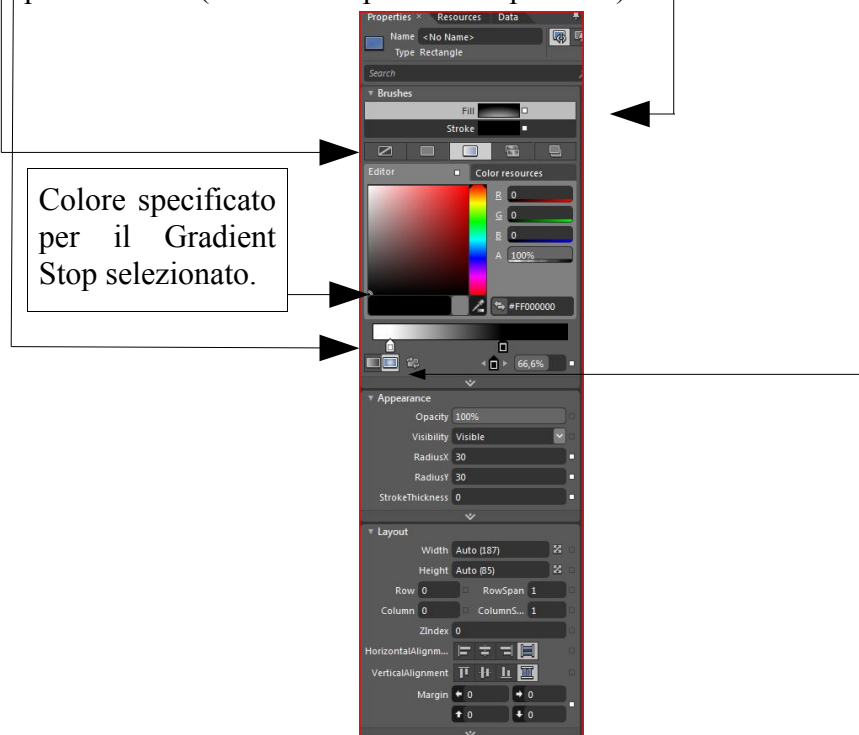


Bottoni, stili, template, stati, transizioni di stato ed animazioni

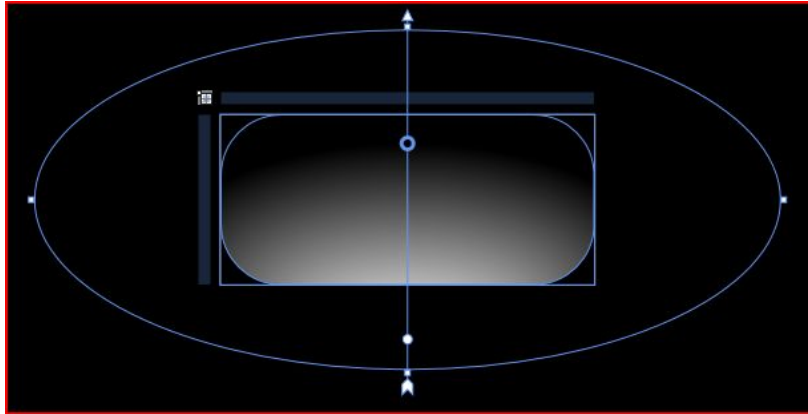
Le foto, i menu ecc. potrebbero non corrispondere perfettamente alla vostra versione di Expression Blend (in inglese) in quanto il docente ha utilizzato la versione gratuita per sviluppare applicazioni Silverlight per Windows Phone 7

Il presente Tutorial mostra una possibile strategia per realizzare dei pulsanti personalizzati, non è ne l'unica ne la migliore. Sta ad ogni singolo studente, utilizzando la propria creatività e la propria intelligenza, la ricerca di strategie alternative. Questo discorso è ancora più vero per quello che riguarda la qualità grafica e l'usabilità del prodotto ottenuto.

1. Nuovo progetto Applicazione Silverlight in Blend.
2. Dalla toolbar a sinistra selezionate una griglia e createla nell'area di lavoro. Le dimensioni al momento non hanno molta importanza in quanto saranno modificabili in seguito. Questa griglia sarà il contenitore del vostro bottone.
3. Nella toolbar sono presenti due tipi di selettori che, come icona, hanno una freccia: il primo (freccia nera) è in genere più utile perché mostra più informazioni e maniglie di lavoro per l'oggetto selezionato.
4. Alcuni pulsanti della toolbar (quelli con il triangolino bianco in basso a destra) presentano altre opzioni che è possibile visionare e scegliere tenendo premuto - su di loro - il pulsante sinistro del mouse per qualche istante, oppure cliccando con il pulsante destro.
5. Inserite un rettangolo entro la griglia ed impostate le seguenti proprietà: Stroke = Black (contorno, è già il valore di default), RadiusX = RadiusY = 30 (ma dipende da quanto potranno essere piccoli e grandi i pulsanti che useranno questo template e dall'effetto che si vuole ottenere), StrokeThickness = 0, tutti i margini a 0 in modo che il rettangolo riempa tutta la griglia.
6. Con il rettangolo sempre selezionato selezioniamo Fill per Brushes, e quindi Gradient Brush (terzo pulsantino nella fila subito sotto alla lista dei Brushes). Spostate il GradientStop del nero a 66,6%, e a 8,9% quello del bianco, scegliete il gradiente radiale (non quello lineare). Se la barra del gradiente non ha il bianco a sinistra ed il nero a destra cliccate sul pulsante per invertirle (a destra del precedente pulsante).



- Dalla toolbar selezionate ora lo strumento "Gradient Tool" e spostate le diverse maniglie in modo da ottenere la situazione di figura.



Maniglie: le 4 maniglie quadrate permettono di ridimensionare l'ellisse o, spostando il mouse leggermente a lato, di ruotarla; il cerchio più spesso (perché è quello selezionato nella foto) sulla freccia permette di spostare il gradient stop del nero (in questo caso), quello sottostante del bianco (si modificano contemporaneamente anche nella finestra delle proprietà); la maniglia iniziale della freccia imposta l'origine del gradiente e quindi la sua orientazione, quella finale (la punta della freccia) ne imposta il raggio.

Tale situazione, nell'XAML, corrisponderà ai seguenti attributi (sono possibili variazioni anche in funzione delle dimensioni del pulsante):

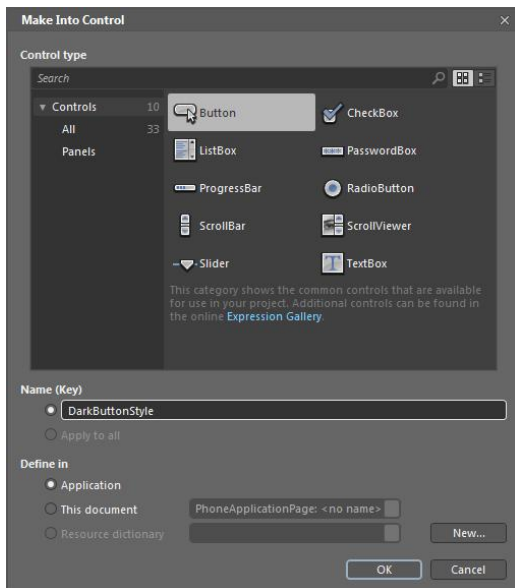
```
<RadialGradientBrush GradientOrigin="0.5,1.5" RadiusY="1" RadiusX="1">
```

L'altro tool del gradiente consente di applicare una trasformazione ulteriore (rotazione, scala e traslazione).

- Inserite una seconda griglia dentro alla prima, con tutti i margini uguali a zero per averne le stesse dimensioni. Cliccando sui due bordi blu (a sinistra e sopra la griglia) è possibile inserire righe e colonne nella griglia stessa (fate attenzione a selezionare quella interna, qualora questo risulti difficile potete utilizzare la tree-view dell'albero degli oggetti, posta in basso a sinistra, oppure l'XAML). Inserite 2 righe e tre colonne. Successivamente cliccate sulle frecce che identificano i confini tra righe e tra colonne (o, ancora meglio, sugli spazi vuoti per righe e colonne tra le suddette bande blu e la griglia), in tal modo nella finestra delle proprietà sarà possibile impostarne le dimensioni. Qui, un secondo attributo, specifica Star (la dimensione in tal caso è intesa come proporzionale), Pixel o Auto (in base al contenuto). Utilizzate Star e impostate 0,45 e 0,55 per la prima e seconda riga, e 0,05 0,9 0,05 per le tre colonne. Questa griglia ci servirà per posizionare, in modo proporzionale a dimensioni e proporzioni dei pulsanti, un altro rettangolo che rappresenterà un riflesso. Si può fare in altri modi?



9. Inserite un nuovo rettangolo dentro alla cella della prima riga, seconda colonna. Per default avrà la stessa brush (pennello) e gli spigoli arrotondati del precedente rettangolo creato. Riducete RadiusX=15 e RadiusY=15, e impostate tutti i margini a 5 (notate come diventa l'attributo XAML in questo caso in cui i 4 valori sono uguali). A questo punto cambiate il tipo di gradiente a lineare. Infine aggiungete una trasformazione di rotazione tridimensionale sull'asse X di -60°.
10. Adesso è il momento di trasformare il vostro "disegno" in un bottone vero. Selezionate la prima griglia inserita, quindi click destro su di lei, e poi "Make Into Control ...". Come tipo selezionate il pulsante, date il nome allo stile (Key) ed impostate la definizione nel file di applicazione (App.xaml) in modo da non avere tutto nello stesso file (ed avere la possibilità di dividerlo tra più pagine dell'interfaccia).



Informazione extra

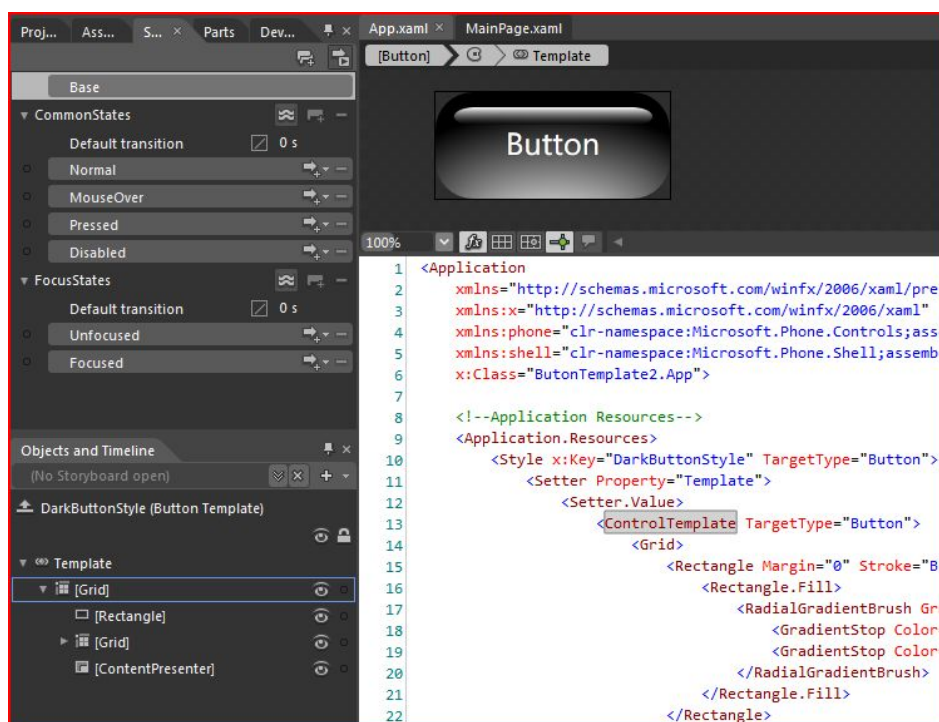
Se da

```
<Style x:Key="DarkButtonStyle"
        TargetType="Button">
```

(in App.xaml) eliminiamo la Key lo stile diventa **implicito**, ovvero il default per ogni pulsante che non specifica una risorsa di stile tramite

```
Style="{StaticResource <NomeKey>}"
```

11. A questo punto si aprirà la pagina App.xaml che vi mostrerà il template del pulsante, ed è qui che comincia il bello 😊.



12. Eseguendo ora l'applicazione si otterrebbe un pulsante molto statico, ma il nuovo aspetto dell'interfaccia che appare ora in Blend vi consente d'intervenire pesantemente ed in modo visuale sui comportamenti dinamici del pulsante. Sopra al pulsante appaiono tre schede ([Button], Stile, Template) che permettono di passare all'editing delle tre parti. Nella modalità style nella colonna a destra di Blend compaiono le solite proprietà generali di ogni controllo, ad esempio modificando l'allineamento verticale nell'XAML comparirebbe il tag

```
<Setter Property="VerticalAlignment" Value="Center"/>
```

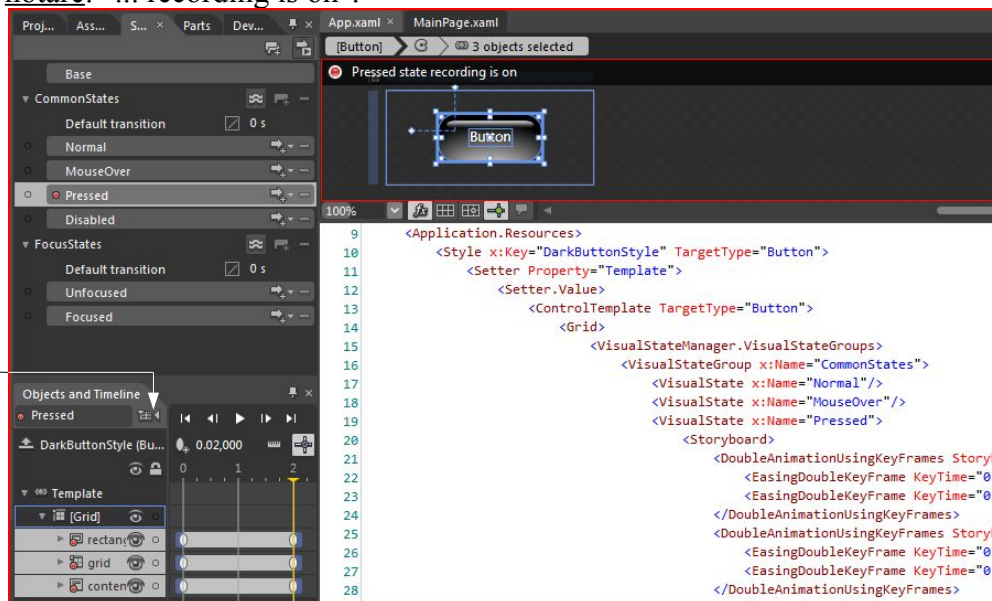
In questo caso è però più utile modificare la proprietà Foreground a bianco:

```
<Setter Property="Foreground" Value="#FFFFFF"/>
```

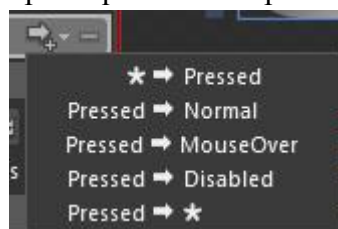
perché è utilizzata per mostrare il Content (il testo del pulsante in questo caso).

Nella modalità template possiamo modificare l'aspetto degli stati (premutato, rilasciato, mouse over ecc.) del pulsante, anche inserendo delle animazioni. Nella colonna di sinistra di Blend è possibile selezionare quale stato editare e, volendo, è possibile anche aggiungere delle animazioni di transizione tra uno stato e l'altro.

13. Selezionate lo stato Pressed. Nella tree-view dei controlli selezionate i tre controlli interni alla nostra prima griglia (content rappresenta in questo caso il testo che sarà visualizzato sul pulsante e che verrà impostato tramite la proprietà Content di ogni pulsante che utilizzerà questo template). Premete il pulsantino per visualizzare la timeline, e create i keyframe al tempo zero con le proprietà attuali dei tre controlli (pulsante con l'ellisse verticale ed il segno +). Spostatevi quindi a 2 secondi (è un valore esagerato, solo per vedere meglio come funziona durante il debug, in seguito selezionare e spostare i keyframe a 0.2) e applicate una trasformazione bidimensionale di scala (0,5 per X e Y come prova, poi conviene "scalare" di meno). In questo modo l'animazione partirà al click sul pulsante, ma si interromperà appena l'utente rilascerà il pulsante del mouse, quindi - di norma - immediatamente. Da notare: "... recording is on".



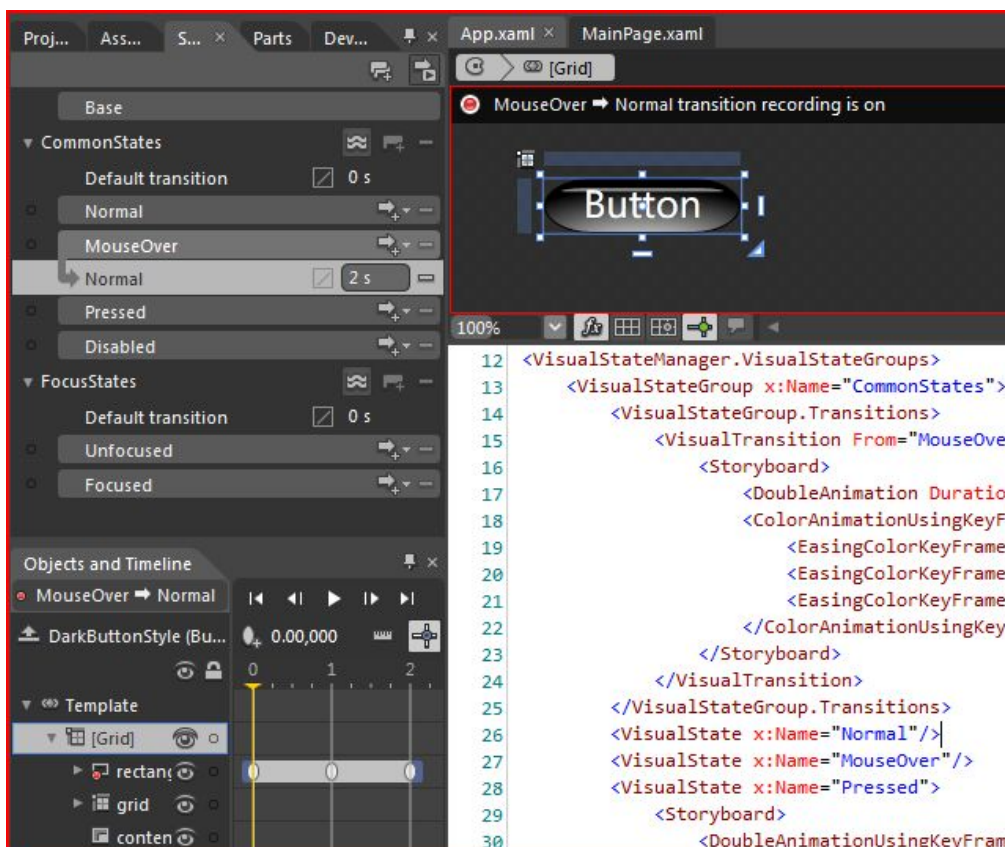
14. A destra dei nomi degli stati di un controllo appare una freccia che potete utilizzare per aggiungere animazioni per i passaggi di stato, per i quali è possibile specificare anche una durata che determina dopo quanto tempo il cambio di stato avviene.



Con l'asterisco si intende uno stato di partenza o di arrivo qualunque (in tal caso l'attributo From o To è omesso nel tag XAML). L'utilizzo dei cambi di stato può essere utile ad esempio nel passaggio in cui il mouse da sopra il bottone è spostato al di fuori (MouseOver => Normal) e si vuole che la relativa animazione duri più a lungo. Provate a replicare la seguente animazione (applicata al primo rettangolo) cercando di capire il seguente XAML, ma utilizzando gli strumenti visuali di Blend (in modo da rendervi conto se siete in grado di interpretare *parzialmente* l'XAML, l'help può aiutare visto che non studiamo i vari tag):

```
<VisualStateManager.Transitions>
  <VisualTransition From="MouseOver" GeneratedDuration="0" To="Normal">
    <Storyboard>
      <DoubleAnimation Duration="0" To="3"
        Storyboard.TargetProperty="(Shape.StrokeThickness)"
        Storyboard.TargetName="rectangle" />
      <ColorAnimationUsingKeyFrames
        Storyboard.TargetProperty="(Shape.Stroke).(SolidColorBrush.Color)"
        Storyboard.TargetName="rectangle">
        <EasingColorKeyFrame KeyTime="0" Value="Black"/>
        <EasingColorKeyFrame KeyTime="0:0:1" Value="#FF683131"/>
        <EasingColorKeyFrame KeyTime="0:0:2" Value="#FFDCE712"/>
      </ColorAnimationUsingKeyFrames>
    </Storyboard>
  </VisualTransition>
</VisualStateManager.Transitions>
```

Se si pone `GeneratedDuration="2"` anziché a 0 l'impostazione finale del bordo resta attiva, altrimenti no.



Prestate attenzione a come combinate queste animazioni.

Happy programming. Stay hungry, stay foolish. ☺