DIAMOND-BACHELOR

MIKU　UI 测试说明与界面分析

UI 测试说明与界面分析

测试员：小石

2011/1/6

目录

测试目的

界面是软件与用户交互的最直接的层，界面的好坏决定用户对软件的第一印象。而且设计良好的界面能够引导用户自己完成相应的操作，起到向导的作用。同时界面如同人的面孔，具有吸引用户的直接优势。设计合理的界面能给用户带来轻松愉悦的感受和成功的感觉，相反由于界面设计的失败，让用户有挫败感，再实用强大的功能都可能在用户的畏惧与放弃中付诸东流。

测试目标

通过用户界面 (UI) 测试来核实用户与软件的交互。UI 测试的目标在于确保用户界面向

用户提供了适当的访问和浏览测试对象功能的操作。除此之外，UI 测试还要确保 UI 功能内部的对象符合预期要求，并遵循公司或行业的标准。

1、通过浏览测试对象可正确反映业务的功能和需求，这种浏览包括窗口与窗口之间、字段与字段之间的浏览，以及各种访问方法 （Tab 健、鼠标移动和快捷键）的使用

2、窗口的对象和特征（例如：菜单、大小、位置、状态和中心）都符合标准。

3、窗口实现遵循标准的界面设计和用户使用习惯的要求。

界面呈现

界面 1


点控动作录制系统
操作说明
操作说明
按下Esc键 停止录制，对回放按钮，只有是存在动作记录文件时才会生效。按返回键将返回选择录制对象界面。
录制动作
回放动作
载入文件
保存文件
键鼠动作的录制，录制开始后窗口将隐藏，Esc键停止录制。

界面 2


点控动作录制系统
记录列表
WM_MOUSEMOVE(443,442
WM_MOUSEMOVE(443,443
WM_MOUSEMOVE(453,445
WM_MOUSEMOVE(459,448
WM_MOUSEMOVE(468,452
WM_MOUSEMOVE(481,456
WM_MOUSEMOVE(494,463
WM_MOUSEMOVE(507,467
录制动作
回放动作
载入文件
保存文件
将当前动作录制的记录文件另存为。

界面 3


点控动作录制系统
记录列表
WM_MOUSEMOVE|(443,442
WM_MOUSEMOVE|(443,443
WM_MOUSEMOVE|(453,445
WM_MOUSEMOVE|(459,448
WM_MOUSEMOVE|(468,452
WM_MOUSEMOVE|(481,456
WM_MOUSEMOVE|(494,463
WM_MOUSEMOVE|(507,467
录制动作
回放动作
载入文件
保存文件
将保存的动作文件载入为当前动作记录文件。

界面 4


点控动作录制系统
记录列表
WM_MOUSEMOVE|(443,442
WM_MOUSEMOVE|(443,443
WM_MOUSEMOVE|(453,445
WM_MOUSEMOVE|(459,448
WM_MOUSEMOVE|(468,452
WM_MOUSEMOVE|(481,456
WM_MOUSEMOVE|(494,463
WM_MOUSEMOVE|(507,467
录制动作
回放动作
载入文件
保存文件
键鼠动作当前录制文件下的动作回放。

界面 5


点控动作录制系统
记录列表
WM_MOUSEMOVE|(443,442
WM_MOUSEMOVE|(443,443
WM_MOUSEMOVE|(453,445
WM_MOUSEMOVE|(459,448
WM_MOUSEMOVE|(468,452
录制动作
回放动作
载入文件
保存文件
另存为
库 ▸ 文档 ▸
搜索 文档
文件名(N):
保存类型(T):
MikuActionScript(*.mact)
浏览文件夹(B)
保存(S)
取消

界面 6


点控动作录制系统
记录列表
确认框
是否要关闭系统？
Yes
No
录制动作
回放动作
WM_MOUSEMOVE|(481,456
WM_MOUSEMOVE|(494,463
WM_MOUSEMOVE|(507,467

界面 7

界面分析

测试成立

所有的功能键不是都放在同一个界面，根据功能层次分在不同层次界面；
相似功能键放在一起，间距合理，功能一目了然；
界面颜色搭配合适，不炫耀也不虚空，不同功能按钮，样式不一，以区别；
界面有说明文字，操作方便；
没最大化窗口就不放最大化窗口按钮；
当鼠标放到功能键上面时，有一定的详细信息显示；
支持 Tab 键浏览功能键和 Enter 键触发功能键；
界面切换流畅，不闪动；
对功能按钮能做到适时触发，不可操作时不响应；
窗口能正确的响应最小化和关闭；
关闭提示框上有确定和取消按钮，而非单独的只有确定按钮，却有文字说明；
功能按钮能准确的响应相应的供能，不多余；
界面布局不堆积，不分散，分散合理；
按钮文字简洁明了；
选择按钮能实现单一选择；

测试不成立

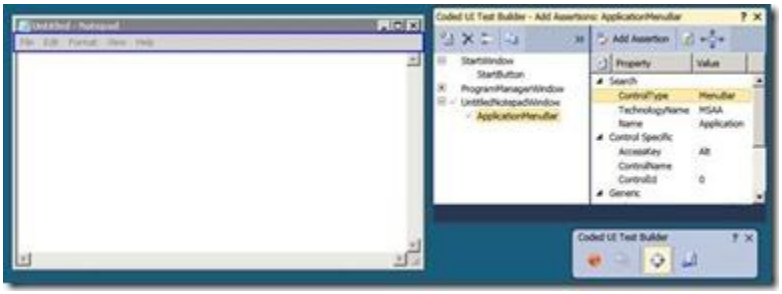
不支持快捷键操作；
选择按钮排布用意不明确；
返回按钮不明显；
说明文字窗口视区和界面搭配不自然；
界面没状态栏和标题栏；
动作记录的文字不清晰，字体不合理；
文件窗口和界面不搭配；
记录列表和操作说明的切换不顺畅，切换按钮不明确，用意不符；
界面不支持全球化和本地化；
界面文字颜色和背景搭配不合理；
界面没帮助按钮和其他关于软件信息；

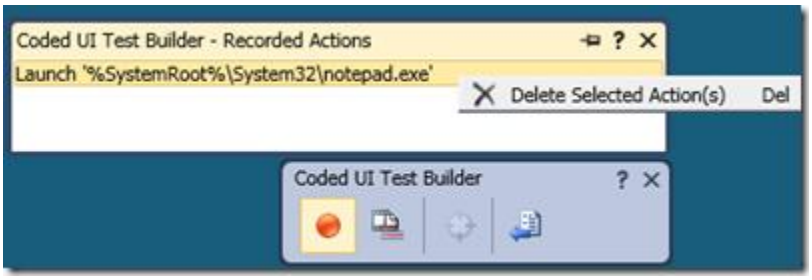
界面 Coded UI Test

```
using Microsoft.VisualStudio.TestTools.UITesting;
using Microsoft.VisualStudio.TestTools.UnitTesting;
using Microsoft.VisualStudio.TestTools.UI
using Keyboard = Microsoft.VisualStudio.T

namespace TestProject1
{
    /// <summary>
    /// Summary description for CodedUITe
    /// </summary>
    [CodedUITest]
    public class CodedUITest2
    {
        public CodedUITest2()
        {
        }

        [TestMethod]
        public void CodedUITestMethod1()
        {
            // To generate code for this test, select "Generate Code for Coded UI Test" from the shortcut menu
            // For more information on generated code, see http://go.microsoft.com/fwlink/?LinkId=179463
        }
```






```
      <Setting Name="ScreenResolutionWidth" Value="1280" WarningLevel="2" />
      <Setting Name="ScreenResolutionHeight" Value="1024" WarningLevel="2" />
      <Setting Name="SystemDPIX" Value="96" WarningLevel="2" />
      <Setting Name="SystemDPIY" Value="96" WarningLevel="2" />
      <Setting Name="Aero" Value="1" WarningLevel="1" />
      <Setting Name="UACEnabled" Value="1" WarningLevel="2" />
      <Setting Name="UACPromptEnabled" Value="5" WarningLevel="1" />
      <Setting Name="WindowsAccessibilityAPIVersion" Value="3.0" WarningLevel="1" />
    </Group>
    <Group Name="TechnologyManagers">
      <Setting Name="Web" WarningLevel="1" />
      <Setting Name="UIA" WarningLevel="1" />
    </Group>
  </Configuration>
  <InitializeActions />
  <ExecuteActions>
    <SetValueAction UIObjectName="UIMap.UIForm1Window.UITextBox1Window.UITextBox1Edit">
      <ParameterName />
      <Value Encoded="false">ddddds</Value>
      <Type>String</Type>
    </SetValueAction>
    <MouseAction UIObjectName="UIMap.UIForm1Window.UIButton1Window.UIButton1Button">
      <ParameterName />
      <ModifierKeys>None</ModifierKeys>
      <IsGlobalHotkey>false</IsGlobalHotkey>
      <Location X="39" Y="9" />
      <WheelDirection>0</WheelDirection>
      <ActionType>Click</ActionType>
      <MouseButton>Left</MouseButton>
    </MouseAction>
    <MouseAction UIObjectName="UIMap.UI提示Window.UI确定Window.UI确定Button">
      <ParameterName />
      <ModifierKeys>None</ModifierKeys>
      <IsGlobalHotkey>false</IsGlobalHotkey>
      <Location X="57" Y="17" />
      <WheelDirection>0</WheelDirection>
      <ActionType>Click</ActionType>
      <MouseButton>Left</MouseButton>
    </MouseAction>
    <TestStepMarkerAction MarkerInformation="RecordedMethod1">
      <ParameterName />
```

```
using Microsoft.VisualStudio.TestTools.UnitTesting;
using Keyboard = Microsoft.VisualStudio.TestTools.UITesting.Keyboard;
using Mouse = Microsoft.VisualStudio.TestTools.UITesting.Mouse;
using MouseButtons = System.Windows.Forms.MouseButtons;


public partial class UIMap
{
}
```

```
using MouseButtons = System.Windows.Forms.MouseButtons;


[GeneratedCode("Coded UITest Builder", "10.0.30319.1")]
public partial class UIMap
{

    /// <summary>
    /// RecordedMethod1 - Use 'RecordedMethod1Params' to pass parameters into this method.
    /// </summary>
    public void RecordedMethod1()
    {
        #region Variable Declarations
        WinEdit uITextBox1Edit = this.UIForm1Window.UITextBox1Window.UITextBox1Edit;
        WinButton uIButton1Button = this.UIForm1Window.UIButton1Window.UIButton1Button;
        WinButton uI确定Button = this.UI提示Window.UI确定Window.UI确定Button;
        #endregion

        // Type 'dddddds' in 'textBox1' text box
        uITextBox1Edit.Text = this.RecordedMethod1Params.UITextBox1EditText;

        // Click 'button1' button
        Mouse.Click(uIButton1Button, new Point(39, 9));

        // Click '确定' button
        Mouse.Click(uI确定Button, new Point(57, 17));
    }

    /// <summary>
    /// RecordedMethod2 - Use 'RecordedMethod2Params' to pass parameters into this method.
    /// </summary>
    public void RecordedMethod2()
    {
        #region Variable Declarations
        WinEdit uITextBox1Edit = this.UIForm1Window.UITextBox1Window.UITextBox1Edit;
        WinButton uIButton1Button = this.UIForm1Window.UIButton1Window.UIButton1Button;
        WinButton uI确定Button = this.UI确定Window.UI确定Button;
        #endregion
```

```
id RecordedMethod2()

on Variable Declarations
it uITextBox1Edit = this.UIForm1Window.UITextBox1Window.UITextBox1Edit;
:ton uIButton1Button = this.UIForm1Window.UIButton1Window.UIButton1Button;
:ton uI确定Button = this.UI确定Window.UI确定Button;
egion

pe 'd' in 'textBox1' text box
ard.SendKeys(uITextBox1Edit, this.RecordedMethod2Params.UITextBox1EditSendKeys, ModifierKeys.None);

ick 'button1' button                                                                    :Keys.None);
 Click(uIButton1Button, new Point(41, 15));

ick '确定' button
 Click(uI确定Button, new Point(59, 6));

pe 'dddd' in 'textBox1' text box
:Box1Edit.Text = this.RecordedMethod2Params.UITextBox1EditText;

ick 'button1' button
 Click(uIButton1Button, new Point(43, 10));

ick '确定' button
 Click(uI确定Button, new Point(56, 9));


:operties
:tual RecordedMethod1Params RecordedMethod1Params


    {
        get
```

总结

界面虽然能实现功能响应，但是布局一般，设计一般，欠缺快捷键操作；但总体来说别前面几个版本的界面要合理。希望后期能继续完善。

注：此界面为最终版本界面，其他界面请细看《界面演迁》文档