**Pontificia Universidad Católica Madre y Maestra**



**Facultad de ingeniería**

**Ingeniería en Sistemas y Computación**

Sistemas Operativos 1

**Nombres:**

Francisco M. Suero N.

Hector Rafael Aristy

**Matriculas:**

2010-6930

2011-5972

**Tema:**

Algoritmos de mapeo con la fundamentación del driver.

**Profesor:**

Rodrigo Orizondo

**Grupo:**

001

**Fecha:**

21 Abril 2016

# Introducción

Este documento especificará a continuación los algoritmos que utilizaremos para el mapeo con respecto a la reenderizacion del cursor en pantalla y la utilización del cursor en base a las acciones del usuario con respecto al mouse. Cabe decir que se utilizó como referencia los algoritmos del filtro del driver del mouse localizados en GIT como base, y como referencia sobre la fricción usamos un stopwatch para ir llamando a una función que disminuye la velocidad (para mas información ver el apéndice de referencia).

Algoritmos:

counterForceActive = FALSE; // Variable para el control de activo/inactivo de la desaceleracion.

mouseFricc = InputDataStart; //asignacion del estado actual del cursor

allowFriction = TRUE; // Variable para el control de activo/inactivo de la desaceleracion.

//MIENTRAS SE ESTE MOVIENDO EL MOUSE....

for (int i = 0; i < InputDataEnd - InputDataStart; i++){

deltaPosX = InputDataStart[i].LastX; // Se asigna el valor actual de la posicion del cursor en el eje X.

deltaPosY = InputDataStart[i].LastY; // Se asigna el valor actual de la posicion del cursor en el eje Y.

curVelx += DEFAULT\_ACCEL\_X \* deltaPosX; // La velocidades en los ejes X y Y aunmentan constantemente, de acuerdo

curVely += DEFAULT\_ACCEL\_Y \* deltaPosY; //a la constante de la acceleracion establecida.

if (InputDataStart[i].LastX > 0) { // En caso de que se mueva el mouse hacia la derecha, el valor del puntero sera positivo, por lo tanto

deltaPosX += curVelx; // se le suma a la posicion del cursor en el eje X el desplazamiento recorrido segun la velocidad.

}

else if (InputDataStart[i].LastX < 0) { // En caso de que se mueva el mouse hacia la izquierda, el valor del puntero sera negativo, por lo tanto

deltaPosX -= curVely; // se le resta a la posicion del cursor en el eje X el desplazamiento recorrido segun la velocidad.

}

if (InputDataStart[i].LastY > 0) { // En caso de que se mueva el mouse hacia arriba, el valor del puntero sera positivo, por lo tanto

deltaPosY += curVely; // se le suma a la posicion del cursor en el eje Y el desplazamiento recorrido segun la velocidad.

}

else if (InputDataStart[i].LastY < 0) { // En caso de que se mueva el mouse hacia bajo, el valor del puntero sera negativo, por lo tanto

deltaPosY -= curVely; // se le resta a la posicion del cursor en el eje Y el desplazamiento recorrido segun la velocidad.

}

deltaPosX = curVelx / STANDARIZER; //se gradúa la posicion del cursor en el eje X.

deltaPosY = curVely / STANDARIZER; //se gradúa la posicion del cursor en el eje Y.

InputDataStart[i].LastX = deltaPosX; //Se le asigna el nuevo valor de la poscuin en X del cursor.

InputDataStart[i].LastY = deltaPosY; //Se le asigna el nuevo valor de la poscuin en X del cursor.

}

Friccion:

hDevice = WdfTimerGetParentObject(Timer);

devExt = FilterGetData(hDevice);

if (counterForceActive == TRUE) {

if (regActive == TRUE && allowFriction == TRUE) {

mouseFricc.LastX = (long)deltaPosX; //asignacion de la ultima posicion en el eje X del cursor.

mouseFricc.LastY = (long)deltaPosY; //asignacion de la ultima posicion en el eje Y del cursor.

(\*(PSERVICE\_CALLBACK\_ROUTINE)devExt->UpperConnectData.ClassService)( //llama al callback enviandole los valores

devExt->UpperConnectData.ClassDeviceObject, // para ir actualizands del puntero.

&mouseFricc,

(&mouseFricc) + 1,

&Consumed

);

}

} // se ejercen las aritmeticamnete la desaceleracion en cada eje, deacuerdo a la friccion.

if (curVelx > 0) {

curVelx -= DEFAULT\_FRICC\_X;

}

if (curVely > 0) {

curVely -= DEFAULT\_FRICC\_Y;

}

if (curVelx < 0) {

curVelx += DEFAULT\_FRICC\_X;

}

if (curVely < 0) {

curVely += DEFAULT\_FRICC\_Y;

}

if (curVelx == 0 && curVely == 0) { //no hacer nada en caso de que el puntero se haya parado.

counterForceActive = FALSE;

}

Bibliografía:

MouFiltr:

<https://github.com/Microsoft/Windows-driver-samples/blob/master/input/moufiltr/moufiltr.c>

Friccion:

<https://msdn.microsoft.com/en-us/library/windows/hardware/ff550050(v=vs.85).aspx>

configuración con el timer:  <https://msdn.microsoft.com/en-us/library/windows/hardware/ff552519(v=vs.85).aspx>

para como comenzar a contar el stopwatch:

<https://msdn.microsoft.com/en-us/library/windows/hardware/ff550054(v=vs.85).aspx>

Como associar el stopwatch con una funcion:

<https://msdn.microsoft.com/en-us/library/windows/hardware/ff541823(v=vs.85).aspx>