**하나의 애니메이션에 모든 동작들이 들어 있는 경우** (4:41 참고) - 확인 완료

실질적으로 가지고 있는 애니메이션 데이터(애니메이션 클립의 전체 크기)를 많이 가지고 있다.

그래서 필요한 부분의 동작 키프레임을 알고 있어야 나눌수가 있다.

동작별로 클립 관리(추가 및 제거로)

해당 애니메이션 리소스 인스펙터 애니메이션 항목 -> Clips에서 +(추가), -(제거) 가능

-> 추가하면 만약 전체 동작이 통으로 있는 경우 전부다 가져오게 된다...그래서 미리 숙지한

키프레임을 기억하여 해당 클립의 스타트, 앤드 키프레임 설정 -> 어플라이

그런데 만약 튄다면...

루프타임 : 체크하면 루프애니메이션 설정

루프포즈 : 마지막에 애니메이션 동작 튀는걸 완화

미러 : 원래 동작을 반대쪽 모션으로 복사

블랜드트리

입력값에 따른 혼합 비율대로 동작 수행하게끔 순서를 맞춰줘야 한다

추가는 스테이트창에서 우클릭 크레이트 스테이트 -> 프롬 뉴 블랜드 트리

동작 추가는 해당 블랜드트리로 들어가서 선택한 채로 인스펙터창의 + -> 에드 모션 필드

추가하는 동작 순서가 중요하다

블랜드(혼합) 할려는 애니메이션 프레임은 동일해야 어색하지 않음

파라메터 좌우 끝의 숫자를 조절하여 입력값 조건을 설정할 수 있다.

※ 입력값 제한을 0.05 로 한 이유 (5:10 참고) - 확인 완료

만약 0으로 설정하게 되면 그 조건을 완료할 수가 없게 된다...

(ex) 앞으로 가는 동작 후 다시 돌아오는 조건이 만약 InputV Less 0 이면

가만히 있어도 앞으로 가는 동작만 실행하게 된다...왜냐면 Less 0 = 0 미만

즉, 0보다 더 작아야 되기 때문이다.

블랜드트리에 또 블랜드트리 추가하기 : 인스펙터창의 + -> 뉴 블랜드 트리

애니메이션 파일의 인스펙터 창의 루트 트랜스폼에 관하여

루트 트랜스폼 로테이션 : 루트의 회전 정보

루트 트랜스폼 포지션 : 위치값은 Y, XZ로 나눠져 있는데 Y는 수직, XZ는 평면 움직임

애니메이터의 어플라이 루트 모션 : 체크하면 애니메이션의 움직임을 트랜스폼에 적용시키겠다는 것

Bake Info Pose

체크하면 루트는 가만히 있고 루트가 움직였던 량은 그 하위에 있는 자식들한테 다 적용됨

(루트가 트랜스폼을 완전히 벗어나지 않는 한에서)

(ex) 앞으로 달리는 동작에서 XZ Position Pose를 체크하면

앞으로 달리는 행동은 하지만 실제로 앞으로 가지는 않는다.

체크를 안 하면 루트에도 적용되므로 (5:37 참고)

(ex) 좌우로 몸을 흔들며 달리는 동작의 Rotation Pose를 해제하면

몸이 좌우로 흔들릴때마다 그쪽으로 방향이 바뀌게 되므로 내가 원하는대로 앞으로만 달리지 않게 된다.

=> 결론은 Bake Into Pose 체크 및 해제를 상황에 맞게 적절하게 잘 사용해야 된다

어베로지 벨로시티와 어베로지 앵글러 : 초당 변화하는 포지션(로컬 기준)

오프셋 : 보정치 설정

만약 어베로지가 원하는 방향에서 어긋났더라면 사용

그러나 어베로지는 평균치이기 때문에 백프로 맞출 수는 없다...

왜냐면 동작 중에 끊길 수도 있으므로

Y 포지션에 베이크 체크되어 있으면 중력을 안 받음(왠만하면 체크를 해주는 것이 좋다)

그러나 XZ 포지션은 체크되면 안됨(움직여야 하므로)

input값의 시작이 0부터 시작한다...그래서 좌우 방향을 갑작스레 바꾸면 튀는것이다

(ex) 좌측방향(-1)에서 바로 우측방향을 누르면 서서히 -1 -> 0 -> 1 순으로 자연스럽게 되야 하는데

그냥 바로 0부터 시작하여 1로 가는 바람에 확 튀어 보이게 된다.

코드로 동작을 제어할 때 해당 동작을 하기 위한 전제 조건 동작이 맞는지를 확실히 확인 한 후에

입력이 되게 해야 한다.

트리거 : 한번 호출이 되면 true가 된다.

트리거로 들어간 것은 해당 동작이 완전히 진입했을 때 자동으로 false가 된다

그래서 트리거는 한타로 하는 동작에 사용(한번 클릭으로 동작 수행 시)

그리고 bool타입은 계속 지속적으로 누르고 있는 동작에 사용

베이즈드 유폰 종류

오리지날

바디 오리엔테이션

좌우로 움직임이 있는 애니 동작의 좌우 변화량을 없애기 위해 오프셋을 이용하여 벨로시티를

수정하였는데도 없어지지 않는다면 이전 동작과의 보간에 따른 오차이기 때문이다..

(즉, 이전 동작에도 좌우 변화량이 존재한다..)

해시 : 중복되지 않는 숫자 정보

readonly : 상수..즉, const와 비슷 (아예 같다는 것은 아님)

컴파일을 하여 해당 클래스를 계산할 때 대입된다.

const : 컴파일 이전에 값을 이미 알고 있는 상태에서 컴파일에 들어간다. 그래서 리터럴 상수만 가능하고

무언가 계산이 들어가는 것은 안됨

리터럴 : 코드에 쓴것(변하지 않는 정보)