import maya.cmds as cmds

from functools import partial

#---external Function Set-----

def createJnt():

    cmds.joint()

def createLoc():

    cmds.spaceLocator()

def createClu():

    cmds.cluster()

def LfHand\_Joint\_Cleaner(\*args):

    Index = cmds.listRelatives('LfIndexRoot\_Jnt', ad=True, type='joint')

    Middle = cmds.listRelatives('LfMiddleRoot\_Jnt', ad=True, type='joint')

    Ring = cmds.listRelatives('LfRingRoot\_Jnt', ad=True, type='joint')

    Thumb = cmds.listRelatives('LfThumbRoot\_Jnt', ad=True, type='joint')

    Pinky = cmds.listRelatives('LfPinkyRoot\_Jnt', ad=True, type='joint')

    if Index:

        for IntheIndex in Index:

            cmds.setAttr(IntheIndex + ".jointOrientX", 0)

            cmds.setAttr(IntheIndex + ".jointOrientY", 0)

            cmds.setAttr(IntheIndex + ".jointOrientZ", 0)

    if Middle:

        for IntheMiddle in Middle:

            cmds.setAttr(IntheMiddle + ".jointOrientX", 0)

            cmds.setAttr(IntheMiddle + ".jointOrientY", 0)

            cmds.setAttr(IntheMiddle + ".jointOrientZ", 0)

    if Ring:

        for IntheRing in Ring:

            cmds.setAttr(IntheRing + ".jointOrientX", 0)

            cmds.setAttr(IntheRing + ".jointOrientY", 0)

            cmds.setAttr(IntheRing + ".jointOrientZ", 0)

    if Thumb:

        for IntheThumb in Thumb:

            cmds.setAttr(IntheThumb + ".jointOrientX", 0)

            cmds.setAttr(IntheThumb + ".jointOrientY", 0)

            cmds.setAttr(IntheThumb + ".jointOrientZ", 0)

    if Pinky:

        for InthePinky in Pinky:

            cmds.setAttr(InthePinky + ".jointOrientX", 0)

            cmds.setAttr(InthePinky + ".jointOrientY", 0)

            cmds.setAttr(InthePinky + ".jointOrientZ", 0)

    for IntheIndex in Index:

        cmds.setAttr(IntheIndex + ".rotateX", 0)

        cmds.setAttr(IntheIndex + ".rotateY", 0)

        cmds.setAttr(IntheIndex + ".rotateZ", 0)

    for IntheMiddle in Middle:

        cmds.setAttr(IntheMiddle + ".rotateX", 0)

        cmds.setAttr(IntheMiddle + ".rotateY", 0)

        cmds.setAttr(IntheMiddle + ".rotateZ", 0)

    for IntheRing in Ring:

        cmds.setAttr(IntheRing + ".rotateX", 0)

        cmds.setAttr(IntheRing + ".rotateY", 0)

        cmds.setAttr(IntheRing + ".rotateZ", 0)

    for IntheThumb in Thumb:

        cmds.setAttr(IntheThumb + ".rotateX", 0)

        cmds.setAttr(IntheThumb + ".rotateY", 0)

        cmds.setAttr(IntheThumb + ".rotateZ", 0)

    for InthePinky in Pinky:

        cmds.setAttr(InthePinky + ".rotateX", 0)

        cmds.setAttr(InthePinky + ".rotateY", 0)

        cmds.setAttr(InthePinky + ".rotateZ", 0)

def ArtPaintSkinWeightsTool():

    cmds.ArtPaintSkinWeightsTool()

def DeleteHisAndFreeze():

    cmds.select(cl=True)

    selA = cmds.ls('\*\_Geo', '\*\_geo', type='transform')

    for Geo in selA:

        cmds.delete(Geo, constructionHistory=True)

        cmds.makeIdentity(Geo, apply=True, t=1, r=1, s=1)

def oneClickMatchJ\_L():

    selected = cmds.ls(selection=True)

    locList = []

    jntList = []

    for obj in selected:

        if '\_Jnt' in obj:

            jntList.append(obj)

        if '\_Loc' in obj:

            locList.append(obj)

    for loc in locList:

        loc\_n\_name = loc.split('\_')[-2]

        for jnt in jntList:

            jnt\_n\_name = jnt.split('\_')[-2]

            if jnt\_n\_name.isdigit() and loc\_n\_name.isdigit() and jnt\_n\_name == loc\_n\_name:

                cmds.matchTransform(loc, jnt)

                break

def oneClickMatchC\_L():

    selected = cmds.ls(selection=True)

    ctrlList = []

    jntList = []

    for obj in selected:

        if '\_Jnt' in obj:

            jntList.append(obj)

        if '\_Ctrl' in obj:

            ctrlList.append(obj)

    for ctrl in ctrlList:

        ctrl\_n\_name = ctrl.split('\_')[-2]

        for jnt in jntList:

            jnt\_n\_name = jnt.split('\_')[-2]

            if jnt\_n\_name.isdigit() and ctrl\_n\_name.isdigit() and jnt\_n\_name == ctrl\_n\_name:

                cmds.matchTransform(ctrl, jnt)

                break

def oneClickblend():

    selected = cmds.ls(selection=True)

    before = []

    after = []

    for obj in selected:

        if '\_Geo' in obj:

            before.append(obj)

        if '\_Dummy' in obj:

            after.append(obj)

    for Geo in before:

        Geo\_n\_name = Geo.split('\_')[-2]

        for Dum in after:

            Dum\_n\_name = Dum.split('\_')[-2]

            if Geo\_n\_name.isdigit() and Dum\_n\_name.isdigit() and Geo\_n\_name == Dum\_n\_name:

                cmds.blendShape(Dum, Geo)

def Findnode(query):

    all\_objects = cmds.ls()

    variable\_obj = []

    for obj in all\_objects:

        if query in obj:

            variable\_obj.append(obj)

    if variable\_obj:

        cmds.select(variable\_obj, replace=True)

    else:

        print("Not found")

#---Window Set-----

def Khn\_Ui():

    if cmds.window('win', exists=True):

        cmds.deleteUI('win')

    cmds.window('win', title='Khn\_Ui', widthHeight=(450, 100))

    cmds.columnLayout(adjustableColumn=True)

#---Internal function-----

    def FindSet(\*args):

        Query = cmds.textField(ConstraintTextset, q=1, tx=1)

        Findnode(Query)

    def executeCreation(\*args):

        selectedRadioButton = cmds.radioButtonGrp(createTypeRadio, q=True, select=True)

        if selectedRadioButton == 1:

            createLoc()

        elif selectedRadioButton == 2:

            createClu()

        elif selectedRadioButton == 3:

            createJnt()

#---Button Set-----

    cmds.columnLayout(adjustableColumn=True)

    cmds.rowColumnLayout(numberOfColumns=3, columnWidth=[(1, 370), (2, 80)])

    createTypeRadio = cmds.radioButtonGrp(label='Create     : ',

                                          labelArray3=['Locator', 'Cluster', 'Joint'],

                                          numberOfRadioButtons=3,

                                          columnWidth=[(1, 75)],

                                          columnAlign=[(1, 'left')])

    cmds.button(label='Enter', command=executeCreation, backgroundColor=[0.0,0.7,1.0])

    cmds.setParent('..')

    cmds.button(label='Paint Skin Weights Tool', command='ArtPaintSkinWeightsTool()')

    cmds.button(label='Hand Joint Clean', command='LfHand\_Joint\_Cleaner()')

    cmds.button(label='Delete History And Freeze', command='DeleteHisAndFreeze()')

    cmds.rowColumnLayout(numberOfColumns=3, columnWidth=[(1, 150), (2, 150), (3, 150)])

    cmds.button(label='one Click Match: Loc->Jnt', command='oneClickMatchJ\_L()', height=50, backgroundColor=[1.0,0.0,0.8])

    cmds.button(label='one Click Match: Ctrl->Jnt', command='oneClickMatchC\_L()', height=50, backgroundColor=[0.0,1.0,1.0])

    cmds.button(label='one Click blend', command='oneClickblend()', height=50, backgroundColor=[1.0,1.0,0.0])

    cmds.setParent('..')

    cmds.rowColumnLayout(numberOfColumns=3, columnWidth=[(1, 100), (2, 270), (3, 80)])

    cmds.text(label='Finding Node   :')

    ConstraintTextset = cmds.textField(tx="Constraint")

    cmds.button(label='Enter', c=lambda x: FindSet(), backgroundColor=[1.0,0.3,0.0])

    cmds.setParent('..')

    cmds.showWindow()

Khn\_Ui()

0. 전체 스크립트.

#---external Function Set-----

def createJnt():

    cmds.joint()

def createLoc():

    cmds.spaceLocator()

def createClu():

    cmds.cluster()

def LfHand\_Joint\_Cleaner(\*args):

    Index = cmds.listRelatives('LfIndexRoot\_Jnt', ad=True, type='joint')

    Middle = cmds.listRelatives('LfMiddleRoot\_Jnt', ad=True, type='joint')

    Ring = cmds.listRelatives('LfRingRoot\_Jnt', ad=True, type='joint')

    Thumb = cmds.listRelatives('LfThumbRoot\_Jnt', ad=True, type='joint')

    Pinky = cmds.listRelatives('LfPinkyRoot\_Jnt', ad=True, type='joint')

    if Index:

        for IntheIndex in Index:

            cmds.setAttr(IntheIndex + ".jointOrientX", 0)

            cmds.setAttr(IntheIndex + ".jointOrientY", 0)

            cmds.setAttr(IntheIndex + ".jointOrientZ", 0)

    if Middle:

        for IntheMiddle in Middle:

            cmds.setAttr(IntheMiddle + ".jointOrientX", 0)

            cmds.setAttr(IntheMiddle + ".jointOrientY", 0)

            cmds.setAttr(IntheMiddle + ".jointOrientZ", 0)

    if Ring:

        for IntheRing in Ring:

            cmds.setAttr(IntheRing + ".jointOrientX", 0)

            cmds.setAttr(IntheRing + ".jointOrientY", 0)

            cmds.setAttr(IntheRing + ".jointOrientZ", 0)

    if Thumb:

        for IntheThumb in Thumb:

            cmds.setAttr(IntheThumb + ".jointOrientX", 0)

            cmds.setAttr(IntheThumb + ".jointOrientY", 0)

            cmds.setAttr(IntheThumb + ".jointOrientZ", 0)

    if Pinky:

        for InthePinky in Pinky:

            cmds.setAttr(InthePinky + ".jointOrientX", 0)

            cmds.setAttr(InthePinky + ".jointOrientY", 0)

            cmds.setAttr(InthePinky + ".jointOrientZ", 0)

    for IntheIndex in Index:

        cmds.setAttr(IntheIndex + ".rotateX", 0)

        cmds.setAttr(IntheIndex + ".rotateY", 0)

        cmds.setAttr(IntheIndex + ".rotateZ", 0)

    for IntheMiddle in Middle:

        cmds.setAttr(IntheMiddle + ".rotateX", 0)

        cmds.setAttr(IntheMiddle + ".rotateY", 0)

        cmds.setAttr(IntheMiddle + ".rotateZ", 0)

    for IntheRing in Ring:

        cmds.setAttr(IntheRing + ".rotateX", 0)

        cmds.setAttr(IntheRing + ".rotateY", 0)

        cmds.setAttr(IntheRing + ".rotateZ", 0)

    for IntheThumb in Thumb:

        cmds.setAttr(IntheThumb + ".rotateX", 0)

        cmds.setAttr(IntheThumb + ".rotateY", 0)

        cmds.setAttr(IntheThumb + ".rotateZ", 0)

    for InthePinky in Pinky:

        cmds.setAttr(InthePinky + ".rotateX", 0)

        cmds.setAttr(InthePinky + ".rotateY", 0)

        cmds.setAttr(InthePinky + ".rotateZ", 0)

def ArtPaintSkinWeightsTool():

    cmds.ArtPaintSkinWeightsTool()

def DeleteHisAndFreeze():

    cmds.select(cl=True)

    selA = cmds.ls('\*\_Geo', '\*\_geo', type='transform')

    for Geo in selA:

        cmds.delete(Geo, constructionHistory=True)

        cmds.makeIdentity(Geo, apply=True, t=1, r=1, s=1)

def oneClickMatchJ\_L():

    selected = cmds.ls(selection=True)

    locList = []

    jntList = []

    for obj in selected:

        if '\_Jnt' in obj:

            jntList.append(obj)

        if '\_Loc' in obj:

            locList.append(obj)

    for loc in locList:

        loc\_n\_name = loc.split('\_')[-2]

        for jnt in jntList:

            jnt\_n\_name = jnt.split('\_')[-2]

            if jnt\_n\_name.isdigit() and loc\_n\_name.isdigit() and jnt\_n\_name == loc\_n\_name:

                cmds.matchTransform(loc, jnt)

                break

def oneClickMatchC\_L():

    selected = cmds.ls(selection=True)

    ctrlList = []

    jntList = []

    for obj in selected:

        if '\_Jnt' in obj:

            jntList.append(obj)

        if '\_Ctrl' in obj:

            ctrlList.append(obj)

    for ctrl in ctrlList:

        ctrl\_n\_name = ctrl.split('\_')[-2]

        for jnt in jntList:

            jnt\_n\_name = jnt.split('\_')[-2]

            if jnt\_n\_name.isdigit() and ctrl\_n\_name.isdigit() and jnt\_n\_name == ctrl\_n\_name:

                cmds.matchTransform(ctrl, jnt)

                break

def oneClickblend():

    selected = cmds.ls(selection=True)

    before = []

    after = []

    for obj in selected:

        if '\_Geo' in obj:

            before.append(obj)

        if '\_Dummy' in obj:

            after.append(obj)

    for Geo in before:

        Geo\_n\_name = Geo.split('\_')[-2]

        for Dum in after:

            Dum\_n\_name = Dum.split('\_')[-2]

            if Geo\_n\_name.isdigit() and Dum\_n\_name.isdigit() and Geo\_n\_name == Dum\_n\_name:

                cmds.blendShape(Dum, Geo)

def Findnode(query):

    all\_objects = cmds.ls()

    variable\_obj = []

    for obj in all\_objects:

        if query in obj:

            variable\_obj.append(obj)

    if variable\_obj:

        cmds.select(variable\_obj, replace=True)

    else:

        print("Not found")

1. 버튼을 클릭했을 때 실행될 함수들을 외부에 정의한다.

1-1) 맨 마지막 Findnode함수의 변수 query를 기억해 놓자.

#---Window Set-----

def Khn\_Ui():

    if cmds.window('win', exists=True):

        cmds.deleteUI('win')

    cmds.window('win', title='Khn\_Ui', widthHeight=(450, 100))

    cmds.columnLayout(adjustableColumn=True)

2. 윈도우를 생성한다.

        cmds.deleteUI('win')

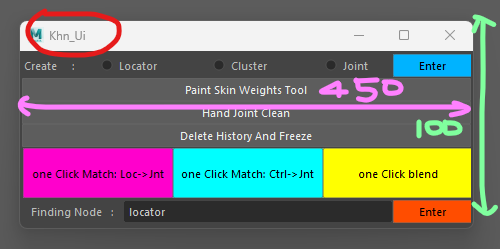
2-1) 이 구문으로 윈도우 창을 계속해서 생성하는 것을 막는다. (한번만 생성되게 함)

    2-2) cmds.window('win', title='Khn\_Ui', widthHeight=(450, 100))

    2-3) cmds.columnLayout(adjustableColumn=True)

2-2) 윈도우의 제목을 Khn\_Ui로 설정하고 생성 되었을 때의 창 크기를 가로 450, 세로100으로 설정한다.

2-3) 저 구문을 추가하여 창의 크기를 사용자가 마음대로 늘렸다가 줄일 수 있다.



#---Internal function-----

    3-1) def FindSet(\*args):

        Query = cmds.textField(ConstraintTextset, q=1, tx=1)

        Findnode(Query)

    3-2) def executeCreation(\*args):

        selectedRadioButton = cmds.radioButtonGrp(createTypeRadio, q=True, select=True)

        if selectedRadioButton == 1:

            createLoc()

        elif selectedRadioButton == 2:

            createClu()

        elif selectedRadioButton == 3:

            createJnt()

3. 윈도우 내부 함수를 정의한다. (라디오 버튼, 텍스트 필드에 들어갈 변수들을 찾는 용도)

3-1) 위에서 본 Query변수를 가져온다. Query는 사용자가 텍스트 필드에 입력한 글자들이 담긴 변수다. 새로운 값이 들어가는 상황엔 전에 저장된 값을 날리고 새로 변수를 지정해야 하기 때문에 이 구문을 추가한다.

3-2) 밑에 나올 라디오 버튼이 작동하기 위해 들어간 함수. 각각의 라디오 버튼에 1부터 3까지의 숫자를 부여한 뒤 그에 맞게 함수를 배치한다.

4-1) cmds.rowColumnLayout(numberOfColumns=3, columnWidth=[(1, 370), (2, 80)])

4-2) createTypeRadio = cmds.radioButtonGrp(label='Create     : ',

                             4-3) labelArray3=['Locator', 'Cluster', 'Joint'],

                                          numberOfRadioButtons=3,

                             4-4)columnWidth=[(1, 75)],

                             4-5)columnAlign=[(1, 'left')])

4-6) cmds.button(label='Enter', command=executeCreation, backgroundColor=[0.0,0.7,1.0])

    cmds.setParent('..')

4. 맨 윗줄에 들어갈 Create항목을 추가한다.

4-1) cmds.rowColumnLayout커맨드는 밑에 줄의 레이아웃을 변경하는 커맨드. 나는 전체를 삼등분 한 뒤, 1번의 가로 넓이를 370, 2번의 가로 넓이를 80으로 설정했다. (이걸로는 파란색 Enter버튼의 가로 길이를 조절함)

4-2) 위에 설정해둔 변수 createTypeRadio을 정의해준다.

labelArray3=['Locator', 'Cluster', 'Joint'],

4-3)세개의 항목을 설정

columnWidth=[(1, 75)],

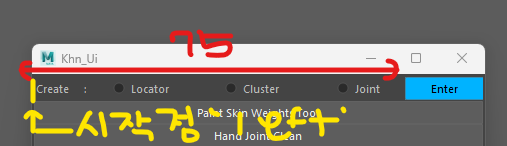
4-4) 길이 조절 나는 가로 75로 조절하여 Enter버튼에 마지막 항목이 가려지지 않게끔 했다. (이걸로 라디오 버튼 항목의 크기를 조절함)

columnAlign=[(1, 'left')]

4-5) 라디오 버튼을 가운데 정렬에서 왼쪽 정렬로 맞춘다.

cmds.button(label='Enter', command=executeCreation, backgroundColor=[0.0,0.7,1.0])

4-6) 파란색 Enter버튼을 만든 뒤 색깔을 지정함. 순서대로 빨, 노, 파. 각각 1이 최댓값이다.

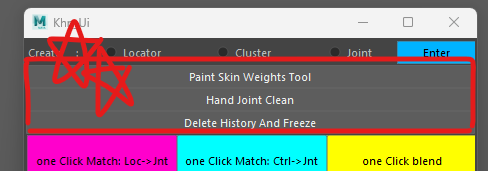


    cmds.button(label='Paint Skin Weights Tool', command='ArtPaintSkinWeightsTool()')

    cmds.button(label='Hand Joint Clean', command='LfHand\_Joint\_Cleaner()')

    cmds.button(label='Delete History And Freeze', command='DeleteHisAndFreeze()')

5. ‘Paint Skin Weights Tool’, ‘Hand Joint Clean’, ‘Delete History And Freeze’ 버튼을 만든다.



6-1) cmds.rowColumnLayout(numberOfColumns=3, columnWidth=[(1, 150), (2, 150), (3, 150)])

    cmds.button(label='one Click Match: Loc->Jnt', command='oneClickMatchJ\_L()', 6-2) height=50, backgroundColor=[1.0,0.0,0.8])

    cmds.button(label='one Click Match: Ctrl->Jnt', command='oneClickMatchC\_L()', height=50, backgroundColor=[0.0,1.0,1.0])

    cmds.button(label='one Click blend', command='oneClickblend()', height=50, backgroundColor=[1.0,1.0,0.0])

6-3) cmds.setParent('..')

6. 'one Click Match: Loc->Jnt', 'one Click Match: Ctrl->Jnt', 'one Click blend' 버튼을 만든다.

cmds.rowColumnLayout(numberOfColumns=3, columnWidth=[(1, 150), (2, 150), (3, 150)])

6-1) 위에서 설명했던 4-1) 번과 같은 역할이다. 대신 이 버튼들은 전체를 삼등분 한뒤 각각 가로를 150로 설정했다.

height=50,

6-2) 다른 버튼들과 구분을 위해 높이를 변경했다.

cmds.setParent('..')

6-3) 위에도 쓰였다. 4-6) 번이다. cmds.rowColumnLayout을 끝내는 역할을 한다. 가로 안에 들어간(‘. .’)이 부호들은 바다토끼 이모티콘이 아니라 자신의 위까지를 cmds.rowColumnLayout에 영향으로 둔다는 뜻이다.



    cmds.rowColumnLayout(numberOfColumns=3, columnWidth=[(1, 100), (2, 270), (3, 80)])

7-1)cmds.text(label='Finding Node   :')

7-2)ConstraintTextset = cmds.textField(tx="Constraint")

7-3)cmds.button(label='Enter', c=lambda x: FindSet(), backgroundColor=[1.0,0.3,0.0])

    cmds.setParent('..')

7. 씬 내에서 입력된 글자가 포함된 항목들을 전부 찾는버튼을 생성한다. (아웃라이너 뿐만 아니라 노드까지.) rowColumnLayout와 setParent('..')은 굳이 한번 더 설명하지 않는다.

cmds.text(label='Finding Node   :')

7-1) 제일 왼쪽에 배치될 제목이다.

ConstraintTextset = cmds.textField(tx="Constraint")

7-2) 위에서 눈여겨 보라고 했던 query변수가 이제야 쓰인다. 3-1) 번이다. tx="Constraint"은 창을 켤 때 미리 타이핑 되어있는 글자다.

cmds.button(label='Enter', c=lambda x: FindSet(), backgroundColor=[1.0,0.3,0.0])

7-3) lamda 함수를 이용해 빨간 Enter버튼을 만든다. 이 버튼은 Findset 함수를 실행시킨다.

