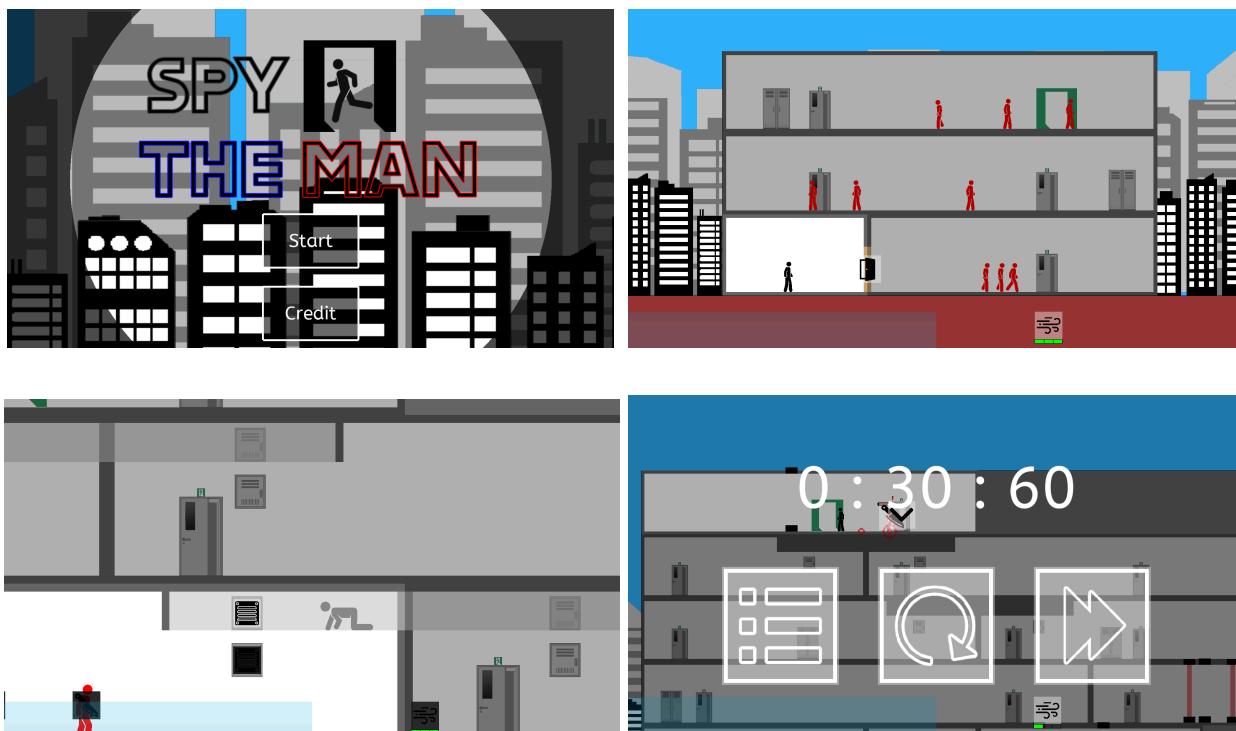

Team Project Morning

Spy The Man

Game Project, Custom Engine Project, C++
2021.9 - 2022.6

"Spy The Man"은 스테이지 형식의 2D 횡스크롤 스파이 액션 게임입니다. 플레이어는 NPC 그리고 맵에 존재하는 오브젝트와의 상호작용을 통해 목표 위치까지 도달하는 구조입니다.



주요 작업

이 프로젝트는 C++ 기반의 간단한 게임 엔진 위에 직접 구현되었으며, 저는 프로그래머로 참여하여 주로 NPC 인공지능 제어, 경로 탐색, 오브젝트 간 상호작용과 같은 시스템 구현을 담당하였습니다.

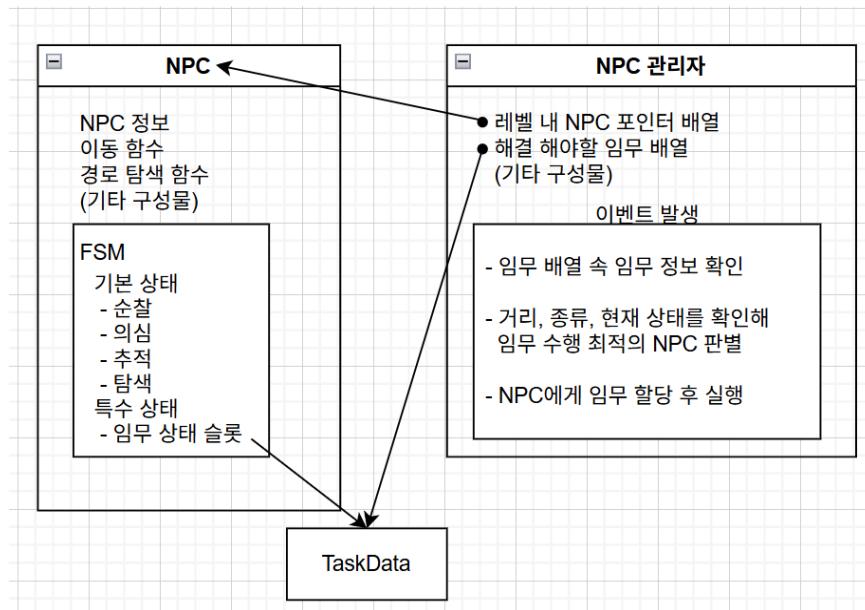
NPC 제어

기존에는 FSM을 사용해, 각 NPC의 행동 흐름을 정적으로 설계하여 제어했습니다. 간단하게, 군인 NPC는 '순찰 → 의심 → 추적 → 탐색 → 순찰'처럼 고정된 상태 흐름을 따르며 동작했습니다. 하지만 게임이 확장되면서 다음과 같이 NPC에게 요구되는 행동이 점점 추가되었습니다.

ex) 플레이어가 경보를 울렸을 때 경보 위치로 이동, 특정 지역에서 침입자가 감지됐을 때 주변 탐색, 다른 NPC의 요청에 응답하여 지원 행동 수행

이처럼 외부의 상황 기반의 행동들이 점차 많아지면서, 참조 관계가 복잡해져 새로운 상태의 구현과 상태 흐름 제어가 기존의 FSM만으로는 이를 유연하게 관리하기 어려워졌습니다.

FSM을 그대로 확장하기보다는 **임무 기반 제어 시스템**으로 구조를 변경했습니다. **외부 이벤트가 발생하지 않았을 때 수행할 기본 상태들 + 임무 수행 상태**로 구성하고, 임무를 할당해줄 관리자 객체를 만들어 임무를 수행할 수 있는 상태의 NPC와 거리와 종류 등 가장 적절한 조건의 NPC를 판별하여 **외부 이벤트가 발생했을 때 임무를 할당해주는** 구조로 변경 하였습니다.

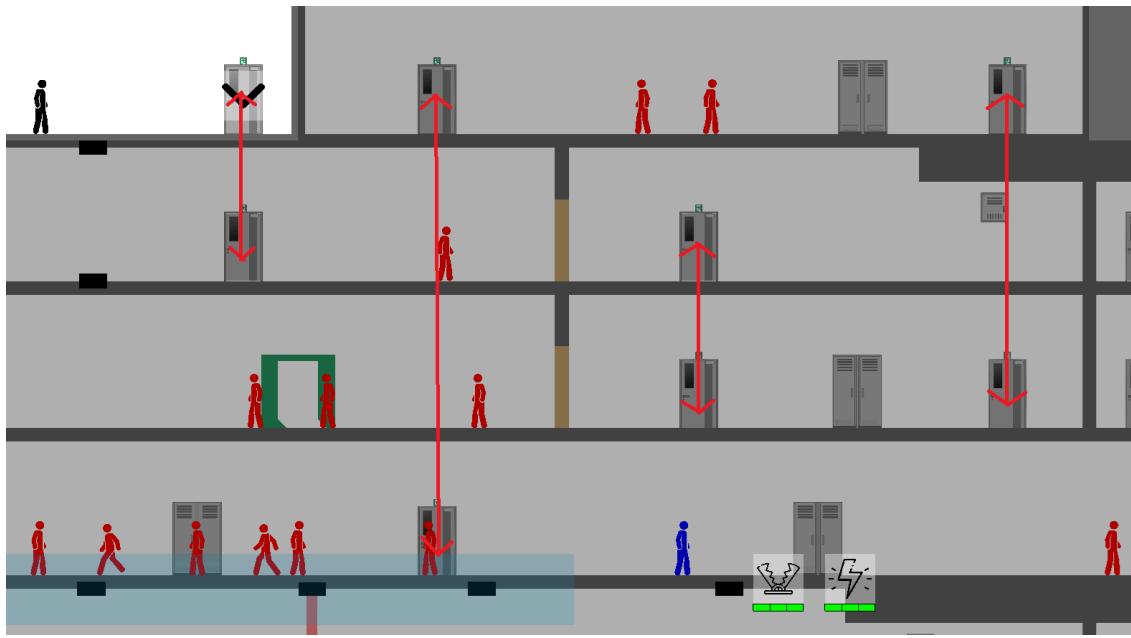


- 임무는 외부에서 할당되므로, **NPC FSM**은 공통된 기본 동작만 유지하고, **행동 확장**은 임무 객체 단위로 제작, 관리 가능
- NPC 간 상호작용은 모두 관리자를 통해 간접 처리되어 NPC 클래스 간의 직접적인 의존도가 낮아짐
- 새로운 임무를 추가할 때, **FSM**을 수정하지 않아도 되므로 확장성과 유지보수성이 향상

경로 탐색

이 게임의 맵은 여러개의 방으로 구성된 2D 구조입니다. 방들은 좌우로만 직접 연결되어 있지만 위아래로는 연결되어 있지 않기 때문에 층간 이동을 원한다면 반드시 계단을 이용해야 합니다. 게다가 계단은 단순히 바로 위나 아래 방으로 연결되는 구조가 아니며, 1층에서 4층, 2층에서 5층처럼 불규칙적인 방식으로 연결되어 있습니다. 또한, 같은 층이라고 해도 중간에 벽이 있다면 해당 방향으로의 이동이 제한됩니다.

이러한 구조로 인해 일반적인 그리드 맵과 달리, 특정 위치로 가는 경로가 불규칙적이며 예외적인 형태를 띠고 있습니다.



구현 할 당시에는 A* 알고리즘에 대한 이해가 부족했고, 다음과 같은 이유들로 인해 이 게임의 맵 구조에서는 휴리스틱 계산이 불가능하거나 부정확할 것이라고 판단했습니다.

- 계단을 통해 도달하는 층이 규칙성이 없어 어느 계단에 더 높은 가중치를 둬야하는지 판단 어려움
- 경로가 막혀있거나 계단을 통해서만 도달 가능한 경로도 있기 때문에, 단순히 거리를 기반으로 한 휴리스틱은 실제 경로의 유효성을 판단할 수 없음

부정확한 휴리스틱은 결국 완전탐색과 비슷해지거나 최적의 경로를 찾아내지 못하는 경우도 있기 때문에 DFS를 통한 완전 탐색으로 구현하게 되었습니다.

구현을 완료하여 게임에서 바라는 방향으로 동작하게 되었지만 모든 가능한 경로를 탐색해야 했기 때문에 맵이 커질수록 성능 저하가 발생할 우려가 있었습니다.

지금이라면 다음과 같은 방식으로 휴리스틱 함수를 구현하여 A*를 시도해 볼 수 있을것 같습니다.

A* 경로 탐색	휴리스틱 함수
<pre> 시작 지점 삽입 while (배열에 경로 존재) { 배열에서 휴리스틱+이동 횟수(최종 비용) 가장 적은 지점 검색 검색된 지점 기록 if(검색된 지점 == 목표 지점) break; 검색된 지점에서 좌,우 경로 휴리스틱 계산 if(현재 방에 계단이 존재) {계단을 통해 도착하는 경로 휴리스틱 계산} 체크한 경로 비용 비교하여 삽입 } 도착지에서 역추적하여 시작점 까지 경로 생성 </pre>	<pre> 휴리스틱 함수 return 목표 지점과의 x 값 차이 절대값 + if(목표 지점과 다른 층) { if(방에 계단 존재) { 계단을 통해 가게 될 층과 목적지 y 값 차이 } else {Max(맵 가로 길이, 맵 세로 길이)} } else {0} </pre>

마무리

이 프로젝트는 제 첫 번째 게임 인공지능 시스템 구현 경험이었습니다. NPC 제어 시스템을 구현하며 오브젝트와 시스템 간 결합으로 인한 유지보수 문제를 경험하였고, 경로 탐색을 구현하며 여러 제약 조건을 통해 수많은 실패를 경험하였습니다. 이런 경험을 통해 구조 설계의 중요성을 이해하였으며, 구현 목적과 제약조건 등의 현재 상황을 이해하는 것의 중요성을 깨달았고 게임 로직에 쓰이는 시스템 구현 및 설계에 더욱 흥미를 느끼게 된 프로젝트였습니다.