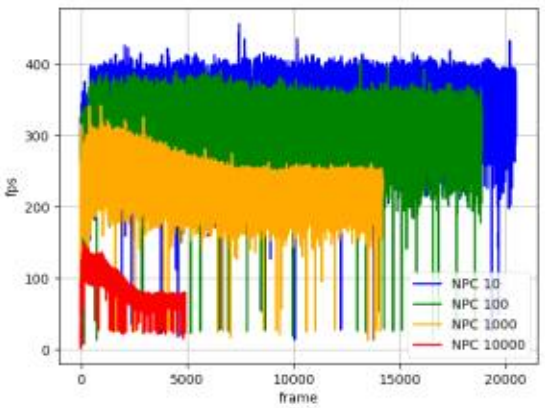

그림 2. NPC 수에 따른 FPS 변화: NPC 개수를 1개, 10개, 100개, 1000개로 달리하며 60초간 측정한 결과

실험 환경은 AMD Ryzen 7 5800H CPU, RTX 3060 Laptop GPU, 16GB RAM으로 구성했다. 그림 2는 FSM 기반 NPC의 수를 1개, 10개, 100개, 1000개로 점진적으로 증가시키면서 각 단계별 FPS를 측정한 결과이다. 실험 분석 결과, FSM 기반 NPC의 수가 100개 이하일 때는 시스템 성능에 큰 영향을 미치지 않는 것을 확인할 수 있었다.

3. 기대효과 및 활용

본 연구에서 제안하는 유니티 엔진과 FSM 기반의 NPC 시스템은 다양한 게임 환경에서 다수의 NPC가 안정적으로 동작하는 기반을 제공한다. 이를 통해 게임의 전반적인 몰입감과 완성도를 크게 향상시킬 수 있다. 또한, FSM의 직관적인 구조는 개발 및 유지보수의 효율성을 높여준다. 결과적으로 새로운 행동 패턴의 추가나 기존 로직의 수정이 용이해지며, 성능 저하 없이 예측 가능한 NPC 행동을 구현하여 플레이어에게 일관된 게임 경험을 제공할 수 있을 것으로 기대된다.

4. 결론

FSM 기반 NPC 인공지능 구조는 행동 트리나 머신러닝 방식에 비해 성능과 개발 효율성 측면에서 뚜렷한 강점을 보인다. 하지만 상태와 전이가 많아질수록 구조가 복잡해지고, NPC의 행동 패턴이 단조롭게 예측된다는 명확한 한계 또한 존재한다.

따라서 향후에는 이러한 단점을 보완하고 더 자연스러우면서도 복잡한 행동을 표현하는 NPC를 구현하기 위한 연구가 필요하다. 구체적인 방안으로는 FSM과 행동 트리, 머신러닝 모듈 등을 결합한 하이브리드 구조를 도입하거나, 계층적 FSM(HFSM)을 적용하여 복잡도를 효과적으로 관리하는 구조적 개선 연구가 요구된다.

감사의 글

“본 연구는 2025년 과학기술정보통신부 및 정보통신기획평가원의 SW중심대학사업의 연구 결과로 수행되었음“(2021-0-01399)

참고 문헌

[1] 김동현, 김경태, "게임 인공지능의 발전 동향과 적용 기술," 한국콘텐츠학회논문지, 2018, 18(1), pp. 1-11.

[2] 이면재. "FSM 기반의 게임 NPC 인공 지능 평가." 한국게임학회 논문지 14.5 (2014): 127-135.