

# 사이드 뷰 2D 게임에서의 고수준 길찾기 알고리즘에 대한 연구

유환진<sup>o</sup>, 이범로<sup>\*</sup>

<sup>\*</sup>청강문화산업대학교 게임콘텐츠스쿨,

<sup>o</sup>청강문화산업대학교 게임콘텐츠스쿨

e-mail: meiroo2@naver.com<sup>o</sup>, redcom@ck.ac.kr<sup>\*</sup>

## A Study on High-level Pathfinding Algorithm in Side View 2D Games

Hwan-Jin You<sup>o</sup>, Bum-Ro Lee<sup>\*</sup>

<sup>\*</sup>School of Game, Chungkang College of Cultural Industries,

<sup>o</sup>School of Game, Chungkang College of Cultural Industries

### ● 요약 ●

게임 플랫폼의 고도화와 게임 구현 기술의 발전을 통해 이루어지고 있는 개발 환경의 효율성 제고를 통해 이전에 게임의 구현 자체에 소요하는 시간을 게임 콘텐츠의 질적 향상을 위해 투입할 수 있는 여건이 형성되고 있다. 본 논문은 게임에서 등장하는 NPC가 보다 사실적인 추론을 통해 플레이어를 추적하도록 지정하는 알고리즘을 고수준의 카테고리로 구분하여 구현하여, 개발 과정에서의 NPC 구현에 효율성을 제고하는 기법을 제안한다. NPC가 플레이어를 추적하는 스토리를 기반으로 하는 게임에서 NPC가 플레이어의 위치에 도달하기 위한 길찾기 알고리즘은 필수적이다. 일반적으로 사용되는 최단 거리 길찾기 알고리즘인 다익스트라(Dijkstra)의 경우, 효율적이지만 플레이어가 NPC의 경로를 예측 가능할 수 있기에 게임의 재미 향상에 걸림돌이 되는 단점이 있다. 본 논문에서는 최단 거리 길찾기를 구현하면서 다양한 게임 스토리 상의 설정으로 지정할 수 있는 다양한 방법론을 연구하였으며, 이를 구현하기 위한 고수준 길찾기에 대한 방법을 제시한다.

**키워드:** 게임(Game), 사이드 뷰(Side View), 길찾기(Pathfinding), 인공지능(A.I.)

## I. Introduction

지금의 게임 산업의 제작 환경은 게임 플랫폼의 고도화와 게임 구현 기술의 발전을 통해 이루어지고 있는 개발 환경의 효율성 제고를 통해 이전에 게임의 구현 자체에 소요하는 시간을 게임 콘텐츠의 질적 향상을 위해 투입할 수 있는 여건이 형성되고 있으며, 이를 통해 개발자들이 게임의 게임성 자체에 집중할 수 있는 환경이 주어지고 있다. 또한 자동화할 수 있는 부분을 기술적 적용을 통해 가능한 시스템이 전담하도록 하는 노력을 통해 개발 기간을 단축하고 개발자의 시행 착오를 줄여 비용면에서도 개발 효율성을 극대화 하는 방향으로 발전하고 있다. 본 논문은 게임에서 등장하는 NPC가 보다 다양한 조건에 대한 사실적인 추론을 통해 플레이어를 추적하도록 지정하는 알고리즘을 고수준의 카테고리로 구분하여 구현하여, 개발 과정에서의 NPC 구현에 효율성을 제고 하는 기법을 제안한다. NPC가 플레이어를 추적하는 스토리를 기반으로 하는 게임에서 NPC가 플레이어의 위치에 도달하기 위한 길찾기 알고리즘은 필수적이다. 일반적으로

사용되는 최단 거리 길찾기 알고리즘인 다익스트라(Dijkstra)의 경우, 효율적이지만 플레이어가 NPC의 경로를 예측 가능할 수 있기에 게임의 재미 요소를 극대화하는 측면에는 걸림돌이 되는 단점이 있다. 본 논문에서는 최단 거리 길찾기를 구현하면서 다양한 게임 스토리상의 설정으로 지정할 수 있는 방법론을 연구하였으며, 이를 활용한 고수준 길찾기에 대한 방법을 제시한다. 본 논문에서는 현재 개발 중인 작품 'Revenant' 프로젝트를 기반으로 테스트를 진행하였다.

## II. Preliminaries

길찾기 알고리즘은 게임에서 매우 빈번하게 사용되는 고전적인 인공지능 기술이며, 이미 많은 연구에서 길찾기 알고리즘 개선에 대한 많은 시도들이 진행되었다. A\*, 다익스트라 알고리즘과 같이

특정 지점에서 다른 지점으로 이동할 때 최단 경로를 탐색하기 위한 많은 방법이 제시되었지만, 사이트 뷰 2D 게임에서 최단 경로 탐색 알고리즘만을 사용하게 되면 플레이어가 해당 경로를 쉽게 예측할 수 있게 되어 더 높은 재미를 끌어내는 데에 한계를 가진다. 본 논문에서는 최단 경로 탐색 알고리즘을 기반으로 하는 다양한 방법을 제시하여 플레이어가 쉽게 해당 경로를 예측하지 못하게 해 게임의 재미를 향상하는 방법론을 제시하는 것에 목적이 있다.

본 논문에서 길찾기 알고리즘을 구현하고 테스트가 이루어지는 환경은 아래 Fig. 1과 같으며, 실험을 효율성과 검증의 편의성을 고려하여 사이트 뷰 2D 게임을 테스트 베드로 선정하였다.

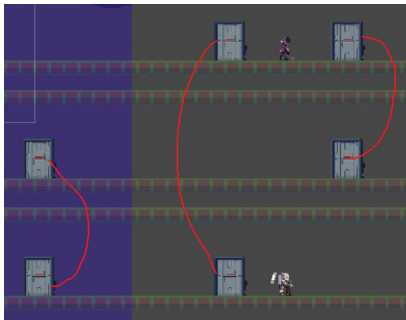


Fig. 1. Path-find verifying environment - Revenant

Fig. 1에서 가장 아래 층에 위치한 캐릭터는 플레이어를 나타내고, 가장 위층에 있는 캐릭터는 NPC를 나타낸다. 각 문은 플레이어나 NPC가 상호작용을 시도하면 해당 상호작용 개체를 연결된 문의 좌표로 이동시키는 역할을 담당한다. 해당 환경에서, 플레이어가 위치한 층에 존재하는 문의 개수는 2개 이상이므로 플레이어에게 도달하는 경로가 여러 가지 도출될 수 있음을 나타낸다. 이를 기반으로 여러 가지 방법론의 적용을 통해 단순한 최단 거리 길찾기 알고리즘뿐만이 아닌, 오락적인 측면을 충분히 고려한 고수준 길찾기에 대한 방법을 제안하고 이를 실제 게임에 적용하여 실증하고자 한다.

### III. The Proposed Scheme

본 논문에서는 크게 기본적인 최단 경로 길찾기, 포위에 의한 길찾기, 그리고 매복을 위한 길찾기를 기본으로 구현하여 2D 사이트 뷰 게임에 적용하여 그 유용성을 검증하였으며, 따라서 경로에 대한 탐색 대상은 2차원 공간으로 제한한다.

#### 1. Shortest Path - 최단경로

본 논문에서는 각 공간을 MapSection, 문을 Section Door라고 정의하였다.

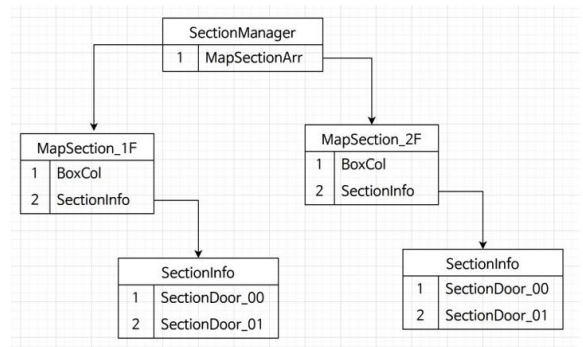


Fig. 2. Class Diagram

이러한 정의를 기반으로 NPC가 플레이어에게 갈 수 있는 다양한 경로를 다음과 같은 알고리즘을 통해 도출하였다.

1. 현재 MapSection에 존재하는 SectionDoor를 파악한다.
2. 이 SectionDoor들 중 하나를 선택하여 List에 삽입 후 건너편 MapSection을 검사한다.
3. 막다른 방일 경우 List에서 제거하고, 아닐 때 해당 방에 있는 SectionDoor를 파악한다.

이를 반복하여 플레이어를 발견하고, 이동 거리에 따라 경로별로 순위를 매긴다. 해당 알고리즘을 토대로 도출된 여러 경로 중, 1순위에 있는 경로가 최단 거리 경로임을 알 수 있다.

#### 2. BackDoor - 포위

도출된 여러 경로를 활용하여 NPC가 플레이어를 포위하도록 길찾기 알고리즘을 수정할 수 있다. 우선 각 NPC의 길찾기 방식을 Enum 타입으로 정의하였으며 이 타입을 기반으로 해당 알고리즘을 적용하도록 설계 하였다.

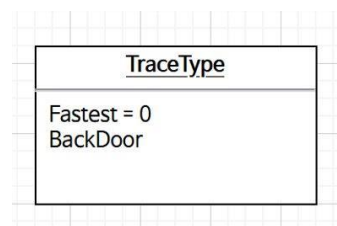


Fig. 3. TraceType

이를 기반으로 각 NPC에게 TraceType를 할당하였다. Fig. 3에 나오는 명칭과 같이 뒷문을 이용해 플레이어를 포위하기 위해서 아래와 같이 뒷문을 정의하였다.

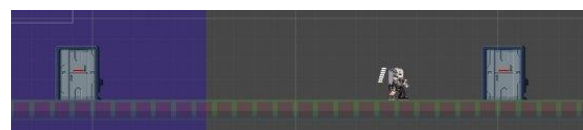


Fig. 4. BackDoor

Fig. 4를 기준으로, 플레이어가 위치한 층에서 플레이어와 가장 멀리 위치한 문의 BackDoor이다. 이를 기반으로 작성된 알고리즘의 순서도는 아래 Fig. 5와 같다.

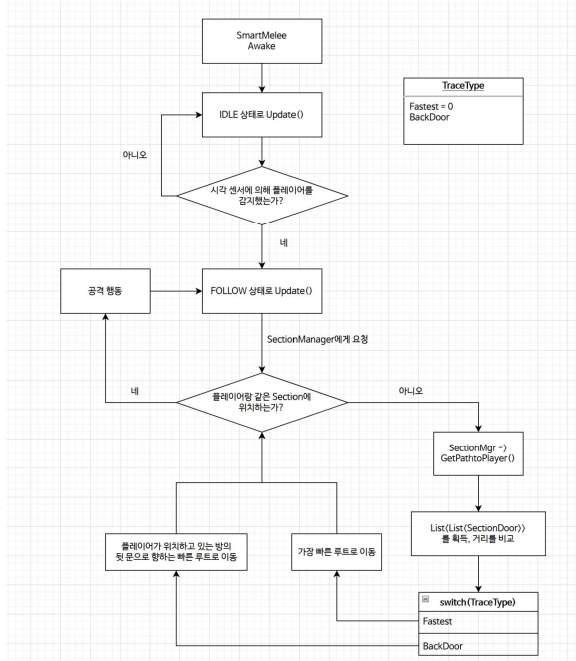


Fig. 5. Shortest Path and Siege flow chart

해당 알고리즘을 토대로 최단 거리 경로 파악을 통한 여러 길찾기 경로 중 BackDoor로 향하는 경로를 선택하여 플레이어를 포위할 수 있음을 알 수 있다. 해당 그림에서 마젠타 색상의 경로는 최단 거리 경로(TraceType-Fastest)를, 연두 색상의 경로는 포위 경로(TraceType-BackDoor)를 나타낸다.

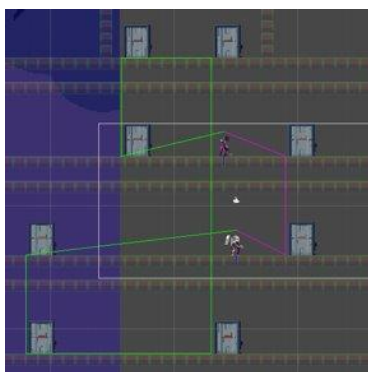


Fig. 6. Hide Type Pathfinding

### 3. Hide - 매복

매복(TraceType-Hide) 방식 길찾기의 경우에는 NPC가 플레이어를 추적할 시, 플레이어를 그대로 따라가는 것이 아닌 특정 MapSection에 대기하고 이동 후, 해당 MapSection에 플레이어가 접근 시 돌격하도록 진행하였다. 이를 구현한 알고리즘을 아래 Fig.

7에 설명하였다.

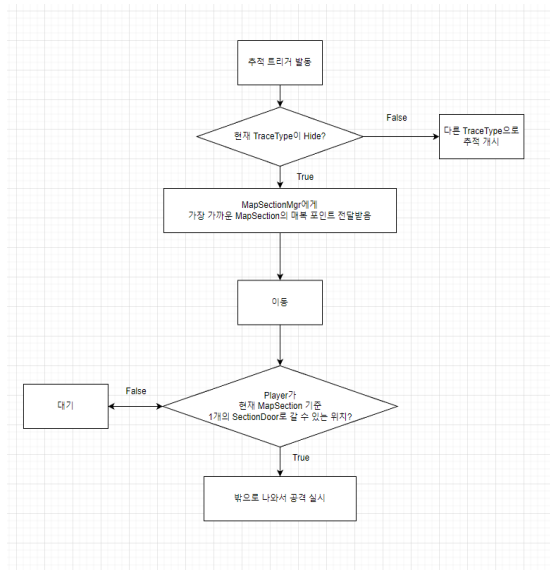


Fig. 7. TraceType-Hide flow chart

구현되는 프레임워크는 추가적으로 정의되어 설계 및 구현되는 다양한 길찾기 기법들을 쉽게 추가하고 관리하기 위해 길찾기 인터페이스를 정의하고 해당 인터페이스와 연결 되는 개별 길찾기 알고리즘 클래스를 통한 구현을 통해 종속성 없이 관리될 수 있도록 구현하였다. 또한 길찾기 알고리즘을 적용하여 테스트를 수행한 샘플 게임의 구현과 알고리즘의 구현은 Unity3D를 통해 구현되었다.

## IV. Conclusions

NPC가 항상 최단 거리로 플레이어를 추적하도록 하는 것은 알고리즘 측면에서 최상의 선택으로 보일지 모르나 게임의 재미 측면에서는 더 나은 방법들이 존재한다. 본 논문에서 제시한 다양한 길찾기 방법들을 이용한다면 플레이어가 NPC의 움직임을 쉽게 예측하지 못하기 때문에 재미 향상에 도움을 줄 수 있을 것이라 예상된다. 추후 과제로서 개별 NPC에게 더 높은 수준의 정교한 지능을 부여하여 NPC가 상황을 판단하고 이를 기반으로 스스로 Trace Type을 유동적으로 변경할 수 있으며, 이러한 선택을 반복적으로 수행함으로써 경쟁학습을 통해 스스로의 판단 능력을 향상시키는 학습형 NPC 모델에 대한 추가 연구를 진행할 예정이다.

## REFERENCES

[1] Jeon Youngjae, and Oh Kyung-soo, "Difficulty Evaluation of Game Levels using A Path-Finding Algorithm" August 2015.  
 [2] Lee Myeonseop, "A Study on Searching a Pass of the

Intelligent Character using Genetic Algorithm“ August. 2009.

- [3] Jong-Hyun, "Efficient Controlling Trajectory of NPC with Accumulation Map based on Path of User and NavMesh in Unity3D", Journal of the Korea Society of Computer and Information ,25(4), 55-61.
- [4] Eunsol Kim, Hyeyeon Kim, Kyeonah Yu, "Implementation of Adaptive Navigation for NPCs in Computer Games", Journal of KIISE,43(2),222-228.
- [5] Kyeonah Yu, Chojung Lee, Inyoung Cho, "Development of a Multi-criteria Pedestrian Pathfinding Algorithm by Perceptron Learning“, Journal of the Korea Society of Computer and Information ,22(12),49-54.