Adding Sound Effects to Mouse Events (Janet Terra)

We have all become very attuned to the sounds of our computer. Games and even serious applications often have different sounds associated with mouse events - a wrong move (buzzer), a successful move (warble), a critical crash (annoying honk). Here's a demo that shows how two separate wave files can signal to the user that the left button has been pressed, the mouse dragged, and the left button released. You will need two small wave files to run this demo. You can either use your own or download sfxMoveMouse.wav and sfxUpMouse.wav from the this Wiki File Archive site. Both files are contained in the zip file

sfxMouseEvents.zip

- Details
- Download
  - 4 KB

. The XOR rule just adds a bit of visual effect. A blank screen would have been boring.

```
Nomainwin
WindowWidth = 407
WindowHeight = 452
UpperLeftX = Int((DisplayWidth - WindowWidth)/2)
UpperLeftY = Int((DisplayHeight - WindowHeight)/2)
Menu #main, "&Options", "E&xit", CloseMainMenu
Graphicbox #main.g, 0, 0, 400, 400
Open "Mouse Sound Effects" for Window as #main
#main, "Trapclose CloseMainTrap"
#main.g, "Down; Fill Black; Flush; Color Darkblue"
#main.g, "Rule ";_R2_NOTXORPEN
#main.g, "When leftButtonDown DownMouse"
Wait
Sub CloseMainMenu
Call CloseMainTrap "#main"
End Sub
Sub CloseMainTrap handle$
Close #main
End
End Sub
Sub DownMouse handle$, xVar, yVar
#main.g, "When leftButtonDown"
#main.g, "Place ";xVar;" ";yVar
#main.g, "Circlefilled 10"
```

```
Playwave "sfxMoveMouse.wav", Loop
#main.g, "When leftButtonUp UpMouse"
#main.g, "When leftButtonMove MoveMouse"
End Sub
Sub MoveMouse handle$, xVar, yVar
HueStrand$ = "Red Yellow Blue Green Darkpink Brown Darkcyan"
hue = Int(Rnd(1) * 7) + 1
#main.g, "Backcolor ";Word$(HueStrand$, hue)
#main.g, "Place ";xVar;" ";yVar
#main.g, "Circlefilled 10"
#main.g, "Discard"
End Sub
Sub UpMouse handle$, xVar, yVar
#main.g, "When leftButtonMove"
#main.g, "When leftButtonUp"
Playwave "sfxUpMouse.wav", Async
#main.g, "When leftButtonDown DownMouse"
End Sub
```