

DIAMOND-BACHELOR

# MIKU 用户界面测试用例

---

用户界面测试用例

测试员：小石

2011/1/6

目录

MIKU 用户界面测试用例 ..... 1

1. 被测试对象的介绍..... 3

2. 测试范围与目的 ..... 3

3. 测试环境与测试辅助工具的描述..... 3

4. 测试驱动程序的设计 ..... 3

5. 测试人员的分类 ..... 6

6. 用于测试用户界面的检查表..... 6

# 1. 被测试对象的介绍



界面

此界面为点控动作录制软件的界面，能实现分别对鼠标、键盘或鼠标和键盘等三种录制对象的动作录制，界面支持以上三种录制对象分别对应的录制、回放、录制文件的载入、录制文件的保存，以及录制记录的显示。

## 2. 测试范围与目的

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 键、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。

## 3. 测试环境与测试辅助工具的描述

测试环境：Window 7 环境且安装有 .Net Framework 4.0

测试工具：MTM (Microsoft Test Manager), Microsoft Test Professional 2010, Visual Nunit

## 4. 测试驱动程序的设计

```

        <Setting Name="ScreenResolutionWidth" Value="1280" WarningLevel="2" />
        <Setting Name="ScreenResolutionHeight" Value="1024" WarningLevel="2" />
        <Setting Name="SystemDPI" 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 />
    </TestStepMarkerAction>
</ExecuteActions>
}

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

public partial class UIMap
{
}

```

```

using MouseEventArgs = System.Windows.Forms.MouseEventArgs;

[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 'ddddd' 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

        /// <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

            // Type 'd' in 'textBox1' text box
            Keyboard.SendKeys(uITextBox1Edit, this.RecordedMethod2Params.UITextBox1EditSendKeys, ModifierKeys.None);

            // Click 'button1' button
            Mouse.Click(uIButton1Button, new Point(41, 15));

            // Click '确定' button
            Mouse.Click(uI确定Button, new Point(59, 6));

            // Type 'dddd' in 'textBox1' text box
            uITextBox1Edit.Text = this.RecordedMethod2Params.UITextBox1EditText;

            // Click 'button1' button
            Mouse.Click(uIButton1Button, new Point(43, 10));

            // Click '确定' button
            Mouse.Click(uI确定Button, new Point(56, 9));
        }

        #region Properties
        public virtual RecordedMethod1Params RecordedMethod1Params
        {
            get

```

## 5. 测试人员的分类

类别	特征
A 类	界面编码人员：熟悉界面代码和后台服务代码；
B 类	非界面编码人员：不熟悉界面代码但能看得懂后台服务代码，并有测试经验；
C 类	用户级人员：不熟悉代码，能对计算机有一定的操作能力；
.....	

## 6. 用于测试用户界面的检查表

指标	检查项	测试人员的类别及其评价
合适性和正确性	用户界面是否与软件的功能融洽	A: 能结合，并实现响应 B: 是 C: 是
	是否所有界面元素的文字和状态都正确无误	A: 是 B: 是 C:
容易理解	对于常用的功能，用户能否不必阅读手册就能使用？	A: 是 B: 是 C: 是
	是否所有界面元素（例如图标）都不会让人误解？	A: 是 B: 是 C: 是
	是否所有界面元素提供了充分而必要的提示？	A: 是 B: 是 C: 是
	界面结构能够清晰地反映工作流程？	A: 是 B: 是 C: 是
	用户是否容易知道自己在界面中的位置，不会迷失方向？	A: 否 B: 否 C: 否

	有联机帮助吗?	<b>A:</b> 否 <b>B:</b> 否 <b>C:</b> 否
指标	检查项	测试人员的类别及其评价
风格一致	同类的界面元素是否有相同的视感和相同的操作 式?	<b>A:</b> 是 <b>B:</b> 是 <b>C:</b> 否
	字体是否一致?	<b>A:</b> 是 <b>B:</b> 是 <b>C:</b> 否
	是否符合广大用户使用同类软件的习惯	<b>A:</b> 是 <b>B:</b> 是 <b>C:</b> 否
及时反馈信息	是否提供进度条、动画等反映正在进行的比较耗时间的过程?	无
	是否为重要的操作返回必要的结果信息?	<b>A:</b> 是 <b>B:</b> 是 <b>C:</b> 是
出错处理	是否对重要的输入数据进行校验?	无
	执行有风险的操作时,有“确认”、“放弃”等提示吗?	<b>A:</b> 是 <b>B:</b> 是 <b>C:</b> 是
	是否根据用户的权限自动屏蔽某些功能?	<b>A:</b> 是 <b>B:</b> 否 <b>C:</b> 是
	是否提供 UNDO 功能用以撤消不期望的操作?	无
适应各种水平的用户	所有界面元素都具备充分必要的操作键盘和鼠标操作吗?	<b>A:</b> 否 <b>B:</b> 否 <b>C:</b> 否
	初学者和专家都有合适的方式操作这个界面吗?	<b>A:</b> 是 <b>B:</b> 是 <b>C:</b> 否
	色盲或者色弱的用户能正常使用该界面吗?	<b>A:</b> 否 <b>B:</b> 否 <b>C:</b> 否
国际化	是否使用国际通行的国标和语言?	<b>A:</b> 否 <b>B:</b> 否 <b>C:</b> 否
	度量单位、日期格式、人的名字等是否符合国际惯例?	<b>A:</b> 是 <b>B:</b> 是

		C: 是
个性化	是否具有与众不同的、让用户记忆深刻的界面设计?	A: 是 B: 是 C: 是
	是否在具备必要的“一致性”的前提下突出“个性化”设计?	A: 是 B: 是 C: 是
合理布局和谐色彩	界面的布局符合软件的功能逻辑吗?	A: 是 B: 是 C: 是
	界面元素是否在水平或者垂直方向对齐?	A: 是 B: 是 C: 是
	界面元素的尺寸是否合理? 行、列的间距是否保持一致?	A: 是 B: 是 C: 是
	是否恰当地利用窗体和控件的空白, 以及分割线条?	A: 一般, 部分合理 B: C:
	窗口切换、移动、改变大小时, 界面正常吗?	A: 是 B: C:
	界面的色调是否让人感到和谐、满意?	A: 是 B: C:
	重要对象是否用醒目的色彩表示?	A: 是 B: C:
	色彩使用是否符合行业的习惯?	A: 一般 B: C:
	界面的布局符合软件的功能逻辑吗?	A: 部分合理 B: C:
	界面元素是否在水平或者垂直方向对齐?	A: 是 B: C: