How to use KeyboardMouseControl:

1. Open a WPF or Windows Form Application project.
2. Add the KeyboardMouseControl class file to the project.
3. Add “mousekeyboardactivitymonitor.dll” reference to the project. If it’s not available, please download it from here- <http://api.256file.com/mousekeyboardactivitymonitor.dll/en-download-21063.html> .
4. Make an instance of the KeyboardMouseControll class.  
     
   KeyboardMouseControl keyMouse = new KeyboardMouseControl();  
     
   The keyboard & mouse hook is initialized in the constructor. The mouseFlag & keyFlag are both set to 0 to enable it. For the value 1 both are disabled.
5. To disable keyboard call the function keyDisable & to disable mouse call the function mouseDisable on the created instance.  
     
   keyMouse.keyDisable();

keyMouse.mouseDisable();  
  
The methods are as following:  
  
public void keyDisable()

{

keyFlag = 1;

Console.WriteLine("disabling keyboard");  
  
}

public void mouseDisable()

{

mouseFlag = 1;

Console.WriteLine("disabling mouse");

}

1. To enable the keyboard call the function keyEnable & to enable the mouse call the function mouseEnable on the created instance of KeyboardMouseControl class.  
     
   keyMouse.keyEnable();  
   keyMouse.mouseEnable();  
     
   The methods are written below:  
     
   public void keyEnable()

{

keyFlag = 0;

Console.WriteLine("enabling key");

}  
  
public void mouseEnable()

{

mouseFlag = 0;

Console.WriteLine("enabling mouse");

}

1. All the key events are suppressed in HookManager\_KeyDown & HookManager\_KeyUp, setting the value of ( KeyEventArgs e ) e.SuppressKeyPress to ‘true’. So the Key press events are not passed to the system to process it, rather it says to the system that the key press has been handled.  
     
   private void HookManager\_KeyDown(object sender, KeyEventArgs e)

{

Console.WriteLine(e.KeyData.ToString() + " Pressed");

if (e.KeyCode == Keys.Escape)

{

keyFlag = 0;

mouseFlag = 0;

}

if (keyFlag == 1)

{

e.SuppressKeyPress = true;

}

}

private void HookManager\_KeyUp(object sender, KeyEventArgs e)

{

Console.WriteLine(e.KeyData.ToString() + " Released");

if (e.KeyCode == Keys.Escape)

{

keyFlag = 0;

mouseFlag = 0;

}

if (keyFlag == 1)

{

e.SuppressKeyPress = true;

}

}

1. The ‘Esc’ key is used to enable both the mouse & keyboard in case you disable both in HookManager\_KeyUp & HookManager\_KeyDown method. So it sets the keyFlag & mouseFlag to ‘0’.
2. All the mouse click events are also suppressed in HookManage\_MouseDown method. When the value of mouseFlag is equal to ‘1’. The mouse click event ( MouseEventExtArgs e) e.Handled is set to ‘true’. So the mouse click event is not passed to the System rather it says to the system that the event has been handled.   
     
   private void HookManager\_MouseDown(object sender, MouseEventExtArgs e)

{

Console.WriteLine(e.Button.ToString() + " Pressed");

//e.Handled = true;

if (mouseFlag == 1)

{

e.Handled = true;

}

}