

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ



Федеральное государственное бюджетное образовательное
учреждение высшего профессионального образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Методические указания к лабораторной работе №8

«Трансформация в WPF-приложениях»

по дисциплине «Технологии разработки пользовательских интерфейсов»

Вичугов В.Н., доцент каф. АиКС

Томск 2012

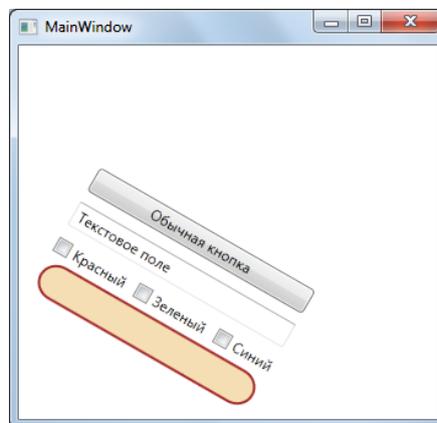
Любая часть пользовательского интерфейса WPF-приложения может быть подвергнута трансформации с помощью свойства `RenderTransform`. Существуют следующие виды трансформации:

- 1) **Вращение** (элемент `RotateTransform`) – поворачивает координатную систему на угол `Angle`, заданный в градусах, относительно центра трансформации. Центр трансформации может быть задан с помощью атрибутов `CenterX` и `CenterY` (0,0 – это левый верхний угол элемента управления). Положительное значение угла соответствует вращению по часовой стрелке.

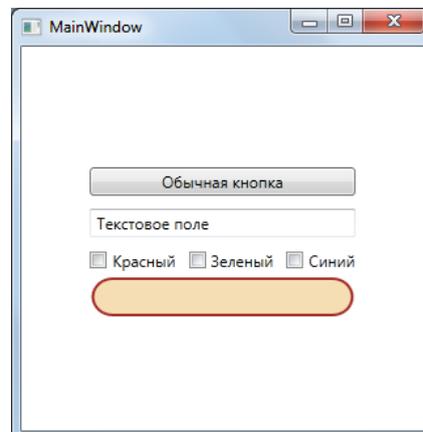
Пример 1 Код XAML

```
<StackPanel HorizontalAlignment="Center" VerticalAlignment="Center">
  <Button>Обычная кнопка</Button>
  <TextBox>Текстовое поле</TextBox>
  <StackPanel Orientation="Horizontal">
    <CheckBox>Красный</CheckBox>
    <CheckBox>Зеленый</CheckBox>
    <CheckBox>Синий</CheckBox>
  </StackPanel>
  <Rectangle Width="200" Height="30" Fill="Wheat" Stroke="Brown" StrokeThickness="2"
  RadiusX="15" RadiusY="15" />
  <StackPanel.RenderTransform>
    <RotateTransform Angle="30" CenterX="0" CenterY="0" />
  </StackPanel.RenderTransform>
</StackPanel>
```

Результат



Без трансформации:



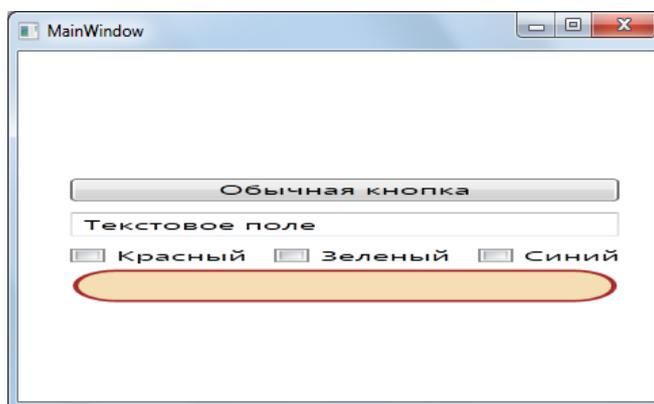
- 2) **Масштабирование** (элемент `ScaleTransform`) – масштабирует координатную систему в большую или в меньшую сторону относительно центра трансформации. Центр трансформации может быть задан с помощью атрибутов `CenterX` и `CenterY` (0,0 – это левый верхний угол элемента управления). Коэффициент масштабирования задается в относительных единицах в атрибутах `ScaleX` и `ScaleY` для разных координатных осей. `ScaleX="2"` означает, что по оси абсцисс элемент управления будет

растянут в два раза; ScaleX="0.5" означает, что по оси ординат элемент управления будет сжат в два раза.

Пример 2 Код XAML

```
<StackPanel.RenderTransform>  
  <ScaleTransform ScaleX="2" ScaleY="0.8" CenterX="100" CenterY="100" />  
</StackPanel.RenderTransform>
```

Результат

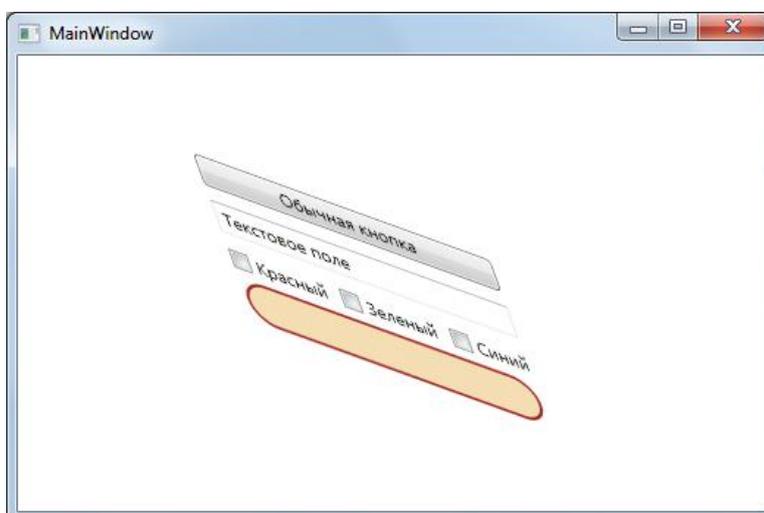


- 3) **Сдвиг** (элемент SkewTranform) – деформирует координатную систему, наклоняя ее относительно центра трансформации на AngleX градусов по оси абсцисс и на AngleY градусов по оси ординат. Центр трансформации может быть задан с помощью атрибутов CenterX и CenterY (0,0 – это левый верхний угол элемента управления).

Пример 3 Код XAML

```
<StackPanel.RenderTransform>  
  <SkewTransform AngleX="20" AngleY="20" CenterX="100" CenterY="100" />  
</StackPanel.RenderTransform>
```

Результат



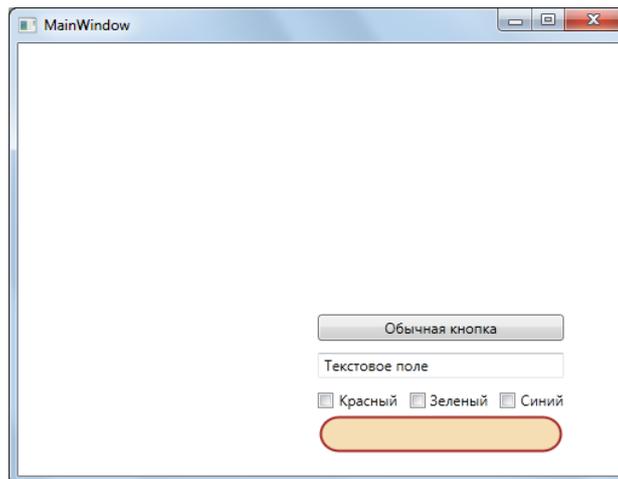
- 4) **Смещение** (элемент TranslateTranform) – смещает координатную систему на величину X по оси абсцисс и на величину Y по оси ординат.

Пример 4

Код XAML

```
<StackPanel.RenderTransform>  
  <TranslateTransform X="100" Y="100" />  
</StackPanel.RenderTransform>
```

Результат

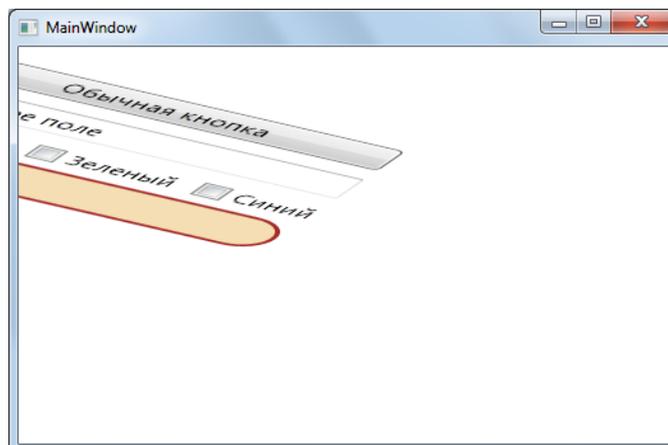


- 5) **Матричное преобразование** (элемент `MatrixTransform`) – модифицирует координатную систему, используя матричное умножение с указанной матрицей.
- 6) **Комбинированная трансформация** (элемент `TransformGroup`) – позволяет объединять несколько трансформаций вместе.

Пример 5 Код XAML

```
<StackPanel.RenderTransform>  
  <TransformGroup>  
    <RotateTransform Angle="30" />  
    <ScaleTransform ScaleX="2" ScaleY="0.8" />  
    <TranslateTransform X="-200" Y="-100" />  
  </TransformGroup>  
</StackPanel.RenderTransform>
```

Результат



Использовать атрибуты `CenterX` и `CenterY` для задания центральной точки трансформации имеет смысл в том случае, когда точно известны размеры элемента управления, подвергаемого трансформации. В том случае, когда размеры неизвестны или изменяются в процессе работы, имеет смысл задавать атрибут

RenderTransformOrigin элемента управления (не трансформации), подвергаемого трансформации, в следующем формате

RenderTransformOrigin="X,Y", где

X – относительная координата X (0 – начало элемента управления; 1 – конец элемента управления);

Y – относительная координата Y (0 – начало элемента управления; 1 – конец элемента управления).

Например, RenderTransformOrigin="0.5,0.5" смещает центр трансформации в центр элемента управления.

Анимированные трансформации

Свойства трансформации могут быть анимированы с помощью объектов анимации. Для этого необходимо добавить в код XAML элементы трансформации (можно без атрибутов), а в элементах анимации изменять следующие свойства трансформации:

- если не используется комбинированная трансформация TransformGroup:

```
RenderTransform.Angle  
RenderTransform.AngleX  
RenderTransform.AngleY  
RenderTransform.CenterX  
RenderTransform.CenterY  
RenderTransform.X  
RenderTransform.Y  
RenderTransform.ScaleX  
RenderTransform.ScaleY
```

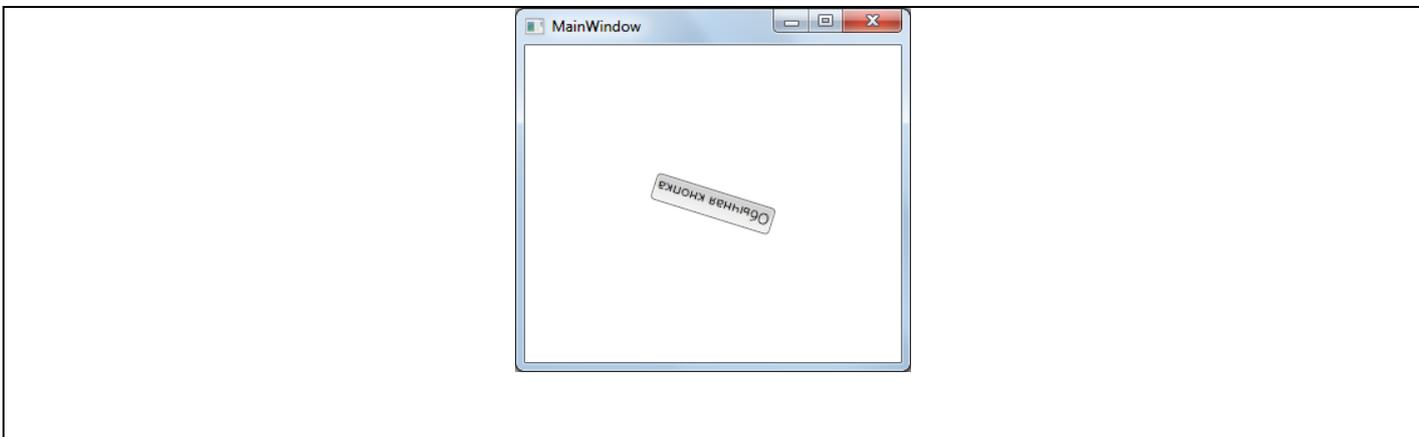
- если используется комбинированная трансформация TransformGroup:

```
RenderTransform.Children[ИНДЕКС_ТРАНСФОРМАЦИИ_В_ГРУППЕ].Angle  
RenderTransform.Children[ИНДЕКС_ТРАНСФОРМАЦИИ_В_ГРУППЕ].AngleX  
RenderTransform.Children[ИНДЕКС_ТРАНСФОРМАЦИИ_В_ГРУППЕ].AngleY  
RenderTransform.Children[ИНДЕКС_ТРАНСФОРМАЦИИ_В_ГРУППЕ].CenterX  
RenderTransform.Children[ИНДЕКС_ТРАНСФОРМАЦИИ_В_ГРУППЕ].CenterY  
RenderTransform.Children[ИНДЕКС_ТРАНСФОРМАЦИИ_В_ГРУППЕ].X  
RenderTransform.Children[ИНДЕКС_ТРАНСФОРМАЦИИ_В_ГРУППЕ].Y  
RenderTransform.Children[ИНДЕКС_ТРАНСФОРМАЦИИ_В_ГРУППЕ].ScaleX  
RenderTransform.Children[ИНДЕКС_ТРАНСФОРМАЦИИ_В_ГРУППЕ].ScaleY
```

Пример 6 Код XAML

```
<Button RenderTransformOrigin="0.5,0.5">  
    Обычная кнопка  
    <Button.RenderTransform>  
        <RotateTransform />  
    </Button.RenderTransform>  
    <Button.Triggers>  
        <EventTrigger RoutedEvent="MouseEnter">  
            <BeginStoryboard>  
                <Storyboard>  
                    <DoubleAnimation Storyboard.TargetProperty="RenderTransform.Angle"  
From="0" To="360" Duration="0:0:1" />  
                </Storyboard>  
            </BeginStoryboard>  
        </EventTrigger>  
    </Button.Triggers>  
</Button>
```

Результат



Задание 1

Для любого WPF-приложения с большим количеством элементов управления, разработанного в одной из предыдущих лабораторных работ, реализуйте масштабирование всего пользовательского интерфейса с помощью ползунка Slider.

Задание 2

Разработайте WPF-приложение «Текстовый редактор» в соответствии с изображениями пользовательского интерфейса, приведенными ниже. Блоки с многострочными текстовыми полями выводятся в нижней части интерфейса «полукругом». При выделении какого-либо блока (событие GotFocus) он плавно перемещается в центральную часть. При потере фокуса (событие LostFocus) блок возвращается в свое исходное состояние. Приложение не должно содержать код на языке C#.

Подсказки:

- используйте диспетчер компоновки Canvas;
- если для анимации не заданы атрибуты To и From, то целевым значением является исходное значение свойства, действующее до начала серии анимаций.

