

```

/*
tool to automate blendshapeMirroring for characters to work with my autorig
*/

// window
global proc bzpBlendShapeMirrorWindow() {

    if (`window -exists myWindowForBlendshapeMirror`) deleteUI myWindowForBlendshapeMirror;
    window -title "BZP PRD BlendshapeMirror Script" -widthHeight 500 400 myWindowForBlendshapeMirror;
    columnLayout -columnAttach "both" 5 -rowSpacing 5 -columnWidth 500;
    text "Type the name of the base mesh";

    global string $textFieldOne;
    $textFieldOne = `textField -text "Type the name of the base mesh Here" `;
    text "Type the mirror axis of the mesh";

    radioButtonGrp
    -numberOfRadioButtons 3
    -label "select type"
    -labelArray3 "scaleX" "scaleY" "scaleZ"
    -select 1
    myRadBtnGrp;

    text "Type the name of the blendShape Node";
    global string $textFieldThree;
    $textFieldThree = `textField -text "Type the name of the blendShape Node Here" `;
    text "Type the name of the blendShape Attribute";
    global string $textFieldFour;
    $textFieldFour = `textField -text "Type the name of the blendShape Attribute Here" `;
    text "Type the name of the group node the new blendShape will be parented";
    global string $textFieldFive;
    $textFieldFive = `textField -text "Type the name of the group node the new blendShape will be parented Here" `;
    text "Type the name of the newly created blendShape Here";
    global string $textFieldSeven;
    $textFieldSeven = `textField -text "Type the name of the newly created blendShape Here" `;
    floatSliderGrp -label "connection strength" -field true
    -minValue .01 -maxValue 20.0
    -value 1
    mySlider;
    string $button = `button -bgc 0 1 0 -command "buttonActivate()" -label "Generate Mirror Blendshape" `;

    showWindow myWindowForBlendshapeMirror;
}

// button to activate procedure
global proc buttonActivate() {

    global string $textFieldOneBase;
    global string $textFieldOne;
    string $textFieldOneBase = `textField -q -text $textFieldOne `;

    global string $textFieldThreeBase;
    global string $textFieldThree;
    string $textFieldThreeBase = `textField -q -text $textFieldThree `;

    global string $textFieldFourBase;
    global string $textFieldFour;
    string $textFieldFourBase = `textField -q -text $textFieldFour `;

    global string $textFieldFiveBase;
    global string $textFieldFive;
    string $textFieldFiveBase = `textField -q -text $textFieldFive `;
}

```

```

global string $textFieldSevenBase;
global string $textFieldSeven;
string $textFieldSevenBase = `textField -q -text $textFieldSeven`;

global string $scaleX = "scaleX";
global string $scaleY = "scaleY";
global string $scaleZ = "scaleZ";

if (`radioButtonGrp -q -select myRadBtnGrp` == 1){
    float $scaleVal = `floatSliderGrp -q -value mySlider`;
    blendShapeGenerator($textFieldOneBase, "scaleX", $textFieldThreeBase, $textFieldFourBase, $textFieldFiveBase, $textFieldSevenBase, $scaleVal);
}
if (`radioButtonGrp -q -select myRadBtnGrp` == 2){
    float $scaleVal = `floatSliderGrp -q -value mySlider`;
    blendShapeGenerator($textFieldOneBase, "scaleY", $textFieldThreeBase, $textFieldFourBase, $textFieldFiveBase, $textFieldSevenBase, $scaleVal);
}
if (`radioButtonGrp -q -select myRadBtnGrp` == 3){
    float $scaleVal = `floatSliderGrp -q -value mySlider`;
    blendShapeGenerator($textFieldOneBase, "scaleZ", $textFieldThreeBase, $textFieldFourBase, $textFieldFiveBase, $textFieldSevenBase, $scaleVal);
}
}

global proc blendShapeGenerator(string $originalName, string $sX_or_sY_or_sZ, string $bShapeName, string $bShapeAttributeName, string $groupTargetsName, string $newBShapeName, float $strengthValue) {

string $original = $originalName;
string $newOriginal[] = `duplicate $original`;
setAttr ($original + "." + $sX_or_sY_or_sZ) -l;
select -r $newOriginal $original;
string $wrapName[] = `doWrapArgList "2" {"I", "O", "B"}`;
setAttr ($bShapeName + "." + $bShapeAttributeName) l;
string $newBlendShapeToAdd[] = `duplicate -n $newBShapeName $newOriginal`;

setAttr ($bShapeName + "." + $bShapeAttributeName) 0;
parent $newBlendShapeToAdd $groupTargetsName;
int $hierarchyGroup[] = `select -hi $groupTargetsName`;
string $pickk[] = `ls -sl -type "mesh"`;
int $newHil = `size $pickk`;
int $newHii = ($newHil - 1);

blendShape -edit -t $original $newHii $newBlendShapeToAdd l.0.$bShapeName;

delete $wrapName;
setAttr ($original + "." + $sX_or_sY_or_sZ) l;
delete $newOriginal;

}

bzipBlendShapeMirrorWindow;

```