A character with dark hair in a bun, wearing a brown vest and a blue gauntlet, stands on a rocky outcrop. They are looking out over a vast, dark, and jagged landscape under a dramatic sunset sky with orange and purple hues. In the distance, there are tall, dark rock formations and some industrial structures.

게임 엔진

LEC 24 AI (2)



한국공학대학교
TECH UNIVERSITY OF KOREA

이대현 교수

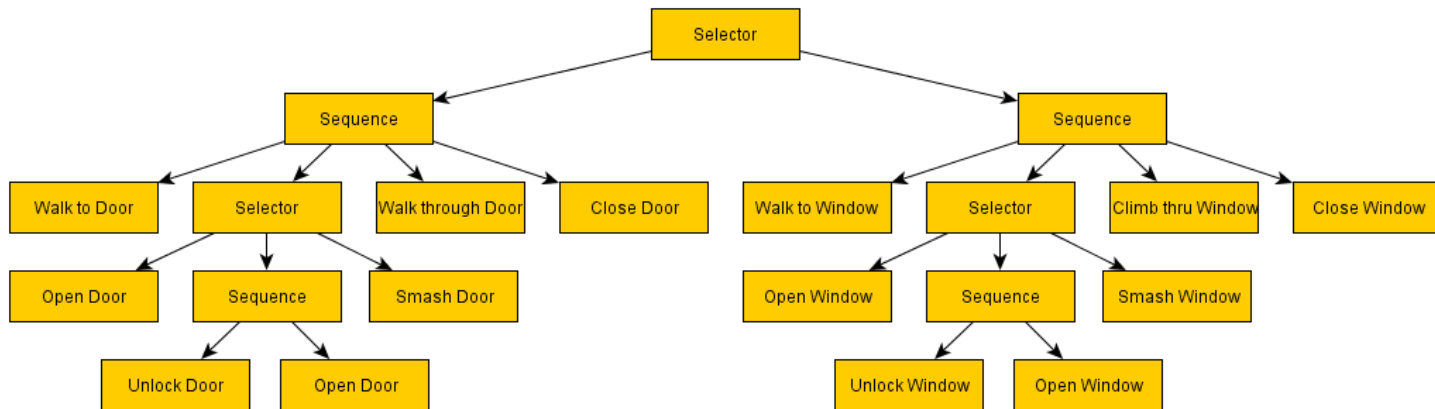
Behavior Tree 시스템의 구성 요소



기본 구조

트리 구조

- 말 그대로, 객체의 행위들을 Tree 구조로 연결하여 나타냄.
- Node는 작업 수행 결과를 반환함.
 - SUCCESS, FAIL, RUNNING
- Node가 자식 노드가 있으면, 자식 노드들을 실행하고, 그 결과를 종합하여 노드의 최종 상태값을 결정함.



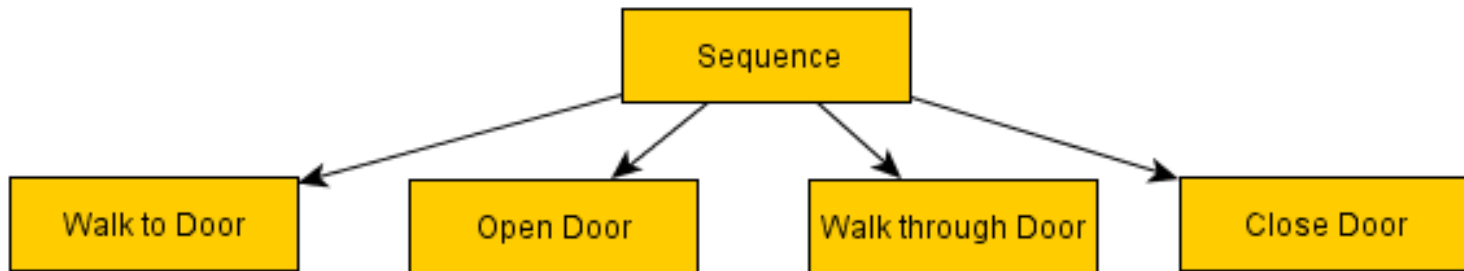
Task Node

- 단위 작업을 수행하는 노드
- 노드의 실행 결과는 Success, Fail, 또는 ND(결정 안됨) 의 세 종류임.
- ND가 있으면, Success 또는 Fail 결과가 나올 때까지 계속 실행됨.
- 언리얼 엔진은 기본 기능의 노드를 제공하고 있음.
- 필요에 따라 목적에 맞는 노드를 제작해야 함.

Finish with Result
Make Noise
Move Directly Toward
Move To
Play Animation
Play Sound
Push Pawn Action
Rotate to Face BBEntry
Run Behavior
Run Behavior Dynamic
Run EQSQuery
Set Tag Cooldown
Wait
Wait Blackboard Time
Auto Arrange

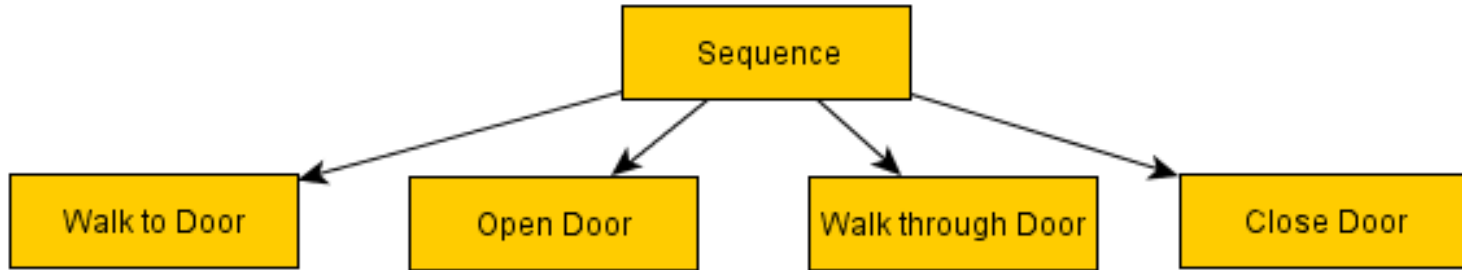
Sequence Node

- 실행은, 맨 왼쪽 자식 노드부터 오른쪽으로 진행하면서 실행됨.
- 모든 자식 노드가 다 SUCCESS 되면, 노드도 성공
- 여러 개의 작업이 모두 다 차근 차근 진행되어야 하는 경우 - AND 조건
- 하나라도 FAIL 되면, 실행 중단. Sequence Node 도 FAIL
- 실행 결과, 처음으로 RUNNING이 나오면, 자식 노드의 위치를 기록함. 결과는 RUNNING임.
- 어떤 목표를 달성하기 위해 수행해야 하는 Task 들을 차례로 모두 완수해야 하는 경우에 사용됨.



복합 노드: Sequence Node

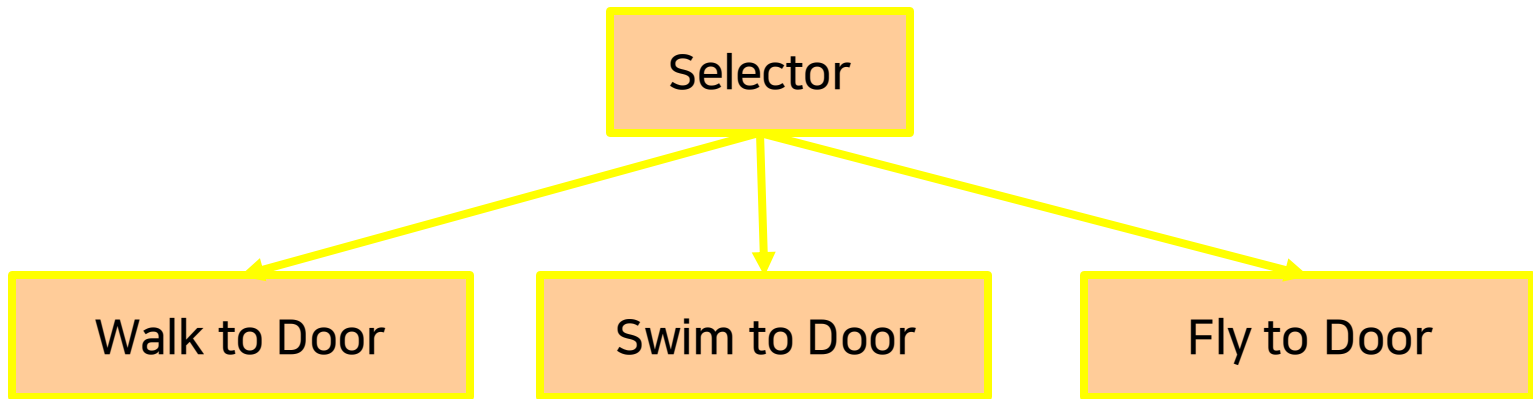
Child 노드를 왼쪽부터 오른쪽으로 차례로 실행시켜나가면서, 모든 child 노드가 다 성공(Success)하게 되는 경우만 Sequence 노드가 성공함. AND 조건으로 볼 수 있음. 어떤 목표를 달성하기 위해 수행해야 하는 Task 들을 차례로 모두 완수해야 하는 경우에 사용됨.



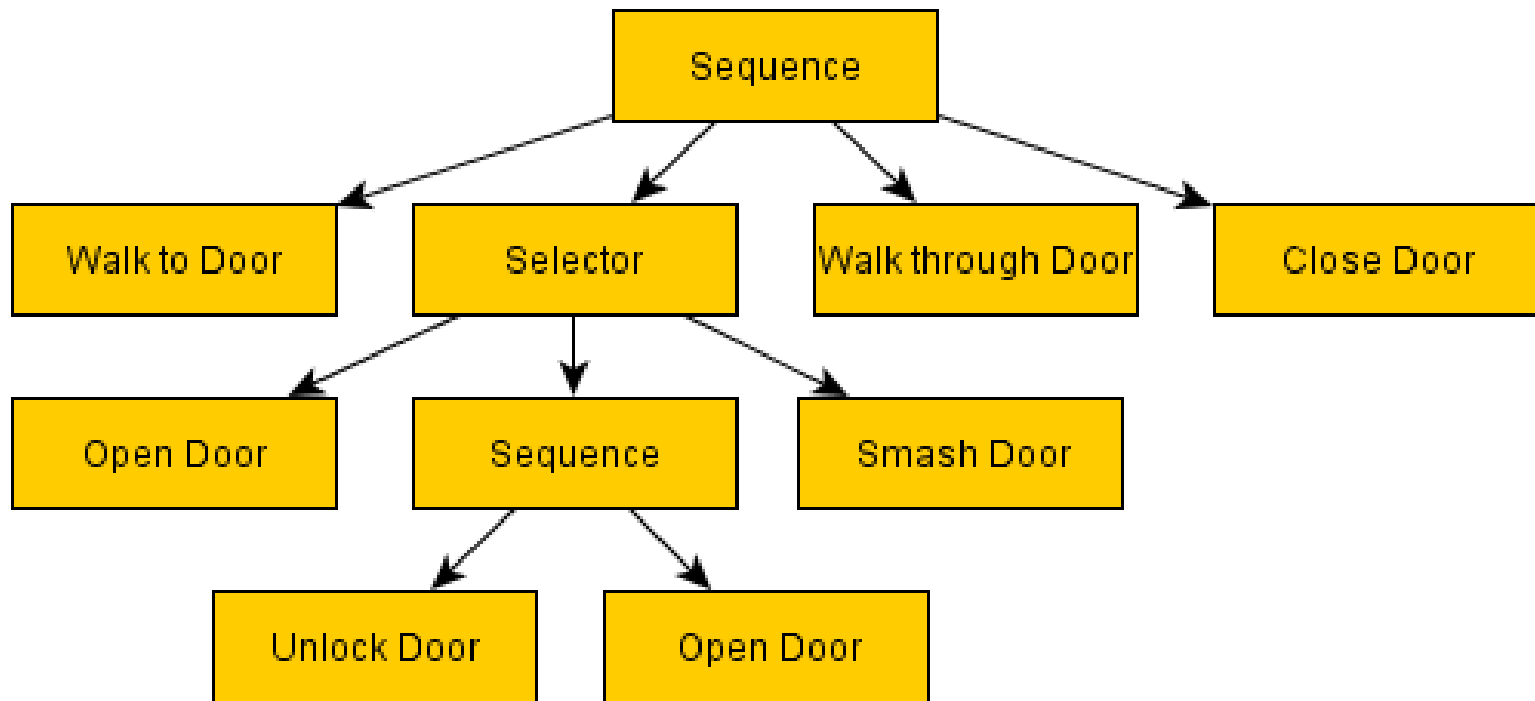
복합 노드: Selector Node

Child 노드를 왼쪽부터 오른쪽으로 차례로 실행시켜나가면서, 성공하는 child 노드가 하나라도 나오면, Selector 노드가 성공함. Child 노드가 모두 다 Fail해야만, 결과가 Fail됨. OR 조건으로 볼 수 있음.

목표를 달성하기 위해 여러 개의 대안이 있을 경우에 사용됨. 대안에 대한 우선 순위를 부여할 때 부여할 수 있음. 즉, 왼쪽에 있는 노드가, 오른쪽에 있는 노드보다 우선 순위가 높음.



BT 예제



Behavior Tree 구현 절차

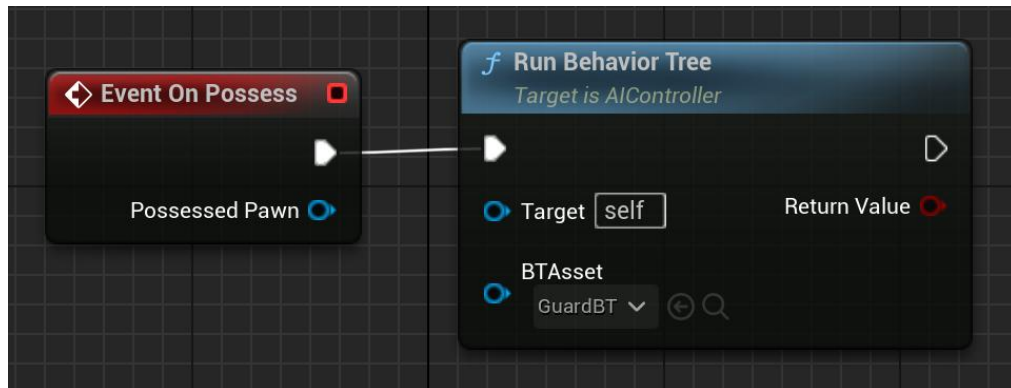
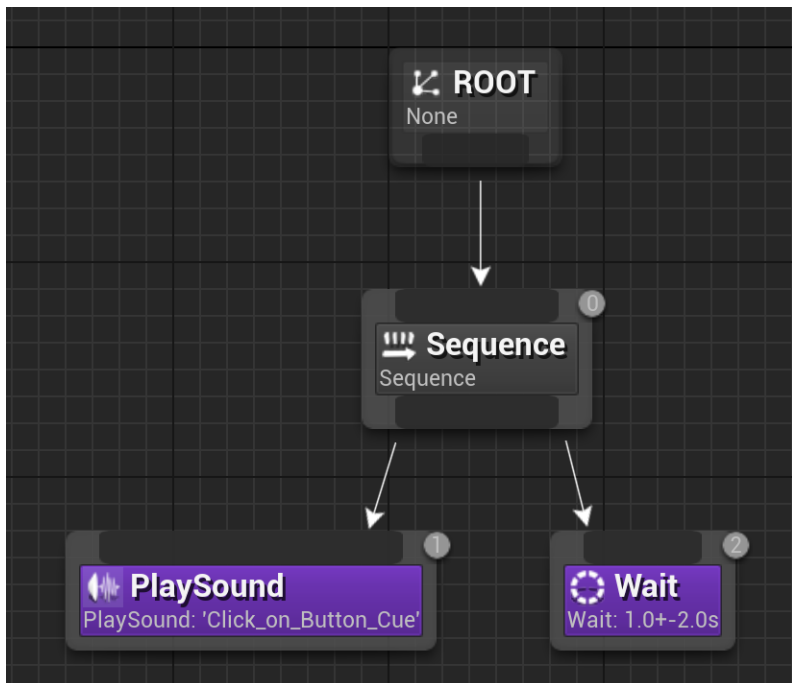
- **Blackboard 준비**
 - BT 구현에 필요한 정보들을 일종의 DB 형식으로 관리(Key-Value Pair)
- **Behavior Tree 구성**
 - Task Node : 단위 작업을 수행
 - Composite(복합) Node: 여러 개의 단위 작업들을 연결
 - Decorator : 조건에 따른 실행 제어
 - Service : 주기적인 상황 파악
- **AI Controller 구현**
 - Blackboard 설정
 - Behavior Tree 실행



실습

Behavior Tree 구성

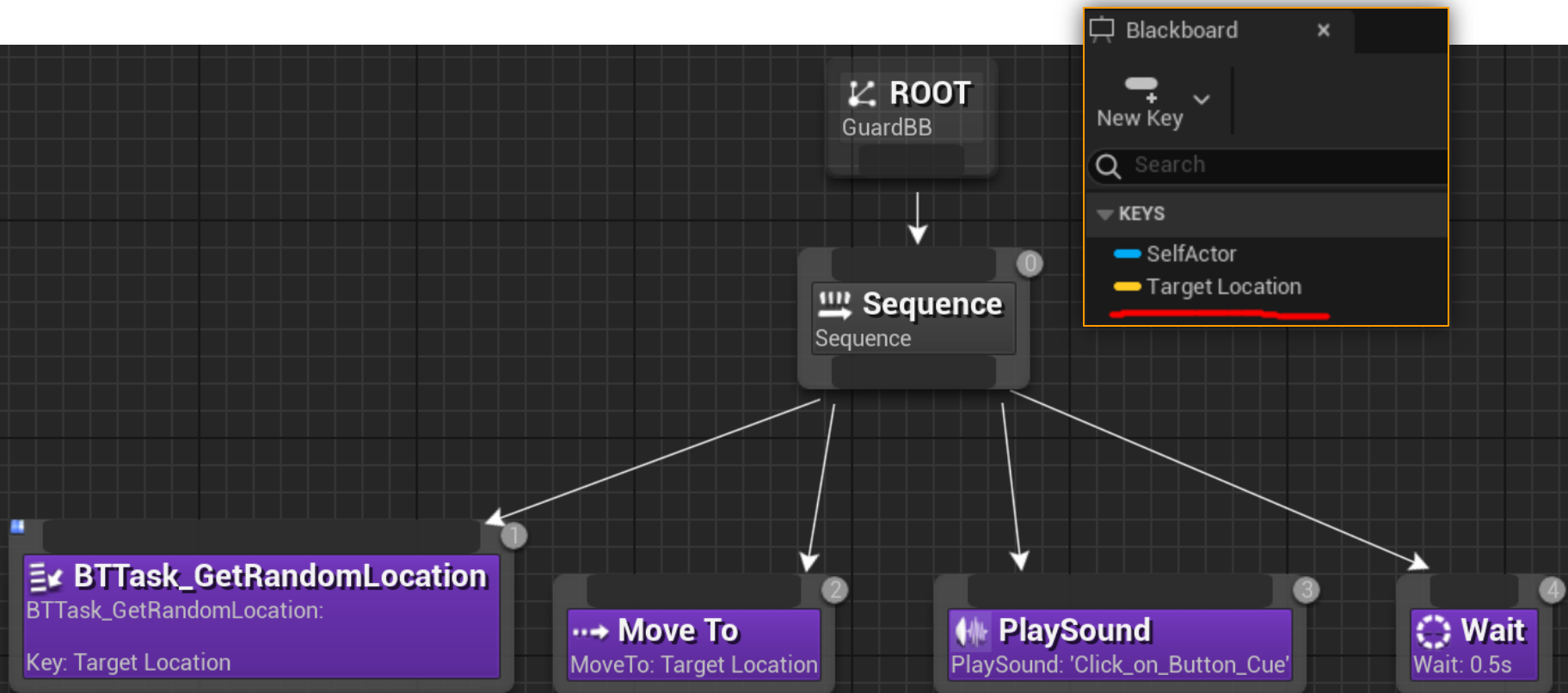
BT 의 작성과 실행



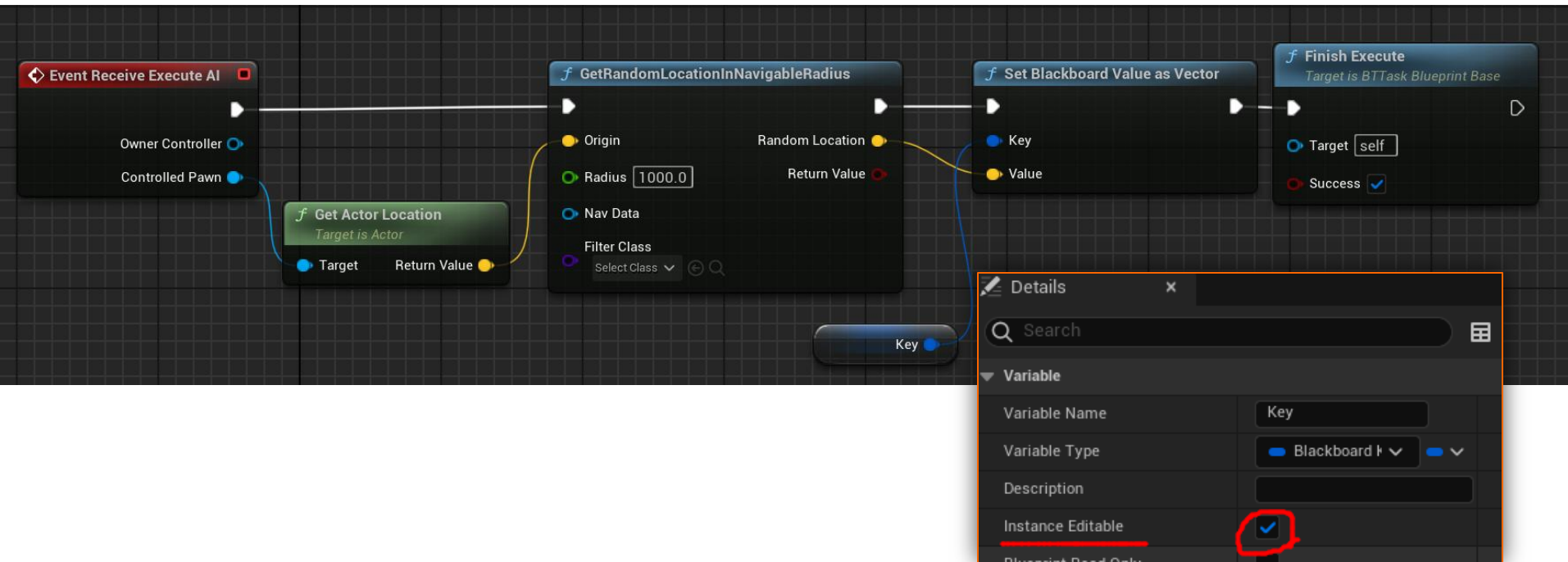


실습 순찰

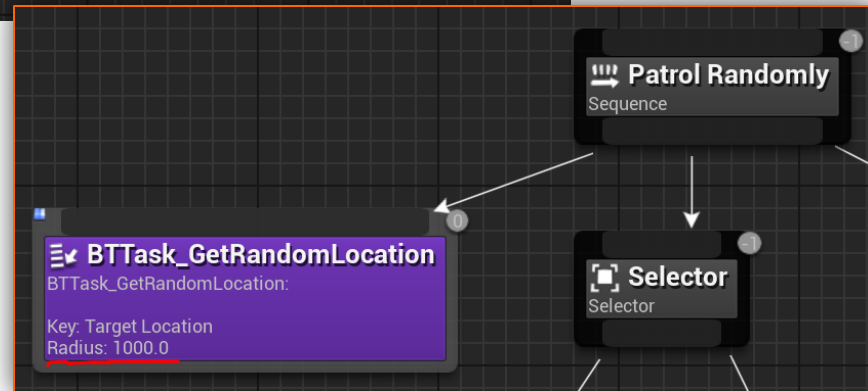
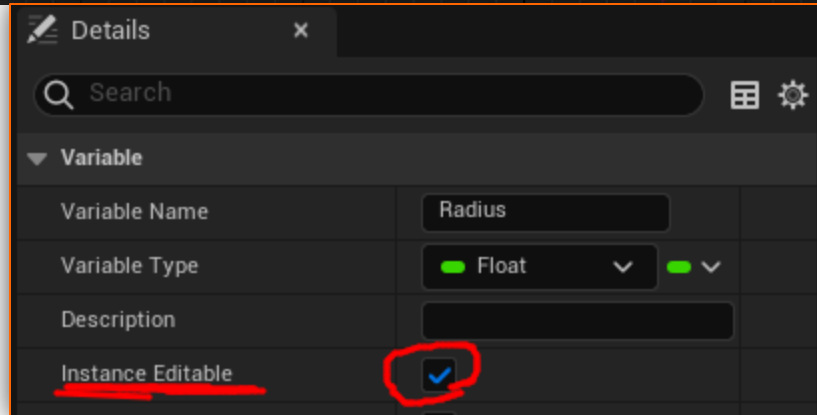
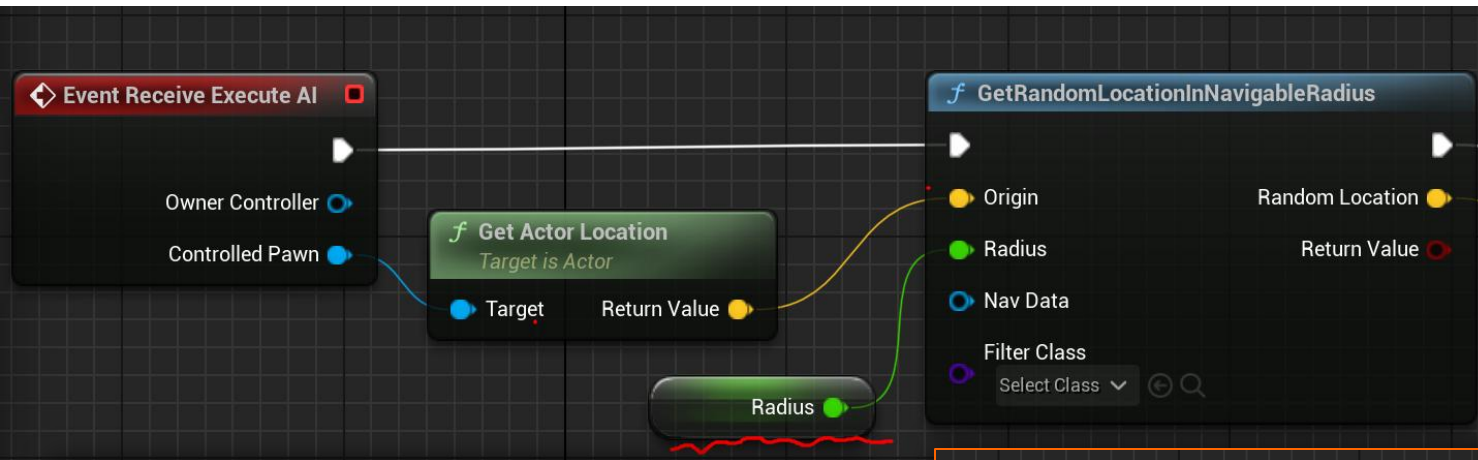
순찰(랜덤 목적지 이동 반복) BT



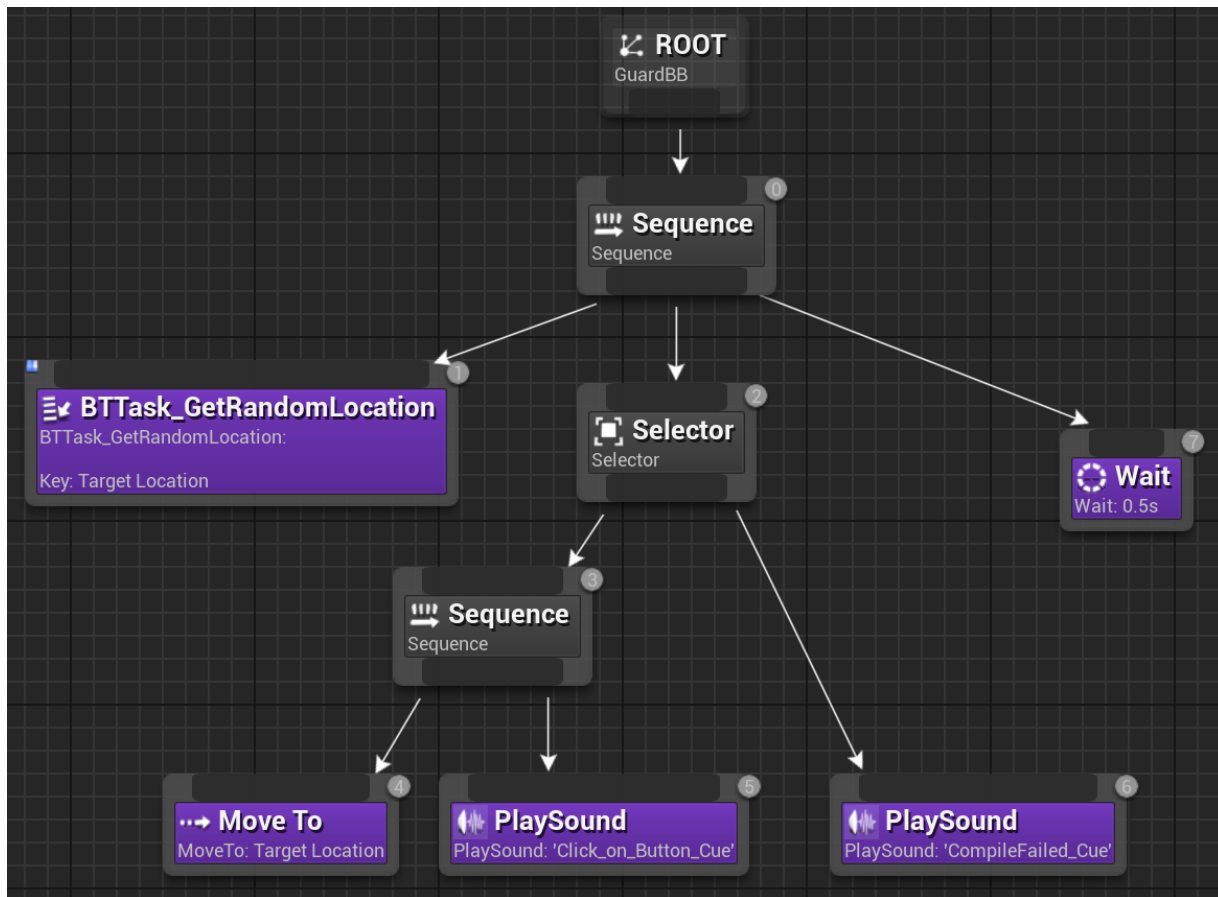
GetRandomLocation Task



순찰 반지름 조절



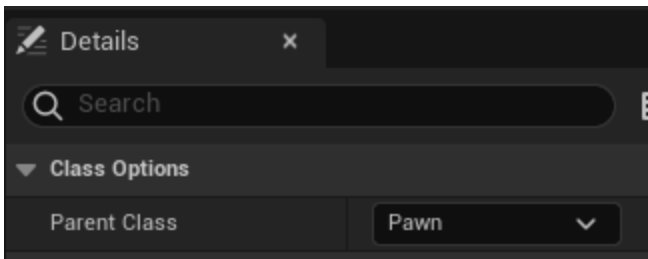
Move To 가 FAIL 일 경우에 대한 처리



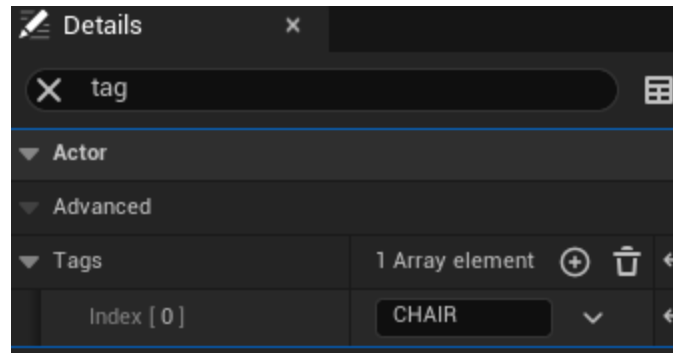


실습 아이템 감지

ChairPawn

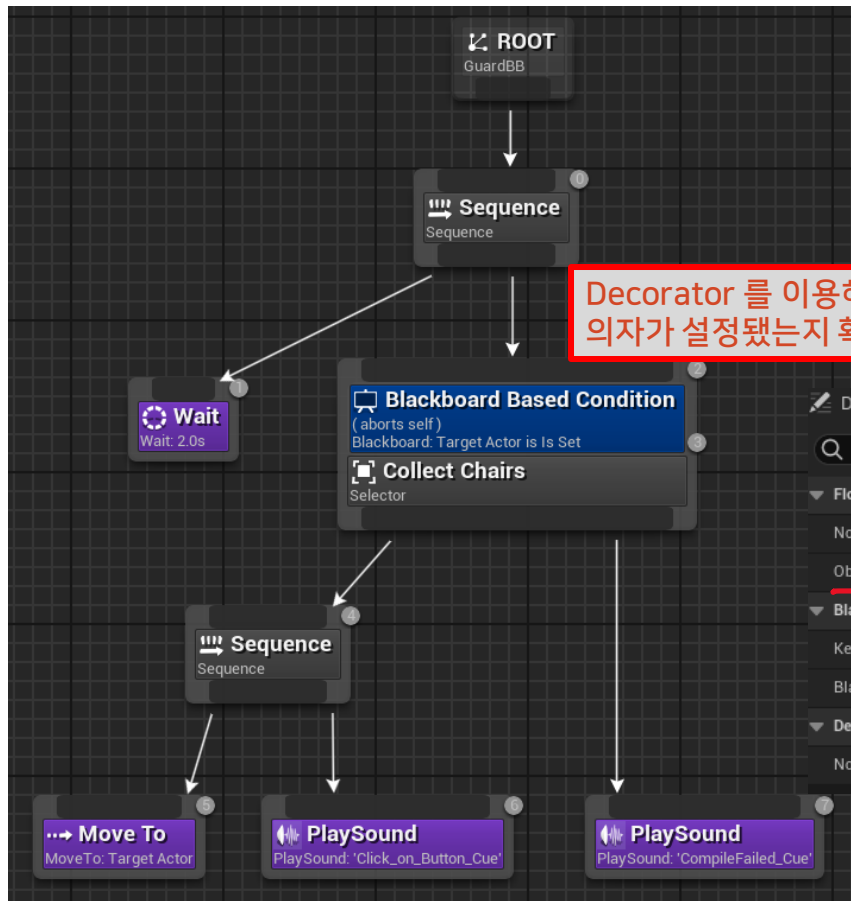


AI Perception 기능을 통해서
감지가 되려면 액터는 Pawn
이어야 함.

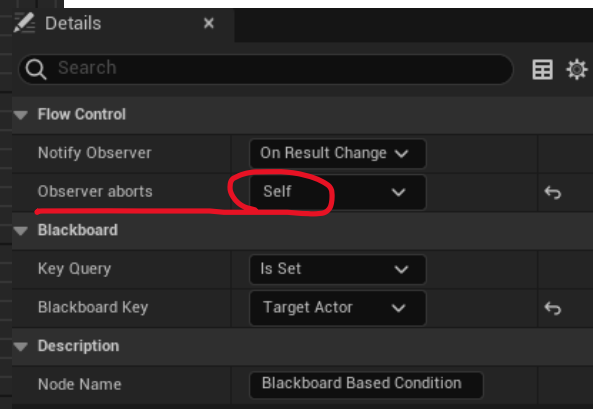
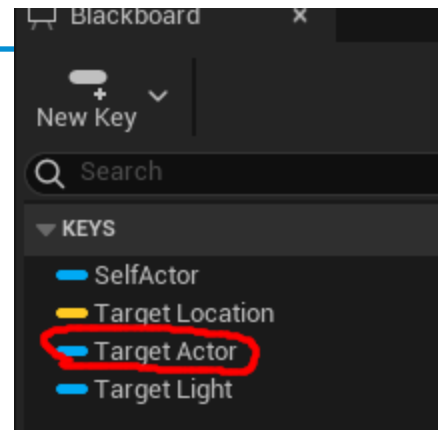


Tag 정보를 넣어서, 액터의
종류를 판별할 수 있도록 함.

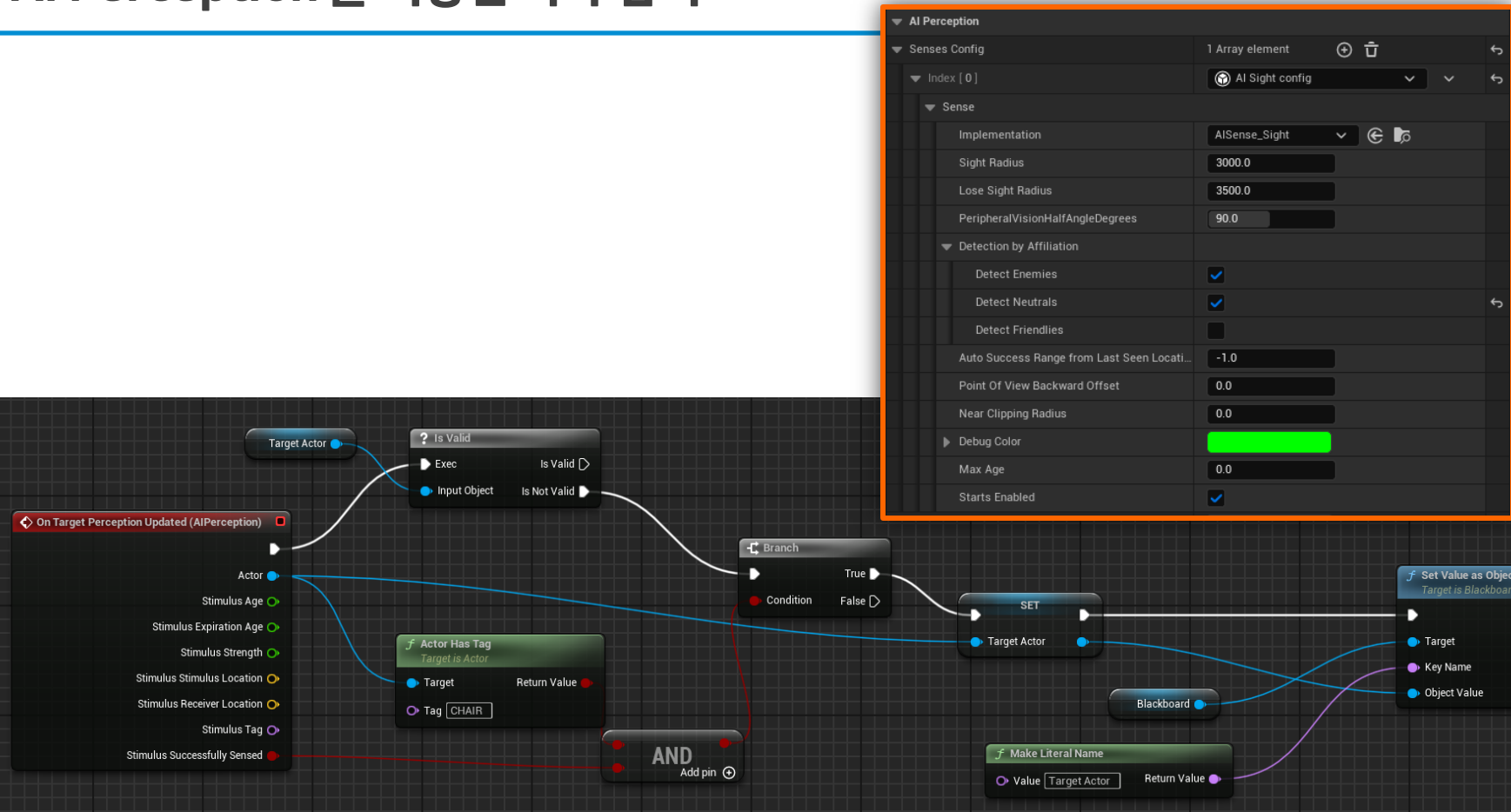
의자 수집 BT



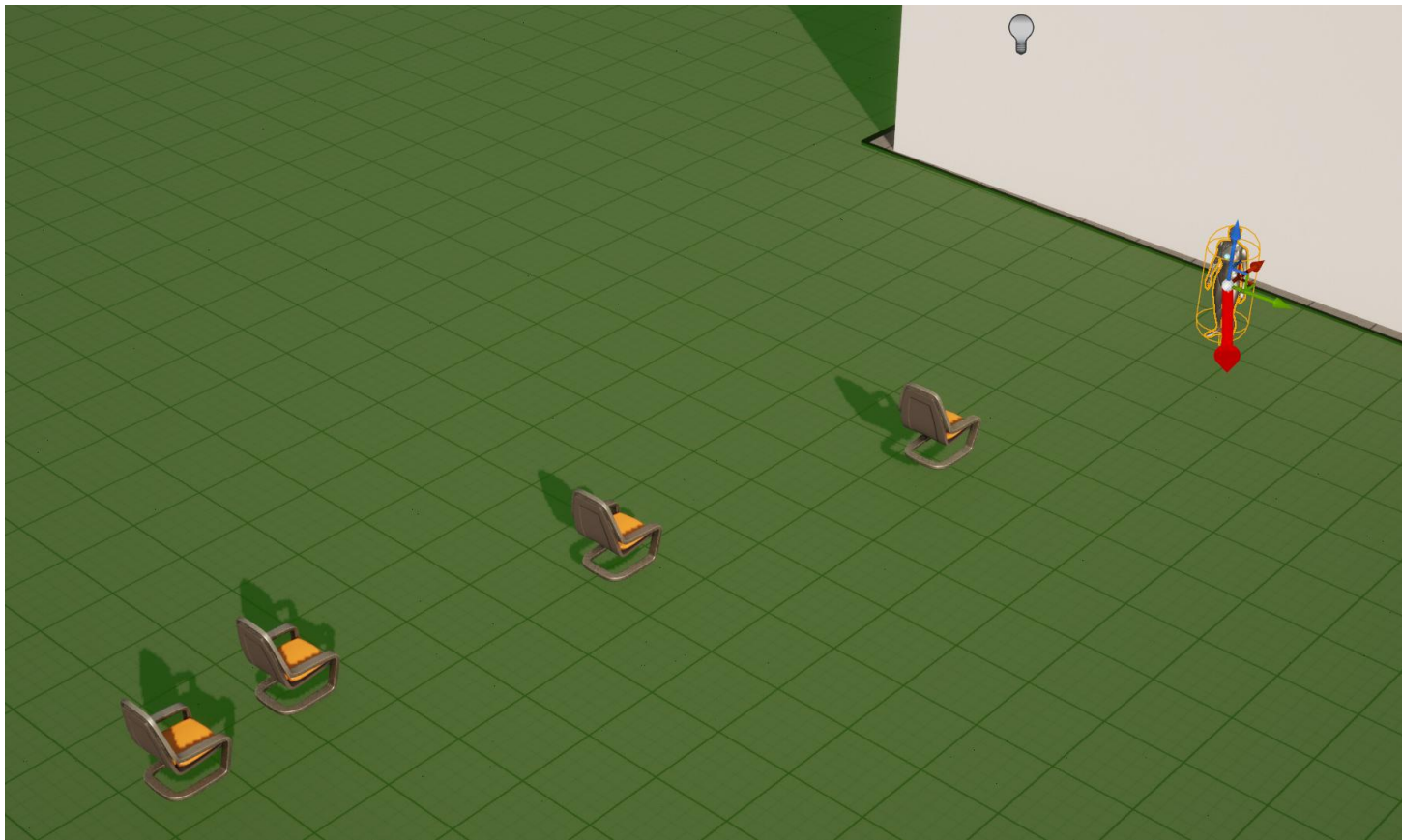
Decorator 를 이용해서 Target Actor 인 의자가 설정됐는지 확인



AI Perception을 이용한 의자 감지



의자 아이템 획득



AI Debug – 작은 따옴표, 숫자패드 키 1,2,3,4 등,

Tap [Apostrophe] to close, use Numpad to toggle categories.
Ctrl+Tilde:HUD Ctrl+Tab:DebugMessages Tab:Spectator
0:Navmesh 1:AI 2:BehaviorTree 3:EQS 4:Perception 5:PerceptionSystem

Time: 6.77s

[CATEGORY: AI]
Controller Name: GuardController_C_0
Pawn Name: QuinnGaurd_C_UAID_C85EA9416E7C9DFB01_1309889867
Movement Mode: Walking, Base: StaticMeshActor_1.StaticMeshComponent0
NavData: Default, Path following: Idle
Behavior: Running, Tree: GuardBT
Active task: GuardBT:Wait[1]
Gameplay tasks: 0
Montage: None

[CATEGORY: BehaviorTree]
Brain Component: BTComponent
Behavior tree: GuardBT
0: Sequence [Sequence]
1: Wait [Wait: 2.0s], remaining: 1.972666s

[CATEGORY: Perception]
Sight: green rangeIN: 3000.00 (green) rangeOUT: 3500.00 (neonpink)

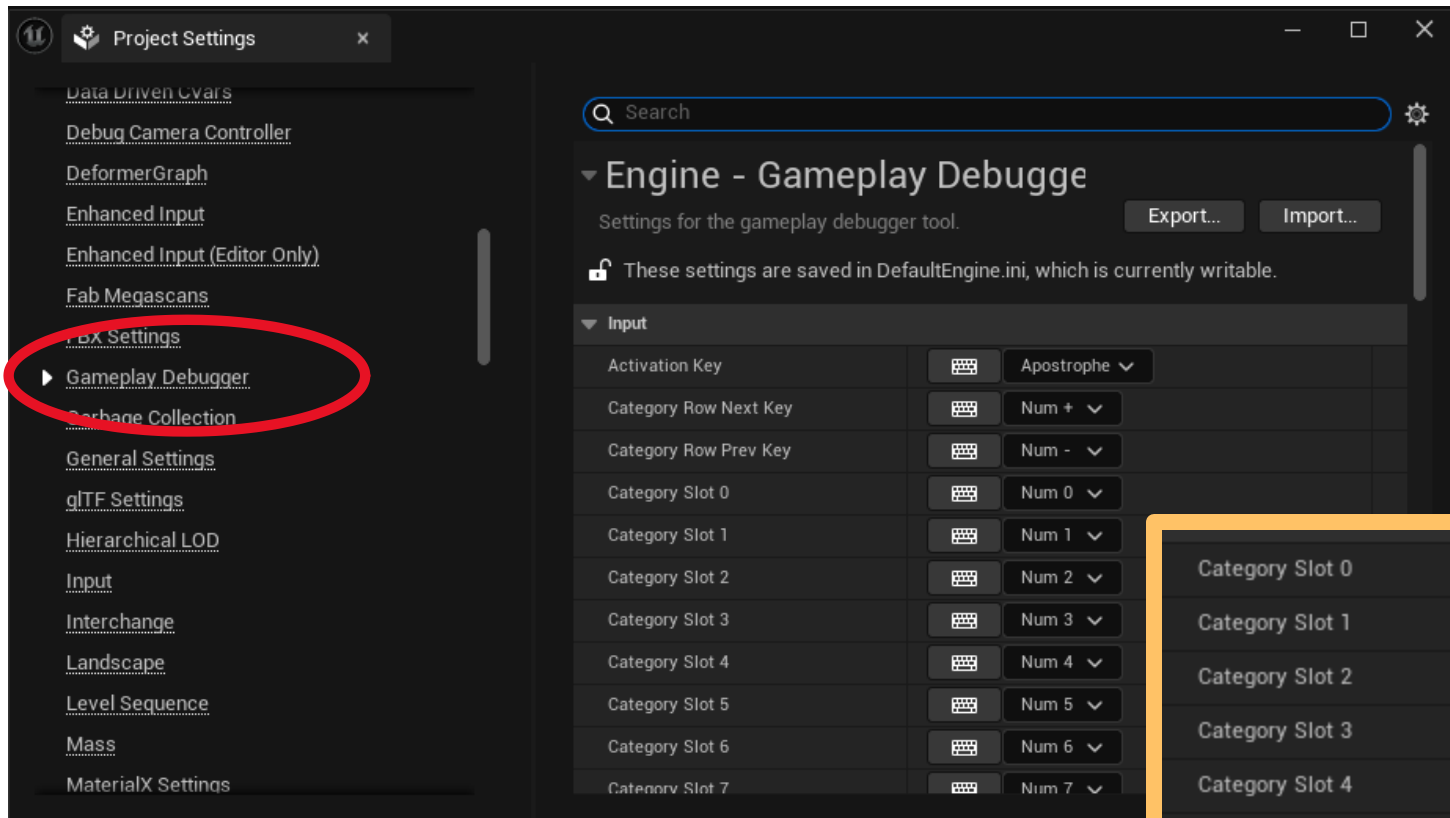
Blackboard (asset: GuardBB)
SelfActor: QuinnGaurd_C_UAID_C85EA9416E7C9DFB01_1309889867
Target Actor: ChairPawn_C_UAID_C85EA9416E7C06FC01_1647270328
Target Light: None
Target Location: (invalid)



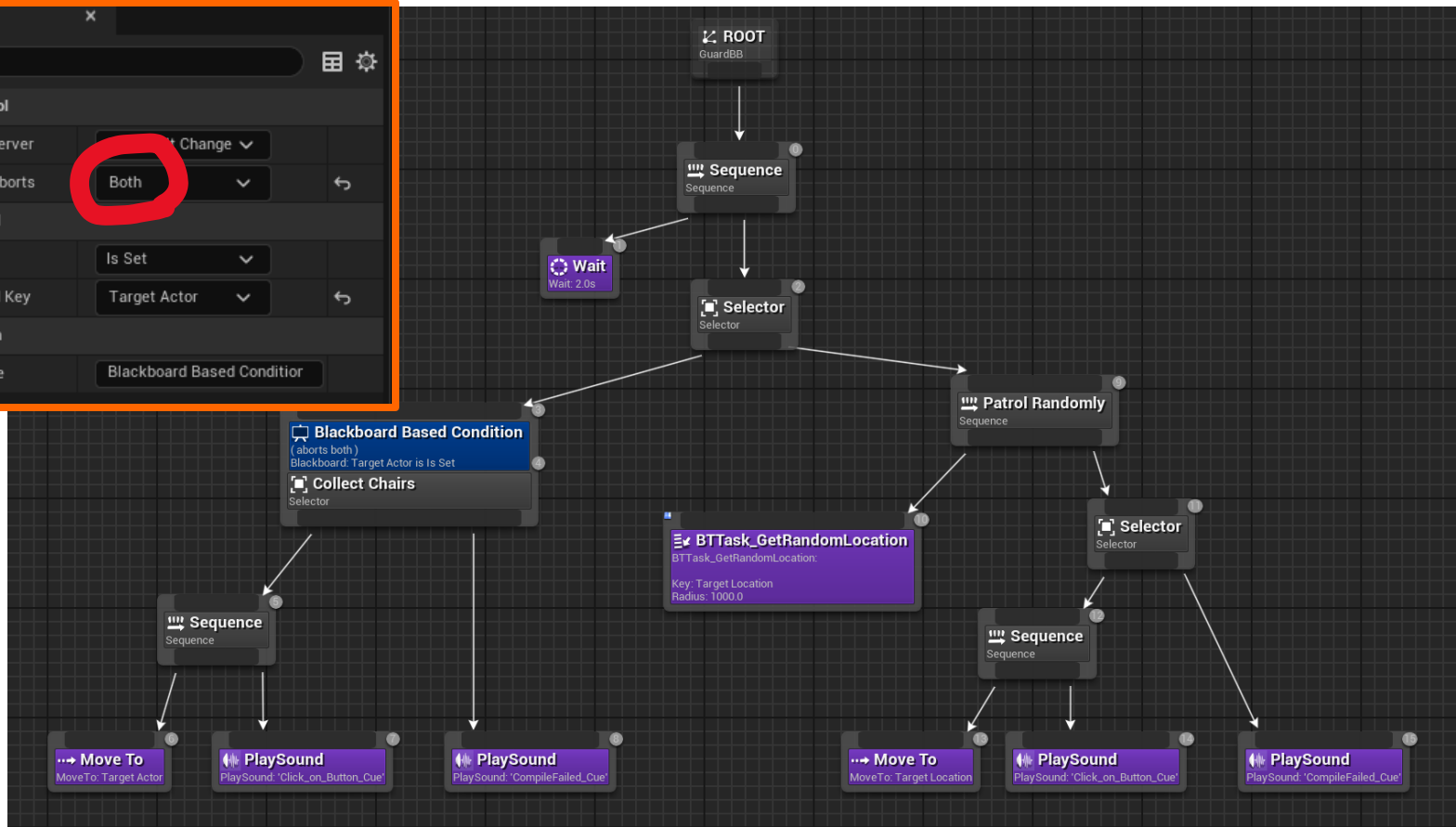
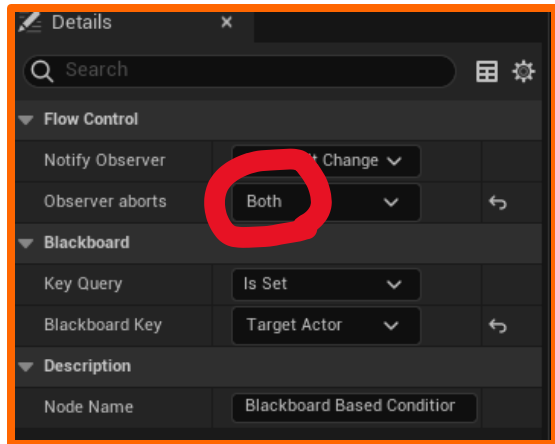
GuardController_C_0 QuinnGaurd_C_UAID_C85EA9416E7C9DFB01_1309889867

Sight: 1.00 - age 0.00
Sight: 1.00 - age 0.00

단축키 변경



랜덤하게 돌아다니면서 의자 수집



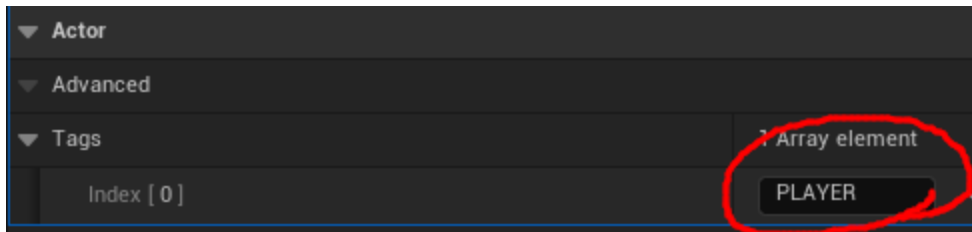
Observer Aborts

None	없음 - 아무것도 중단하지 않습니다.
Self	자신 - 자신과 이 노드 아래 실행중인 서브 트리도 중단합니다.
Lower Priority	하위 우선권 - 이 노드보다 오른쪽에 있는 노드를 중단합니다.
Both	양쪽 - 자신, 그 아래 실행중인 서브 트리, 이 노드 오른쪽에 있는 노드를 중단합니다.

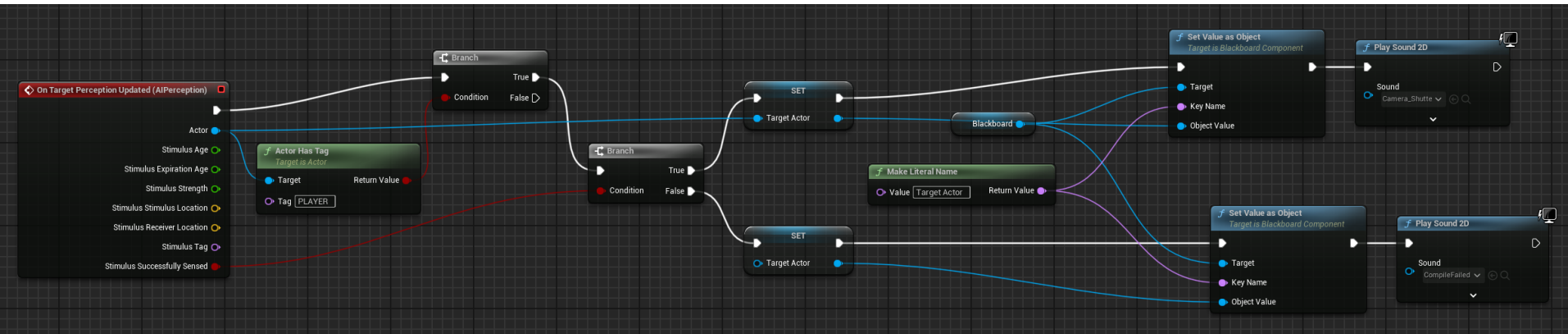


실습 플레이어 감지

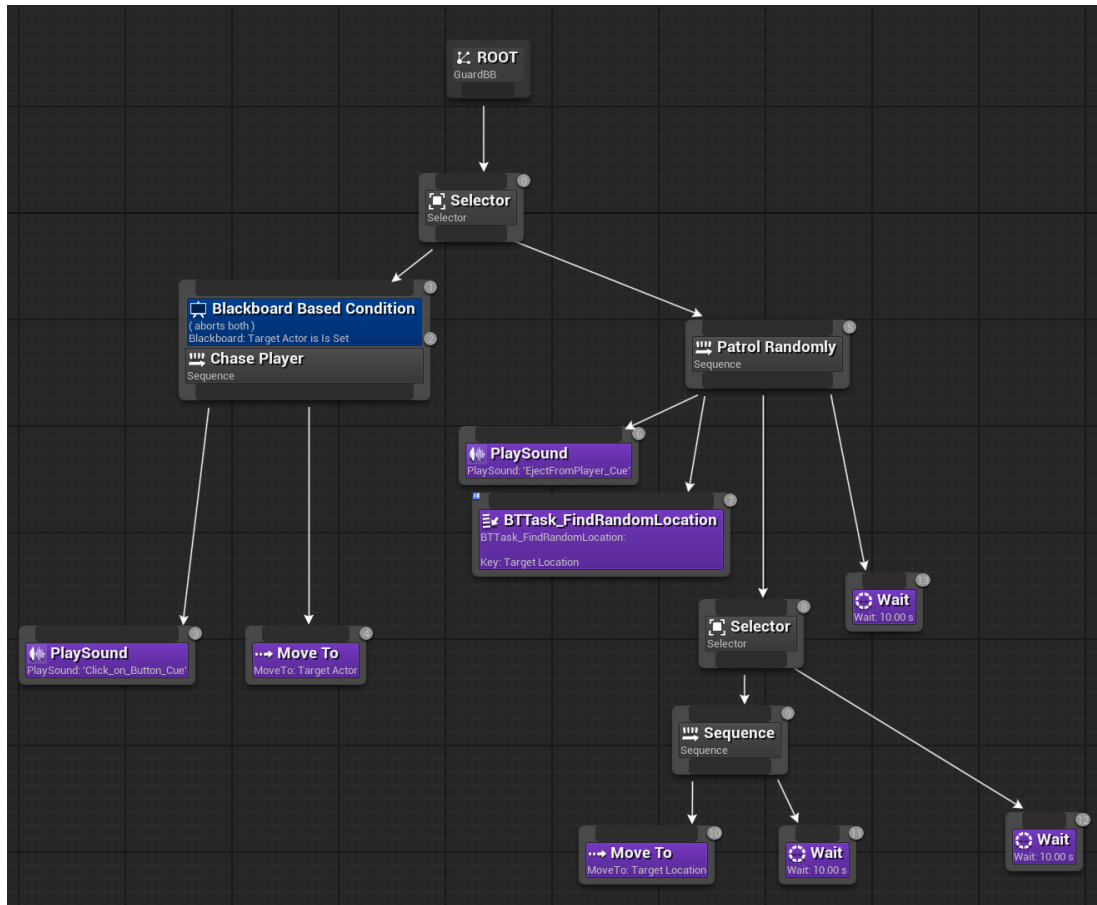
PLAYER tag 부여



AI Perception



Player 추적 Guard



Decorator 유의 사항

