

Làm việc với điều khiển Nút (giao diện nút)

Hãy chuyển sang nghiên cứu điều khiển Nút, thường được sử dụng trong phát triển giao diện người dùng. Điều khiển Nút thường hiển thị một chuỗi văn bản ngắn cho biết lệnh sẽ được thực thi khi bạn nhấn hoặc bấm vào điều khiển bằng ngón tay. Điều khiển cũng có thể hiển thị hình ảnh bitmap hoặc kết hợp văn bản và hình ảnh để chỉ định lệnh sẽ được thực thi.

Nút có một số thuộc tính ảnh hưởng đến hình thức của nó:

- `TextColor` - màu của văn bản bên trong nút;
- `BackgroundColor` - màu nền của nút;
- `BorderColor` - màu của vùng xung quanh nút;
- `FontFamily` - họ phông chữ dùng cho văn bản;
- `FontSize` - kích thước văn bản trong nút;
- `FontAttribut` - cho biết văn bản in nghiêng hay in đậm;
- `BorderWidth` - chiều rộng đường viền nút;
- `CornerRadius` - bán kính làm tròn các góc của nút;
- `CharacterSpacing` - khoảng cách giữa các ký tự văn bản bên trong nút;
- `TextTransform` - chuyển đổi văn bản bên trong nút.

Thuộc tính `IsEnabled` của điều khiển Nút phải được đặt thành `true` để Nút có thể nhấp được.

Điều khiển Nút kích hoạt sự kiện Đã nhấp, xảy ra khi người dùng nhấp vào điều khiển Nút bằng ngón tay hoặc con trỏ chuột. Ngoài sự kiện Đã nhấp, điều khiển Nút cũng xác định các sự kiện Đã nhấn và Đã phát hành. Sự kiện `Pressed` xảy ra khi một ngón tay nhấn vào Nút hoặc nút được nhấn bằng con trỏ chuột. Sự kiện Đã phát hành xảy ra khi một nút chuột hoặc ngón tay được thả ra. Thông thường, sự kiện `Clicked` cũng kích hoạt cùng lúc với sự kiện `Released`.

Để làm việc với các điều khiển Nút, Bộ chọn, Thanh trượt và Công tắc Tạo dự án `ButtonDemo` mới để phát triển ứng dụng phần mềm `Xamarin.Forms` đa nền tảng bằng mẫu Trống. Để làm việc với bộ chọn màu trong ứng dụng phần mềm, trước tiên hãy tạo

lớp `NamedColor`. Để thực hiện việc này, hãy di chuyển con trỏ chuột đến dự án `ButtonDemo`, nhấp chuột phải, chọn "Thêm" rồi chọn "Lớp". Hãy mở tệp `NamedColor.cs` đã tạo và nhập mã chương trình được đưa ra trong bản trình bày. Mã chương trình tạo các thuộc tính Tên, `FriendlyName`, Màu sắc, `RgbDisplay`, các thuộc tính này sẽ được mã chương trình truy cập từ các tệp

`MainPage.xaml` và `MainPage.xaml.cs`. Hàm `NamedColor()` được cung cấp,

tạo thành một mảng các màu hệ thống có sẵn cũng như tên tương ứng của chúng. Danh sách tên màu là cần thiết cho hoạt động của các điều khiển Picker, được sử dụng để chọn màu từ danh sách (mã chương trình được cung cấp trong tệp MainPage.xaml)

```
sử dụng Hệ
thống; sử dụng System.Collections.Generic;
sử dụng System.Reflection;
sử dụng System.Text;
sử dụng Xamarin.Forms;

không gian tên ButtonDemo {

    lớp công khai NamedColor {

        StringBuilder tĩnh stringBuilder = new StringBuilder();

        riêng tư NamedColor()

        { }

        Tên chuỗi công khai { tập riêng tư ; lấy; }

        chuỗi công khai FriendlyName { tập riêng tư ; lấy; }

        Màu công cộng Màu { bộ riêng; lấy; }

        chuỗi công khai RgbDisplay { tập riêng tư ; lấy; }

        tĩnh NamedColor() {

            Danh sách<NamedColor> all = Danh sách mới<NamedColor>();

            foreach (PropertyInfo propInfo in typeof(Color).GetRuntimeProperties()) {

                if (propInfo.GetMethod.IsPublic &&
                    propInfo.GetMethod.IsStatic &&
                    propInfo.PropertyType == typeof(Color))
                {
                    all.Add(CreateNamedColor(propInfo.Name, (Color)propInfo.GetValue(null)));
                }
            }

            foreach (FieldInfo fieldInfo in typeof(Color).GetRuntimeFields()) {

                if (fieldInfo.IsPublic &&
                    fieldInfo.IsStatic &&
                    fieldInfo.FieldType == typeof (Màu))
                {
```

```

        all.Add(CreateNamedColor(fieldInfo.Name, (Color)fieldInfo.GetValue(null)));
    }
}
all.TrimExcess();

Tất cả = tất cả;
}

public static IList<NamedColor> Tất cả { tập riêng tư ; lấy; }

tính NamedColor CreateNamedColor(tên chuỗi, Màu màu)
{
    // Chuyển đổi tên thành FriendlyName.
    stringBuilder.Clear();
    chỉ số int = 0;

    foreach (char ch trong tên)
    {
        if (index != 0 && Char.IsUpper(ch))
        {
            stringBuilder.Append(' ');
        }
        stringBuilder.Append(ch);
        chỉ mục++;
    }

    NamedColor có tênColor = New NamedColor
    {
        Tên = tên,
        FriendlyName = stringBuilder.ToString(),
        Màu sắc = màu sắc,
        RgbDisplay = String.Format("{0:X2}-{1:X2}-{2:X2}",
            (int)(255 * màu.R),
            (int)(255 * màu.G),
            (int)(255 * màu.B))
    };
    trả về tênColor;
}
}
}

```

Hãy mở tệp MainPage.xaml và nhập mã chương trình được cung cấp sau trong bản trình bày. Trang này chứa StackLayout chứa các điều khiển Nhấn và Nút, cũng như hai StackLayouts. Mã chương trình của điều khiển Nút chứa các thuộc tính TextColor,

BackgroundColor, BorderColor, được liên kết bằng cách sử dụng thuộc tính x:Reference cho các điều khiển Picker. Nút điều khiển cũng

được liên kết với các trình xử lý sự kiện Đã nhấn và Đã phát hành, mã chương trình được cung cấp trong tệp MainPage.xaml.cs.

Bố cục StackLayout bên trong đầu tiên chứa hai điều khiển Nhấn hiển thị văn bản với các giá trị thuộc tính FontAttribut sẽ được áp dụng cho văn bản trong điều khiển Nút, cũng như điều khiển Chuyển đổi để chuyển đổi giữa các giá trị thuộc tính FontAttribut. Điều khiển Switch cũng được liên kết với trình xử lý sự kiện OnSwitchToggled, được mã hóa trong tệp MainPage.xaml.cs.

StackLayout bên trong thứ hai chứa ba điều khiển Thanh trượt để chọn kích thước phông chữ của văn bản bên trong nút, để chọn độ dày của đường viền của nút và để chọn bán kính góc của nút. Mã chứa bố cục StackLayout liên kết với điều khiển Nút bằng BindingContext. Tiếp theo, điều khiển Thanh trượt sử dụng liên kết Ràng buộc với các thuộc tính FontSize, BorderWidth và CornerRadius của điều khiển Nút.

Mã này cũng bao gồm ba điều khiển Nhấn để hiển thị các giá trị được chọn bởi các điều khiển Thanh trượt. Để thực hiện việc này, bằng cách sử dụng thuộc tính x:Reference, các điều khiển Nhấn được liên kết với các điều khiển Thanh trượt tương ứng.

Các điều khiển bộ chọn được thiết kế để chọn màu cho các thuộc tính TextColor, BackgroundColor, BorderColor từ danh sách được tạo bằng lớp NamedColor, mã chương trình đã được cung cấp trong tệp NamedColor.cs.

```
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    xmlns:local="clr-namespace:ButtonDemo"
    x:Class="ButtonDemo.MainPage"
    BackgroundColor="Vôi">
    <StackLayout>
        <Nhấn x:Tên="nhấn"
            Text="Nhấn xoay"
            FontSize="20" FontAttribut="Đậm"
            VerticalOptions="CenterAndExpand"
            HorizontalOptions="Trung tâm"
            BackgroundColor="Đỏ" TextColor="Vàng"
            Phần đệm="10"
            Ký quỹ="50" />

        <Nút x:Tên="nút"
            Text="Đã nhấn và phát hành"
            VerticalOptions="CenterAndExpand"
            HorizontalOptions="Trung tâm"
            TextColor="{Binding Source={x:Reference textColorPicker},
                Đường dẫn=SelectedItem.Color}"
```

```
        BackgroundColor="{Binding Source={x:Reference BackgroundColorPicker},
                                Đư ờng dẫn=SelectedItem.Color}"

        BorderColor="{Binding Source={x:Reference borderColorPicker},
                                Đư ờng dẫn=SelectedItem.Color}"

        Đã nhấn="OnButtonPressed"
        Đã phát hành="OnButtonReleased" />
<StackLayout Orientation="Ngang">
    <Label Text="Button.FontAttribution=None"
            VerticalOptions="Trung tâm"
            HorizontalOptions="CenterAndExpand"
            FontSize="12" FontAttribut="Đậm"
            TextColor="Đen" />
    <Chuyển x:Name="BoldSwitch"
            Toggled="OnSwitchToggled"
            HorizontalOptions="CenterAndExpand" />
    <Label Text="Button.FontAttribution=Đậm"
            VerticalOptions="Trung tâm"
            HorizontalOptions="CenterAndExpand"
            FontSize="12" FontAttribut="Đậm"
            TextColor="Black" /> </
StackLayout>
<StackLayout BindingContext="{x:Reference Button}"
    Phần đệm="10">

    <Thanh trượt x:Name="fontSizeSlider"
        Tối đa="30"
        Tối thiểu="1"
        Value="{Binding FontSize}"
        Tối thiểuTrackColor="Vàng"
        MaximumTrackColor="Xanh" />

    <Label Text="{Binding Source={x:Reference fontSizeSlider},
            Đư ờng dẫn=Giá trị,
            StringFormat='FontSize = {0:F0}}'"
            HorizontalTextAlignment="Trung tâm"
            Kích thước phông chữ="16"
            TextColor="Đen" />

    <Thanh trượt x:Name="borderWidthSlider"
        Tối thiểu="-1"
        Tối đa="12"
        Value="{Binding BorderWidth}"
        Tối thiểuTrackColor="Vàng"
        MaximumTrackColor="Xanh" />

    <Label Text="{Binding Source={x:Reference borderWidthSlider},
            Đư ờng dẫn=Giá trị,
            StringFormat='BorderWidth = {0:F0}}'"
            HorizontalTextAlignment="Trung tâm"
```

```
Kích thước phông chữ="16"

TextColor="Đen"/>

<Thanh trượt x:Name="cornerRadiusSlider"
    Tối thiểu="-1"
    Tối đa="24"

    Value="{Binding CornerRadius}"
    Tối thiểuTrackColor="Vàng"
    MaximumTrackColor="Xanh"/>
<Label Text="{Nguồn liên kết={x:Góc tham chiếuRadiusSlider},
    Đưa ởng dẫn=Giá trị,
    StringFormat='CornerRadius = {0:F0}'}"
    HorizontalTextAlignment="Trung tâm"
    Kích thước phông chữ="16"
    TextColor="Đen"/>
<Lưu ý>
<Grid.RowDefinitions>
    <RowDefinition Height="Auto" />
    <RowDefinition Height="Auto" />
    <RowDefinition Height="Auto" /> </
Grid.RowDefinitions>

<Grid.ColumnDefinitions>
    <ColumnDefinition width="*" />
    <ColumnDefinition width="*" /> </
Grid.ColumnDefinitions>

<Grid.Resources>
    <Style TargetType="Label">
        <Setter Property="VerticalOptions" Value="Center" /> </
    Style> </
Grid.Resources>

<Label Text="Màu văn bản:"
    Grid.Row="0" Grid.Column="0"
    TextColor="Đen"
    FontSize="16"/>

<Picker x:Name="textColorPicker"
    ItemsSource="{Binding Source={x:Static local:NamedColor.All}}"
    ItemDisplayBinding="{Binding FriendlyName}"
    Đã chọnIndex="0"
    Grid.Row="0" Grid.Column="1" />

<Label Text="Màu nền:"
    Grid.Row="1" Grid.Column="0"
    TextColor="Đen"
    FontSize="16"/>
```

```

<Picker x:Name="backgroundColorPicker"
    ItemsSource="{Binding Source={x:Static local:NamedColor.All}}"
    ItemDisplayBinding="{Binding FriendlyName}"
    Đã chọnIndex="0"
    Grid.Row="1" Grid.Column="1" />

<Label Text="Màu đư ờng viền:"
    Grid.Row="2" Grid.Column="0"
    TextColor="Đen"
    FontSize="16"/>

<Picker x:Name="borderColorPicker"
    ItemsSource="{Binding Source={x:Static local:NamedColor.All}}"
    ItemDisplayBinding="{Binding FriendlyName}"
    Đã chọnIndex="0"
    Grid.Row="2" Grid.Column="1" />
</Grid>
</StackLayout>
</StackLayout>
</Trang nội dung>

```

Hãy mở tệp MainPage.xaml.cs và nhập mã chương trình đư ợc đư a ra trong bản trình bày. Mã chứa ba trình xử lý sự kiện đư ợc tham chiếu trong tệp MainPage.xaml.

Khi điều khiển Nút đư ợc bấm và sự kiện Đã nhấn kích hoạt, điều khiển Nhãn sẽ đư ợc xoay và có thuộc tính x:Name đư ợc áp dụng cho nó trong tệp MainPage.xaml và thuộc tính Tên của nó đư ợc đặt thành nhãn.

Để thực hiện xoay, bộ hẹn giờ đư ợc khởi động bằng cách sử dụng khả năng định giờ của thiết bị. Khoảng thời gian giữa các lần thực hiện của bộ đếm thời gian là 20 mili giây. Một phiên bản của lớp Đồng hồ bấm giờ đư ợc sử dụng để xác định góc xoay của điều khiển Nút. Khi nút đư ợc nhả ra, sự kiện Đã phát hành sẽ xảy ra và điều khiển Nhãn sẽ ngừng xoay.

Khi Công tắc đư ợc bật và sự kiện Toggled xảy ra, trình xử lý sự kiện OnSwitchToggled sẽ kích hoạt. Sự kiện Toggled có đối tượng ToggledEventArgs có thuộc tính đư ợc gọi là Giá trị. Khi sự kiện xảy ra, giá trị của thuộc tính Giá trị phản ánh giá trị mới của thuộc tính IsToggled.

sử dụng Hệ thống;

sử dụng System.Diagnostics;

sử dụng Xamarin.Forms;

không gian tên ButtonDemo

```

{
    lớp một phần công khai MainPage: ContentPage
    {

```

```
bool animationInProgress = false; bool
in đậm = sai;
Đồng hồ bấm giờ đồng hồ bấm giờ = Đồng hồ bấm giờ mới();

MainPage công khai()
{
    Khởi tạoComponent();
}

void OnButtonPressed(ngư ời gửi đối tư ợng, EventArgs args) {

    đồng hồ bấm
    giờ.Start(); hoạt hìnhInProgress
    = true; Device.StartTimer(TimeSpan.FromMilliseconds(20), () => {

        nhân.Rotation = 360 * (đồng hồ bấm giờ.Elapsed.TotalSeconds % 1);

        trả lại hoạt ảnhInProgress; }));

} void OnButtonReleased(ngư ời gửi đối tư ợng, EventArgs args) {

    animationInProgress = false; đồng
    hồ bấm giờ.Stop();
}

void OnSwitchToggled(ngư ời gửi đối tư ợng, ToggledEventArgs args) {

    nếu (đậm)
    {
        in đậm = sai;
        nút.FontAttribution = FontAttribut.None;

    }

    khác {
        in đậm =
        đúng; nút.FontAttribution = FontAttribut.Bold;
    }

}

}
```

Hãy khởi chạy một ứng dụng phần mềm để hoạt động với các điều khiển Nút, Bộ chọn, Công tắc và Thanh trượt. Ứng dụng phần mềm cho phép bạn thử nghiệm giao diện của điều khiển Nút bằng cách đặt giá trị cho các thuộc tính của nó bằng cách sử dụng các điều khiển Bộ chọn, Công tắc và Thanh trượt. Điều khiển bộ chọn đặt màu nền, màu văn bản hiển thị và màu đường viền của điều khiển Nút. Điều khiển thanh trượt đặt kích thước phông chữ

của văn bản đư ợc hiển thị trong điều khiển Nút, bán kính góc và độ dày của đư ờng viền xung quanh điều khiển Nút. Điều khiển Công tắc đặt kiểu hiển thị văn bản trong điều khiển Nút. Việc bấm và giữ điều khiển Nút sẽ xoay điều khiển Nhấn. Khi bạn ngừng bấm vào điều khiển Nút, điều khiển Nhấn sẽ ngừng xoay.

