

### 크로스헤어 부분

- CrossHair.TBA import
- 위젯 블루프린트를 생성하여 Image로 CrossHair를 할당
- 앵커를 가운데로 설정 후 위치를 (0,0)으로 조절
- 정사각형의 Size를 조절하기 위하여 Size to Content로 조절한다. 이미 이미지에서 정사각형으로 설정이 되어있기 때문에 가능
- 이미지와 앵커가 잘 맞지않기 때문에 Alignment를 (0.5,0.5)로 수정, Pivot과 같은 기능을 수행하지만 Widget의 위치를 수정하는 것이기 때문에 다른 듯?
- PlayerBase에서 CrossHair는 항상 쓰기 때문에 계속 표시하기 위하여 Begin Play이벤트를 호출하여 Create Widget노드를 생성한다.
- 이때 Player 자신에게 넣어야하는데 Get Player Controller에 넣는다.
- Player Controller는 0번 Index가 항상 자기 자신이기 때문에 0번을 넣는다.
  - \* 왜 이렇게 할까?
    1. 데디케이티드 서버 기반이기 때문에 각각의 Client를 Clone하여 다른 Client에 넣어 줘야하기 때문?
- Return값을 변수로 승격시킨다.
- 그 후 Aiming 부분에서 True시에 Add to Viewport로 화면에 출력시킨다.
  - \* 이 때 Create Widget부분은 생성한 WidgetBP\_CrossHair를 받는 부분이고 이것은 PlayerBase로 가져와서 Set하여 화면에 보여주는 부분이 Add to Viewport인 듯
- Aiming이 False시에 Widget변수를 가져와서 Remove from Parent 노드를 생성한다.
  - \* 여기서 추측할 수 있는 것은 Widget은 하나의 클래스로써 PlayerBase에 어떠한 경로로 상속을 하여 쓰인다는점
  - \* 정확히 말하자면 WidgetBP\_CrossHair는 PlayerBase에 friend효과 내지는 상속에 가까운 기능을 가진다?

### 카메라 부분

- PlayerBase의 카메라 Component 부분에서 FOV 설정 가능
- Aiming시에 FOV가 확대 되게끔 하기 위하여 Camera Component를 가져와 Set FOV를 하고 False시에 다시 원상복구 시킨다.
- 너무 프레임이 없이 바로 실행되기 때문에 부드러운 느낌을 주기 위하여 Animation을 넣어야 하는데 이 때 TimeLine 기능을 사용하여 넣는다.
- 이 때 설계를 1초에 걸쳐서 FOV가 40에서 70으로 설정하게끔 하는 것을 목표로 한다.
- TimeLine 노드 생성시에 그래프가 나오는데 전체 길이를 1로 설정 후 Float 트랙을 추가하여 CurveFloat를 추가한다. 각각 Curve는 (0,70), (1,40)을 설정
- 각각의 Src, des Curve를 선택 후 우클릭으로 자동 보간 설정
- 위에서 FOV가 확대 되게끔 한 Set FOV를 삭제 후 TimeLine에 연결 시켜서 True시 Play하고, False시 Reverse 하게 하면 TimeLine이 값을 기억하여 중간에 동작을 끊어도 다시 설정한 Curve대로 돌아가게 된다.