

<https://doi.org/10.7236/JIIBC.2024.24.2.113>  
JIIBC 2024-2-17

## 지능형 NPC의 행동 메커니즘에 따른 계층적 유한 상태 기계와 행동 트리의 효율성 평가

### Efficiency Evaluation of Hierarchical Finite-State Machines and Behavior Trees according to Behavior Mechanism of Intelligent NPCs

이정민\*, 김정이\*\*

Jung-Min Lee\*, Jung-Yi Kim\*\*

**요약** 본 연구에서는 지능형 NPC의 행동을 효과적으로 구현하기 위한 두 가지 주요 구조, 계층적 유한 상태 기계(Hierarchical Finite State Machine, HFSM)와 행동 트리(Behavior Tree)의 구조를 설계하고 실험용 게임을 제작하여 비교 및 분석하였다. 계층적 유한 상태 기계는 복잡한 상호작용 중심의 동작에 적합하여 상태 변화와 그 전이가 중요한 행동 메커니즘에서 효율적이고, 행동 트리는 동적 환경에서의 수정과 확장이 용이하여 다양한 조건 하에서의 동적 반응이 중요한 행동 메커니즘에서 효율적이라는 분석 결과를 도출하였다. 이 두 구조를 유니티 엔진을 활용하여 실험적으로 적용하고 그 효율성을 검증하였다. 본 연구는 기본적인 구조 설계에 초점을 맞췄으며, 추후 개발 예정인 액션 어드벤처 탈출 게임에 이 구조를 적용할 예정이다. 본 연구의 결과는 게임 개발자들이 지능형 NPC를 효율적으로 구현하는데 도움을 주며, 게임의 퀄리티와 플레이어 만족도 향상에 기여할 것으로 기대된다.

**Abstract** In this study, we designed and analyzed two main structures for effectively implementing the behavior of intelligent NPCs the Hierarchical Finite State Machine (HFSM) and the Behavior Tree, by creating experimental games. The HFSM was found to be efficient for complex interaction-centered actions where state changes and transitions are crucial, while the Behavior Tree was effective in dynamic environments where ease of modification and expansion are required for dynamic responses under various conditions. These structures were experimentally applied using the Unity engine to verify their efficiency. This study focused on the basic structure design and plans to apply these structures to an upcoming action-adventure escape game. The results of this research are expected to assist game developers in efficiently implementing intelligent NPCs, thereby contributing to the improvement of game quality and player satisfaction.

**Key Words** : Behavior Tree, Hierarchical Finite State Machine, Intelligent NPCs, Structure Design

\*학생회원, 성결대학교 미디어소프트웨어학과  
\*\*정회원, 성결대학교 미디어소프트웨어학과 조교수  
접수일자 2024년 2월 8일, 수정완료 2024년 3월 8일  
게재확정일자 2024년 4월 5일

Received: 8 February, 2024 / Revised: 8 March, 2024 /  
Accepted: 5 April, 2024  
Corresponding Author: ecesss@sungkyul.ac.kr  
Dept of Media Software, Sungkyul Univ., Korea

## I. 서 론

최근 몇 년 동안 비디오 게임 분야에서 인공지능 기술의 발전이 두드러지고 있으며, 특히 GPT와 같은 언어 모델을 활용한 인공지능은 큰 주목을 받고 있다. 이러한 언어 모델 기반 인공지능의 발전은 게임 산업에만 국한되지 않고, 메타버스(Metaverse) 안에서 플레이어와 자연스러운 상호작용을 하기 위한 NPC(Non-Player Character, 비플레이어 캐릭터)[1]와 플레이어가 원하는 시나리오로 웹드라마를 제작하는[2] 것과 같은 창의적인 분야에서도 새로운 가능성을 탐색하는 데 큰 기여를 하고 있다. 사용자들은 이러한 기술의 발전 덕분에 훨씬 더 정교하고 재미있는 플레이 경험을 누릴 수 있게 되었다. 이와 같은 기술 발전의 핵심에는 강화학습을 통한 인공지능의 발전도 있지만[3], 인공지능의 행동 상태를 정의하기 위한 지능형 NPC(Intelligent Non-Player Character, 지능형 비플레이어 캐릭터)가 필요하다. 지능형 NPC는 복잡한 인간의 행동을 모방하여 게임 내에서 다양한 반응과 행동들로 플레이어의 몰입도를 향상하는 역할을 한다.

Unity의 ML-Agent를 사용하여 목표 추적 AI를 개발하고, 강화학습으로 다양한 환경에서의 효율성을 검증하고 몰입도를 높일 수 있는 방법론[4]과 NavMesh 기능과 헤비사이드 계단 함수를 통해 가중치를 계산하여 NPC를 제어하는 기법[5] 등 게임 내 다양한 상황에 유연하게 대응할 수 있는 지능형 NPC의 구현은 게임 개발의 중요한 연구 주제가 되었다.

본 연구는 지능형 NPC의 행동을 효과적으로 구현하기 위해 계층적 유한 상태 기계(Hierarchical Finite State Machine, HFSM)와 행동 트리(Behavior Tree)의 설계와 검증을 목표로 한다. 연구 방법은 다음과 같다. 첫째, HFSM과 행동 트리의 구조 설계 후, 유니티 엔진을 사용하여 PC 콘솔 게임 기반의 실험용 게임을 개발한다. 둘째, HFSM은 상태와 전환을 기반으로 한 계층적 구조에, 행동 트리는 조건 반환 결과에 따른 나무 형식 구조에 초점을 맞춰 NPC를 설계 및 개발한다. 셋째, 두 구조의 효율성과 적용성을 비교 분석한다. 이 과정에서 유니티 엔진의 네비게이션 메쉬 기능을 활용하는데 이는, 넓은 가상공간에서 길을 찾는 것은 어렵기 때문에 지형지물의 메쉬 형태의 데이터를 생성하여 NPC가 이동할 수 있는 구역을 시각적으로 나타내주는 기능이다. 이를 활용해 NPC의 효과적인 길 찾기 시스템을 개발하고, 실험용 게임에 적용해 각 구조의 성능을 평가함으로써 지능형 NPC의 움직임 제어에 관한 효율적인 구조를 탐구

하고 분석하고자 한다.

## II. 문헌 고찰

### 1. 유한 상태 기계(FSM)와 계층적 유한 상태 기계(HFSM)

유한 상태 기계(Finite State Machines, FSM)는 각 상태와 그 상태의 전환을 결정하는 외부 이벤트를 정의하여 상태 전이를 구현하는 기법이다. FSM 기법을 적용한 지능형 NPC의 행동에 대한 상태는 한 개만 유지할 수 있고 행동 메커니즘에 따라 그 상태에 맞는 이벤트나 상황이 발생했을 때 다른 상태로 전이하여 일을 수행하는 특징을 가진다[6]. 하지만, FSM은 구조가 간단하고 단순하여 소규모 프로젝트의 모델링 및 분석에 적합하나, 프로젝트 규모가 커질 경우 시스템의 복잡도가 급격히 증가하는 문제점이 있다[7].

이러한 제약을 극복하기 위해 문제점들로 인해 계층적 유한 상태 기계(Hierarchical Finite State Machine, HFSM)가 도입되었다. 계층적 유한 상태 기계는 지능형 NPC의 특정 조건이나 전환 구조의 흐름을 그룹화하여 조건과 전이 및 전환을 반복 사용하지 않아 상태 전이의 재사용을 높이고 코드의 중복성을 줄이는 구조로 발전하였다[8]. 즉, 지능형 NPC의 행동 상태가 여러 구조에 걸쳐 결합한 계층구조를 가지고 다른 상태들을 포괄하는 상위상태로 구조화하여 기존 유한 상태 기계와는 다르게 코드의 재사용이 쉽고 용이하다[9]. 그러나 계층적 유한 상태 기계는 확장성을 가지고 있지만, 하위상태를 구조화하는 방식이 모호하다는 단점이 있다. 특정 상태에서 전이가 되어야 하는 조건이 필요해 같은 상태를 다른 구조에서는 사용할 수 없다[10].

### 2. 행동 트리(Behavior Tree)

행동 트리 방식(Behavior Tree)는 지능형 NPC를 위한 그래픽 기반 표현 방식으로, 다양한 조건과 행동을 노드 형태로 연결하여 표현한다. 지능형 NPC의 행동에 대한 조건과 노드의 우선순위 등 관련된 노드들의 연결로 완성된다[11]. 각 노드는 지능형 NPC의 행동 메커니즘에 따른 상태 우선순위에 따라 노드의 우선순위가 정해진다. 트리에서 상위 우선순위 노드의 조건이 만족할 경우 하위 우선순위의 노드의 동작을 중지하고 해당 노드의 동작을 시작한다[12]. 이러한 행동 트리를 구성하는

노드의 다양성을 이용하여 개발자들이 추가적인 파생 노드들을 생성 및 제작할 수 있어 확장성이 좋다는 장점이 있다. 또한, 가시적인 트리의 형태로 표현할 수 있고 직관성이 좋아 복잡한 지능형 NPC의 시스템 설계 및 구현에 적합하다[13].

### III. 설계 및 구현

#### 1. 계층적 유한 상태 기계(HFSM) 구조 설계

본 연구에서는 두 개의 FSM으로 구성되어 계층적으로 표현한 기본적인 HFSM 구조로 설계되었다. 이는 주요 동작을 정의하는 Root FSM과 전투 동작을 상세하게 나타내는 Fight FSM, 두 개의 FSM으로 구성된다. Root FSM에는 순찰(Patrol), 추적(Chase), 전투(Fight), 검색(Search)의 네 가지 주요 상태가 있으며, Fight 상태 내에는 추가로 전투 동작을 세분화하는 Wait, Telegraph, Hit의 세 가지 상태를 포함하는 Fight FSM이 있다. 이 구조는 지능형 NPC가 플레이어를 순찰, 추적, 전투, 플레이어를 놓쳤을 때 탐색하는 동작을 효과적으로 수행하게 되어있다.

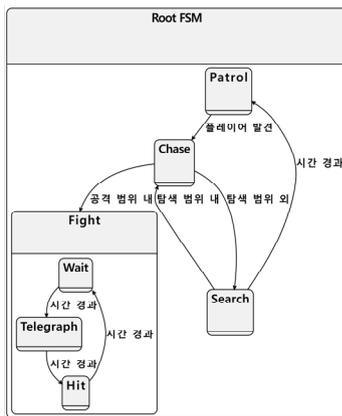


그림 1. 지능형 NPC의 계층적 유한 상태 기계 구조 설계  
 Fig. 1. Hierarchical Finite State Machine Structure Design for Intelligent NPCs

#### 2. 행동 트리(Behavior Tree) 구조 설계

본 연구에서 설계된 행동 트리 구조는 대기 노드와 추적 노드에 대한 하위 액션 노드들로 구성되어 있다. 행동 트리에서 노드의 실행 순서는 위에서 아래로, 왼쪽에서 오른쪽으로 실행된다. 시작점인 Base Logic 선택자 노드는 자식 노드가 참을 반환할 때까지 여러 자식 노드를

실행한다. Can Chase 조건 노드는 Success와 Failed를 반환하여 타겟 추적 가능성을 판단하며, 타겟을 감지하면 Chase Target 액션 노드가 실행되어 NPC가 타겟을 추적한다. 타겟을 놓치거나 없으면, Wander 시퀀스 노드가 실행되어 NPC가 무작위 위치로 이동한다. 결과적으로, 이 구조는 캐릭터가 주변에 추적할 타겟이 있는 경우 추적하고, 그렇지 않은 경우 주변을 무작위로 탐색하는 로직을 나타낸다.

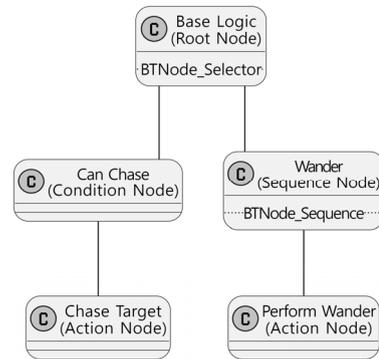


그림 2. 지능형 NPC의 행동 트리 구조 설계  
 Fig. 2. Behavior Tree Structure Design for Intelligent NPCs

### IV. 실험 및 비교 분석

#### 1. 실험 방법

본 연구에서는 네비게이션 매쉬를 이용하여 지능형 NPC가 플레이어를 추적하기 위한 시스템을 구축하였고, 계층적 유한 상태 기계와 행동 트리 두 가지 구조를 통해 지능형 NPC 동작을 설계하고, 유니티 엔진을 활용한 실험용 게임에서 구현 및 비교 분석하였다. HFSM 실험에서는 플레이어와의 상호작용 중심의 동작, 특히 발견, 추적, 공격 등의 주요 상태 전환을 중점적으로 관찰했으며, 행동 트리 실험에서는 부모-자식 노드 형태로 구성된 복잡한 행동 패턴과 조건 기반의 동작 판단을 중심으로 분석하였다.

#### 2. 실험용 게임 - 계층적 유한 상태 기계(HFSM)

계층적 유한 상태 기계 실험에서 기대하는 지능형 NPC의 행동 변화는 플레이어와의 다양한 상호작용을 중심으로 한 다양한 반응을 보여야 한다. 플레이어 발견 시 추적을 시작하고, 일정 거리 내 접근하면 공격을 초기화

하는 등의 반응이 포함된다. NPC는 순찰 중 플레이어를 발견하면 추적을 시작하고, 근접 시 공격하며, 플레이어를 잃어버린 경우 마지막 위치에서 주변 탐색을 진행한다. 이러한 동작은 유니티의 StateMachine을 이용해 여러 상태(Patrol, Chase, Fight, Search)로 구성되며, 플레이어 위치 및 NPC와의 거리에 따라 상태 전환이 이루어진다. Patrol 상태에서는 지정된 경로 순찰, Chase 상태에서는 플레이어 추적, Fight 상태에서는 공격 동작을 세부적으로 구현하며, Search 상태에서는 플레이어를 찾지 못했을 때 무작위 탐색을 한다. 이 구조는 지능형 NPC의 동작을 다양하고 유동적으로 제어할 수 있다.



그림 3. HFSM 구조 실험용 게임 화면  
Fig. 3. Game Screen for HFSM Structure Experiment

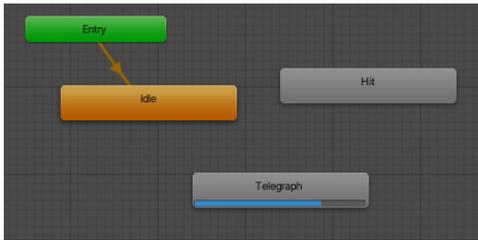


그림 4. HFSM 구조 실험용 게임의 상태 변환 화면  
Fig. 4. State Transition Screen of the Game for HFSM Structure Experiment

### 3. 실험용 게임 - 행동 트리(Behavior Tree)

행동 트리 실험에서 기대하는 지능형 NPC의 행동 변화는 부모 노드에서 하위 노드로의 이동을 위한 참, 거짓 판별 및 해당 노드들의 액션 노드로의 전이가 정상적으로 되며 행동 상태를 동작하는 것이다. 각 노드는 특정한 행동을 나타내며, 상위 노드의 행동이 하위 노드들의 행동으로 세분화되어 구성되었다. 이러한 방식으로, 노드는 나무의 형태처럼 서로 연결되어 복잡한 행동 패턴을 모델링하게 된다. 또한, 시야각 기법을 이용하여 플레

어의 시야를 판정하는 범위를 정의하였다. 이를 통해, 지능형 NPC는 플레이어와의 상대적 위치와 거리를 기반으로 자신의 시야 내에 플레이어가 있는지를 판단하게 된다. 이러한 판정 범위는 더욱 자연스러운 움직임과 반응을 구현하는 데 큰 도움이 된다. 지능형 NPC는 행동 트리 내의 노드들을 순회하며 각 노드에 정의된 조건이나 행동을 수행하게 된다. 각 노드는 수행 결과에 따라 Success 또는 Failed 상태를 반환하며, 이 반환 값은 다음 행동의 결정에 영향을 미치게 된다.

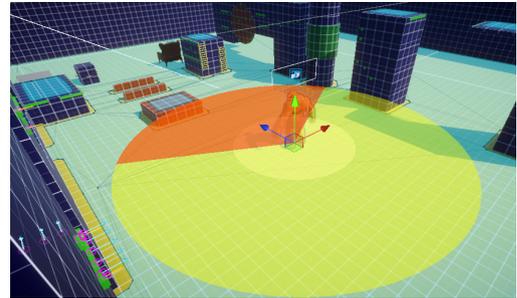


그림 5. 행동 트리 구조 실험용 게임 화면  
Fig. 5. Game Screen for Behavior Tree Structure Experiment

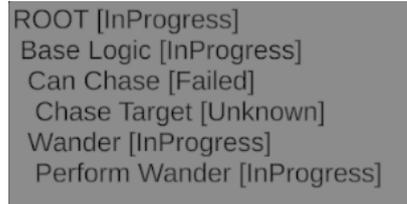


그림 6. 행동 트리 구조 노드의 상태  
Fig. 6. Node State for Behavior Tree Structure Experiment

### 4. 실험 결과 및 비교 분석

본 연구의 목적은 지능형 NPC의 동작을 계층적 유한 상태 기계(HFSM)와 행동 트리(Behavior Tree)를 이용하여 모델링하고 이를 비교하는 것이었다. 유니티 엔진을 활용해 두 구조를 실험하며 다음과 같은 주요 결과 및 분석을 얻었다.

HFSM은 상위 및 하위 동작을 구분하는 데 강점을 보였으며, 계층적 구조를 통해 복잡한 동작 패턴을 명확하게 표현하여 하위수준의 동작 흐름과 하위 수준의 세부 동작을 명확히 분리하게 했다. 반면, 행동 트리는 부모-자식 노드 구조를 통해 우선순위나 조건에 따른 행동을 명확하게 표현하는 데 유용했으며, NPC의 의사 결정 과

정과 노드들을 시각적으로 설계할 수 있는 장점이 있었다. 또한, 노드 간의 명확한 관계와 조건은 빠른 수정과 확장을 가능하게 했다. 이는 실시간 NPC 행동 조절과 다양한 조건에 대응하는 능력을 향상시켰다. 따라서, 프로젝트의 특성과 요구사항에 따라 HFSM이나 행동 트리 중 적합한 모델링 기법을 선택하는 것이 중요하다는 결론을 얻었다. 복잡한 동작을 모델링할 때는 HFSM이, 동적 환경에서의 빠른 수정과 확장을 원할 때는 행동 트리가 더 적합할 수 있다.

**표 1. HFSM의 실험 분석 결과**  
**Table 1. Experimental Analysis Results of HFSM**

항목	HFSM (계층적 유한 상태 기계)
정의	상태 간의 전이를 계층적으로 모델링한 구조
특징	상태 간 전이의 명확성 계층적 분리로 복잡성 관리
적용 시스템	상태 변화와 그 전이가 중요한 NPC 시스템 계층적인 동작 분리 필요한 시스템
장점	계층적 구조로 복잡한 동작 모델링 용이 상태 전이의 명확성 제공

**표 2. BT의 실험 분석 결과**  
**Table 2. Experimental Analysis Results of BT**

항목	Behavior Tree (행동 트리)
정의	조건과 행동을 나무 형태의 노드로 구성한 구조
특징	노드 기반의 유연성 조건에 따른 동적 반응
적용 시스템	다양한 조건 하에서의 동적 반응이 중요한 NPC 시스템 빠른 수정 및 확장이 필요한 시스템
장점	동적 환경에 대한 높은 유연성 모듈식 노드로 쉽게 확장 및 수정 가능

## V. 결 론

비디오 게임 내 지능형 NPC는 게임 경험을 향상하는 중요한 역할을 한다. 따라서, 본 연구에서는 지능형 NPC의 행동을 효과적으로 구현하기 위한 두 가지 주요 구조 계층적 유한 상태 기계(Hierarchical Finite State Machine)와 행동 트리(Behavior Tree)를 설계하고 각 구조의 성능과 효율성을 비교 및 분석에 중점을 두었다. 계층적 유한 상태 기계는 상태 중심의 구조를 확장하여 코드의 재사용성을 높이는 반면, 행동 트리는 노드 기반의 구조로 확장성과 직관성을 갖는다. 즉, 계층적 유한 상태 기계는 복잡한 상호작용 중심의 동작에 적합하여 상태 변화와 그 전이가 중요한 행동 메커니즘에서 효율

적이고, 행동 트리는 동적 환경에서의 수정과 확장이 용이하여 다양한 조건 하에서의 동적 반응이 중요한 행동 메커니즘에서 효율적이라는 분석 결과를 도출하였다.

그 결과, 두 기법 모두 각자의 장점을 지니며, 각각의 상태가 자연스럽게 전이되어 게임 개발 상황에 따라 최적화된 기법을 선택할 수 있음을 확인하였다. 특히, 본 연구에서 사용된 실험 환경은 이러한 결론을 뒷받침하는 근거로 제공된다. 따라서, 게임 개발자나 연구자들은 이 연구의 결과를 기반으로 지능형 NPC의 동작을 효율적으로 모델링하는 데 필요한 기법을 선택할 수 있을 것이다.

하지만 본 연구에는 아직 지능형 NPC의 행동을 구현한 상태 및 행동 노드들이 기본적인 구조들로만 설계하였다는 한계가 있으므로 향후 연구에서는 더욱 다양한 상태 전이와 행동 노드들로 구성하여 복잡한 환경에서의 움직임을 구현해 추가적인 분석을 할 것이다. 또한, 두 가지 기법에 초점을 맞추어 분석하였기에, 다른 인공지능 기술들과의 통합 및 비교는 제한적이었고, 구체적인 게임 시나리오나 다양한 상호작용 따른 분석은 이루어지지 않았다. 이러한 한계점을 극복하기 위한 추가 연구가 필요하다. 향후 연구를 통해 기대되는 결과는 지능형 NPC의 더욱 다양하고 복잡한 행동 패턴 구현이며, 이를 통해 플레이어의 게임 경험을 향상하는 데 큰 기여를 할 것으로 기대된다. 더 나아가, 추후 개발 예정인 액션 어드벤처 탈출 게임에서 지능형 NPC의 동적 행동을 구현하기 위해 적용하고, 특정 상황에 맞게 해당 메커니즘에 맞는 구조를 적용해 보다 효율적인 지능형 NPC의 움직임의 구현을 시도할 예정이다.

본 연구와 그 결과물인 두 구조에 대한 비교 및 효율성 평가의 의미는, 게임 개발자들이 지능형 NPC를 구현할 때 효율적인 방법을 선택할 수 있게 한다는 점에서 찾을 수 있다. 또한, 게임 인공지능 분야에서의 기술적인 발전을 바탕으로, 플레이어들에게 더욱 실감 나고 동적인 게임 환경을 제공함으로써, 전반적인 게임의 퀄리티와 플레이어의 만족도를 향상할 수 있는 자료로서의 의미가 있다. 이 연구는 비디오 게임 산업에서의 지능형 NPC의 중요성과 그 구현 방법을 이해하여, 유사한 기능을 설계, 개발하는 게임 개발자들에게 도움이 될 것으로 예상된다.

## References

- [1] Shi-Jie Sun, Seung-Keun Song, "A Study on NPC Using

- ChatGpt in the Metaverse: Focusing on Facerig," Proceedings of the Comprehensive Korea Contents Association Conference, pp. 359-360. May 2023.
- [2] Hyun-Su Lee, Jung-Yi Kim, "Development of Story Recommendation through Character Web Drama Cliché Analysis", The Journal of The Institute of Internet, Broadcasting and Communication(JIIBC), Vol. 23, No. 4, pp. 17-22, Aug 2023.  
DOI: <https://doi.org/10.7236/JIIBC.2023.23.4.17>
- [3] Hyun-Su Kim, "Development of Reinforcement Learning Environment for Smart Base Isolation Control System using Unity", Journal of the Korea Academia-Industrial cooperation Society(JKAIS), Vol. 24, No. 5 pp. 433-440. May 2023.  
DOI: <https://doi.org/10.5762/KAIS.2023.24.5.433>
- [4] Deok-Hyung Kim, Hyun-Jun Jung, "Performance Analysis of Target Tracking AI based on Unity ML-Agents", The Journal of KIIT, Vol. 19, No. 12, pp. 19-26, Dec 2023.  
DOI: <https://doi.org/10.14801/jkiit.2021.19.12.19>
- [5] Jong-Hyun Kim, Soo-Kyun Kim, "Efficient Path Tracking of Non-Player Character with Controlling NavMesh Based on Smoothed Heaviside Step Function", Proceedings of the Korean Society of Computer Information Conference, Vol. 30, No. 1, pp. 339-340, Jan 2022.
- [6] Ho-Ryul Kim, Jong-Ho Yoo, Chang-Min Han, Syoung-Og An, Soo-Kyun Kim, "Development of Mobile RPG using Unity 3D Engine", in Proceedings of the Korean Society of Computer Information Conference, Vol. 27, No. 2, pp. 81-82, Jul 2019.
- [7] Woo-Jin Lee, "Compositional Safety Analysis for Embedded Systems using the FSM Behavioral Equivalence Algorithm", The KIPS Transactions : Part D, Vol. 14, No. 6, pp. 633-640, Oct 2007.  
DOI: <https://doi.org/10.3745/KIPSTD.2007.14-D.6.633>
- [8] Jun-O Choi and Hyun-Joo Park, "DEVELOPMENT OF GAME AI USING BEHAVIOR TREE : BASED ON THE UNITY5 ENGINE", Proceedings of the Korea Entertainment Industry Association Conference, pp. 9-17, Jul 2018.
- [9] Kyeong-Jin Baek, "Design and Implementation of Artificial Intelligence Agent for Real-Time Simulation Football Game in a Mobile Environment", Proceedings of the Korea Information Processing Society Conference, Vol. 23, No. 1, pp. 636-639, Apr 2016.  
DOI: <https://doi.org/10.3745/PKIPS.y2016m04a.636>
- [10] Jae-Won Jo, Jung-Won Bang, "A study on The Implementation of Monster AI using Finite-State Machine", Proceedings of the Korean Society of Computer Information Conference, Vol. 27, No. 1, pp. 349-350, Jan 2019
- [11] Myoun-Jae Lee, "A Proposal on Game Engine Behavior Tree", Journal of Digital Convergence, Vol. 14, No. 8, pp. 415-421, Aug 2016.  
DOI: <https://doi.org/10.14400/JDC.2016.14.8.415>
- [12] Sung-Jin Bae and Myung-Ju Kang, "Design And Development of Game AI Using Unreal Engine 4 Behavior Tree", Proceedings of the Korea Information Processing Society Conference, Vol. 24, No. 1, pp. 267-269, Jan 2016
- [13] Min-Ji Kwon, Jin-Sek Seo, "Game AI Agents using Deliberative Behavior Tree based on Utility Theory", Journal of Korea Multimedia Society, Vol. 25, No. 2, pp. 432-439, Feb 2022.  
DOI: <https://doi.org/10.9717/kmms.2022.25.2.432>

## 저 자 소 개

### 이 정 민(학생회원)



- 2018년 3월 ~ 현재 : 성결대학교 미디어소프트웨어학과(재학)
- 관심분야 : 게임 프로그래밍, 인공지능

### 김 정 이(정회원)



- 2012년 2월 : 이화여자대학교 대학원 디지털미디어학부(박사)
- 현재 : 성결대학교 미디어소프트웨어학과 조교수
- 관심분야 : UX Design