

Bài: Đổi trạng thái inject

Xem bài học trên website để ủng hộ Kteam: [Đổi trạng thái inject](#)

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage [How Kteam](#) nhé!

Dẫn nhập

Megaman X4 là một tựa game rất thú vị gắn bó với mình một thời gian dài qua điện tử bấm, khi mà máy tính hay lập trình còn là gì đó vô cùng xa vời. Mình còn nhớ nguyên mấy tháng mùa hè đi đầu đi cổ vào game để luyện tay qua màn, mãi mà không phá đảo được. Giờ để thử thách bản thân, mình đã tạo ra một giải pháp giúp mình chinh phục "giấc mơ tuổi thơ này". Đó chính là **TOOL CHEAT GAME MEGAMAN X4**. :v :v

Tuy nhiên, trước khi làm tool bạn cần có một giao diện đẹp "kinh khủng khiếp" để có thể xin xò ngang với game. Chính vì vậy, để hỗ trợ cho khóa học **CHEAT GAME MEGAMAN X4**, đồng thời củng cố khả năng lập trình giao diện với [WPF](#). Kteam sẽ cùng bạn bắt đầu [Lập trình giao diện Megaman tool với WPF](#).

Ở bài trước chúng ta đã cùng nhau [TẠO THANH CONTROLBAR VÀ HÌNH LOGO CHO TOOL MEGAMAN](#) với khá khá kinh nghiệm hữu ích trong việc xử lý animations cho controlbar. Trong bài này, chúng ta sẽ tìm hiểu cách **Đổi trạng thái inject**

Nội dung

Để theo dõi bài này một cách tốt nhất, bạn nên có kiến thức về:

- [LẬP TRÌNH WPF](#)
- Đã xem qua bài [IMPORT ASSEST TỪ DESIGN](#)
- Đã xem qua bài [TẠO THANH CONTROLBAR VÀ HÌNH LOGO](#)

Trong bài này , chúng ta sẽ cùng tìm hiểu các vấn đề:

- Cách đổi trạng thái inject
- Một số lưu ý

Source code tham khảo

App.xaml

xml:

```
<Application x:Class="MegamanUI.App"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:local="clr-namespace:MegamanUI"
    StartupUri="MainWindow.xaml">
    <Application.Resources>
        <ResourceDictionary>
            <ResourceDictionary.MergedDictionaries>
                <ResourceDictionary Source="XamlResources/TriggerRes.xaml"></ResourceDictionary>
                <ResourceDictionary Source="XamlResources/BaseRes.xaml"></ResourceDictionary>
            </ResourceDictionary.MergedDictionaries>
        </ResourceDictionary>
    </Application.Resources>
</Application>
```

App.xaml.cs

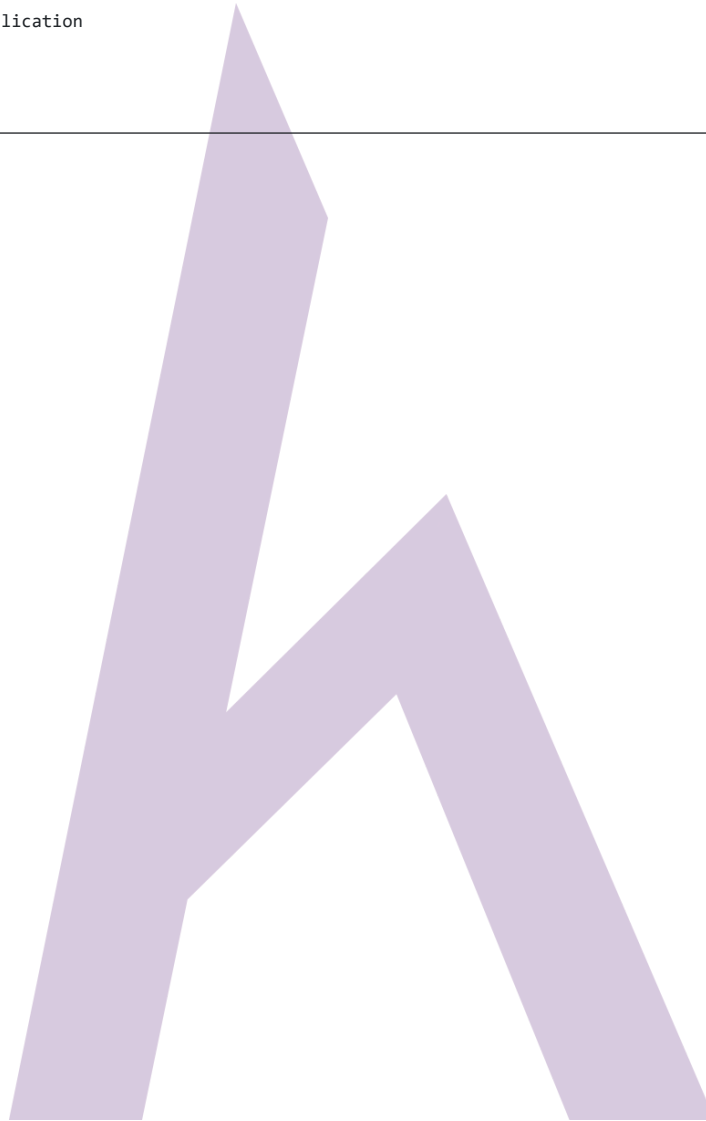
C#:

```
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Threading.Tasks;
using System.Windows;

namespace MegamanUI
{
    /// <summary>
    /// Interaction logic for App.xaml
    /// </summary>
    public partial class App : Application
    {
    }
}
```

MainWindow.xaml

xml:



```

<Window x:Class="MegamanUI.MainWindow"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  xmlns:local="clr-namespace:MegamanUI"
  mc:Ignorable="d"
  WindowStyle="None"
  ResizeMode="CanMinimize"
  WindowStartupLocation="CenterScreen"
  Height="350" Width="550">
  <Window.Background>
    <ImageBrush ImageSource="Images/bg-kteam.png"></ImageBrush>
  </Window.Background>
  <Grid>
    <DockPanel PreviewMouseDown="DockPanel_PreviewMouseDown"
      HorizontalAlignment="Stretch" LastChildFill="False" VerticalAlignment="Top" Background="Transparent"
      Height="25">
      <Image DockPanel.Dock="Right" Source="Images/x.png"
        Cursor="Hand"
        PreviewMouseDown="Image_PreviewMouseDown_1"
        Margin="0 0 15 0">
        <Image.Style>
          <Style TargetType="Image">
            <Setter Property="Width" Value="12"></Setter>
            <Style.Triggers>
              <Trigger Property="Image.IsMouseOver" Value="True">
                <Setter Property="Width" Value="15"></Setter>
              </Trigger>
            </Style.Triggers>
          </Style>
        </Image.Style>
      </Image>
      <Image DockPanel.Dock="Right" Source="Images/- .png"
        PreviewMouseDown="Image_PreviewMouseDown"
        Margin="0 0 10 0" Cursor="Hand">
        <Image.Style>
          <Style TargetType="Image">
            <Setter Property="Width" Value="12"></Setter>
            <Style.Triggers>
              <Trigger Property="Image.IsMouseOver" Value="True">
                <Setter Property="Width" Value="15"></Setter>
              </Trigger>
            </Style.Triggers>
          </Style>
        </Image.Style>
      </Image>
    </DockPanel>

    <Image Source="Images/megaman-logo-kteam.png"
      Height="30"
      Margin="28 3 0 0"
      Cursor="Hand"
      HorizontalAlignment="Left" VerticalAlignment="Top"></Image>

    <Image Source="Images/kteam-logo.png"
      Height="30"
      Margin="0 0 40 10"
      Cursor="Hand"
      HorizontalAlignment="Right" VerticalAlignment="Bottom"></Image>

    <Image Height="220"
      Margin="0 0 40 10"
      Cursor="Hand"
      MouseDown="Image_MouseDown"
      MouseEnter="Image_MouseEnter"
  
```

```

        HorizontalAlignment="Center" VerticalAlignment="Center">
    <Image.Style>
        <Style TargetType="Image">
            <Setter Property="Source" Value="Images/Megaman-Kteam.png"></Setter>
            <Style.Triggers>
                <MultiDataTrigger>
                    <MultiDataTrigger.Conditions>
                        <Condition Binding="{Binding IsInjected}" Value="True"></Condition>
                    </MultiDataTrigger.Conditions>
                    <Setter Property="Source" Value="Images/Megaman-Gaea-Armor-Kteam.png"></Setter>
                </MultiDataTrigger>
                <MultiDataTrigger>
                    <MultiDataTrigger.Conditions>
                        <Condition Binding="{Binding IsInjected}" Value="False"></Condition>
                        <Condition Binding="{Binding RelativeSource={RelativeSource Mode=Self}, Path=IsMouseOver}"
Value="True"></Condition>
                    </MultiDataTrigger.Conditions>
                    <Setter Property="Source" Value="Images/Megaman-Gaea-Armor-Kteam.png"></Setter>
                </MultiDataTrigger>
            </Style.Triggers>
        </Style>
    </Image.Style>
</Image>
</Grid>
</Window>

```

MainWindow.xaml.cs

:



```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Linq;
using System.Media;
using System.Runtime.CompilerServices;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace MegamanUI
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window, INotifyPropertyChanged
    {
        public event PropertyChangedEventHandler PropertyChanged;

        protected virtual void OnPropertyChanged([CallerMemberName] string propertyName = null)
        {
            PropertyChangedEventHandler handler = PropertyChanged;
            if (handler != null)
                handler(this, new PropertyChangedEventArgs(propertyName));
        }

        #region properties
        private bool _IsInjected;
        public bool IsInjected
        {
            get => _IsInjected;
            set
            {
                _IsInjected = value;
                OnPropertyChanged();
            }
        }

        SoundPlayer sp_active;
        SoundPlayer sp_hover;
        #endregion

        public MainWindow()
        {
            InitializeComponent();
            this.DataContext = this;

            FirstLoad();
        }

        #region methods
        void FirstLoad()
        {
            IsInjected = false;

            LoadSound();
        }
    }
}
```

```

void LoadSound()
{
    sp_active = new SoundPlayer(Properties.Resources.hover_2);
    sp_hover = new SoundPlayer(Properties.Resources.hover);
}

void PlaySoundActive()
{
    sp_active.Play();
}

void PlaySoundHover()
{
    sp_hover.Play();
}
#endregion

#region events
private void DockPanel_PreviewMouseLeftButtonDown(object sender, MouseButtonEventArgs e)
{
    base.OnMouseLeftButtonDown(e);
    this.DragMove();
}

private void Image_PreviewMouseDown(object sender, MouseButtonEventArgs e)
{
    WindowState = WindowState.Minimized;
}

private void Image_PreviewMouseDown_1(object sender, MouseButtonEventArgs e)
{
    this.Close();
}

private void Image_MouseDown(object sender, MouseButtonEventArgs e)
{
    IsInjected = !IsInjected;
    PlaySoundActive();
}

private void Image_MouseEnter(object sender, MouseEventArgs e)
{
    if(!IsInjected)
        PlaySoundHover();
}
#endregion
}
}

```

XamlResources/BaseRes.xaml

xml:

```

<ResourceDictionary xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:local="clr-namespace:MegamanUI.XamlResources">
    <FontFamily x:Key="mainFont">/Megaman UI;component/font/HP-Yard Regular.ttf#HP-Yard</FontFamily>
</ResourceDictionary>

```

XamlResources/TriggerRes.xaml

xml:

```
<ResourceDictionary xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
                    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
                    xmlns:local="clr-namespace:MegamanUI.XamlResources">

</ResourceDictionary>
```

Assest file

Nhằm giúp bạn có thể trải nghiệm hiệu ứng tool một cách tốt nhất, và chủ động trong việc sử dụng source. Kteam hỗ trợ full source code và các file liên quan tại phần [ASSEST](#) bên dưới nhé



Trong file này có gì???

- Đầu tiên bạn sẽ có 1 file chứa assest gồm âm thanh, font, hình ảnh và icon.
- Tiếp theo là File build hoàn chỉnh để bạn dễ dàng tham khảo các hiệu ứng.
- Project hoàn chỉnh để tham khảo
- Một file game exe dành cho bạn nào chưa từng chơi game megaman có thể thử qua để thêm hứng thú.

Kết luận

Qua bài học này, bạn đã nắm được cách đổi trạng thái inject và thêm âm thanh cho trạng thái.

Trong bài sau, mình sẽ hướng dẫn các bạn [ANIMATION CHO CREATED BY](#).

Cảm ơn các bạn đã theo dõi bài viết. Hãy để lại bình luận hoặc góp ý của mình để phát triển bài viết tốt hơn. Đừng quên "**Luyện tập – Thử thách – Không ngại khó**".