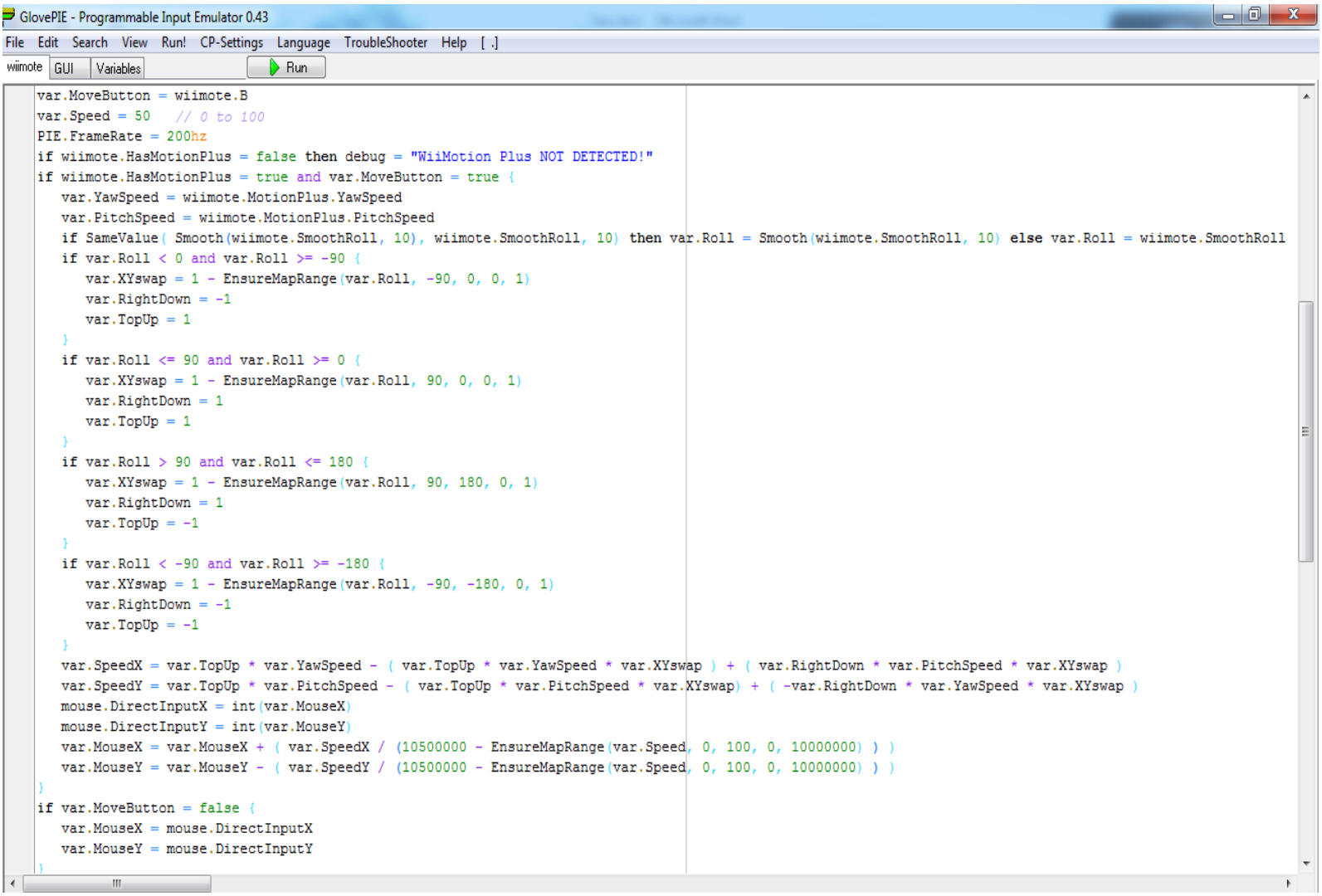
5.2. Análisis y diseño de interfaces gráficas para el Wiimote.

5.2.1. Análisis de la aplicación GlovePIE para programación de Wiimote.

Para conocer los avances en la programación del Wiimote primero se investigaron aplicaciones ya implementadas para funcionar con el Wiimote.

La mejor encontrada durante el desarrollo de este proyecto fue esta, una aplicación que se puede encontrar de forma gratuita en la red y está diseñada para programar dispositivos como el Wiimote con el propósito de personalizar las funciones de los mismos; sin embargo, para realizar esto, la aplicación requiere del conocimiento previo de su lenguaje de programación específico y por ende esto es un obstáculo para el usuario promedio al intentar configurar un dispositivo. A continuación se muestra una ventana de esta aplicación con el respectivo código para simular el movimiento del ratón con el Wiimote:



Además, se requiere de código similar al siguiente para asignar funciones, como los botones del ratón y del teclado, al Wiimote:

```
Key.I = Wiimote.Up
Key.J = Wiimote.Left
Key.K = Wiimote.Down
Key.L = Wiimote.Right
Key.U = Wiimote.One
Key.O = Wiimote.Two
Mouse.RightButton = Wiimote.home
Mouse.WheelUp = Wiimote.plus
Mouse.WheelDown = Wiimote.minus
mouse.LeftButton = wiimote.A

Key.Z = Nunchuk.Z
if nunchuk.C = true {
   WASD = nunchuk.Joy
 }
```

La desventaja de esta aplicación es que el usuario debe conocer el código necesario para modificar la configuración del Wiimote cada vez que requiera ajustarla a sus necesidades.

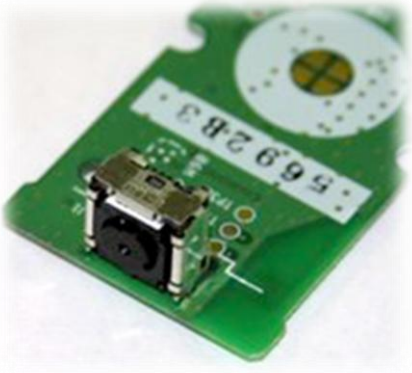
5.2.2 Análisis del Wiimote como puntero.

Existen dos dispositivos que componen al Wiimote y que permiten realizar un muestreo y procesamiento de señales capaces de indicar su posición.

El primero es el acelerómetro. Este nos permite saber el sentido de la gravedad respecto al Wiimote y por lo tanto es posible saber la posición del mismo.

El segundo es una cámara infrarroja. Esta nos permite detectar fuentes de luz infrarroja y el Wiimote nos proporciona hasta 4 coordenadas de distintas fuentes de luz. Por lo tanto, es posible realizar cálculos para detectar la orientación del Wiimote respecto a estas luces.

A continuación se muestra la vista frontal del Wiimote, donde se puede apreciar la ventana de la cámara (izquierda) y la parte interna del Wiimote donde se encuentra la misma (derecha):



Sin embargo, para esta tesis se pretende utilizar el acelerómetro ya que, a diferencia de la cámara infrarroja, este no requiere de apuntar siempre a un lugar específico. Además, las luces infrarrojas deben estar fijas arriba o debajo de la pantalla, por lo que esta configuración le resta movilidad al Wiimote y provoca incomodidad a algunos usuarios que no encuentran factible utilizar luces infrarrojas cerca de sus pantallas.

5.2.3 Análisis de requerimientos de la interfaz gráfica a diseñar.

La interfaz gráfica de la aplicación que funcionará en conjunto con el Wiimote debe constar de tres tareas básicas para asistir la navegación virtual, con la facilidad para que cualquier usuario pueda manipularla. Esas tres tareas son:

- La capacidad de facilitar la programación de los botones del Wiimote de acuerdo a las necesidades del usuario, utilizando menús que permitan visualizar las distintas opciones a seleccionar para cada botón;

- Poder ajustar la sensibilidad del acelerómetro del Wiimote para permitir un movimiento adecuado en cualquier tamaño de pantalla y para cada tipo de usuario, con la ayuda de una barra de control graduada que permita visualizar el nivel de sensibilidad; y

- La activación del Wiimote desde la misma interfaz gráfica a través de un botón, para iniciar su funcionamiento y que el usuario detecte este suceso con facilidad.

5.2.4 Diseño de la interfaz gráfica de la aplicación.

Para crear una interfaz gráfica fácil de comprender por cualquier usuario y compatible con la todos los sistemas operativos de Windows, en esta tesis se pretende utilizar WinAPI, ya que sus componentes, similares a los de una ventana promedio de Windows, permitirán al usuario sentirse familiarizado con la aplicación del Wiimote, la cual llamaremos **WiimoteAPI** en esta tesis.

El componente principal de la WiimoteAPI será el **combobox**, el cual se muestra a continuación:

Este permitirá desplegar y seleccionar la configuración deseada de una lista de opciones de funciones a asignar a cada botón del Wiimote, ya sean de ratón (click izquierdo por ejemplo) o de teclado (letras, números, entrar, escape, pausa, etc.)

También se utilizará un **botón** para iniciar la simulación de movimiento del ratón con el Wiimote, así como para inicializar las funciones definidas de los botones. Dicho botón tendrá la siguiente apariencia:

Para indicar a qué botón corresponde cada combobox se requieren etiquetas de texto, también para dar mensajes al usuario o incluso para indicar escalas o numeraciones que llegan a ser necesarias en elementos como el **slider**, el cual se muestra a continuación:

El slider permite asignar un valor a cierto parámetro; en este caso es necesario utilizar un slider para definir la sensibilidad del movimiento del acelerómetro del Wiimote para simular el control de sensibilidad de un ratón, y así ajustar adecuadamente este movimiento ya que para cada tamaño de pantalla existe una velocidad de puntero distinta.