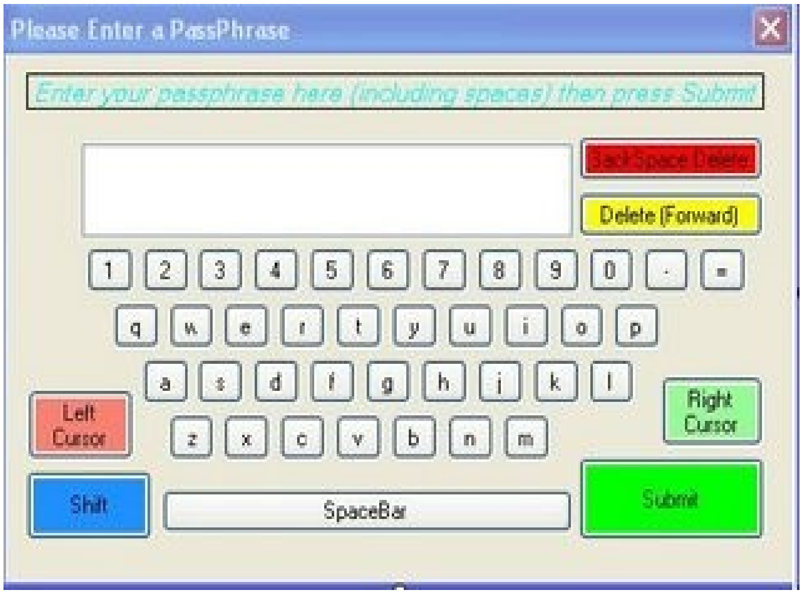
Avoiding KeyStrokes in Windows Applications using C#

In keeping with the bcrypt.exe example cited elsewhere on this site, we seek a method of avoiding using the keypad to enter pass words and/or phrases. A version of this appeared on the Treasury Direct site, but I had the idea earlier, and herein implement it here.

The code is much to large to enter here, so I've removed parts to highlight what must be done. We've created a class (Typer) and it puts up a form which has the standard keyboard implemented as buttons, along with one label and one textwindow. Each button is dragged over from the palette, placed and sized appropriately, re-named so that the internal code references are self-identifying. Notice that this is the simplest programming possible, i.e., no indexing of arrays has been used to make the code compact and/or swift. This is simple mindedness carried to an extreme! Here is the layout of the final screen which I've employed:



The beginning code, with parts excluded, especially those generated by the Forms generator, begins here:

```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Text.RegularExpressions;
namespace Typer {
///
/// Summary description keystroke eliminator for password input.
///
public class TyperDlg : System.Windows.Forms.Form {
private Button bq;
private Button bw;
private Button be;
private Button br;
private Button bt;
private Button by;
private Button bu;
private Button bi;
private Button bo;
private Button bp;
private Button b0;
private Button b9;
private Button b8;
private Button b7;
private Button b6;
private Button b5;
private Button b4;
private Button b3;
private Button b2;
private Button b1;
private Button bminus;
private Button bequals;
private Button bl;
private Button bk;
private Button bj;
private Button bh;
private Button bg;
private Button bf;
private Button bd;
private Button bs;
private Button ba;
private Button bm;
private Button bn;
private Button bb;
private Button bv;
private Button bc;
private Button bx;
private Button bz;
private Button Shift;
private Button Submit;
private TextBox textBox1;
private Label label1;
private Button BackSpace;
private Button Delete;
private Button buttonLeft;
private Button buttonRight;
private Button SpaceBar;
///
/// Required designer variable.
///
private System.ComponentModel.Container components = null;
public TyperDlg() {
    //
    // Required for Windows Form Designer support
    //
    InitializeComponent();
```

We're going to use the vertical separator as a running cursor, so we make our output equal to it at the outset.

```
    resultant = "|";
   textBox1.Text = resultant;
```

```
    //
    // TODO: Add any constructor code after InitializeComponent call
    //
    }
///
/// Clean up any resources being used.
///
protected override void Dispose(bool disposing) {
    if (disposing) {
        if (components != null) {
            components.Dispose();
            }
        }
    base.Dispose(disposing);
    }
#region Windows Form Designer generated code
///
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
///
private void InitializeComponent() {
    this.bq = new System.Windows.Forms.Button();
    this.bw = new System.Windows.Forms.Button();
    this.be = new System.Windows.Forms.Button();
  this.br = new System.Windows.Forms.Button();
```

We've omitted lots of similar code here for each button.

```
    this.SuspendLayout();
```

Each key is created by dragging a button from the palette, re-naming it, and putting in the lower case text which will be shown (unshifted). However, note below that number keys are specially handled, since their upcase/lowercase functionality is not present the way it is with simple letters.

Later, each key will also be doubleclicked, leading to an EventHandler whose definition shows up here, but which is actually inserted later in the creation process. Be that as it may, to get to that code, double click on the button whose code you want to deal with (see below).

```
    //
    // bq
    //
    this.bq.Location = new System.Drawing.Point(62, 130);
    this.bq.Name = "bq";
    this.bq.Size = new System.Drawing.Size(26, 23);
    this.bq.TabIndex = 0;
    this.bq.Text = "q";
    this.bq.UseVisualStyleBackColor = true;
    this.bq.Click += new System.EventHandler(this.bq_Click);
    //
    // bw
    //
    this.bw.Location = new System.Drawing.Point(94, 130);
    this.bw.Name = "bw";
    this.bw.Size = new System.Drawing.Size(26, 23);
    this.bw.TabIndex = 1;
    this.bw.Text = "w";
    this.bw.UseVisualStyleBackColor = true;
    this.bw.Click += new System.EventHandler(this.bw_Click);
    //
    // be
    //
    this.be.Location = new System.Drawing.Point(126, 130);
    this.be.Name = "be";
    this.be.Size = new System.Drawing.Size(26, 23);
    this.be.TabIndex = 2;
    this.be.Text = "e";
    this.be.UseVisualStyleBackColor = true;
    this.be.Click += new System.EventHandler(this.be_Click);
    //
    // br
    //
    this.br.Location = new System.Drawing.Point(158, 130);
    this.br.Name = "br";
    this.br.Size = new System.Drawing.Size(26, 23);
    this.br.TabIndex = 3;
    this.br.Text = "r";
    this.br.UseVisualStyleBackColor = true;
    this.br.Click += new System.EventHandler(this.br_Click);
    //
    // bt
    //
    this.bt.Location = new System.Drawing.Point(190, 130);
    this.bt.Name = "bt";
    this.bt.Size = new System.Drawing.Size(26, 23);
    this.bt.TabIndex = 4;
    this.bt.Text = "t";
    this.bt.UseVisualStyleBackColor = true;
    this.bt.Click += new System.EventHandler(this.bt_Click);
    //
    // by
    //
    this.by.Location = new System.Drawing.Point(222, 130);
    this.by.Name = "by";
    this.by.Size = new System.Drawing.Size(26, 23);
    this.by.TabIndex = 5;
    this.by.Text = "y";
    this.by.UseVisualStyleBackColor = true;
    this.by.Click += new System.EventHandler(this.by_Click);
    //
    // bu
    //
    this.bu.Location = new System.Drawing.Point(254, 130);
    this.bu.Name = "bu";
    this.bu.Size = new System.Drawing.Size(26, 23);
    this.bu.TabIndex = 6;
    this.bu.Text = "u";
    this.bu.UseVisualStyleBackColor = true;
    this.bu.Click += new System.EventHandler(this.bu_Click);
    //
    // bi
```

Again, we eliminate lots of code here, one set of 7 lines per key.

```
    //
    // bequals
    //
    this.bequals.Location = new System.Drawing.Point(399, 101);
    this.bequals.Name = "bequals";
    this.bequals.Size = new System.Drawing.Size(26, 23);
    this.bequals.TabIndex = 21;
    this.bequals.Text = "=";
    this.bequals.UseVisualStyleBackColor = true;
this.bequals.Click += new System.EventHandler(this.bequals_Click);
    //
    // bl
    //
```

And we again eliminate code which teaches us nothing:

Here we treat some special keys:

```
    //
    // Shift
    //
    this.Shift.BackColor = System.Drawing.Color.DodgerBlue;
    this.Shift.Location = new System.Drawing.Point(12, 217);
    this.Shift.Name = "Shift";
    this.Shift.Size = new System.Drawing.Size(72, 36);
    this.Shift.TabIndex = 39;
    this.Shift.Text = "Shift";
    this.Shift.UseVisualStyleBackColor = false;
    this.Shift.Click += new System.EventHandler(this.Shift_Click);
    //
    // Submit
    //
    this.Submit.BackColor = System.Drawing.Color.Lime;
    this.Submit.Location = new System.Drawing.Point(329, 210);
    this.Submit.Name = "Submit";
    this.Submit.Size = new System.Drawing.Size(107, 43);
    this.Submit.TabIndex = 40;
    this.Submit.Text = "Submit";
    this.Submit.UseVisualStyleBackColor = false;
    this.Submit.Click += new System.EventHandler(this.Submit_Click);
    //
    //textBox1
    //
    this.textBox1.Location = new System.Drawing.Point(44, 47);
    this.textBox1.Multiline = true;
    this.textBox1.Name = "textBox1";
    this.textBox1.Size = new System.Drawing.Size(281, 48);
    this.textBox1.TabIndex = 41;
    //
    // label1
    //
    this.label1.AutoSize = true;
    this.label1.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle;
    this.label1.Font = new System.Drawing.Font("Microsoft Sans Serif", 10F,
System.Drawing.FontStyle.Italic, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
    this.label1.ForeColor = System.Drawing.Color.Aqua;
    this.label1.Location = new System.Drawing.Point(12, 10);
    this.label1.Name = "label1";
    this.label1.Size = new System.Drawing.Size(424, 19);
    this.label1.TabIndex = 42;
    this.label1.Text = "Enter your passphrase here (including spaces) then press
```

```
Submit";
    //
    // BackSpace
    //
    this.BackSpace.BackColor = System.Drawing.Color.Red;
    this.BackSpace.Location = new System.Drawing.Point(329, 43);
    this.BackSpace.Name = "BackSpace";
    this.BackSpace.Size = new System.Drawing.Size(106, 23);
    this.BackSpace.TabIndex = 43;
    this.BackSpace.Text = "BackSpace Delete";
    this.BackSpace.UseVisualStyleBackColor = false;
    this.BackSpace.Click += new System.EventHandler(this.BackSpace_Click);
    //
    // Delete
    //
    this.Delete.BackColor = System.Drawing.Color.Yellow;
    this.Delete.Location = new System.Drawing.Point(329, 72);
    this.Delete.Name = "Delete";
    this.Delete.Size = new System.Drawing.Size(106, 23);
    this.Delete.TabIndex = 44;
    this.Delete.Text = "Delete (Forward)";
    this.Delete.UseVisualStyleBackColor = false;
    this.Delete.Click += new System.EventHandler(this.Delete_Click);
    //
    // buttonLeft
    //
    this.buttonLeft.BackColor = System.Drawing.Color.Salmon;
    this.buttonLeft.Location = new System.Drawing.Point(13, 175);
    this.buttonLeft.Name = "buttonLeft";
    this.buttonLeft.Size = new System.Drawing.Size(60, 36);
    this.buttonLeft.TabIndex = 45;
    this.buttonLeft.Text = "Left Cursor";
    this.buttonLeft.UseVisualStyleBackColor = false;
    this.buttonLeft.Click += new System.EventHandler(this.buttonLeft_Click);
    //
    // buttonRight
    //
    this.buttonRight.BackColor = System.Drawing.Color.PaleGreen;
    this.buttonRight.Location = new System.Drawing.Point(377, 168);
    this.buttonRight.Name = "buttonRight";
    this.buttonRight.Size = new System.Drawing.Size(58, 36);
    this.buttonRight.TabIndex = 46;
    this.buttonRight.Text = "Right Cursor";
    this.buttonRight.UseVisualStyleBackColor = false;
    this.buttonRight.Click += new System.EventHandler(this.buttonRight_Click);
    //
    // SpaceBar
    //
    this.SpaceBar.Location = new System.Drawing.Point(90, 227);
    this.SpaceBar.Name = "SpaceBar";
    this.SpaceBar.Size = new System.Drawing.Size(235, 22);
    this.SpaceBar.TabIndex = 47;
    this.SpaceBar.Text = "SpaceBar";
    this.SpaceBar.UseVisualStyleBackColor = true;
    this.SpaceBar.Click += new System.EventHandler(this.SpaceBar_Click);
    //
    // TyperDlg
    //
    this.AutoScaleBaseSize = new System.Drawing.Size(5, 13);
    this.ClientSize = new System.Drawing.Size(452, 275);
    this.Controls.Add(this.SpaceBar);
    this.Controls.Add(this.buttonRight);
    this.Controls.Add(this.buttonLeft);
    this.Controls.Add(this.Delete);
    this.Controls.Add(this.BackSpace);
    this.Controls.Add(this.label1);
    this.Controls.Add(this.textBox1);
    this.Controls.Add(this.Submit);
    this.Controls.Add(this.Shift);
    this.Controls.Add(this.bm);
    this.Controls.Add(this.bn);
    this.Controls.Add(this.bb);
    this.Controls.Add(this.bv);
    this.Controls.Add(this.bc);
    this.Controls.Add(this.bx);
```

And again, we eliminate repetitive coding which is obvious.

```
    this.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
    this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedDialog;
    this.MaximizeBox = false;
    this.MinimizeBox = false;
    this.Name = "TyperDlg";
    this.SizeGripStyle = System.Windows.Forms.SizeGripStyle.Hide;
    this.StartPosition = System.Windows.Forms.FormStartPosition.CenterParent;
    this.Text = "Please Enter a PassPhrase";
    this.Load += new System.EventHandler(this.typerDlg_Load);
    this.ResumeLayout(false);
    this.PerformLayout();
    }
#endregion
private void typerDlg_Load(object sender, System.EventArgs e) {
    }
```

Here we indicate the EventHandler for making the display react to the Shift button. Notice that the top line of keys has to be handled separately, since the display tokens are different between upper case and lower case. Shift_engaged is a flag used to remember whether or not the shift key was hit (CAPS-LOCK).

```
private void Shift_Click(object sender, EventArgs e) {
    if (!Shift_engaged) {
        safety_color = Shift.BackColor;
        Shift.BackColor = Color.Red;
        bq.Text = bq.Text.ToUpper();
        bw.Text = bw.Text.ToUpper();
        be.Text = be.Text.ToUpper();
        br.Text = br.Text.ToUpper();
        bt.Text = bt.Text.ToUpper();
        by.Text = by.Text.ToUpper();
        bu.Text = bu.Text.ToUpper();
        bi.Text = bi.Text.ToUpper();
        bo.Text = bo.Text.ToUpper();
        bp.Text = bp.Text.ToUpper();
        //row 2
        ba.Text = ba.Text.ToUpper();
        bs.Text = bs.Text.ToUpper();
        bd.Text = bd.Text.ToUpper();
        bf.Text = bf.Text.ToUpper();
        bg.Text = bg.Text.ToUpper();
        bh.Text = bh.Text.ToUpper();
        bj.Text = bj.Text.ToUpper();
        bk.Text = bk.Text.ToUpper();
        bl.Text = bl.Text.ToUpper();
        //row 2
        bz.Text = bz.Text.ToUpper();
        bx.Text = bx.Text.ToUpper();
        bc.Text = bc.Text.ToUpper();
        bv.Text = bv.Text.ToUpper();
        bb.Text = bb.Text.ToUpper();
        bn.Text = bn.Text.ToUpper();
        bm.Text = bm.Text.ToUpper();
        b1.Text = "!";
        b2.Text = "@";
        b3.Text = "#";
        b4.Text = "$";
        b5.Text = "%";
        b6.Text = "^";
        b7.Text = "&";
        b8.Text = "*";
        b9.Text = "(";
        bminus.Text = ")";
        bequals.Text = "_";
        Shift_engaged = !Shift_engaged;
        return;
        }
    else {
        Shift.BackColor = safety_color;
        bq.Text = bq.Text.ToLower();
        bw.Text = bw.Text.ToLower();
        be.Text = be.Text.ToLower();
        br.Text = br.Text.ToLower();
        bt.Text = bt.Text.ToLower();
        by.Text = by.Text.ToLower();
        bu.Text = bu.Text.ToLower();
        bi.Text = bi.Text.ToLower();
        bo.Text = bo.Text.ToLower();
        bp.Text = bp.Text.ToLower();
        //row 2
        ba.Text = ba.Text.ToLower();
        bs.Text = bs.Text.ToLower();
        bd.Text = bd.Text.ToLower();
        bf.Text = bf.Text.ToLower();
        bg.Text = bg.Text.ToLower();
        bh.Text = bh.Text.ToLower();
        bj.Text = bj.Text.ToLower();
        bk.Text = bk.Text.ToLower();
        bl.Text = bl.Text.ToLower();
        //row 3
        bz.Text = bz.Text.ToLower();
        bx.Text = bx.Text.ToLower();
        bc.Text = bc.Text.ToLower();
        bv.Text = bv.Text.ToLower();
        bb.Text = bb.Text.ToLower();
        bn.Text = bn.Text.ToLower();
        bm.Text = bm.Text.ToLower();
        //row 0
        b1.Text = "1";
        b2.Text = "2";
        b3.Text = "3";
        b4.Text = "4";
        b5.Text = "5";
        b6.Text = "6";
        b7.Text = "7";
        b8.Text = "8";
        b9.Text = "9";
        bminus.Text = "-";
        bequals.Text = "=";
        Shift_engaged = !Shift_engaged;
        return;
        }
    }
public Boolean Shift_engaged = false;
public Color safety_color;
public String resultant = "|";
```

When the user clicks Submit, we want the resultant to be prepared and ready for whatever use the program wants. For testing purposes, we have a MessageBox which we use to tell us what pass phrase is being assembled, but we comment it out when in "production".

```
private void Submit_Click(object sender, EventArgs e) {
//            resultant = "|";//set up for next time?
    if (resultant.Length == 0) {
        MessageBox.Show(this,
            "Please enter a valid password here.", "Input Error",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    DialogResult = DialogResult.OK;
    resultant = resultant.Replace("|","");
    resultant = resultant.Trim();
    /*
    MessageBox.Show("Your password was:\n\r" + resultant, "Result from Submit
Button ",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
     * */
    Close();
    }
```

Notice that in the above code, we've removed our cursor, and trimmed off outlying blanks. This is our choice!

```
public System.String getPass() {
    return resultant;
    }
```

What follows is the EventHandler for some of the keys, beginning with the "q" key. Each key is handled identically, but with a different letter being used. We are not trying to be sophisticated here, just transparent in the coding!

```
private void bq_Click(object sender, EventArgs e) {
    if (Shift_engaged) {
        resultant = resultant.Replace("|", "Q|");
        }
    else {
        resultant = resultant.Replace("|", "q|");
        }
    textBox1.Text = resultant;
    return;
    }
private void bw_Click(object sender, EventArgs e) {
    if (Shift_engaged) {
        resultant = resultant.Replace("|", "W|");
        }
    else {
        resultant = resultant.Replace("|", "w|");
        }
    textBox1.Text = resultant;
    return;
    }
private void be_Click(object sender, EventArgs e) {
    if (Shift_engaged) {
        resultant = resultant.Replace("|", "E|");
        }
    else {
        resultant = resultant.Replace("|", "e|");
        }
    textBox1.Text = resultant;
    return;
    }
private void br_Click(object sender, EventArgs e) {
    if (Shift_engaged) {
        resultant = resultant.Replace("|", "R|");
        }
    else {
        resultant = resultant.Replace("|", "r|");
        }
    textBox1.Text = resultant;
    return;
    }
private void bt_Click(object sender, EventArgs e) {
    if (Shift_engaged) {
        resultant = resultant.Replace("|", "T|");
        }
    else {
        resultant = resultant.Replace("|", "t|");
        }
    textBox1.Text = resultant;
    return;
    }
```

We need a handler for each key, but the coding is so repetitive and simpleminded that there's no point in repeating it all here.

```
       private void buttonRight_Click(object sender, EventArgs e) {
           int index = resultant.IndexOf(@"|");
           if (index < resultant.Length-1) {
                 String mychar = resultant.Substring(index + 1, 1);
                 String subsin =  "|" + mychar;
                 String subsout = mychar + "|" ;
                 resultant = resultant.Replace(subsin, subsout);
textBox1.Text = resultant;
                 }
             }
```

Text eliminated here for other buttons, but you get the idea.

```
}
```

We've attempted to indicate enough code so that you can follow the logic without seeing each and every key and its functioning.

There are two aspects of key stroke handling being employed here.
First, we have the appearance, which is controlled by the "Shift" button, which here functions as a CAPS-LOCK toggle. Each key's symbol changes on the keyboard itself from one case to the other when the "Shift" key is toggled.

The second aspect is the handling of each mouse click on a key, which is handled with separate handlers. Several of them are shown above.

Carl David
Department of Chemistry
University of Connecticut
Storrs, Connecticut 06269-3060
Carl.David@uconn.edu