|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **주차** | 18 주차 | **기간** | 2020.4.27 ~ 2020.05.03 | **지도교수 이용희** | (서명) |
| 이번주 한일 요약 | -행동트리 기초작업 | | | | |

<상세 수행내용>

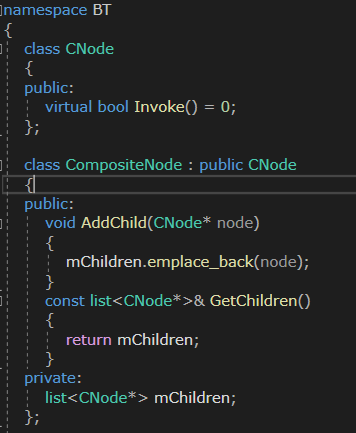
깃 주소 : <https://github.com/zzcx88/DKS_Project.git>

**이재원 :**

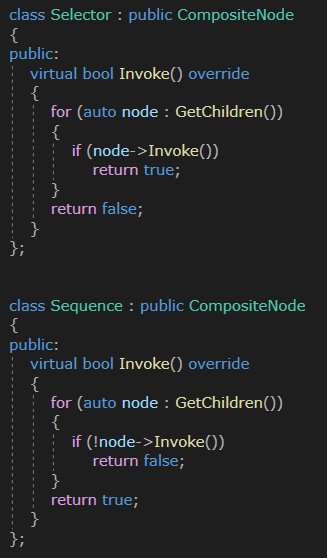
**행동트리를 정의하기 위한 기초 작업을 하였다.**

**1. CNode 클래스 작성**

행동 트리의 기본 노드가 되는 CNode클래스를 작성한다.



CNode는 행위가 성공했는지 못했는지를 판단하는데 필요한 invoke함수와 CNode를 상속받아 작성된 CompositeNode는 자식 노드들을 저장하기 위한 리스트와 자식 노드를 추가, 불러오기 위한 함수를 정의한다.

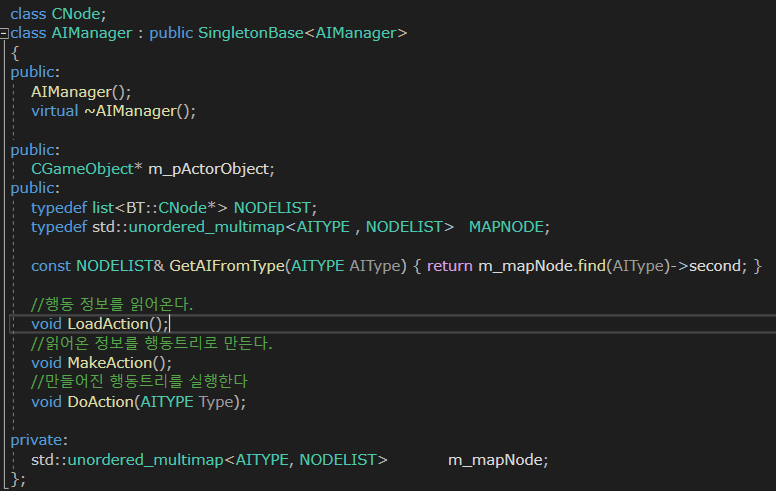


어떤 행위를 실행 할 것인지 결정하기 위한 SelectorNode와 행위를 모두 실행해야 True가 반환되어 SelectorNode의 결정에 영향을 주는 SequenceNode를 정의한다.

SequenceNode노드는 모든 자식 노드가 True를 반환해야 본 노드도 True를 반환하기 때문에 하나만 True여도 True를 반환하는 SelectorNode와 Invoke함수가 차이를 둔다.

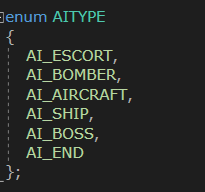
**2. AIManager 클래스 작성**

추후 행동트리 툴에서 만들어진 행위에 대한 정보를 읽어와 행동 트리를 구성하고 게임 오브젝트의 Animate함수에서 호출하여 행동을 적용할 함수를 작성한다.



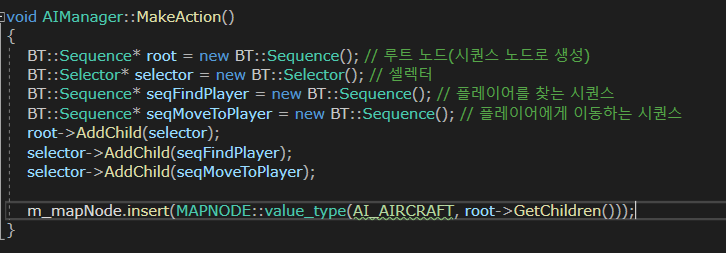
AIManager클래스는 하나의 행동 트리를 정의한 노드가 담긴 리스트를 AITYPE을 키로 가지고 있는 맵 자료 구조를 가지고 있으며 게임 오브젝트는 이 리스트의 노드를 순회하여 행위를 실행한다.

물론 게임 오브젝트는 DoAction 함수를 통해 행위를 호출하며 필요에 따라 GetAIFromType이라는 함수를 통해 직접 맵 노드에 접근할 수도 있다.



AITYPE은 enum으로 정의되어 있으며 하나의 행동트리는 하나의 AITYPE에 대응된다.

따라서 필요한 행동트리의 수만큼 AITYPE의 enum값도 추가 혹은 삭제가 될것이다.



임시로 노드를 구성하여 MakeAction함수를 채워보았다.

노드만 구성하였고 노드 내 행위는 구성하지 않았음으로 아무 동작도 하지 않을 것이지만 우선 트리를 어떻게 구성할지 개념을 잡을 필요가 있었다.

각 노드를 생성한 후에 루트부터 자식 노드까지 설정하고 최종적으로 루트 노드의 자식 노드 리스트를 맵 자료구조에 넣는 것으로 마무리하는 구조로 작성하였다.

추후 툴에서 생성된 노드 정보를 가지고 유동적으로 생성될 수 있게 변경할 것이다.

|  |  |  |  |
| --- | --- | --- | --- |
| **문제점 정리** | 이재원 :  - 아직 행동 트리 툴이 개발되지 않았으며 행위 또한 정의되지 않음 | | |
| **해결방안** | 이재원 :  - 아무 행위나 만들어서 트리를 테스트하고 완성된 양식에 맞춰서 노드를 정의하여 저장하는 행동 트리 툴을 제작한다. | | |
| **다음주차** | 19 주차 | **다음기간** | 2020.05.04 ~ 2020.05.10 |
| **다음주 할일** | 이재원 : Behavior 툴 제작, 기총발사 제작 | | |
| **지도 교수**  **Comment** |  | | |