```
////////////// Curve Blend Tool v1.2 //////////

//////////////        Mat Clark      /////////

//

//use this tool quickly blend multiple curves

//it also applies keyframes to blended curves

//Other useful features include.......
//

//- make curves dynamic

//- randomize selection

//- apply paint effects

//- randomize keyframes
//
//Feel free to use this script in your work,
//please let me know how you have used it
//and please credit me.

////////////////////////////////////////////



if (`window -exists Curve_Blend_Tool`){

         deleteUI Curve_Blend_Tool;

}



int $buttonPush;

int $degree;

int $span;

int $numBlends;

string $base;

string $target;

int $active = 0;

int $tCount = 0;

int $bCount = 0;
```

```
windowProc;


proc windowProc(){


        global int $active;

        global int $aCount;

        global int $bCount;


        $win = `window -sizeable true Curve_Blend_Tool`;


                columnLayout;

                        rowLayout -numberOfColumns 2

                                -columnWidth2 150 150

                                -columnAlign  1 "right"

                                -columnAttach 1 "both"     0

                                -columnAttach 2 "both"  0;

                                button -w 75 -h 40 -en 1 -label
($aCount+"  target curves") -command "tCountProc" -annotation
"Select all target curves and click to count" aCountBtn;

                                button -w 75 -h 40 -en 1 -label
($bCount+"  base curves") -command "bCountProc" -annotation "Select
all base curves and click to count" bCountBtn;

                        setParent ..;

                        columnLayout;

                                button -label "create indovidual
blend"  -w 302 -h 40 -command "individualProc" -en $active -
annotation "  Shift select TARGET and BASE then click to create
blend" individualBtn;

                        setParent ..;

                        columnLayout;
```

```
                                button -label "mass create blends"
-w 302 -h 40 -command "massProc" -en $active -annotation "  Click to
create blendshapes between all target and base nodes" massBtn;

                        setParent ..;

                        rowLayout -numberOfColumns 2

                                -columnWidth2 150 150

                                -columnAlign  1 "right"

                                -columnAttach 1 "both"     0

                                -columnAttach 2 "both"  0;

                                button -w 75 -h 40 -en $active -
label "randomise blend timing" -command "randomKeyProc" -annotation
"Select all blended curves and click to randomise timing"
randomKeyBtn;

                                button -w 75 -h 40 -en $active -
label "randomise selection" -command "randomSelProc" -annotation
"Select all base curves and click to deselect a random quantity"
randomSelBtn;

                        setParent ..;

                        columnLayout;

                                button -label "make selected
dynamic"  -w 302 -h 40 -command "dynProc" -en $active -annotation "
select blend curves and click to convert them to nHair" dynBtn;

                        setParent ..;

                        //pfx

                        rowLayout -numberOfColumns 2

                                -columnWidth2 150 150

                                -columnAlign  1 "right"

                                -columnAttach 1 "both"     0

                                -columnAttach 2 "both"  0;

                                button -w 75 -h 40 -en $active -
label "attach PFX" -command "pfxProc" -annotation "Select the curves
to which you would like to attach a PFX brush and click" pfxBtn;

                                string $pfxOptions =
`optionMenuGrp -en $active -changeCommand "print #1" pfxList`;
```

```
                                    menuItem -label "charcoalThick" -
da 1 item1;

                                    menuItem -label "charcoalWet" -da
1 item2;

                                    menuItem -label
"pencilScribbleDark" -da 1 item3;

                                    menuItem -label "pencilSmooth" -da
1 item4;

                                    menuItem -label "pencilTexDark" -
da 1 item5;

                                    menuItem -divider true item6;

                                    menuItem -label
"bPointScribbleIndigo" -da 2 item7;

                                    menuItem -label "calligraphic" -da
2 item8;

                                    menuItem -label "inkGlop" -da 2
item9;

                                    menuItem -label "inkScribble" -da
2 item10;

                                    menuItem -label "inkSplash" -da 2
item211;

                            setParent ..;


                showWindow $win;

                select -cl;

        }


        proc tCountProc(){

                global int $tCount;

                global string $aCurveSel[];

                string $aCurveSel[] = `ls -sl`;

                $tCount = size($aCurveSel);
```

```
                if($tCount == 0){

                        button -edit -label ($tCount+" target
curves") ("aCountBtn");

                        warning("You have no curves selected,
please select all of the base curves in this scene"+"\n");

                }

                else{

                        print ("you have selected "+$tCount+" base
curves"+"\n");

                        color -rgb 1 0.75 0.07 $aCurveSel;

                        select -cl;

                        compareCurvesProc;

                }

        }


        proc bCountProc(){

                global int $bCount;

                global string $bCurveSel[];

                string $bCurveSel[] = `ls -sl`;

                $bCount = size($bCurveSel);

                if($bCount == 0){


                        button -edit -label ($bCount+" base
curves") ("bCountBtn");

                        warning("You have no curves selected,
please select all of the target curves in this scene"+"\n");

                }

                else{

                        print ("you have selected "+$bCount+"
target curves"+"\n");
```

```
                        color -rgb 0.3 1 0 $bCurveSel;
                        select -cl;
                        compareCurvesProc;
                }
        }

        proc compareCurvesProc(){
                global int $tCount;
                global int $bCount;
                global int $active;
                if ($tCount == $bCount){
                        $active = 1;
                        button -edit -label ($tCount+"  target
curves") ("aCountBtn");
                        button -edit -label ($bCount+"  base
curves") ("bCountBtn");
                        button -edit -en $active
("individualBtn");
                        button -edit -en $active ("massBtn");
                        button -edit -en $active ("randomKeyBtn");
                        button -edit -en $active ("randomSelBtn");
                        button -edit -en $active ("dynBtn");
                        button -edit -en $active ("pfxBtn");
                        optionMenuGrp -edit -en $active
("pfxList");

                }
                else{
                        $active = 0;
```

```
                        button -edit -label ($tCount+"  target
curves") ("aCountBtn");
                        button -edit -label ($bCount+"  base
curves") ("bCountBtn");
                        button -edit -en $active
("individualBtn");
                        button -edit -en $active ("massBtn");
                        button -edit -en $active ("randomKeyBtn");
                        button -edit -en $active ("randomSelBtn");
                        button -edit -en $active ("dynBtn");
                        button -edit -en $active ("pfxBtn");
                        optionMenuGrp -edit -en $active
("pfxList");
                }

        }


        proc individualProc(){
                global int $buttonPush;
                global string $base;
                global string $target;

                $buttonPush = 1;
                string $objectSel[] = `ls -sl`;
                int $selected = size($objectSel);
                if ($selected == 2){
                        $target = $objectSel[0];
```

```
                $base = $objectSel[1];
                cvCountProc;
        }
        else{
                warning "Please ensure that you have 2
objects selected – the TARGET and the BASE";
        }
}


proc massProc(){

        global string $aCurveSel[];
        global string $bCurveSel[];
        global string $target;
        global string $base;
        global int $buttonPush;
        global int $tCount;
        global int $bCount;

        $buttonPush = 2;

        if ($tCount > 0){
                $target = $aCurveSel[0];
                $base = $bCurveSel[0];
                cvCountProc;
        }
```

```
		else{
			print ("job done"+"\n");
		}
	}

	proc cvCountProc(){
		global string $base;
		global string $target;
		global int $baseCount;
		global int $targetCount;

		//count cv's in base
		rename $base baseCurve;
		$baseCount=`getAttr baseCurve.degree`+`getAttr
baseCurve.spans`;
		rename baseCurve $base;

		//count cv's in blend
		rename $target targetCurve;
		$targetCount=`getAttr targetCurve.degree`+`getAttr
targetCurve.spans`;
		rename targetCurve $target;

		matchCVsProc;
	}

	proc matchCVsProc(){
```

```
		global int $baseCount;

		global int $targetCount;

		global string $base;

		global string $target;


		print ($baseCount+"\n"+$targetCount+"\n");


		if ($baseCount == $targetCount){

			//do nothing

		}

		if ($baseCount > $targetCount){

			rename $base testCurve;

			select -r $target;

			rebuildCurve -ch 1 -rpo 1 -rt 0 -end 1 -kr
0 -kcp 0 -kep 1 -kt 0 -s ($baseCount-`getAttr testCurve.degree`) -d
3 -tol 0.01;

			select -cl;

			rename testCurve $base;

		}

		if ($baseCount < $targetCount){

			rename $target testCurve;

			select -r $base;

			rebuildCurve -ch 1 -rpo 1 -rt 0 -end 1 -kr
0 -kcp 0 -kep 1 -kt 0 -s ($targetCount-`getAttr testCurve.degree`) -
d 3 -tol 0.01;

			select -cl;

			rename testCurve $target;

		}
```

```
                newBlendProc;

        }

        proc newBlendProc(){

                global string $base;
                global string $target;
                global int $numBlends;



                //count blend nodes
                select $base;
                string $sels[] = `ls -sl`;
                string $selsHistory[] = `listHistory $sels[0]`;
                string $findBlend[] = `ls -typ blendShape
$selsHistory`;
                $numBlends = size($findBlend);

                if ($numBlends == 0){
                        blendShape -n ($base+"Blend_1") $target
$base ;
                        color -rgb 0 0.55 1 $base;

                        //set keys at frame 1 and frame 25
                        select $base;
                        string $sels[] = `ls -sl`;
                        string $selsHistory[] = `listHistory
$sels[0]`;
```

```
                        string $findBlend[] = `ls -typ blendShape
$selsHistory`;



                        setKeyframe -v 0 -t 1
($findBlend[0]+".w[0]") ;
                        setKeyframe -v 1 -t 25
($findBlend[0]+".w[0]") ;
                        select -cl;


                        if (`objExists blendsLyr`){
                                parent $base blendGrp;
                                select $base;
                                layerEditorAddObjects blendsLyr;
                        }
                        else{
                                group -n blendGrp $base;
                                select -r $base;
                                createDisplayLayer -name
"blendsLyr" -number 1 -nr;
                        }
                        if (`objExists targetLyr`){
                                parent $target targetGrp;
                                select $target;
                                layerEditorAddObjects targetLyr;
                        }
                        else{
                                group -n targetGrp $target;
                                select -r $target;
```

```
                                createDisplayLayer -name
"targetLyr" -number 1 -nr;


                        }


                }

                else{

                        warning "There is already a blendshape on
this curve, plaease select a different one";

                }

                removeProc;

        }


        proc removeProc(){

                global string $aCurveSel[];

                global string $bCurveSel[];

                global string $target;

                global string $base;

                global int $buttonPush;

                global int $tCount;

                global int $bCount;


                //remove target curve from array

                select $target;

                string $targetItem[] = `ls -sl`;

                string $aCurveSel[] =
stringArrayRemove($targetItem, $aCurveSel);
```

```
                //remove base curve from array
                select $base;
                string $baseItem[] = `ls -sl`;
                string $bCurveSel[] = stringArrayRemove($baseItem,
$bCurveSel);
                if ($buttonPush == 1){
                        $tCount = ($tCount-1);
                        $bCount = ($bCount-1);
                        button -edit -label ($tCount+"  target
curves") ("aCountBtn");
                        button -edit -label ($bCount+"  base
curves") ("bCountBtn");
                        print ("job done"+"\n");
                }
                if ($buttonPush == 2){
                        $tCount = ($tCount-1);
                        $bCount = ($bCount-1);
                        button -edit -label ($tCount+"  target
curves") ("aCountBtn");
                        button -edit -label ($bCount+"  base
curves") ("bCountBtn");
                        print ("still some work to do"+"\n");
                        massProc;
                }

                select -cl;

        }
```

```
	proc randomKeyProc(){


		//select blendGrp;

		//$o = `ls -sl -o`;

		layerEditorSelectObjects blendsLyr;


		//$ch = `listRelatives -c $o`;

		//select -r $ch;


		int $count;

		//string $newPFX[];

		string $curveSel[] = `ls -sl`;

		int $numSel = size ($curveSel);


		for ($count = 0; $count<$numSel; $count++){

			int $rand = rand (0,16);

			string $selHistory[] = `listHistory
$curveSel[$count]`;

			string $findBlend[] = `ls -typ blendShape
$selHistory`;

			string $findWeights[] = `listAttr -m
($findBlend[0] + ".w")`;

			keyframe -e -iub true -r -o over -tc $rand
($curveSel[$count]+"Blend_1_"+$findWeights[0]) ;

		}

		select -cl;

	}
```

```
        proc randomSelProc(){

                global string $gMainProgressBar;  // This is
defined on maya startup

                string $currentSelection[] = `ls -selection -
flatten`;

                string $deselect[];



                //CHANGE THE BELOW VALUE OF $percentageToSelect TO
CHANGE THE PERCENTAGE WHICH WILL BE DESELECTED.

                float $percentageToSelect = 75;

                float $i = 0;

                int $numDesel = 0;



                progressBar -edit

                -beginProgress

                -isInterruptable true

                -status "Randomizing..."

                -maxValue `size($currentSelection)`

                $gMainProgressBar;



                $percentageToSelect /=100;

                for ($each in $currentSelection)

                {

                        $i = rand(0,1);

                        if ($i > $percentageToSelect) {

                                $deselect[$numDesel++] = $each;

                        }

                }
```

```
                progressBar -edit
                -step 1 $gMainProgressBar;

                select -d $deselect;

                progressBar -edit
                -endProgress
                $gMainProgressBar;
        }

        proc dynProc(){
                int $countA;
                int $countB = 0;
                string $dynSel[] = `ls -sl`;
                int $selCount = size ($dynSel);

                if ($selCount == 0){
                        warning "no curves Selected";
                }
                else{

                        for ($countA = 0; $countA < $selCount;
$countA ++){
                                string $selsHistory[] =
`listHistory $dynSel[$countA]`;
                                string $findBlend[] = `ls -typ
blendShape  $selsHistory`;
                                $numBlends = size($findBlend);
```

```
                                if ($numBlends == 0){
                                        $countB = $countB+1;
                                }
                        }
                        if ($countB > 0){
                                warning "one or more of the curves
you have selected has no blend applied";
                        }
                        else{
                                if(`objExists dynLyr`){
                                        //do nothing
                                }
                                else{
                                        createDisplayLayer -name
"dynLyr" -number 1 -empty;
                                }

                                if (`objExists "hairSystem1"`){

                                        select -tgl hairSystem1;
                                        makeCurvesDynamic 2
{ "1", "0", "1", "1", "0"};
                                }

                                else{

                                        makeCurvesDynamic 2
{ "1", "0", "1", "1", "0"};
```

```
                                    }
                                    for ($countA = 0; $countA <
$selCount; $countA ++){
                                            if (`objExists
"originalBlendGrp"`){

                                                    parent
$dynSel[$countA] originalBlendGrp;

                                                    select -cl;
                                            }
                                            else{

                                                    parent -w
$dynSel[$countA];
                                                    group -n
originalBlendGrp $dynSel[$countA];
                                                    select
hairSystem1OutputCurves;
                                                    select -cl;

                                            }
                                    }
                            }
                    }
            }
```

```
	proc pfxProc(){


		string $curveChk[] = `ls -sl`;
		int $numSel = size ($curveChk);
		if ($numSel == 0){
			warning "please ensure that you have at
least one curve selected";
		}
		else{
			string $cbLocation = `getenv
MAYA_LOCATION`;
			string $pfxSel = `optionMenuGrp -q -v
pfxList`;
			int $menuNum = `optionMenuGrp -q -select
pfxList`;
			int $dataVal = `menuItem -q -data ("item"+
$menuNum)`;


			for($count = 0; $count<$numSel; $count++)
{


				if (`objExists pfxGrp`){
					//do nothing
				}
				else{
					group -em -name pfxGrp;
				}


				if (`objExists pfxLyr`){
					//do nothing
```

```
                                        }
                                        else{
                                                createDisplayLayer -name
"pfxLyr" -number 1;
                                        }

                                        select $curveChk[$count];

                                        if ($dataVal == 1){
                                                eval("source \""+
$cbLocation+"/Examples/Paint_Effects/Pencils/"+$pfxSel+".mel\"");
                                                AttachBrushToCurves;

                                        }
                                        if ($dataVal == 2){
                                                eval("source \""+
$cbLocation+"/Examples/Paint_Effects/Pens/"+$pfxSel+".mel\"");
                                                AttachBrushToCurves;
                                        }

                                        string $newPFX[] = `ls -sl`;
                                        print ($count+"  ");
                                        print $newPFX;

                                        if (`objExists ($pfxSel+"Grp")`){

                                                parent $newPFX
($pfxSel+"Grp");
```

```
                                        }
                                        else{
                                                group -n ($pfxSel+"Grp")
$newPFX;
                                                parent ($pfxSel+"Grp")
pfxGrp ;
                                        }
                                }
                        }
                        SelectTool;
                }
```