

졸업프로젝트

(이상한 실험실)

보고서

B677034

조광형

목차

1. Stage 플로우 차트

1-1 Stage 1

1-2 Stage 2

1-3 Boss Stage

2. 클래스 다이어그램

2-1 Characters

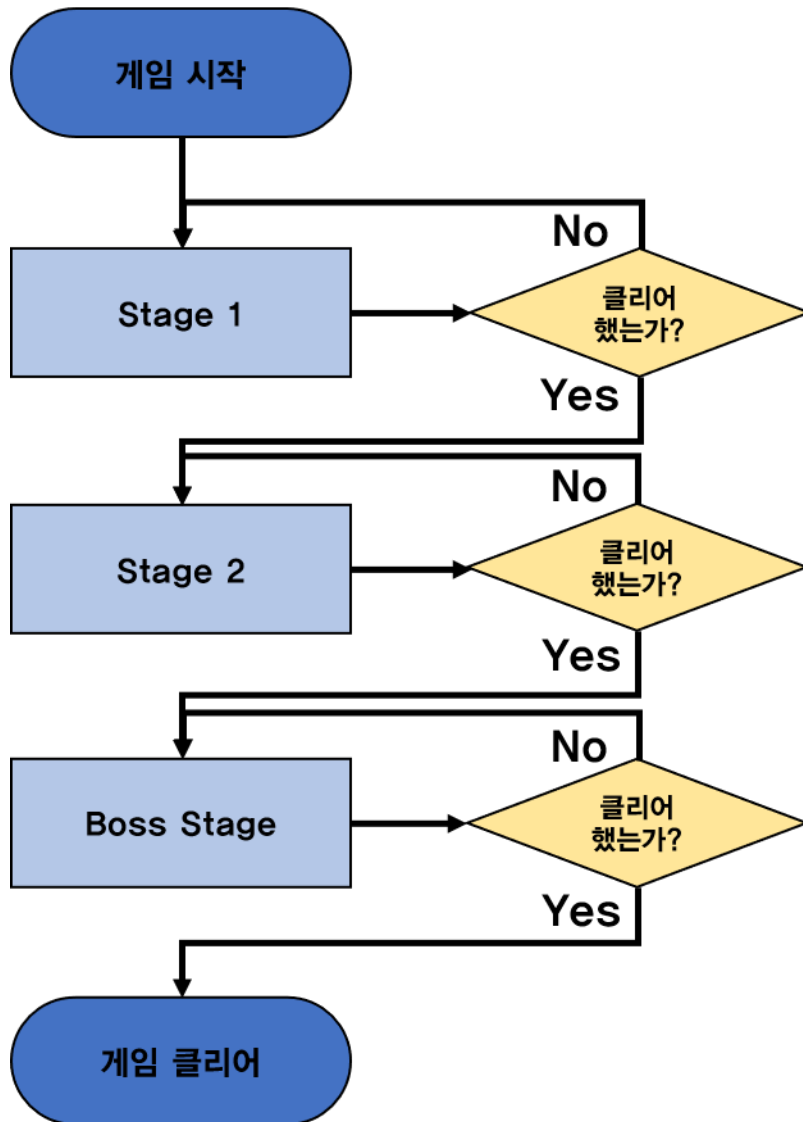
2-2 Weapon

2-3 Stat

2-4 Enemy AI

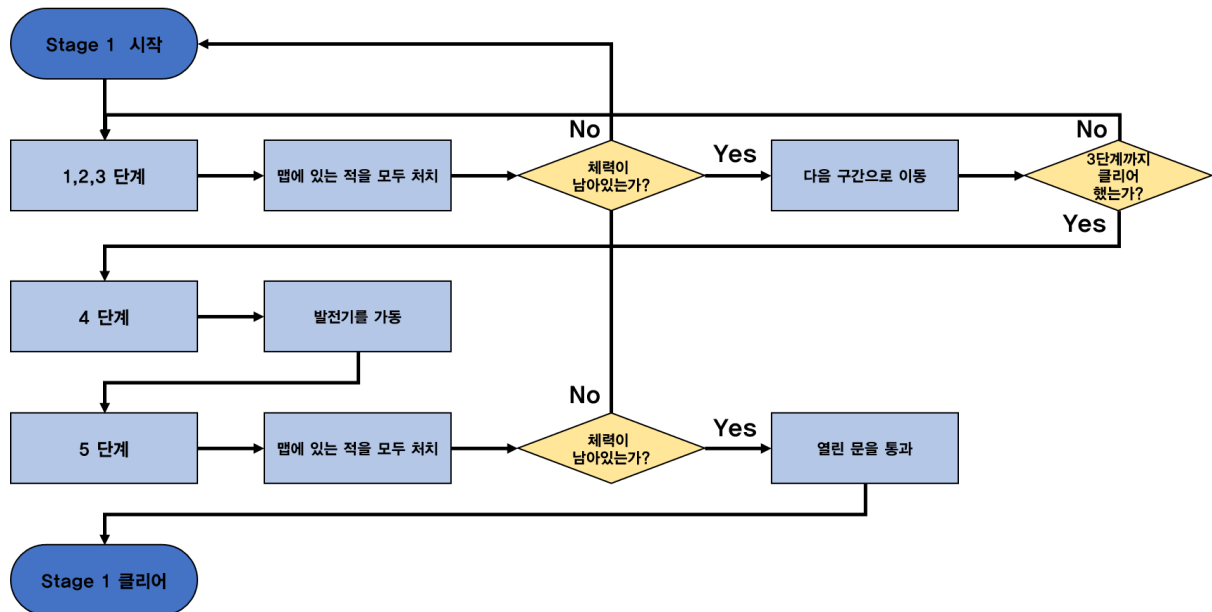
2-5 Stage Controller

1. Stage 플로우 차트



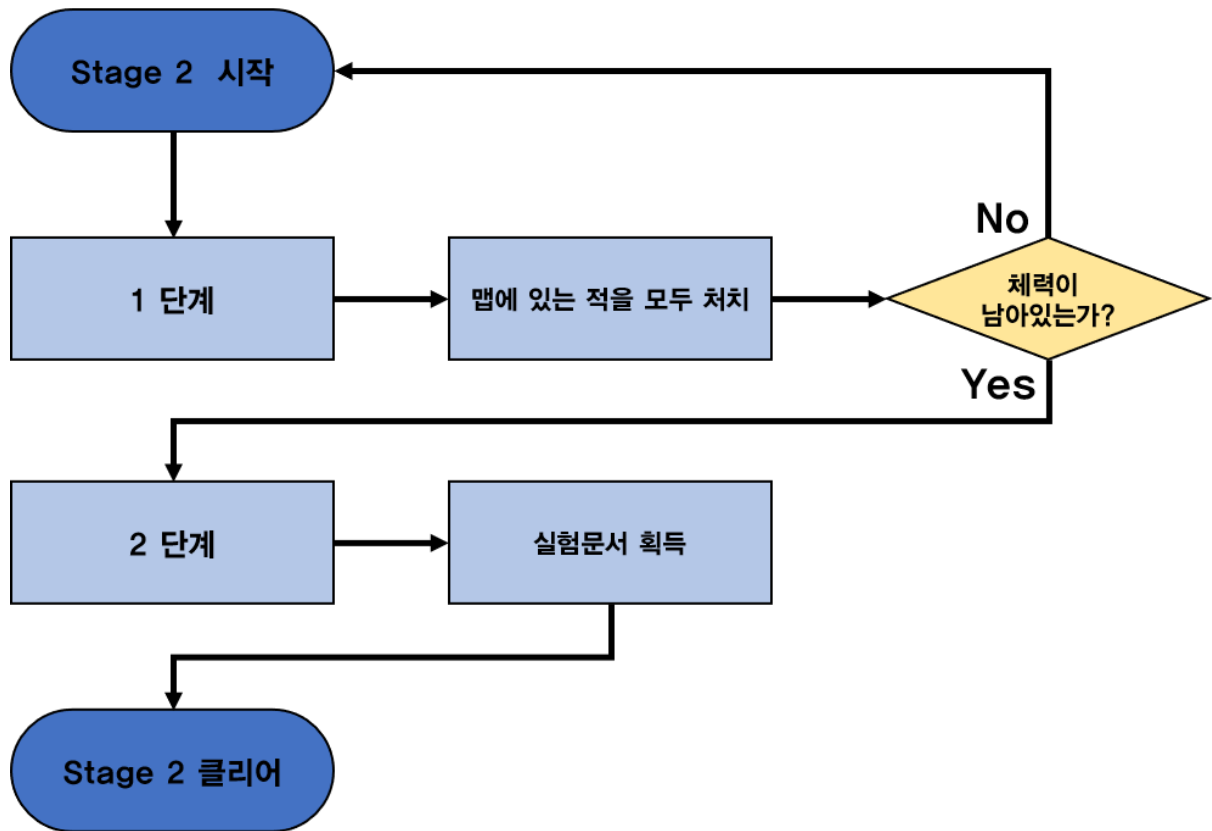
총 3가지 스테이지로 해당 스테이지에서 클리어 하지 못하면 해당 스테이지에서 다시 플레이 하게 됩니다.

1-1 Stage 1



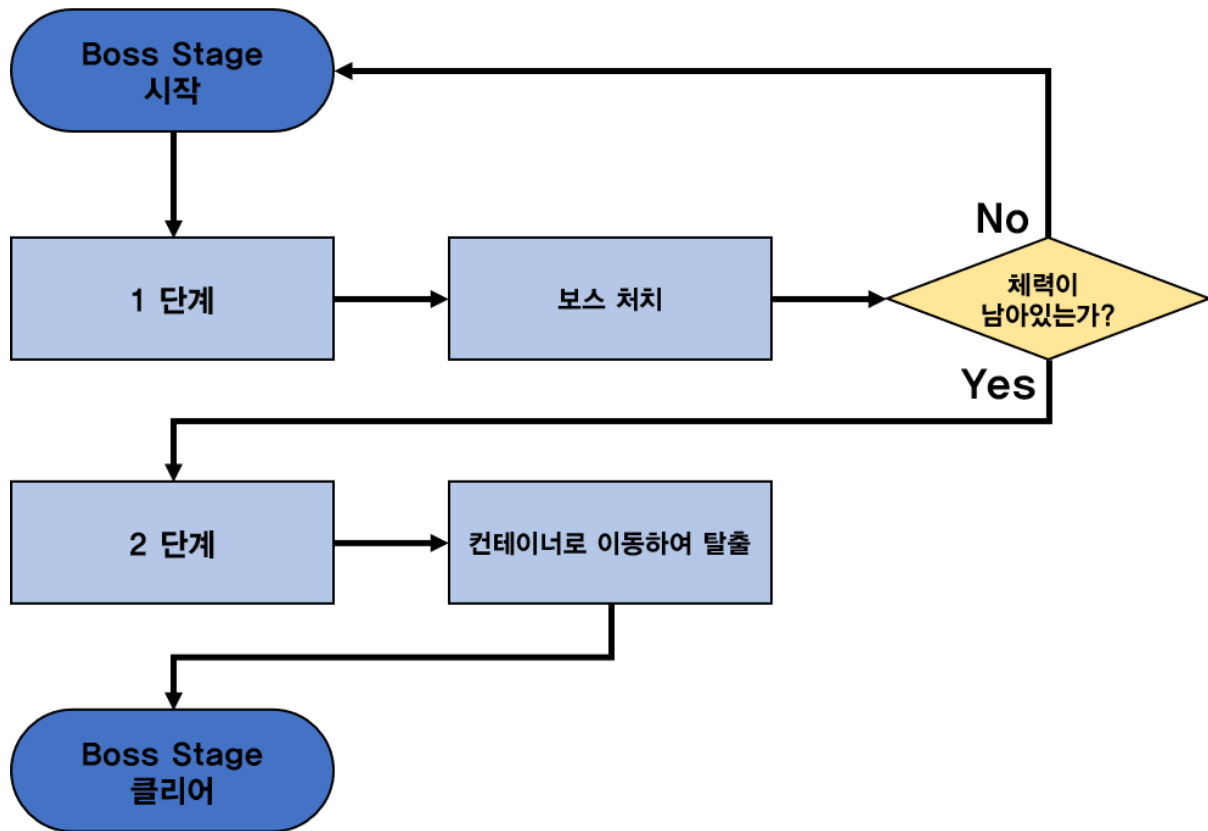
Stage 1은 원거리 공격을 하는 적이 등장하며 주어진 무기를 사용해 맵에 있는 모든 적을 처치 후 다음 구간으로 넘어가는 단계가 대부분이며 플레이어의 무기의 정확도 및 숙련도를 늘리기 위하여 구성되어 있습니다.

1-2 Stage 2



Stage 2는 근접 공격을 하는 적과 폭발하는 적으로 구성되어 있으며 밀려오는 적을 모두 처치한 후 실험문서를 획득하여 클리어하는 곳입니다. 폭발하는 적은 미리 처치한다면 달려와서 터지지 않지만 미리 처치하지 못한다면 플레이어의 위치로 달려와 폭발하게 됩니다. 폭발 후 산성 안개가 펼쳐지며 안개 안에 위치하게 되면 중독 데미지를 입게 됩니다.

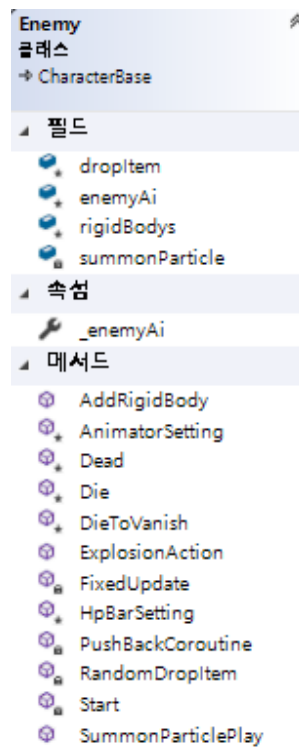
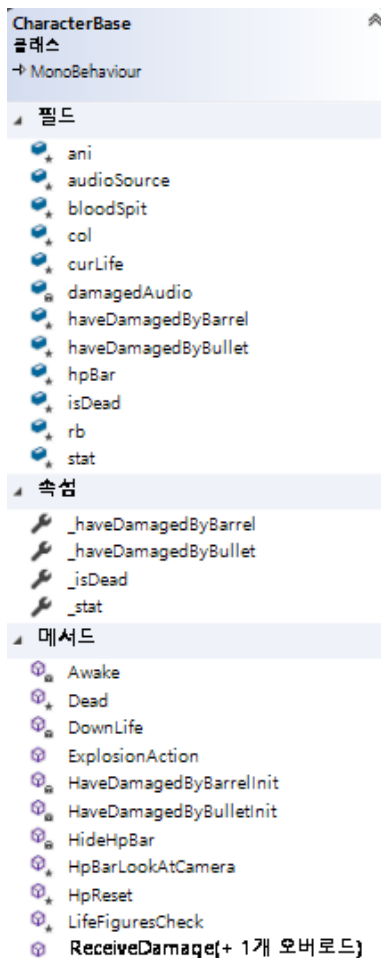
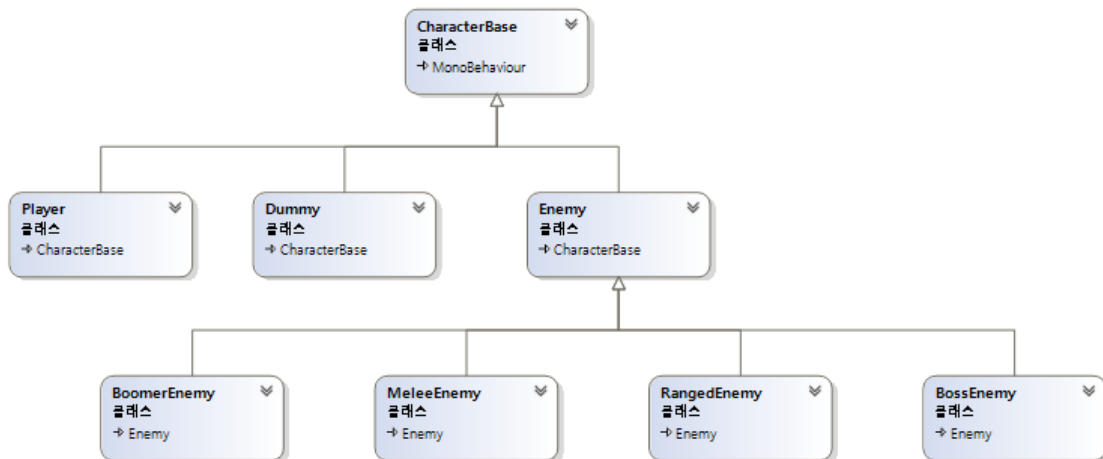
1-3 Boss Stage



Boss는 총 3가지 패턴으로 일정 거리에 위치하면 산성 공격, 일정 거리를 벗어난 위치라면 점프 공격, 돌진 공격을 합니다. 공격을 한 후 쉬는 시간을 가져 플레이어가 공격할 수 있는 시간이 주어집니다.

2. 클래스 다이어그램

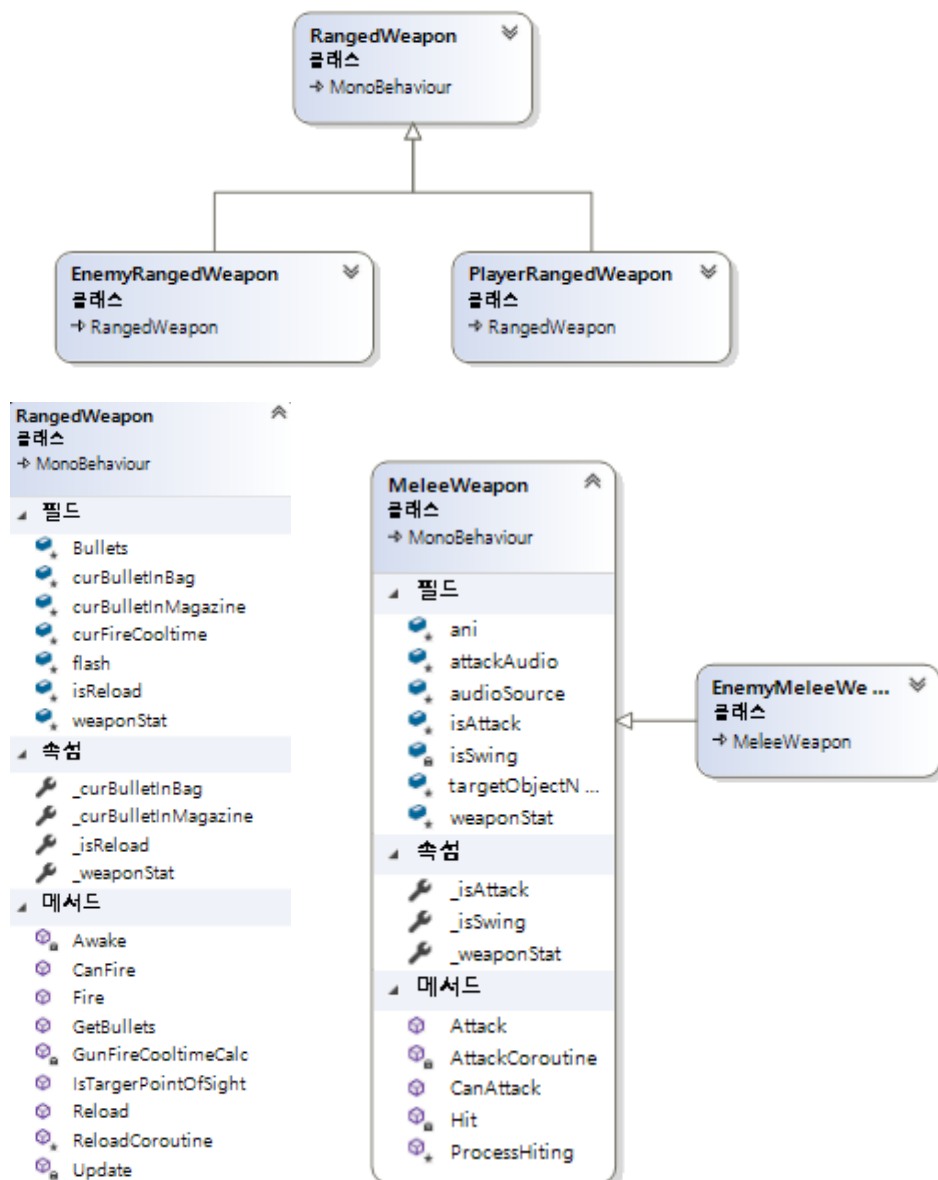
2-1. Characters



Character들이 데미지 받는 것, 죽은 것, hpBar 등과 같이 공통되는 변수 및 함수가 있어 CharacterBase와 다른 Character와의 상속 관계로 코드를 했습니다.

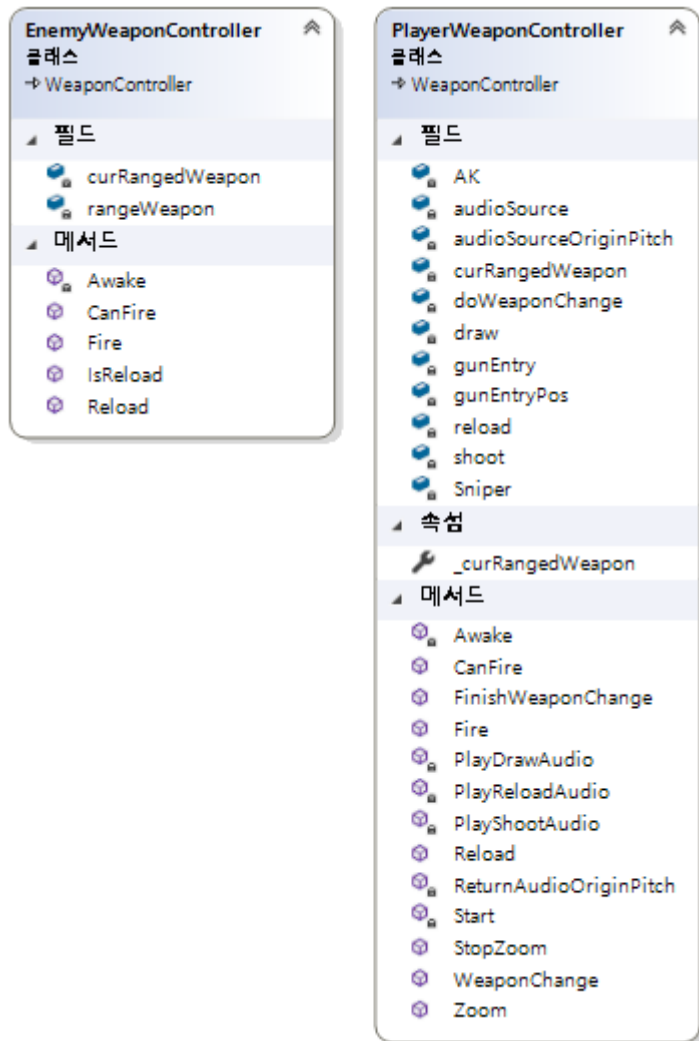
Enemy들은 NavMesh로 이루어져있고 RagDoll로 구현이 되어있기 때문에 그에 따라 공통 되는 것들을 위해서 부모 클래스와 자식 클래스로 구성을 했습니다. NavMesh와 Rigidbody를 동시에 사용하면 충돌이 일어나기에 Enemy의 Rigidbody의 isKnematic을 true로 두었다가 죽었을 경우에 false로 돌려 RagDoll의 기능이 나올 수 있도록 구현했습니다. Enemy에서 같은 함수지만 다른 작용을 해야하는 경우에는 virtual, override를 통해 구현을 했습니다.

2-2. Weapon



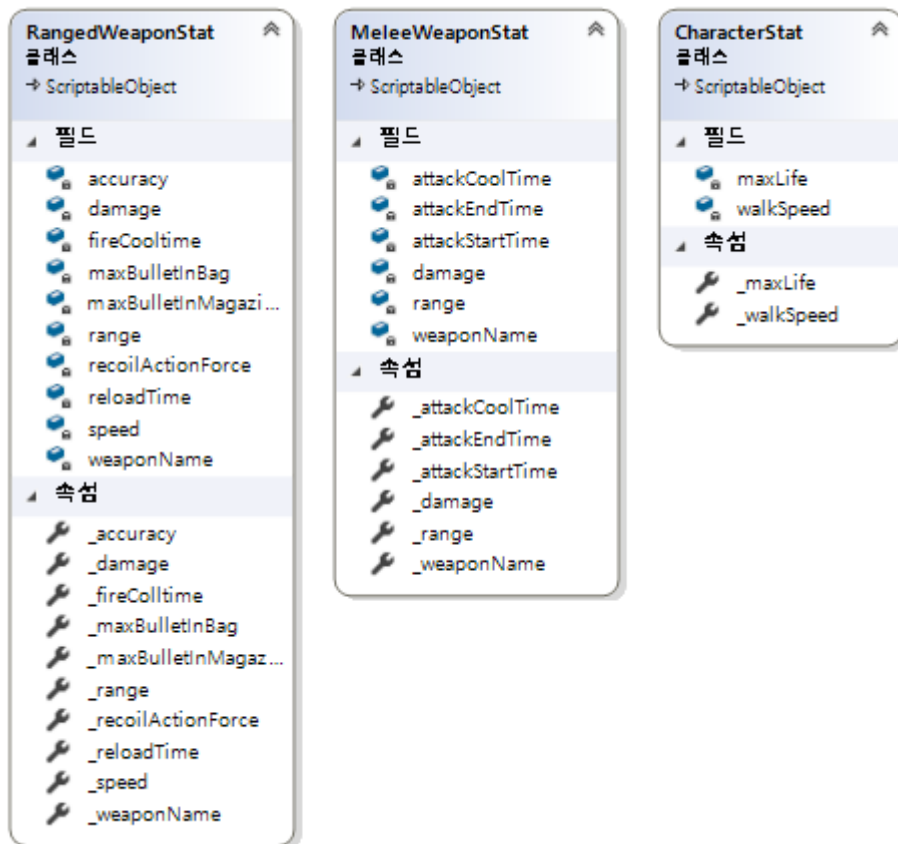
플레이어와 Enemy가 사용하는 Weapon의 기본 구조가 같기에 부모 클래스를 작성하여 그

에 따른 자식 클래스로 구현했습니다. Player의 근접공격은 기획 단계에서 예정에 없어서 구현을 하지 않았지만 나중에 구현이 필요하다면 쉽고 빠르게 구현을 하기 위하여 MeleeWeapon 스크립트를 부모로 받아 스크립트를 짤 수 있도록 했습니다.



WeaponController는 Enemy와 Player의 공통된 부분을 찾고 그에 맞게 구현을 하려 했으나 공통된 부분이 많지 않아 따로 작성하였고 필요하다면 부모 클래스를 구현하여 상속을 받게 할려고 합니다.

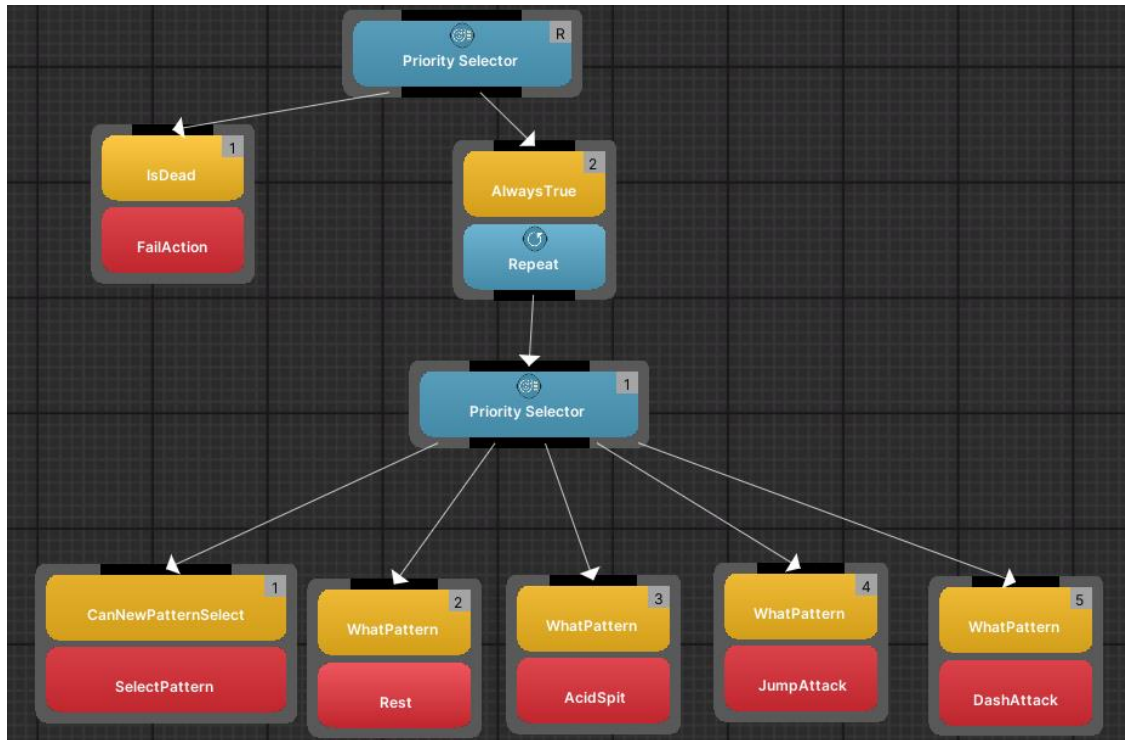
2-3. Stat



여러 무기나, 캐릭터들이 만들어졌을 때 일일이 스탯을 인스펙터 창 또는 스크립트에서 넣기보다는 `ScriptableObject`를 사용하여 구현하는게 편하고 좋을 것이라 판단되어 Stat들을 `ScriptableObject`로 구현을 했습니다. 변수들은 전부 `private`으로 선언하여 값을 임의적으로 바꾸는 것을 막았고 `Get` 지정자를 사용하여 Stat에 있는 값을 받을 수 있도록 구현했습니다.

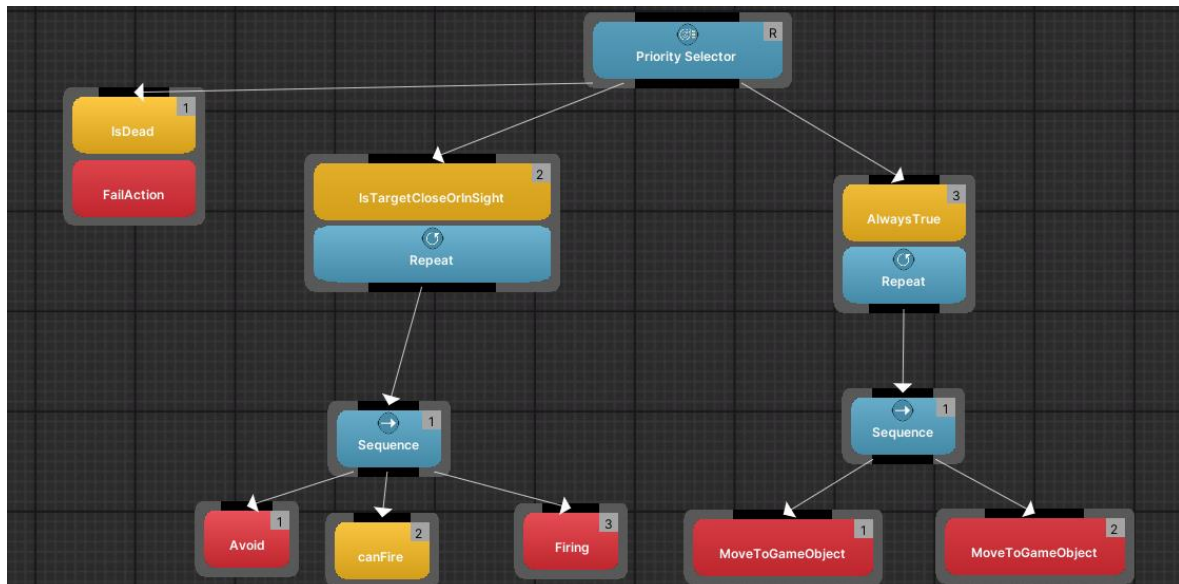
2-4. Enemy AI

1) 보스



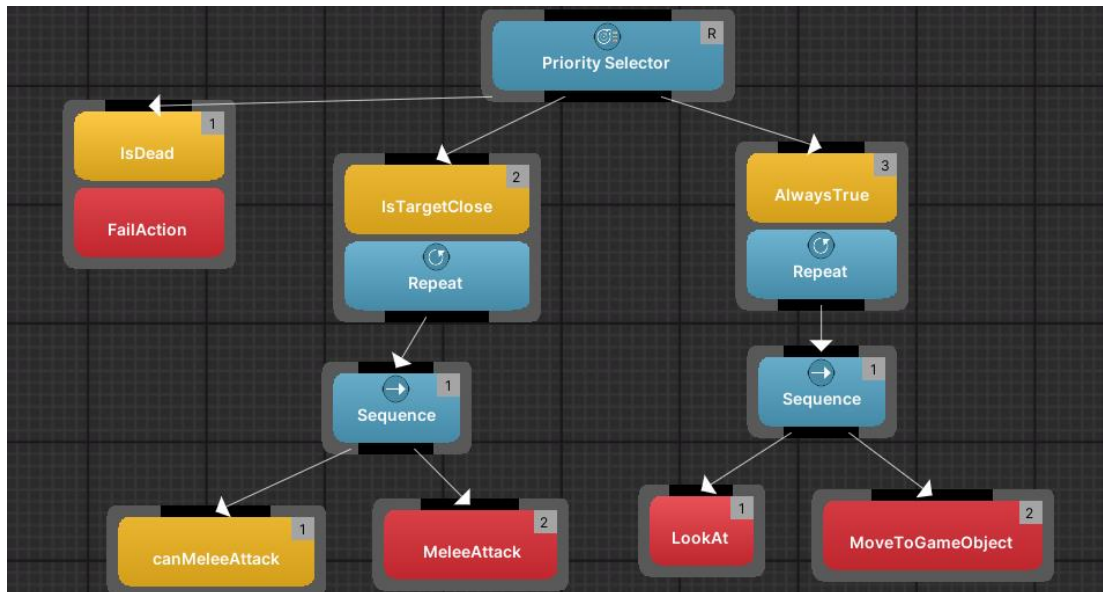
Pattern을 랜덤으로 생성한 후 그에 따른 패턴으로 구성된 비헤이비어 트리입니다. 오브젝트가 죽었는지 안 죽었는지를 우선적으로 나눈 후 살아있을 경우 현재 선택된 패턴이 종료 될 때 패턴을 선택하여 그 패턴에 따른 노드로 넘어가 노드에 있는 함수들이 실행됩니다.

2) 원거리 공격 적



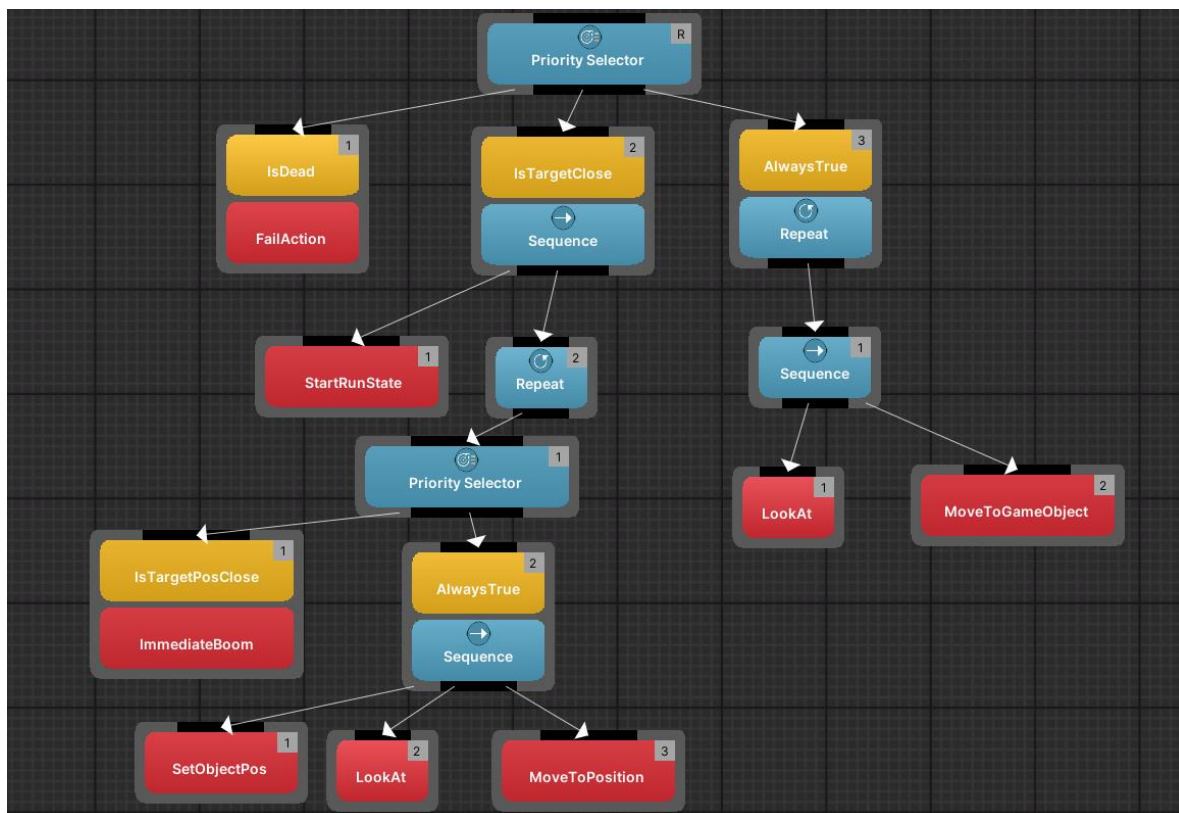
총 3가지 서브트리로 구성이 되며 타겟(플레이어)이 시야에 있거나 또는 일정거리 안에 존재한다면 사격하고, 사격 쿨타임일 경우에는 피하게 구성이 되어있고 타겟이 시야 또는 일정거리 안에 존재하지 않다면 미리 정해진 순찰구역을 순찰하게 됩니다.

3) 근접 공격 적



플레이어를 일방적으로 따라다니고 플레이어가 일정거리에 위치한다면 공격 쿨타임에 맞춰 공격을 하게 됩니다.

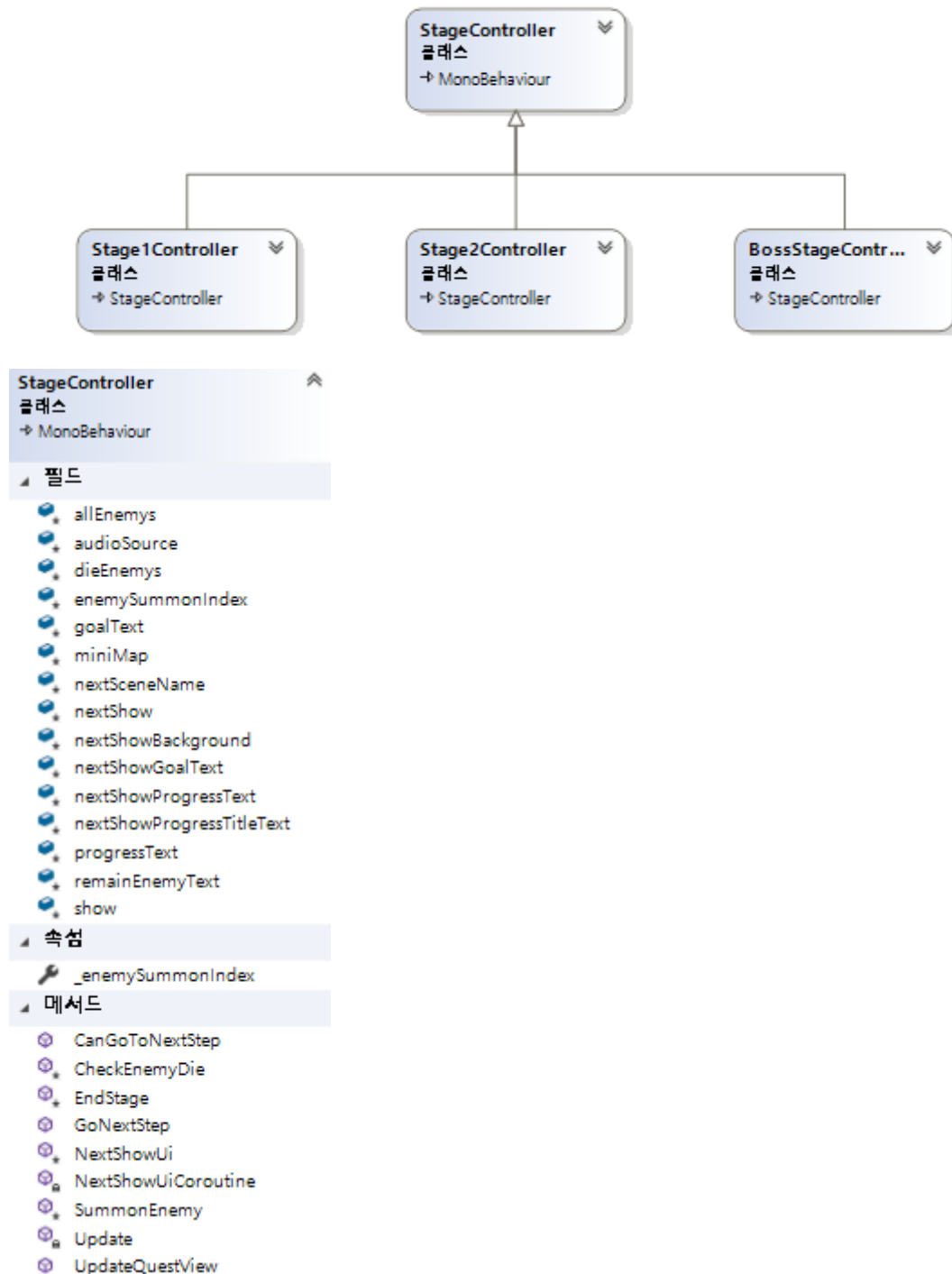
4) 폭발 적



플레이어를 따라다니다가 일정거리에 도달하면 이동속도가 빨라지게 됩니다. 이동속도가 빨라진 상태로 플레이어의 최근 위치에 도달하게 되면 폭발을 하며 산성 안개를 뿌리게 됩니다. 플레이

어에게 접근하기 전에 체력이 0이 되면 그 자리에서 폭발을 합니다.

2-5. Stage Controller



각 스테이지 별 구성도가 비슷하기에 상속을 통해 비슷한 부분을 재사용할 수 있도록 코드를 구현했습니다. 구성도는 비슷하나 그 안에 세세한 내용을 다르기에 세세한 내용을 포함한 함수에는 **override** 키워드를 통해 구현했습니다.