

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Drawing;
using System.Data;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace RememberEachWord
{
    public partial class WordEditorField : UserControl
    {
        private Font curVisFont;
        private Font curUnvFont;
        private Color curVisColor;
        private Color curUnvColor;
        public Font curFont;
        private Color curColor;
        private bool state = false;
        private string itemText = "";
        private bool edit = false;
        private int index = 0;
        private Point origLocation;
        private Size origSize;
        private float zoom = 1.0f;

        //BackColor 168, 183, 205

        public event WordEditorFieldEventHandler FieldChanged;
        public event WordEditorFieldEventHandler ItemDoubleClick;

        public bool State
        {
            get
            {
                return state;
            }
            set
            {
                state = value;
                RefreshState();
            }
        }

        public string ItemText
        {
            get
            {
                return itemText;
            }
            set
            {
                itemText = value;
                textBox1.Text = itemText;
                Refresh();
            }
        }

        public int ItemIndex
        {
            get { return index; }
            set { index = value; }
        }

        public bool Edit
        {
            get { return edit; }
            set
            {
                edit = value;
                RefreshState();
            }
        }
    }
}
```

```

    }

    public Point OrigLocation
    {
        set { origLocation = value; }
        get { return origLocation; }
    }

    public Size OrigSize
    {
        set { origSize = value; }
        get { return origSize; }
    }

    public Font CurrentVisibleFont
    {
        set
        {
            curVisFont = value;
            textBox1.Font = curVisFont;
            textBox1.Location = new Point(3, (this.Height - textBox1.Height) / 2);
            textBox1.Width = this.Width - 6;
            RefreshState();
        }
    }

    public Font CurrentUnvisibleFont
    {
        set
        {
            curUnvFont = value;
            RefreshState();
        }
    }

    public Color CurrentVisibleColor
    {
        set
        {
            curVisColor = value;
            RefreshState();
        }
    }

    public Color CurrentUnvisibleColor
    {
        set
        {
            curUnvColor = value;
            RefreshState();
        }
    }

    //public Font CurrentFont //Delete!!!!
    //{
    //    get
    //    {
    //        return curFont;
    //    }
    //}

    public float Zoom
    {
        set
        {
            zoom = value;
            RefreshState();
        }
    }

    public WordEditorField()
    {
        InitializeComponent();
    }

```

```

        FontFamily family = new FontFamily("Times New Roman");

        curVisFont = new Font(family, 16.0f, FontStyle.Bold);
        curUnvFont = new Font(curVisFont, FontStyle.Regular);
        curVisColor = Color.Black;
        curUnvColor = Color.Gray;
    }

    public WordEditorField(string text, bool st, int ind)
        : this()
    {
        ItemText = text;
        State = st;
    }

    private void WordEditorField_MouseDoubleClick(object sender, MouseEventArgs e)
    {
        if (edit)
        {
            if (e.Button == MouseButtons.Left)
            {
                if (ItemDoubleClick != null)
                {
                    WordEditorFieldEventArgs ea = new WordEditorFieldEventArgs();
                    ea.Index = this.index;
                    ea.State = false;
                    ea.Text = "";
                    ItemDoubleClick(this, ea);
                }
                textBox1.Text = ItemText;
                textBox1.Visible = true;
                textBox1.Focus();
            }
        }
    }

    private void RefreshState()
    {
        curColor = (state) ? curVisColor : curUnvColor;
        //curFont = (state) ? new Font(curVisFont.Name, curVisFont.Size * zoom, curVisFont.Style,
        curVisFont.Unit, curVisFont.GdiCharSet, curVisFont.GdiVerticalFont) :
        // new Font(curUnvFont.Name, curUnvFont.Size * zoom, curUnvFont.Style, curUnvFont.Unit,
        curUnvFont.GdiCharSet, curUnvFont.GdiVerticalFont);
        curFont = (state) ? new Font(curVisFont.FontFamily, curVisFont.Size * zoom, curVisFont.Style) :
            new Font(curUnvFont.FontFamily, curUnvFont.Size * zoom, curUnvFont.Style);

        if (edit)
        {
            this.Visible = true;
        }
        else
        {
            if (state)
            {
                this.Visible = true;
            }
            else
            {
                this.Visible = false;
            }
        }
        Refresh();
    }

    protected override void OnPaint(PaintEventArgs e)
    {
        base.OnPaint(e);

        StringFormat sf = new StringFormat();
        sf.Alignment = StringAlignment.Center;
        sf.LineAlignment = StringAlignment.Center;

        Graphics gr = e.Graphics;
    }

```

```

        gr.DrawString(itemText, curFont, new SolidBrush(curColor), this.ClientRectangle, sf);
    }

    private void textBox1_KeyDown(object sender, KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Enter)
        {
            OnKeyDown(e);

            StoreNewText();
        }
    }

    public void EndEdit()
    {
        if (textBox1.Visible)
        {
            StoreNewText();
        }
    }

    private void StoreNewText()
    {
        ItemText = textBox1.Text;
        if (ItemText == "") State = false;
        textBox1.Visible = false;

        WordEditorFieldEventArgs ea = new WordEditorFieldEventArgs();
        ea.Index = this.index;
        ea.State = this.state;
        ea.Text = this.itemText;
        if (FieldChanged != null) FieldChanged(this, ea);
    }

    private void WordEditorField_MouseClick(object sender, MouseEventArgs e)
    {
        if (edit)
        {
            if (this.itemText != "")
            {
                if (e.Button == System.Windows.Forms.MouseButtons.Right)
                {
                    State = !state;

                    WordEditorFieldEventArgs ea = new WordEditorFieldEventArgs();
                    ea.Index = this.index;
                    ea.State = this.state;
                    ea.Text = this.itemText;
                    if (FieldChanged != null) FieldChanged(this, ea);
                }
            }
        }
    }

    private void WordEditorField_Resize(object sender, EventArgs e)
    {
        textBox1.Location = new Point(3, (this.Height - textBox1.Height) / 2);
        textBox1.Width = this.Width - 6;
        RefreshState();
    }

    public void StoreOrigParameters(Point loc)
    {
        origLocation = loc;
        origSize = this.Size;
    }
}

public class WordEditorFieldEventArgs : EventArgs
{
    public int Index;
    public string Text;
    public bool State;
}

```

```
}  
  
    public delegate void WordEditorFieldEventHandler(object sender, WordEditorFieldEventArgs e);  
}
```