

Below is sample Code from Joseph Illoreta's Final Project "Pittsburgh SoundScape" for the Interactive Multimedia Programming class. Joseph was Psychology, Design and Architecture with an HCI double major with very little experience in programming in the beginning of the semester. He will continue his studies in the School of Design Master in Interaction Design program Fall 2006.

```
global hoveredSprite, mySerialConnection, analogSensorData, buttonData,
touchSensorData,tilesList, myTile
global firstTileChan, lastTileChan, firstBuildingChan, lastBuildingChan,
soundsList
property entered
property spriteChan, hdist, vdist, ldist, rdist, panValue , panSide -- (-1, 1)
property counter
property mouseLoc, mouseLocX, mouseLocY, locH, locV, mouseH, mouseV
property spriteLoc, spriteLocX, spriteLocY
property d, tileChan, lookatSprite, buildChan
property myNewOpacity, myTileProperties, myOpacity
global blendVal, oldMouse, newMouse, soundPlay1

property mouseNowx, mouseNowy, mouseThenx, mouseTheny
property soundChan, soundSymbolChan
property deltaS, deltaV, panVal, volVal

global firstSoundSymbolChan, lastSoundSymbolChan, firstSoundSpriteChan,
lastSoundSpriteChan

-- DATA Format from Basic STAMP
-- DEC3 ADCBITS, DEC1 ButtonState, DEC2 TouchState, RETURN
-- ADCBITS is the information coming from the analog sensor

--there are a total of 31-69: 38 building sprites/71-109 for the tile sprites

on enterFrame me

    --detect whether the mouse has moved or not
    checkMouse()

end

on checkMouse

    oldMouse = mouseH

    if mouseThenx = the mouseH and mouseTheny = the mouseV then
    end if

    --ONCE movement in mouse is detected, we calculate the distance
    --between each building and the mouse position

    if mouseThenx <> the mouseH or mouseTheny <> the mouseV then

        repeat with buildChan = firstBuildingChan to lastBuildingChan
            checkDistance(buildChan)--(buildChan)
        end repeat

        checkSoundDistance
    end if

end
```

```

on checkDistance (buildChan) --spriteChan

    --CALCULATE DISTANCES and Calculate (left/right) (pos/neg)
    hdist = sprite(buildChan).locH - the mouseH
    vdist = sprite(buildChan).locV - the mouseV

    --CALCULATE absolute values //begins DETERMINING VOLUME for soundChannels
    hdist = abs(hdist)
    vdist = abs(vdist)

    -- CALC the CLOSEST value (hdist or vdist) and send as the smallest value as
    the blend value

    if hdist < vdist then
        hdist = vdist
    end if

    if hdist > 201 then

    else if hdist > 20 and hdist < 201 then

        --put "blending"
        if hdist < 21 then blendVal = 100 --20 pixels from the center is full blend
            blendVal = (100 - ((hdist - 20)/2))

        if buildChan >= firstBuildingChan and buildChan<=lastBuildingChan then
            changeOpacity(buildChan)
        end if
    end if
end

on checkSoundDistance

--this handler checks the soundsymbol sprites' distances from the MOUSE
repeat with n= firstSoundSymbolChan to lastSoundSymbolChan

    --check for l/r status
    if the mouseH < sprite(n).locH then

        panSide = 1

    else if mouseH = sprite(n).locH then

        panSide = 0

    else

        panSide = -1

    end if

    deltaS = (abs(sprite(n).locH - the mouseH))
    deltaV = (abs(sprite(n).locV - the mouseV))

    --calculate PanValue

    if deltaS>255 then

        --distances greater than 255 pixels out default to 255 (end of spectrum)

        panVal = 255

    else if deltaS<10 then

        --distances within 10 pixels default to 100 (no pan)

        panVal = 0

    else

        panVal = deltas

```

```

end if
--calculate VolValue
if deltaV > deltaS then

    deltaS = deltaV
    --the shorter distance must be used to calculate the volume
end if
if deltaS > 200 then --if mouse is more than 400 there is no volume
    volVal = 0
else if deltaS<10 then
    --if mouse is within 10 pixels, automatically set vol to 100
    volVal = 100
else
    volVal = integer((200-deltaS)/2) -- volVal is a rounded integer of the
function
end if
myTempPan = panVal * panSide

-- retrieve soundproperty listat soundsList at n (soundChannel)
mySoundProperties = getaProp(soundsList, n)

setat (mySoundProperties , 3, myTempPan)
setat (mySoundProperties, 2, volVal)

-- replace list property into soundsList
setaProp soundsList, n, mySoundProperties

end repeat

playSounds

end

on playSounds

--PLAYSOUNDS processe the soundsList and applies to the soundchannels/files
what changes were made in the previous handler

repeat with s = firstSoundSymbolChan to lastSoundSymbolChan
    ch = s-80

    mySoundProperties = getaProp(soundsList, s)

    if sound(ch).isbusy()then
        if getAt(mySoundProperties, 2) = 0 then
--if volume is zero and sound is play, stop the play

            sound(ch).fadeout()
            sound(ch).stop()

```

```

else
    sound(ch).volume = getAt(mySoundProperties, 2)
    sound(ch).pan = getAt(mySoundProperties, 3)
end if
else if not sound(ch).isbusy() then
    if getAt(mySoundProperties, 2) > 0 then
--sound channel is not playing and volume is greater than zero
        sound(ch).pan = getAt(mySoundProperties, 3)
        sound(ch).play(member(getAt(mySoundProperties, 1)))
        sound(ch).fadein(getAt(mySoundProperties, 2))
    end if
end if
end repeat
end

on changeSoundValues(buildChan, blendVal)
    mySoundProperties = getaProp(soundsList, buildChan)
    mySoundChannel = getat (mySoundProperties,1)

    if not sound(1).isbusy() then
        sound(1).play(member(mySoundChannel))
    end if
end

on changeOpacity (buildChan)
    myTileProperties = getaProp(tilesList, buildChan)
    myTile = getat (myTileProperties , 1)
    --put myTile

    sprite(myTile).blend = blendVal

    -- to update the opacity value
    --based on the sprite of the building, it is retrieving the properties
    -- of the tile that matches the building in the list.
    --then sets the second index of the tile properties list (opacity) that
    --corresponds to the tile of the building that is calculating the
    --distancefrom the mouse

    --Change TILE Opacity in tilesList
    myTileProperties = getaProp(tilesList, buildChan)
    setat (myTileProperties , 2, blendVal)
    setaProp tilesList, buildChan, myTileProperties
end

on exitFrame me
--store the location of the mouse for comparison on checkMouse in exitFrame

```

```
mouseThenx = the mouseH  
mouseTheny = the mouseV  
go to the frame  
end
```