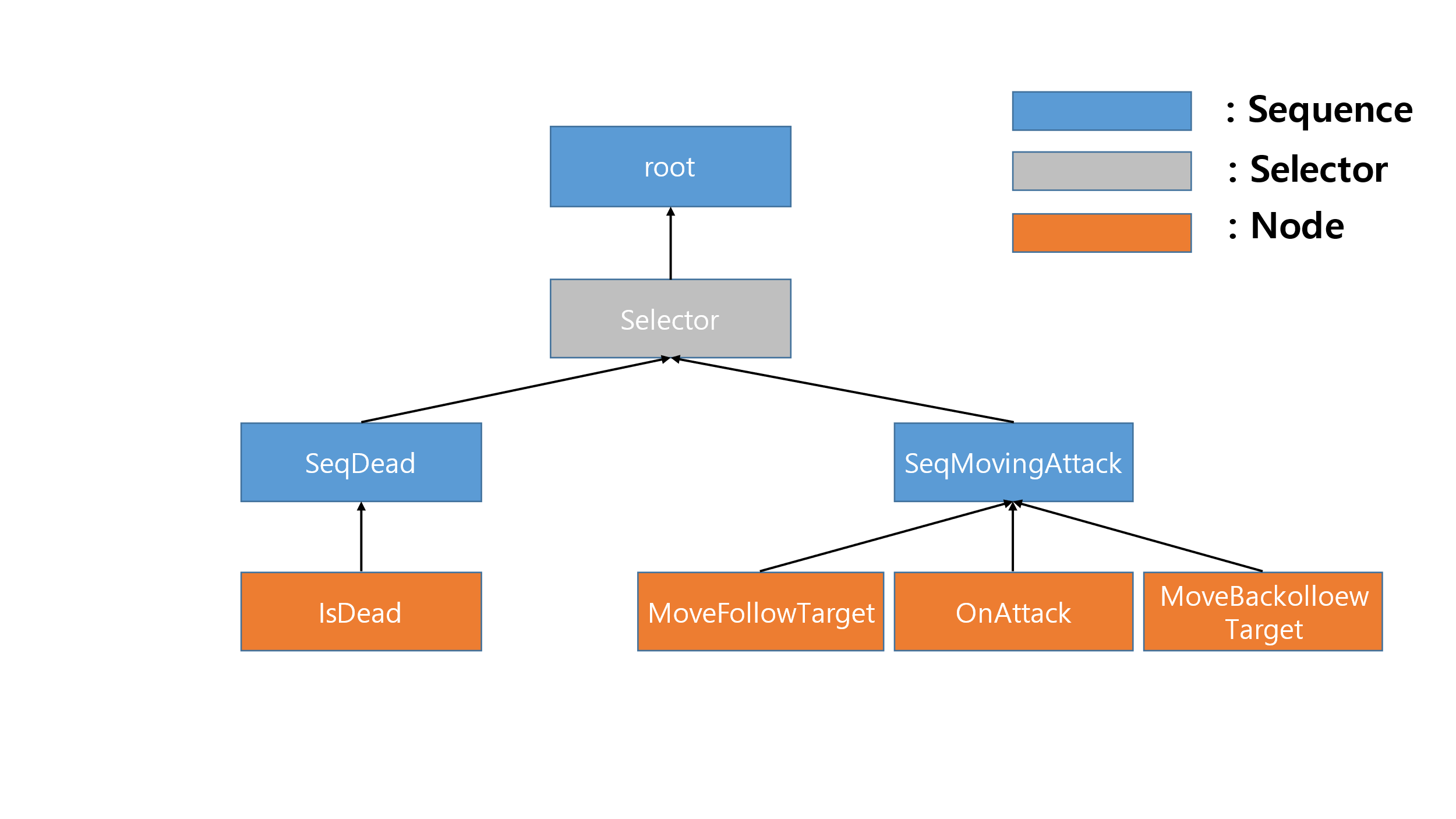
|  |
| --- |
| MoveFollowTarget() |
| INPUT : *EnemyHP, HeroPos, speed*  IF *EnemyHP* >=50 do;  Dir <- *HeroPos* – this.position  Dir.Normalize()  Rot <- LookRotation(Dir).Angley  This.Rotation <- Rot  this.position <- this.position + Dir \* *speed*  RETURN TRUE  End IF  RETURN FALSE: |



적 AI는 위 그림과 같은 구조로 형성되었다.

Selector가 적이 죽음 조건부터(SeqDead) 체크하고, 죽었다고 판단되면 root에게 전달하여 적을 소멸시킨다. 죽음 조건이 충족되지 않았다면 Selector가 다음 시퀀스로(SeqMovingAttack) 넘어가 MoveFollowTarget, MoveBackollowTarget, AddBullet( OnAttack 노드에서 실행되는 함수 ) 함수들 중 조건에 알맞는 함수를 실행시킨다.

1수도코드는 MoveFollowTarget 함수에 대해 설명한다. MoveFollowTarget 함수는 적의 HP가 50이상일때 적이 플레이어를 쫓아가는 처리를 하기 위한 함수이다.

EnemyHp는 적의 HP를 의미한다.

HeroPos는 플레이어의 위치를 의미한다.

Speed는 적의 속력을 의미한다.

Normalize()는 벡터를 단위벡터화하는 함수이다.

|  |
| --- |
| MoveBackollowTarget() |
| INPUT : *EnemyHP, HeroPos, speed*  IF *EnemyHp* < 50 && *EnemyHp*>0 do;  Dir <- *HeroPos* - this.position  Dir.Normalize()  Rot <- LookRotation(Dir).Angley  this.Rotation <- Rot  this.position <- this.position - Dir \* *speed*  RETURN TRUE  End IF  RETURN FALSE: |

2수도코드는 MoveBackollowTarget 함수에 대해 설명한다. MoveBackollowTarget 함수는 적의 HP가 0과 50 사이일때 적이 플레이어에게서 멀어지는 처리를 하기 위한 함수이다.

EnemyHp는 적의 HP를 의미한다.

HeroPos는 플레이어의 위치를 의미한다.

Speed는 적의 속력을 의미한다.

Normalize()는 벡터를 단위벡터화하는 함수이다.

|  |
| --- |
| IsDead() |
| INPUT : *EnemyHp*  IF *EnemyHp* <= 0 do;  RETURN FALSE  End IF  RETURN TRUE |

|  |
| --- |
| AddBullet() |
| INPUT : *EnemyHp, Dir, nTime*  IF *EnemyHp*> 0 do;  IF *nTime* %100 ==0 do;  bullet <- Prefab\_bullet  bullet.Dir<- this.*Dir*  bullet.parent <- null  bullet.layer <- "Enemy"  bullet.position <- this.position  RETURN TRUE  End IF  End IF  RETURN FALSE: |

3수도코드는 IsDead 함수에 대해 설명한다. IsDead 함수는 적의 HP가 0이하가 될 시 false를 반환하며, 이는 IsDead 노드에서 true로 변환되어 selector에게 전달하는 기능을 가진다.

EnemyHp는 적의 HP를 의미한다.

4수도코드는 AddBullet 함수에 대해 설명한다. AddBullet 함수는 적이 살아있을 때 일정 주기마다 총알을 발사하는 처리를 하기 위한 함수이다.

EnemyHp는 적의 HP를 의미한다.

Dir은 적이 바라보고 있는 방향을 의미한다.

nTime은 게임이 시작된 후 일정주기로 1씩 증가한다.