

Sample Code Logic for NRC Virtual Orientation Center

```
Accessibility.isActive();
#include "mc_tween2.as"

// Function to send to appropriate frame of pan upon return
officeMC.gotoAndStop(_root.playFrame);

// Y position of movieMC on Screen
var yPos:Number = 282.85;

// Number of Name Plates selected for Avatar interaction
var namePlates:Number = 0;

// Total number of frames in Room Clip
var beginFrame:Number = 1;
var endFrame:Number = 126;
avatarPlaying = false;
var popUp:Boolean = false;

//Check if instruction panel should be up...
if (!_root.firstPlay){
    instructionPanel._visible = false;
} else {
    instructionPanel.gotoAndPlay(2);
}

// Initial Settings * Do not Change *
officeMC.stop();
var frame:Number = 1;
var inCenter:Boolean = true;
popup._visible = false;

// Hot Zone Distance Settings
var stageWidth:Number = Stage.width;
var eighth:Number = stageWidth/12;

var fastForward:Number = stageWidth-eighth;
var slowForward:Number = fastForward-eighth;

var fastReverse:Number = eighth;
var slowReverse:Number = fastReverse*2;

var initialForward:Number = slowForward-(eighth/2);
var initialReverse:Number = slowReverse+(eighth/2);
```

// Play functions

//will stop 'play', 'gotoAndPlay' and 'rewind', 'gotoAndRewind' functions

```
MovieClip.prototype.stopAll = function () {  
    delete this.onEnterFrame;  
    this.stop();  
}
```

//will stop 'rewind', 'gotoAndRewind' functions

```
MovieClip.prototype.stopRewind = function () {  
    delete this.onEnterFrame;  
}
```

//will play the timeline backwards and continal to loop until 'stopAll', 'stopRewind' or 'pause' are called

//note: the 'rewind' function will cancel the 'play' function BUT the 'play' function will not cancel the 'rewind' function

//you must call 'stopAll' or 'stopRewind' before using 'play' when your timeline is playing backwards

```
MovieClip.prototype.rewind = function () {  
    this.stop();  
    this.onEnterFrame = function () {  
        if(this._currentframe == 1){this.gotoAndStop(endFrame);}  
        if (this._currentframe > 0) {  
            this.prevFrame ();  
        } else {  
            this.gotoAndStop (this._totalframes);  
        }  
    }  
}
```

//extends the typical play allowing the play to loop at the end without a goto and stop function call

```
MovieClip.prototype.scrollplay = function () {  
    this.stop();  
    this.onEnterFrame = function () {  
        if(this._currentframe == endFrame){this.gotoAndStop(beginFrame);}  
        if (this._currentframe < endFrame) {  
            this.nextFrame ();  
        } else {  
            this.gotoAndStop (this._totalframes);  
        }  
    }  
}
```

//will goto the specified frame and play the timeline backwards and continal to loop until 'stopAll', 'stopRewind' or 'pause' are called

```
MovieClip.prototype.gotoAndRewind = function (frame) {  
    this.gotoAndStop (frame);  
    this.rewind();  
}
```

```

//will pause the timeline for the specified amount of time. will always play forwards when the
pause has finished
MovieClip.prototype.pause = function (time) {
    delete this.onEnterFrame;
    secondsToPause = time;
    pauseInt = setInterval (this, "restart", secondsToPause * 1000);
}

restart = function () {
    clearInterval (pauseInt);
    play();
}

////////////////////////////////////
// CC Window Fucntions (do not change)
////////////////////////////////////

// CC window function

import flash.external.ExternalInterface;

function ccCheck(){
    if (ccON){
        ccUp(_root.mainMC.ccWindow);
    }
}

function ccUp(ccWindow){

    screenWidth = System.capabilities.screenResolutionX;
    screenHeight = System.capabilities.screenResolutionY;
    leftPostion = (screenWidth/2)-276;
    topPosition = (screenHeight/2)-200;

    if (ExternalInterface.available) { ExternalInterface.call("window.open", ccWindow, "win",
    "height=366,width=552,toolbar=no,scrollbars=no,top="+topPosition+",left=" + leftPostion); }

}

ccButton.onRelease = function(){

    //ccUp(_root.mainMC.ccWindow);
    if (ccON){
        ccON = false;
        ccButton.gotoAndPlay("off");
    }else{
        ccON = true;
        ccButton.gotoAndPlay("on");
    }
    if (avatarPlaying){
        ccCheck();
    }
}

```

```
}  
}
```

```
////////////////////////////////////  
// Audio Button Fucntions (do not change)  
////////////////////////////////////
```

```
audioON = true;
```

```
function setAvatar(currentAvatar){  
  currentAvatarMC = eval(currentAvatar);  
}
```

```
audioBTN.onPress = function(){
```

```
  trace ("Audio Setting = " + audioON);  
  trace ("avatarPlaying = " + avatarPlaying);  
  if (audioON){
```

```
    trace("..but turning it off");  
    audioON = false;  
    audioBTN.gotoAndPlay("off");  
    //_root.music.stop();  
    if(avatarPlaying == false){  
      fadeOut();  
    }  
  }else{
```

```
    trace("..but turning it on");  
    audioON = true;  
    audioBTN.gotoAndPlay("on");  
    if(avatarPlaying == false){  
      music.stop();  
      music.setVolume(50);  
      music.start();
```

```
      //fadeIn();  
    }  
  }
```

```
  if (avatarPlaying){  
    checkAudio();  
  }  
}
```

```
audioBTN.onRollOver = function(){  
  if (audioON){  
    audioBTN.gotoAndStop("onover");  
  }else{  
    audioBTN.gotoAndStop("offover");  
  }  
}
```

```

audioBTN.onRollOut = function(){
if (audioON){
    audioBTN.gotoAndStop("on");
}
else{
    audioBTN.gotoAndStop("off");
}
}

#include "mc_tween2.as"

function checkAudio(setting){
if (audioON){
    currentAvatarMC.snd.volumeTo(100, 1, "linear");
    //currentAvatarMC.snd.setVolume(100);
}
else{
    currentAvatarMC.snd.volumeTo(0, 1, "linear");
    //currentAvatarMC.snd.setVolume(0);
}
}

////////////////////////////////////
// PopUp Window Function
////////////////////////////////////

function callPopUp(popWindow){
    trace(popWindow);

    screenWidth = System.capabilities.screenResolutionX;
    screenHeight = System.capabilities.screenResolutionY;
    leftPostion = (screenWidth/2)-276;
    topPosition = (screenHeight/2)-200;

    if (ExternalInterface.available) { ExternalInterface.call("window.open", popWindow,
"win", "height=366,width=552,toolbar=no,scrollbars=no,top="+topPosition+",left=" +
leftPostion); }

}

////////////////////////////////////
// Background Music and Fade Functions
////////////////////////////////////

// initiate sound
music = new Sound();

music.attachSound("audio_bed");

if(_root.comingFromIntro){
    trace("First time in from Intro");
}

if (!_root.comingFromIntro){

```

```

        music.start(0, 999999);
        music.setVolume(50);
        vol = 50;
        comingFromIntro = false;
    }

```

```

/*function fadeOut(){

```

```

    _root.fOut = setInterval(fade2,5);
}

```

```

function fadeIn(){
    _root.fIn = setInterval(fade1,5);
}*/

```

```

/*function fadeOut(){
    import mx.transitions.Tween;
    import mx.transitions.easing.*;
    musicTween = new Tween(music, "_volume", Strong.easeIn, 50,0, 0.5, true)
    musicTween.onMotionChanged=function(){
        music.setVolume(music._volume)
    }
    musicTween.onMotionComplete = function(){
        music.stop();
        music.setVolume(50);
    }
}
*/

```

```

function fadeOut(){
    music.volumeTo(0, 0.5, "linear");
    music.onTweenComplete = function(){
        music.stop();
        music.setVolume(100);
    }
}

```

```

function fadeIn(){
    music.volumeTo(100, 1, "linear");
    music.onTweenComplete = function(){
        music.stop();
        music.setVolume(100);
    }
}

```

```

function fade1() {
    vol += 2;
    music.setVolume(vol);
}

```

```

        if(vol>100){
            clearInterval(_root.fIn);
            //trace("in");

        }
    }

    function fade2() {
        vol -= 2;
        music.setVolume(vol);

        if(vol<1){
            clearInterval(_root.fOut);
            music.stop();
            music.setVolume(100);
            //trace("out");
        }
    }

    function checkAudio2(setting){
    if (audioON){
        currentAvatarMC.snd.setVolume(100);
    }else{
        currentAvatarMC.snd.setVolume(0);
    }
    }

    _root.firstPlay = false;

    //////////////////////////////////////
    // SCORM                                //
    //////////////////////////////////////

    if (_root.applicationType == 1){

        totalComplete = 0;
        roomComplete = 0;

        //////////////////////////////////////
        // Get overall count of complete rooms //
        //////////////////////////////////////

        overallData = ExternalInterface.call("getLessonData", 1);
        numberRooms = overallData.length;

        for (i=0; i<numberRooms; i++) {
            trace(overallData.charAt(i));

            if (overallData.charAt(i)>"1"){
                totalComplete++;
            }
        }
    }

```

[illegible]


```

totalComplete = 0;
roomComplete = 0;

////////////////////////////////////
// Get overall count of complete rooms //
////////////////////////////////////

overallData = ExternalInterface.call("getLessonData", 1);
numberRooms = overallData.length;

for (i=0; i<numberRooms; i++) {
    trace(overallData.charAt(i));

    if (overallData.charAt(i)>"1"){
        totalComplete++;
    }
}

totalTXT.text = totalComplete+" of "+numberRooms+" areas completed"

////////////////////////////////////
// Get count of items in room //
////////////////////////////////////

roomNumber = 4;
var room_array:Array = new Array()

roomData = ExternalInterface.call("getLessonData", roomNumber);
roomSize = roomData.length;

for (i=0; i<roomSize; i++) {
    if (roomData.charAt(i)>"0"){
        roomComplete++;
    }
    room_array[i] = roomData.charAt(i);
}

roomTXT.text = roomComplete+" of "+roomSize+" area items viewed"

} // close application type check

////////////////////////////////////
// Set new count of items in room //
////////////////////////////////////
function updateRoom() {

if (_root.applicationType == 1){

    var newRoom:String = "";
    var roomFinished:String = "";
    for (i=0; i<roomSize; i++){
        newRoom = newRoom + room_array[i];
        roomFinished = roomFinished + "1";
    }
}
}

```

```

    }

    ExternalInterface.call("setLessonData", roomNumber, newRoom);

    if (newRoom == roomFinished){
        ExternalInterface.call("setLessonComplete", roomNumber);
        totalComplete++;
        totalTXT.text = totalComplete+ " of "+numberRooms+" areas completed";
    }
}
}

import mx.transitions.Tween;
import mx.transitions.easing.*;

initialYpos = 282.85;

// Detect keypress

var keyListener:Object = new Object();
keyListener.onKeyDown = function() {
    if (Key.isDown(Key.LEFT)) {
        officeMC.rewind();
    } else if (Key.isDown(Key.RIGHT)) {
        officeMC.scrollplay();
    }
}
keyHit = Key.getAscii();
trace (keyHit);

switch(keyHit){

    case 51:
        _root.currentMovie="hall2h.swf";
        _root.gotoAndPlay(3);
        _root.playFrame = 125;
        break;
    case 50:
        _root.currentMovie="room2a.swf";
        _root.gotoAndPlay(3);
        _root.playFrame = 125;
        break;
    case 49:
        _root.currentMovie="room2b.swf";
        _root.gotoAndPlay(3);
        _root.playFrame = 125;
        break;
    case 52:
        _root.currentMovie="elevator3.swf";
        _root.gotoAndPlay(3);
        _root.playFrame = 125;
        break;
    case 97:
        _root.currentMovie="room1a.swf";
        _root.gotoAndPlay(3);

```

```

        _root.playFrame = 125;
        break;
        case 98:
            _root.currentMovie="room1b.swf";
            _root.gotoAndPlay(3);
            _root.playFrame = 125;
            break;
        case 99:
            _root.currentMovie="room1c.swf";
            _root.gotoAndPlay(3);
            _root.playFrame = 125;
            break;
        case 101:
            _root.currentMovie="lobby_east.swf";
            _root.gotoAndPlay(3);
            _root.playFrame = 125;
            break;
        case 119:
            _root.currentMovie="lobby_west.swf";
            _root.gotoAndPlay(3);
            _root.playFrame = 125;
            break;
        case 118:
            _root.currentMovie="elevator.swf";
            _root.gotoAndPlay(3);
            _root.playFrame = 125;
            break;
        case 104:
            _root.currentMovie="hall1h.swf";
            _root.gotoAndPlay(3);
            _root.playFrame = 125;
            break;
        default:
            //trace ("The number is not equal to 0, 2, 4 or 8");
    }
};

// Detect keyrelease

keyListener.onKeyUp = function() {
    officeMC.stopAll();
};

Key.addListener(keyListener);

// Detect mouse position

mouseListener = new Object();
mouseListener.onMouseMove = function () {
    stagexSpot=_xmouse;

```

```

        stageySpot=_ymouse;
        popup.prompt.text = "";
        popup._visible = false;
        moveFunction();

        updateAfterEvent();

        // trace("Mouse X Position = "+stagexSpot);
    }
    Mouse.addListener(mouseListener);

    function moveFunction(){
        updateAfterEvent();
        // Move to the Left functionality

        if(stagexSpot<slowReverse and stageySpot<529 and stageySpot>66 and !popUp and !
        helperOver){
            officeMC.rewind();
        }

        // Move to the Right functionality

        if(stagexSpot>slowForward and stageySpot<529 and stageySpot>66 and !popUp and !
        helperOver){
            officeMC.scrollplay();
        }

        // Stop when in center

        if(stagexSpot<slowForward and stagexSpot>slowReverse){
            officeMC.stopAll();
        }

        if(stageySpot>529 or stageySpot<66){
            officeMC.stopAll();
        }

        // Return to Center functionality

        if(stageySpot>505 and inCenter and !popUp){
            var vertTween:Tween = new Tween
            (officeMC, "_y",Strong.easeOut,initialYpos,initialYpos-56,3,true);
            inCenter = false;
            vertTween.onMotionFinished = function (){
                this.stop()
            }
        }

        if(stageySpot<95 and inCenter and !popUp){
            var vertTween:Tween = new Tween
            (officeMC, "_y",Strong.easeOut,initialYpos,initialYpos+56,3,true);

```

```

        inCenter = false;
        vertTween.onMotionFinished = function (){
            this.stop()
        }
    }
    if(stageySpot>95 and stageySpot<505 and !inCenter and !popUp){
        var vertTween:Tween = new Tween
        (officeMC, "_y",Strong.easeOut,officeMC._y,initialYpos,3,true);
        inCenter = true;
        vertTween.onMotionFinished = function (){
            this.stop()
        }
    }

    previousXspot = stagexSpot;
    previousYspot = stageySpot;
}

```

```

mouseListener.onMouseMove = function ():Void { }
trace("Begin Transition");

stop();

```