using System;

using System.Runtime.InteropServices;

using System.Windows.Forms;

namespace Prueba

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

[System.Runtime.InteropServices.DllImport("user32.dll")]

static extern bool SetWindowPos(IntPtr hWnd, IntPtr hWndInsertAfter, int X, int Y, int cx, int cy, uint uFlags);

[System.Runtime.InteropServices.DllImport("user32.dll")]

private static extern bool ShowWindow(IntPtr hWnd, int nCmdShow);

[DllImport("USER32.DLL", CharSet = CharSet.Unicode)]

public static extern IntPtr FindWindow(String lpClassName, String lpWindowName);

[DllImport("USER32.DLL")]

public static extern bool SetForegroundWindow(IntPtr hWnd);

private bool BringApp(string \_Process, int x, int y)

{

// Checks if process name \_Process exists

System.Diagnostics.Process[] pname = System.Diagnostics.Process.GetProcessesByName(\_Process);

if (pname.Length==0) return false;

// Set normal size

ShowWindow(pname[0].MainWindowHandle, 1); // 1 normal, 2 minimized , 3 maximized

// Bring to front

SetForegroundWindow(pname[0].MainWindowHandle);

// Move to x,y location

SetWindowPos(pname[0].MainWindowHandle, new IntPtr(0), x, y, 0, 0, 0x0001);

return true;

}

private void Form1\_Load(object sender, EventArgs e)

{

BringApp("calc",200,200);

}

}

}

In Hotbar, the win app could be minimized

* Clicking the taskbar icon
* Clicking the minimize icon on the form

When it was minimized by clicking the icon on the form then it was not possible to the the mainWindowHandle

So that value was previously store in a global variable IntPtr Handle1 and then the code is

If (pname[0].Title == !”) GlobalData.Handle = pname[0].MainWindowHandle;

IntPtr LocalHandle = GlobalData.Handle1;

ShowWindow(LocalHandle, 1); // and it set Normal

SetForegroundWindow(LocalHandle);

SetWindowPos(LocalHandle, new IntPtr(0), x, y, 0, 0, 0x0001);

Once I have the win app in front, I proceed to paste text, click button, this is done with

Install-Package InputSimulator

using WindowsInput.Native;

using WindowsInput;

InputSimulator enties = new InputSimulator();

entries.SimulateTextEntry(“Information Now”);

entries.SimulateTextEntry(VirtualKeyCode.TAB);

entries.SimulateTextEntry(GlobalData.ATTPassword);

entries.SimulateTextEntry(VirtualKeyCode.TAB);

entries.SimulateTextEntry(VirtualKeyCode.RETURN);

Also, in order to paste the text it is needed to click where the textbox is, and that is why it is important to set first where the win app is going to be with

SetWindowPos(LocalHandle, new IntPtr(0), x, y, 0, 0, 0x0001);

To set the position is by clicking the mouse

LeftMouseClick(491, 265);

For this you need to include the API libraries previously

[System.Runtime.InteropServices.DllImport("user32.dll")]

static extern bool SetCursorPos(int x, int y);

[System.Runtime.InteropServices.DllImport("user32.dll")]

public static extern void mouse\_event(int dwFlags, int dx, int dy, int cButtons, int dwExtraInfo);

public const int MOUSEEVENTF\_LEFTDOWN = 0x02;

public const int MOUSEEVENTF\_LEFTUP = 0x04;

public static void LeftMouseClick(int xpos, int ypos)

{

SetCursorPos(xpos, ypos);

mouse\_event(MOUSEEVENTF\_LEFTDOWN, xpos, ypos, 0, 0);

mouse\_event(MOUSEEVENTF\_LEFTUP, xpos, ypos, 0, 0);

}