**Ik fk 스위치(+스트레치)**

1단계 구상->2단계 분할->3단계 메소드 작성->4단계 수정

최상위 - class Arm\_build

실행 조건- 디폴트 조인트가 씬 내에 있어야 함. 혹은 선택한 오브젝트에 디폴트 조인트가 있어야 함. (최소 두개, 이름은 Df1\_Jnt, Df2\_Jnt, Df3\_Jnt, 축방향 y)

메소드0. def \_\_init\_\_(self):

메소드1. def Base\_build\_function()

1) Df1\_Jnt 위치(만) 참조하여 전체 컨트롤러 만들기 (어트리뷰트에 스트레치와 스위치 기능을 추가, 이름: Root\_Ctrl)

1-1) 커브를 만든 다음 매치트랜스 폼

메소드2. def ik\_BaseJnt\_function():

2) 디폴트 조인트 위치를 기준으로 ik 조인트 생성 (Arm1Ik\_Jnt, Arm2Ik\_Jnt, Arm3Ik\_Jnt)

메소드3. def fk\_BaseJnt\_function():

3) 디폴트 조인트 위치를 기준으로 fk 조인트 생성 (Arm1Fk\_Jnt, Arm2Fk\_Jnt, Arm3Fk\_Jnt)

메소드4. def Ctrl\_Build():

4) ik 컨트롤러 만들기 (이름: Arm1Ik\_Ctrl, Arm2Ik\_Ctrl)

5) fk 컨트롤러 만들기 (이름: Arm1Fk\_Ctrl, Arm2Fk\_Ctrl, Arm3Fk\_Ctrl)

메소드5. def ik\_function()

1) 로테이트 플레인 솔버로 핸들 만들기

2) Arm2Ik\_Ctrl에 ik핸들(3)번) 하위노드로 집어넣기

3) Arm1Ik\_Jnt를 Arm1Ik\_Ctrl에 페런츠 컨스트레인

메소드6. def fk\_function()

1) 만든 fk 컨트롤러와 조인트를 컨스트레인

메소드7. def Switch\_function()

1) ikfk조인트들을 df조인트에 같이 컨스트레인

2) 전체 컨트롤러의 어트리뷰트를 블렌드 컬러에 연결 (1R-0, 1G-1, 1B-0, 2R-1, 2G-0, 2B-0)

3) 블렌드 컬러(메소드1)와 cmds.curve( p=[(0, 0, 0), (3, 5, 6), (5, 6, 7), (9, 9, 9)] ) 컨트롤러 최상위 그룹 노드를 연결 (스위칭에 따라 컨트롤러들도 감춰지고 보이게끔)

메소드8. def stretch\_function()

1) Lf\_Df1\_Jnt, Lf\_Df3\_Jnt의 거리를 구하기 (DT\_Python 스크립트 사용)

2) ScaleDefault(Loc) 생성,

3) multi 생성, ScaleDefault(Loc)와 연결

5) 블랜드 컬러생성, condition노드 생성. 그 후 condition을 각각의 ik조인트들에 연결함