

# 언리얼 프로그래밍 Part2-11

## 제목: 행동트리 모델의 이해

\*\*강의 내용 : 행동 트리 모델의 개요와 설계

### 강의 내용

#### 행동 트리 모델의 개요와 설계

이번 강의에서는 NPC의 인공지능을 구현하는 데 사용할

-강의에서 다루는 게임프레임워크 요소

### 강의에서 다루는 게임프레임워크 요소

게임	월드	모드	상태	
기믹	트리거	스폰	물리	
플레이어	입력	카메라	HUD	상태
폰	이동	모션	액션	위젯
데이터	애셋	테이블	설정	저장
인공지능	길찾기	BT		

## \*\*강의 목표

### 강의 목표

---

- 행동 트리 모델의 역사를 살펴보고 행동 트리 모델이 가진 장점을 이해
- 행동 트리 모델의 구성 요소와 이를 활용한 설계 방법의 학습



1138208

이득우의 언리얼 프로그래밍 Part2 - 언리얼 게임 프레임워크의 이해  
이득우

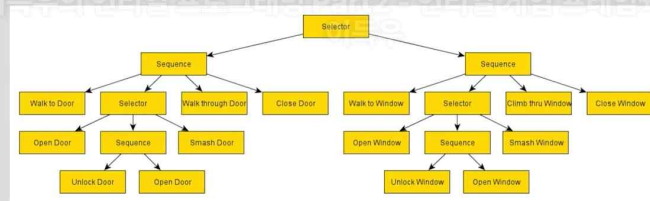
두 번째 목표로는 행동 트리 모델의 구성 요소와

-행동트리를 사용하여 행동에 따른 NPC의 인공지능을 단순한 행동에서부터 복잡한 행동까지 다양하게 지정한 것을 볼 수 있다.

## \*\*행동 트리 모델의 장점

### 행동 트리 모델의 장점

- FSM이 가지고 있던 문제를 해결하는 새로운 게임 인공지능 모델의 수립.
- 행동 트리 모델의 장점
  - 모듈화가 잘 되어 있어 확장이 자유롭다.
  - 트리를 기반으로 계층화가 잘 되어 있어, 복잡한 인공지능 모델을 쉽게 설계할 수 있다.
  - 다이어그램으로 인공지능 모델을 효과적으로 표현할 수 있다.
  - 제공되는 여러 편리한 부가 기능을 활용해 다양한 상황에 대해 손쉽게 제어할 수 있다.



유한 상태 기계모델(FSM)이 가지고 있었던

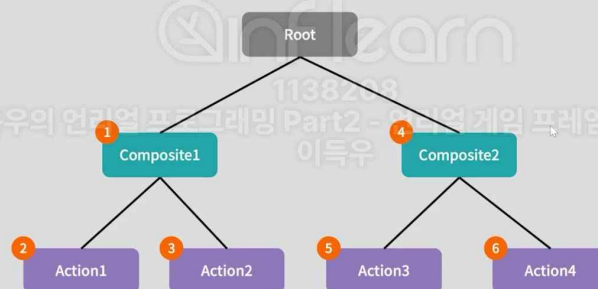
-행동 트리 모델은 당시에 일반적으로 사용되는 FSM이 가지고 있던 고질적인 문제를 해결해주는 새로운 방법론이었음.

-이런 행동 트리의 모델은 FSM에 비하여 가지는 장점이 많다.

## \*\*행동 트리 모델의 구성 요소

### 행동 트리 모델의 구성 요소

- 트리에서 항상 왼쪽에 있는 노드에 우선 순위를 부여한다.
- 시작 상태를 설정할 필요 없이 왼쪽에서부터 깊이 우선 탐색을 시작한다.



그렇다면 행동 트리 모델은 어떻게 구성되는지

-행동 트리 모델에는 언제나 (Root)라고 하는 최상위 노드가 있음.

-행동 트리 모델은 일정 간격마다 루트로부터의 의사결정이 일어난다.

-왼쪽에 높은 우선권을 부여하여 왼쪽에 배치한 노드를 먼저 처리하도록 설정함.

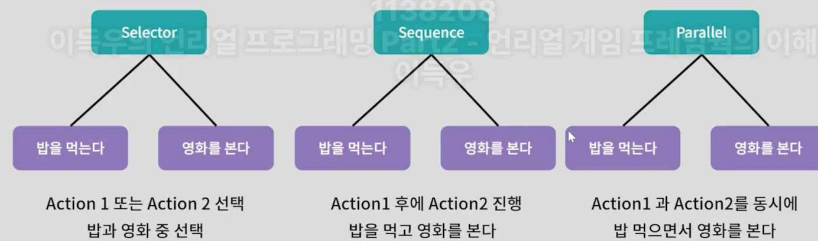
-지금 보이는 그림은 루트노트에서 파생된 자식 노드의 우선권을 숫자로 표시한 예시임.

-그림에서 보이는 것처럼 행동 트리는 의사결정을 위해 DFS(깊이 우선 탐색)를 수행 함.

## \*\*행동 트리 모델의 구성 요소

### 행동 트리 모델의 구성 요소

- 행동을 중심으로 설계한다.
- 단 부모 노드에서 다수의 행동을 컨트롤 함. 이를 컴포짓(Composite)이라고 함
  - 셀렉터 (여러 행동 중 하나의 행동을 지정)
  - 시퀀스 (여러 행동을 모두 수행)
  - 패러렐 (여러 행동을 함께 수행)



### 이러한 행동 트리는

-행동 트리는 단어가 의미하는 것처럼 캐릭터가 수행한 행동을 중심으로 인공지능을 설계한다.

-행동 트리에서 중심이 되는 행동은 독자적으로 실행 되지 않고, 행동을 어떻게 수행할것인지 지정하는 Composite이라고 하는 상위 노드를 통해서만 실행이 된다.

-하나의 Composite노드에는 여러개의 행동을 묶을 수가 있다. 또한 Composite노드에 속한 여러 행동들을 어떻게 수행하느냐에 따라서 Selector 노드, Sequence노드 Parallel노드라고 하는 세 가지 컴포짓 중 하나를 선택 할 수 있다.

-Selector 노드 : 말 그대로 여러 행동중에 하나의 행동을 선택하는데 사용됨.

-Sequence 노드 : 여러 행동을 차례대로 수행하도록 지정하는 노드

-Parallel 노드 : 행동을 동시에 수행하도록 지정하는 노드

ex) 밥을 먹는다는 행동과 영화를 본다라는 두가지 행동이 있다고 가정

(1) 셀렉터 컴포짓 노드에 두가지 행동을 배치하면 밥먹거나 영화보거나 둘중에 하나를 선택해야함

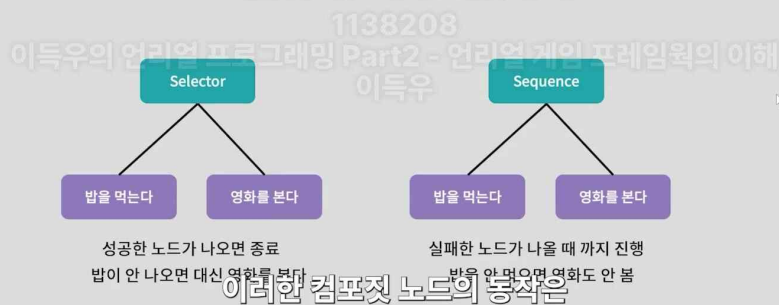
(2)시퀀스 노드에 배치하면 밥을 먹고 영화를 보는 두가지 행동을 차례대로 해야함.

(3)패러렐 노드에 두 행동 노드를 배치하면 밥을 먹는 행동과 영화를 보는 행동을 동시에 수행함.

## \*\*행동 트리 모델의 구성 요소

### 행동 트리 모델의 구성 요소

- 행동에 대한 다양한 결과
  - 성공(Succeeded) : 행동의 성공
  - 실패(Failed) : 행동의 실패
  - 중지(Aborted) : 외부 요인으로 인한 행동의 실패
  - 진행 중(InProgress) : 행동 결과를 홀딩
- 컴포짓 노드마다 다른 행동 결과 처리



-컴포짓 노드의 동작은 컴포짓 노드가 관리하는 행동 노드의 결과에 따라 다양하게 변화 된다.

-행동 노드는 수행한 결과에 따라 성공, 실패, 중지, 진행 총 네 가지 정보로 구분하여 컴포짓 노드에 보고한다.

- (1) '성공 결과'는 현재 행동을 성공적으로 완료했음을 알려주는 값
- (2) '실패'는 현재 행동이 완료했지만, 실패했음을 알려주는 값
- (3) '중지'는 현재 행동을 진행하고 있었는데 외부 요인에 의해 중지되었음을 알려주는 값
- (4) '진행 중'은 아직 행동이 끝나지 않고 진행 중임을 알려주는 값

-이러한 결과값을 보고 받은 컴포짓 노드는 값에 따라서 다른 행동을 취한다.

-셀렉터 컴포짓 노드는 자기가 관리하는 여러 행동중 하나만 선택하는 형태로 여러 행동중 성공한 액션이 하나라도 나오면 다른 행동을 살펴볼 필요가 없다. 따라서 성공 액션이 하나라도 나오면 바로 종료하고 컴포짓 노드도 성공으로 종료 시킴.

-시퀀스 노드는 모두 다 차례대로 실행하는 형태로 성공한 액션이 나오면 그 다음 액션을 실행하고 실패한 노드가 나오면 전체 행동을 종료하도록 설계 됨.

ex) 셀렉터 노드에 '밥을 먹는다'와 '영화를 본다'를 배치함.

왼쪽에 있는 밥을 먹는다가 우선순위를 가짐. 그리고 밥을 먹는다가 성공을 하면 밥을 먹었으므로 셀렉터 노드는 종료가 됨. 하지만 밥을 먹지 못하면 밥을 먹는 대신에 영화를 보고 해당 노드를 종료함

ex) 시퀀스 노드에 '밥을 먹는다'와 '영화를 본다'를 배치함.

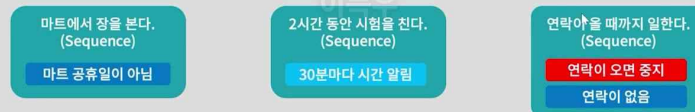
밥을 먹는다와 영화를 본다가 모두 수행이 되어 성공으로 인식이 되기 때문에 밥을 먹은 후에 영화를 보고 영화를 잘 봤으면 시퀀스 노드는 성공이 되는것이고 밥을 먹는데 실패했으면 해당 노드는 영화도 안보고 바로 실패로 끝내게 됨.

## \*\*행동 트리 모델의 구성 요소

### 행동 트리 모델의 구성 요소

- 컴포짓 노드에 부착하는 다양한 추가 기능
- 데코레이터(Decorator) : 컴포짓 노드가 실행되는 조건을 지정
- 서비스(Service) : 컴포짓 노드가 활성화될 때 주기적으로 실행하는 부가 명령
- 관찰자 중단(Abort) : 데코레이터 조건에 부합되면 컴포짓 내 활동을 모두 중단

이득우의 언리얼 프로그래밍 Part2 - 언리얼 게임 프레임워크의 이해



### 다음은 행동 트리 모델에서 제공하는

-행동 트리의 컴포짓 노드에는 다양한 부가 기능을 부착 가능하다.

(1) 데코레이터 : 컴포짓 노드에 조건을 명시하여 부착하고 해당 조건을 만족하는 경우에만 컴포짓이 수행되도록 설정하는 부가 기능이다.

ex) 마트에서 장을 보는 컴포짓 노드가 있다고 가정하자. 여기에 조건을 하나 부착을 하는데 마트 공휴일이 아닌 때의 조건의 데코레이터를 부착하면 공휴일에 장을 보는 일이 발생하지 않게 된다.

(2) 서비스 : 컴포짓 노드가 실행 되는 동안 계속해서 무언가를 주기적으로 실행하는 기능이다.

ex)만일 두 시간 동안 시험을 치르는 컴포짓이 있다고 가정할 때 30분마다 남은 시간을 알려주는 서비스를 장착하게 되면 시험을 치는 시간중에 30분 마다 남은 시간을 알려줄 수가 있게 된다.

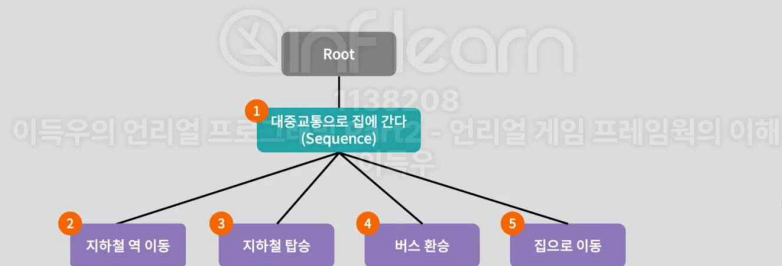
(3) 관찰자 중단 : 이 기능은 컴포짓 노드에서 실행되는 행동을 즉시 멈추고 처음부터 행동 트리를 루트에서부터 재실행할 때 유용하게 사용할 수가 있다.

ex) 만약에 긴급한 연락이 올 때까지 일한다는 설정이 있다고 했을 때 관찰자 중단 기능을 설정하지 않으면 일하는 중에 긴급한 연락이 와도 하는일을 멈추지 않고 일을 모두 끝낸 후에 연락을 하게 된다. 하지만 관찰자 중단 기능을 사용하면 하던일을 멈추고 루트에서부터 다시 실행하여 우선순위에 따라서 다른 일을 할 수 있도록 설정 할 수 있다.

## \*\*행동 트리 모델의 예시

### 행동 트리 모델의 예시

- 퇴근하고 집에 가는 행동 트리 모델의 설정
  - 지하철 역까지 이동한 후, 지하철을 타고, 버스를 환승한 후, 문 앞에 도착한다.
  - 시퀀스 컴포짓을 사용해 이들을 묶는다.



### 회사에서 퇴근하고 집에 가는 행동에 대한

\*회사에서 퇴근하고 집에 가는 행동에 대한 행동 트리 예시

-대중 교통을 이용해서 회사에서 집까지 가는 행동은 회사에서 집으로 가는 구역에 따라서 4가지 정도의 영역으로 구분할 수 있다.

- (1) 회사에서 지하철역까지 이동하는 코스
- (2) 회사의 지하철역에서 집 근처의 지하철역까지 이동하는 구역
- (3) 집근처 지하철역에서 내려서 집 근처 까지 가는 버스를 환승
- (4) 버스 정류장에서 집까지 이동하는 총 4가지 구역을 생각해볼 수 있음.

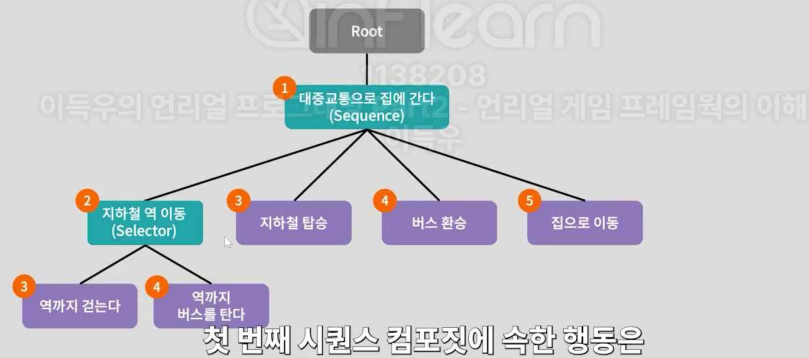
이들 행동은 모두 연결이 되어야 회사에서 집으로 도착할 수 있기에 시퀀스 컴포짓으로 묶어 주는 것이 좋다.



## \*\*행동 트리 모델의 예시

### 행동 트리 모델의 예시

- 트리가 가진 깊이를 활용해 단계별로 세부적인 행동을 설계한다.
- 퇴근하고 집에 가기
  - 1단계 : 지하철 역까지 이동한 후, 지하철을 타고, 버스를 환승한 후, 문 앞에 도착한다.
  - 2단계 : 셀렉터 컴포짓을 사용해 지하철 역까지 걸거나 버스를 타거나 둘 중의 하나를 선택한다.



첫 번째 시퀀스 컴포짓에 속한 행동은

\*첫 번째 시퀀스 컴포짓에 속한 행동은 다시 세부적으로 설계가 가능하다.

-만약에 회사에서 지하철역까지 이동하는 구역에 대한 행동은 회사에서 지하철역까지 걸거나 회사에서 버스를 타서 지하철역까지 가는 행동 중 하나를 선택하도록 설계할 수도 있다. 이런 경우에는 두가지 행동중 하나만 선택하도록 셀렉터 컴포짓을 추가해주고 이것을 사용하여 걷는 행동과 버스를 타는 행동을 묶어줄 수 있다.

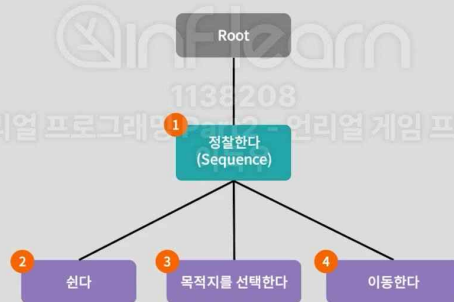
-이렇게 하여 트리의 특정 과제에서 필요한 만큼 추가적으로 깊이를 발전하면 보다 상세한 행동을 우리가 설계할 수가 있게 된다.

\*\*\*NPC전투를 위한 행동트리 모델 예시

\*\*행동 트리 모델의 구현 예시

## 행동 트리 모델의 구현 예시

- 전형적인 RPG 게임 NPC에 대한 행동 트리 모델 예시
- 쉬기(IDLE)와 정찰(PATROL)에 상태에 대한 행동 트리



-전형적인 RPG게임의 NPC에 대한 행동 트리 모델을 설계할 예정임.

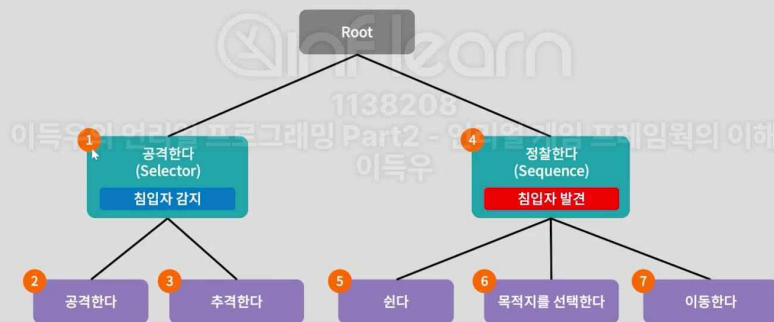
-먼저 NPC의 경우 정찰한다 라는 컴포짓 노드가 있는데 이는 시퀀스 노드로 구성이 되어 있다.

-이렇게 되면 쉬는 애니메이션과 정찰 애니메이션에 대응하는 행동 트리 모델을 설계할 수가 있게 된다.

## \*\*행동 트리 모델의 구현 예시

### 행동 트리 모델의 구현 예시

- 전형적인 RPG 게임 플레이어에 대한 행동 트리 모델 예시
- 추가적으로 추격(CHASE)과 공격(ATTACK) 상태에 대한 행동 트리



먼저 공격한다라고 하는 컴포짓 노드를 추가하고

-공격 하는 기능을 추가

-‘공격 한다’ 라고 하는 컴포짓 노드를 추가하고 이것에 대한 우선순위를 높여서 왼쪽에 배치한다.

-이 ‘공격 한다’라고 하는 컴포짓 노드의 경우 공격을 하거나 감지된 침입자를 추격하는 두가지 행동으로 구성이 되어 있는데 이렇게 침입자를 현재 감지하고 있는 상황인지에 대한 데코레이터 노드를 사용하여 컴포짓 노드에 부착을 한다. 이렇게 하면 NPC가 침입자를 감지하고 있는 동안에는 계속해서 공격한다 컴포짓 노드가 실행이 될 것이고 이것은 셀렉터로 구성이 되어 있어 침입자와의 거리에 따라 공격을 하거나 추격을 하는 두 가지 액션중 하나를 선택하여 실행을 할 것이다.

-기존의 ‘정찰한다’라는 컴포짓 노드의 경우에는 관찰자 중단 노드를 추가해줘야지만 쉬고 정찰지로 이동하는 동안에 침입자가 발견이 되면 그 즉시 정찰하는 컴포짓 노드를 중단하고 다시 루트에서부터 판단을 수행할 것이다.

-이렇게 하여 기본적으로 RPG게임에 사용하는 행동트리 모델을 살펴봄

**\*\*실습**

-Artificial Intelligence에 있는 blackboard는 인공지능 모델에서 의사결정을 하기 위한 기본 데이터를 제공하는 데이터 저장소를 의미한다.

\*\*정리

## 행동 트리 모델의 이해

---

1. 행동 트리 모델이 가지는 장점
2. 행동 트리 모델의 구성 요소와 설계 방법
3. 언리얼 엔진의 행동트리를 구동시키기 위한 기본 설정



1138208

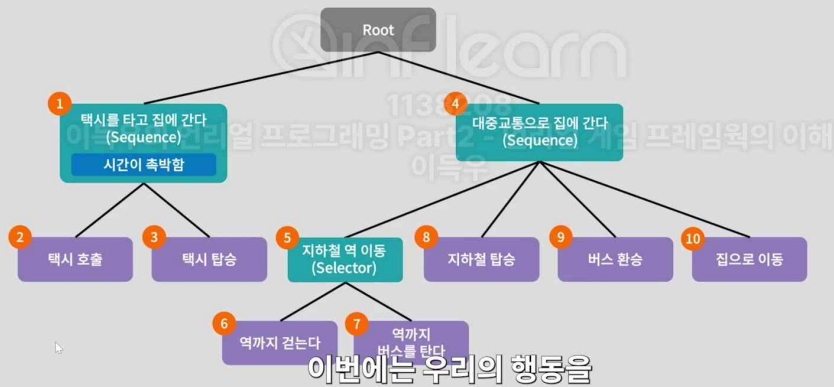
이득우의 언리얼 프로그래밍 Part2 - 언리얼 게임 프레임워크의 이해  
이득우

행동 트리 모델에 대해서 살펴보았는데요

\*\*컴포짓의 데코레이터 설정(Root노드 아래에 바로 Selector노드가 설정된다는 가정에 설명을 진행)

## 컴포짓의 데코레이터 설정

- 컴포짓에 조건을 걸어 선택의 폭을 넓힐 수 있도록 확장한다.
- 우선 순위가 높은 컴포짓을 왼쪽에 배치하고 데코레이터를 설정한다.



\*퇴근 후에 집에서 즐겨보는 TV드라마의 본방을 사수 하는 취미 활동이 있다고 가정 그런데 잔업이 생겨 본방까지의 시간이 계속 촉박해 졌다고 가정하자. 따라서 기존에 작업한 행동트리를 확장하여 이에 대한 상황을 대처하기 위한 행동 트리를 설계해 보자.

-먼저 시간이 촉박한지를 판단하는 데코레이터와 택시를 타고 집에 빨리 가는 새로운 행동을 추가함.

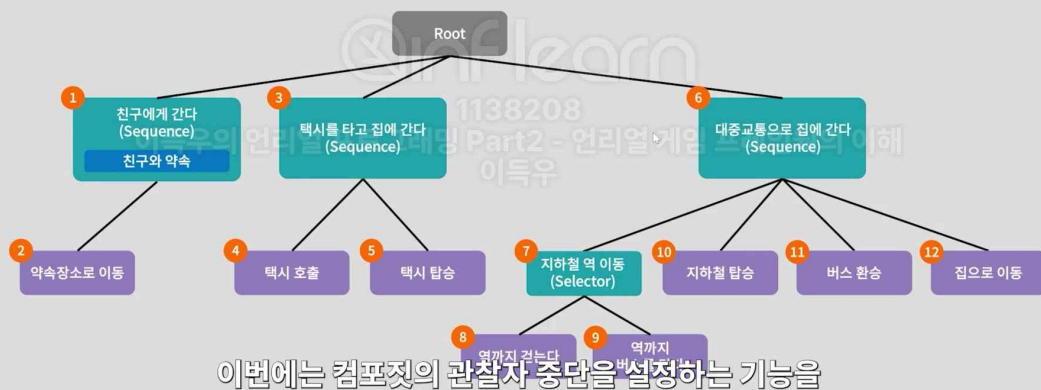
-시간이 촉박한 경우에는 택시를 타고 빨리 집으로 가야 하기 때문에 이것이 더 우선순위가 높다.(택시를 타고 집에 가는 상황을 컴포짓 노드 왼쪽에 배치) 여기에는 시간이 촉박한지 판단하는 데코레이터 추가 데코레이터를 만족하면 택시를 호출하고 택시가 도착하면 바로 탑승하여 집으로 가는 행동 수행.

-택시를 호출하는 행동이 실패하면 해당 노드는 실패를 띄우고 바로 우선순위를 다음 컴포짓 노드로 이동을 시켜주고 어쩔 수 없이 대중교통으로 집에가는 컴포짓이 실행이 된다.

**\*\*컴포짓에 관찰자 중단(Abort) 설정**

## 컴포짓에 관찰자 중단(Abort) 설정

- 비상 상황에 대한 처리. 예) 친구로부터 함께 놀자는 연락이 왔다.



\*집으로 이동하는 중에 친구로부터 연락이 왔다 퇴근 이후에 친구와 노는 것은 드라마를 떠나서 지금으로서는 가장 주요한 일이기 때문에 집으로 이동하는 모든 행동을 중지하고 친구한테 가야한다.

-왼쪽에 가장 높은 우선순위로 친구에게 가는 컴포짓 노드를 배치함.

-그래서 최초에 친구에게 약속이 있었다면 집으로 가지 않고 약속장소로 이동하는 행동이 수행이 됨.

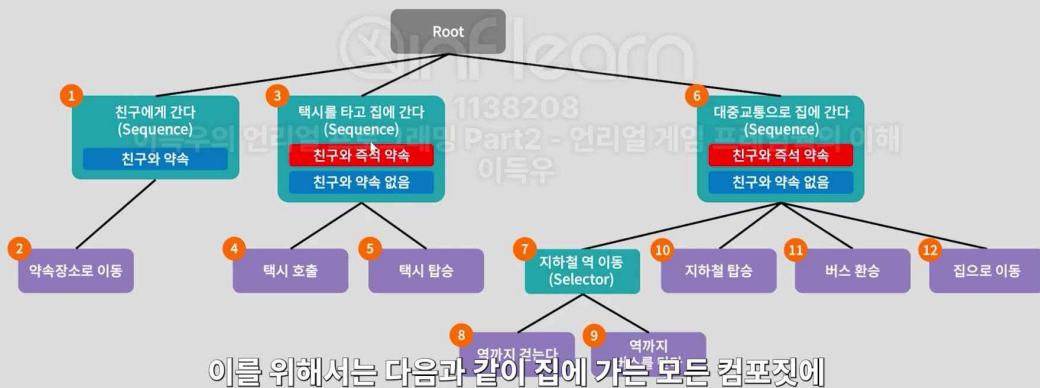
-그런데 퇴근시간 전까지 약속이 없어서 집으로 가는데 갑자기 친구로부터의 연락이 왔다. 하지만 지금까지의 행동 트리 설계상으로 집으로 도착해야지만 모든 액션이 종료되고 다시 루트로부터 판단을 수행하게 된다. 그렇게 되면 이미 집에 도착할때까지 많은 시간이 흘러 친구와 놀 수 없게 된다.

-그래서 이 행동 트리 모델을 바꿔서 집에 가는 길을 멈추고 즉시 친구에게 가도록 설정해주자.

\*\*컴포짓에 관찰자 중단(Abort) 설정

## 컴포짓에 관찰자 중단(Abort) 설정

- 비상 상황에 대한 처리. 예) 친구로부터 함께 놀자는 연락이 왔다.
- 현재 진행 중인 컴포짓을 즉시 취소하고 루트에서부터 다시 점검한다.



\*퇴근시간 전까지 약속이 없어서 집으로 가는데 갑자기 친구로부터의 연락이 왔다. 하지만 지금까지의 행동 트리 설계상으로 집으로 도착해야지만 모든 액션이 종료되고 다시 루트로부터 판단을 수행하게 된다. 그렇게 되면 이미 집에 도착할때까지 많은 시간이 흘러 친구와 놀 수 없게 된다.

-위 그림과 같이 모든 컴포짓에 친구와 약속이 없다는 조건을 부착을 하고 친구로부터의 연락이 발생하면 현재 집으로 가는 모든 행동을 중지하도록 관찰자 중단 기능을 설정해준다.

-이렇게 설계하면 집으로 가는 중에 친구와 연락이 오면 그 즉시 모든 행동을 취소하고 루트에서부터 다시 의사결정을 진행하게 된다.