



Behavior Tree : Service, Decorator

1. Decorator (데코레이터)

- **역할:** 데코레이터는 특정 조건에 따라 행동 트리의 분기를 제어
즉, 트리의 특정 노드나 브랜치가 실행될지 여부를 결정
- **위치:** 데코레이터는 행동 트리의 특정 노드에 첨부
- **조건 평가:** 데코레이터는 조건을 평가하고, 조건이 만족되면 해당 노드를 실행시키고, 그렇지 않으면 실행을 막음.
- **주 사용 예:** 조건에 따라 행동을 실행하거나 실행하지 않을 때 사용
예를 들어, 플레이어가 특정 거리 내에 있을 때만 공격 행동을 실행하는 경우, 데코레이터가 이 조건을 평가합니다.

2. Service (서비스)

- **역할:** 서비스는 정기적으로 실행되어 행동 트리에 정보를 업데이트하거나 특정 작업을 수행
- **위치:** 서비스는 행동 트리의 특정 브랜치(노드 계층 구조)에 첨부
- **정기적 실행:** 서비스는 설정된 시간 간격마다 주기적으로 실행, 이 시간 간격은 서비스의 설정에 따라 조정할 수 있음
- **주 사용 예:** 지속적으로 상태를 체크하거나 행동 트리를 업데이트할 때 사용
예를 들어, 적의 위치를 계속 추적하여 블랙보드 값을 업데이트하는 경우 서비스가 이를 처리

차이점 요약

- **조건 평가 vs. 주기적 업데이트:**
 - **데코레이터**는 조건을 평가하여 노드의 실행을 제어
 - **서비스**는 정기적으로 실행되어 상태를 업데이트하거나 특정 작업을 수행

- 위치와 실행 빈도:

- **데코레이터**는 특정 노드에 붙어서 그 노드의 실행 여부를 결정합니다.
- **서비스**는 브랜치에 붙어서 일정 시간 간격마다 실행됩니다.

예시

데코레이터 예시

플레이어가 100 미터 이내에 있을 때만 공격 행동을 실행:

```
plaintext코드 복사
Selector
├─ Sequence (Attack)
│   └─ Decorator: PlayerInRange(100)
│       └─ Task: PerformAttack
```

- 여기서 **Decorator: PlayerInRange(100)** 은 플레이어가 100 미터 이내에 있는지 조건을 평가,
조건이 참일 때만 **PerformAttack** 작업이 실행됩니다.

서비스 예시

적의 위치를 지속적으로 업데이트:

```
plaintext코드 복사
Selector
├─ Sequence (Chase)
│   └─ Service: UpdateEnemyLocation
│       └─ Task: MoveToEnemy
```

- 여기서 **Service: UpdateEnemyLocation** 은 주기적으로 실행되어 블랙보드에 적의 위치를 업데이트→ 이를 통해 **MoveToEnemy** 작업이 항상 최신 정보를 기반으로 동작