

Windows Phone 7 - Tutoriel 7 : Lucky Dice

Créer un jeu de dés aléatoires simple en utilisant Silverlight sur Windows Phone 7

par Peter Bull ([Blog](#)) Deepin Prayag (Traduction) ([Home](#))

Date de publication : 26/03/2012

Dernière mise à jour : 26/03/2012

Ce tutoriel fait partie d'une série de niveau débutant-intermédiaire pour apprendre Windows Phone 7 par le biais d'exemples pas à pas.

Langage : [C#](#)

Public visé : niveau *Débutant*

Commentez ce tutoriel :

Traduction.....	3
Introduction.....	3
Créer un jeu de dés aléatoires simple en utilisant Silverlight sur Windows Phone 7.....	3
Étape 1.....	3
Étape 2.....	4
Étape 3.....	4
Étape 4.....	6
Étape 5.....	7
Étape 6.....	8
Étape 7.....	9
Étape 8.....	10
Étape 9.....	11
Étape 10.....	11
Étape 11.....	12
Étape 12.....	13
Conclusion.....	13
Liens.....	13
Remerciements.....	14

Traduction

Ce tutoriel est la traduction la plus fidèle possible du tutoriel original de **Peter Bull**, **Windows Phone 7 - Create simple random Dice rolling game using Silverlight on Windows Phone 7**.

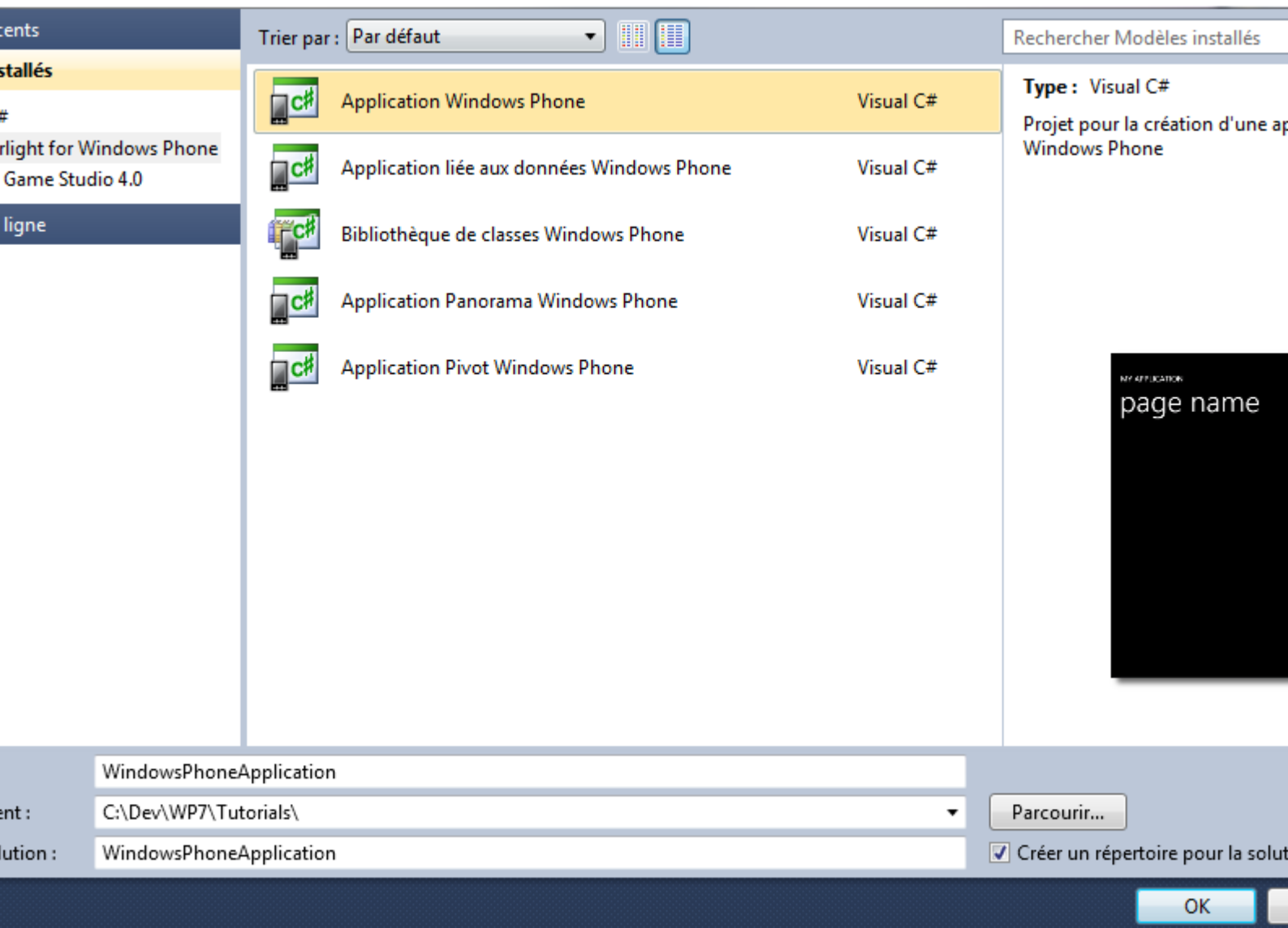
Introduction

Lucky Dice est un jeu de dés simple avec des nombres aléatoires en utilisant Silverlight sur Windows Phone 7.

Créer un jeu de dés aléatoires simple en utilisant Silverlight sur Windows Phone 7

Étape 1

Démarrez Microsoft Visual Web Developer 2010 Express pour Windows Phone, puis sélectionnez *Fichier* puis *Nouveau Projet...* Sélectionnez *Visual C#* puis *Silverlight for Windows Phone* puis *Application Windows Phone* dans les modèles installés, sélectionnez un emplacement si vous le souhaitez, puis entrez un nom pour le projet et appuyez sur OK :



Étape 2

Une page *Application Windows Phone* nommée *MainPage.xaml* devrait alors apparaître :



Étape 3

Dans le volet XAML pour *MainPage.xaml* entre les lignes `<Grid x:Name="ContentPanel" Grid.Row="1">` et `</Grid>` , tapez le code XAML suivant :

```
<Grid x:Name="ContentMain">
  <Grid.RowDefinitions>
    <RowDefinition Height="150"/>
    <RowDefinition Height="*"/>
    <RowDefinition Height="Auto"/>
  </Grid.RowDefinitions>
  <Grid Grid.Row="0" x:Name="Header">
    <Grid.ColumnDefinitions>
      <ColumnDefinition Width="150"/>
      <ColumnDefinition Width="*"/>
      <ColumnDefinition Width="150"/>
    </Grid.ColumnDefinitions>
    <!-- Header -->
  </Grid>
```

```
<!-- Footer -->  
<Button Grid.Row="2" Width="160" Content="new" Click="New_Click"/>  
</Grid>
```

XAML :

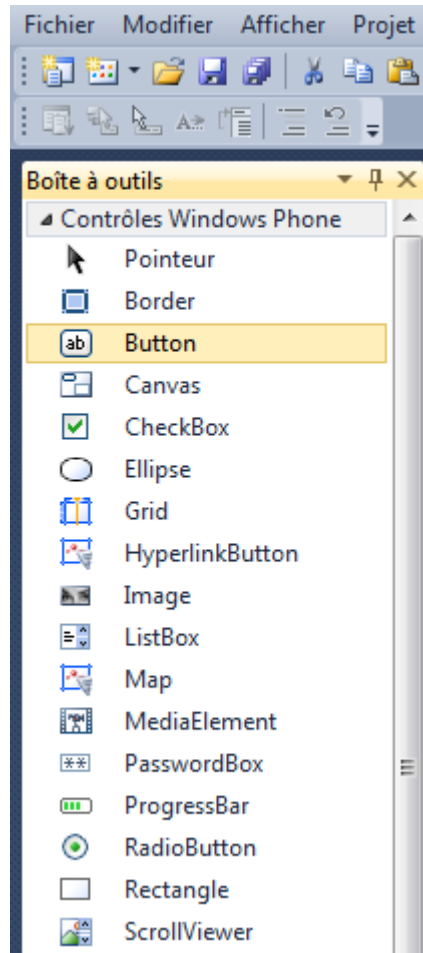
```
<Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0">  
  <Grid x:Name="ContentMain">  
    <Grid.RowDefinitions>  
      <RowDefinition Height="150"/>  
      <RowDefinition Height="*/>  
      <RowDefinition Height="Auto"/>  
    </Grid.RowDefinitions>  
    <Grid Grid.Row="0" x:Name="Header">  
      <Grid.ColumnDefinitions>  
        <ColumnDefinition Width="150"/>  
        <ColumnDefinition Width="*/>  
        <ColumnDefinition Width="150"/>  
      </Grid.ColumnDefinitions>  
      <!-- Header -->  
      <Button Grid.Column="0" Name="DiceOne" Click="DiceOne_Click"/>  
      <Button Grid.Column="2" Name="DiceTwo" Click="DiceTwo_Click"/>  
  
    </Grid>  
    <!-- Footer -->  
    <Button Grid.Row="2" Width="160" Content="new" Click="New_Click"/>  
  </Grid>  
</Grid>
```

Design :



Étape 4

Puis dans la section *Contrôles Windows Phone* de la *Boîte à outils*, sélectionnez le contrôle *Button* :



Étape 5

Dessinez deux boutons sur la section *Header* du *Grid*, la grande section en dessous du *titre de la page*, en glissant les boutons sur la partie supérieure du *Grid*. Ensuite dans le volet XAML en dessous du `<!-- Header -->`, modifiez les lignes *Button* comme ceci :

```
<Button Grid.Column="0" Name="DiceOne" Click="DiceOne_Click"/>
<Button Grid.Column="2" Name="DiceTwo" Click="DiceTwo_Click"/>
```

Voir ci-dessous :



Étape 6

Faites un clic droit sur la page ou sur l'entrée pour *MainPage.xaml* dans l'*Explorateur de solutions* et choisissez l'option *Afficher le code*. Dans la *vue de code* au-dessus de `public MainPage()` tapez la méthode suivante :

```
private void Add(Grid grid, int row, int column)
{
    Ellipse _dot = new Ellipse();
    _dot.Width = 20;
    _dot.Height = 20;
    _dot.Fill = new SolidColorBrush(Colors.White);
    _dot.SetValue(Grid.ColumnProperty, column);
    _dot.SetValue(Grid.RowProperty, row);
    grid.Children.Add(_dot);
}
```



```

private void Add(Grid grid, int row, int column)
{
    Ellipse _dot = new Ellipse();
    _dot.Width = 20;
    _dot.Height = 20;
    _dot.Fill = new SolidColorBrush(Colors.White);
    _dot.SetValue(Grid.ColumnProperty, column);
    _dot.SetValue(Grid.RowProperty, row);
    grid.Children.Add(_dot);
}
    
```

Étape 7

Puisque nous en sommes à la *vue de code* pour *MainPage.xaml.cs*, en dessous du « } » de la méthode `private void Add(...)`, tapez la méthode suivante :

```

private Grid Dice(int Value)
{
    Grid _grid = new Grid();
    _grid.Height = 100;
    _grid.Width = 100;
    for (int index = 0; (index <= 2); index++) // 3 by 3 Grid
    {
        _grid.RowDefinitions.Add(new RowDefinition());
        _grid.ColumnDefinitions.Add(new ColumnDefinition());
    }
    switch (Value)
    {
        case 0:
            // No Dots
            break;
        case 1:
            Add(_grid, 1, 1); // Middle
            break;
        case 2:
            Add(_grid, 0, 2); // Top Right
            Add(_grid, 2, 0); // Bottom Left
            break;
        case 3:
            Add(_grid, 0, 2); // Top Right
            Add(_grid, 1, 1); // Middle
            Add(_grid, 2, 0); // Bottom Left
            break;
        case 4:
            Add(_grid, 0, 0); // Top Left
            Add(_grid, 0, 2); // Top Right
            Add(_grid, 2, 0); // Bottom Left
            Add(_grid, 2, 2); // Bottom Right
            break;
        case 5:
            Add(_grid, 0, 0); // Top Left
            Add(_grid, 0, 2); // Top Right
            Add(_grid, 1, 1); // Middle
            Add(_grid, 2, 0); // Bottom Left
            Add(_grid, 2, 2); // Bottom Right
            break;
        case 6:
            Add(_grid, 0, 0); // Top Left
            Add(_grid, 0, 2); // Top Right
            Add(_grid, 1, 0); // Middle Left
            Add(_grid, 1, 2); // Middle Right
            Add(_grid, 2, 0); // Bottom Left
            Add(_grid, 2, 2); // Bottom Right
            break;
    }
    return _grid;
}
    
```

```

private Grid Dice(int Value)
{
    Grid _grid = new Grid();
    _grid.Height = 100;
    _grid.Width = 100;
    for (int index = 0; (index <= 2); index++) // 3 by 3 Grid
    {
        _grid.RowDefinitions.Add(new RowDefinition());
        _grid.ColumnDefinitions.Add(new ColumnDefinition());
    }

    switch (Value)
    {
        case 0:
            // No Dots
            break;
        case 1:
            Add(_grid, 1, 1); // Middle
            break;
        case 2:
            Add(_grid, 0, 2); // Top Right
            Add(_grid, 2, 0); // Bottom Left
            break;
        case 3:
            Add(_grid, 0, 2); // Top Right
            Add(_grid, 1, 1); // Middle
            Add(_grid, 2, 0); // Bottom Left
            break;
        case 4:
            Add(_grid, 0, 0); // Top Left
            Add(_grid, 0, 2); // Top Right
            Add(_grid, 2, 0); // Bottom Left
            Add(_grid, 2, 2); // Bottom Right
            break;
        case 5:
            Add(_grid, 0, 0); // Top Left
            Add(_grid, 0, 2); // Top Right
            Add(_grid, 1, 1); // Middle
            Add(_grid, 2, 0); // Bottom Left
            Add(_grid, 2, 2); // Bottom Right
            break;
        case 6:
            Add(_grid, 0, 0); // Top Left
            Add(_grid, 0, 2); // Top Right
            Add(_grid, 1, 0); // Middle Left
            Add(_grid, 1, 2); // Middle Right
            Add(_grid, 2, 0); // Bottom Left
            Add(_grid, 2, 2); // Bottom Right
            break;
    }
    return _grid;
}
    
```

Étape 8

Puisque nous en sommes toujours à la *vue de code* pour *MainPage.xaml.cs*, en dessous du « } » de la méthode `private Grid Dice(...)` tapez la méthode suivante :

```
private int Roll()
{
    Random random = new Random();
    return random.Next(1,7);
}
```

```
private int Roll()
{
    Random random = new Random();
    return random.Next(1, 7);
}
```

Étape 9

En restant toujours dans la *vue de code* pour *MainPage.xaml.cs*, en dessous du « } » de la méthode `public MainPage()` tapez les *gestionnaires d'évènements* suivants :

```
private void New_Click(object sender, RoutedEventArgs e)
{
    DiceOne.Content = Dice(0);
    DiceTwo.Content = Dice(0);
}

private void DiceOne_Click(object sender, RoutedEventArgs e)
{
    DiceOne.Content = Dice(Roll());
}

private void DiceTwo_Click(object sender, RoutedEventArgs e)
{
    DiceTwo.Content = Dice(Roll());
}
```

```
private void New_Click(object sender, RoutedEventArgs e)
{
    DiceOne.Content = Dice(0);
    DiceTwo.Content = Dice(0);
}

private void DiceOne_Click(object sender, RoutedEventArgs e)
{
    DiceOne.Content = Dice(Roll());
}

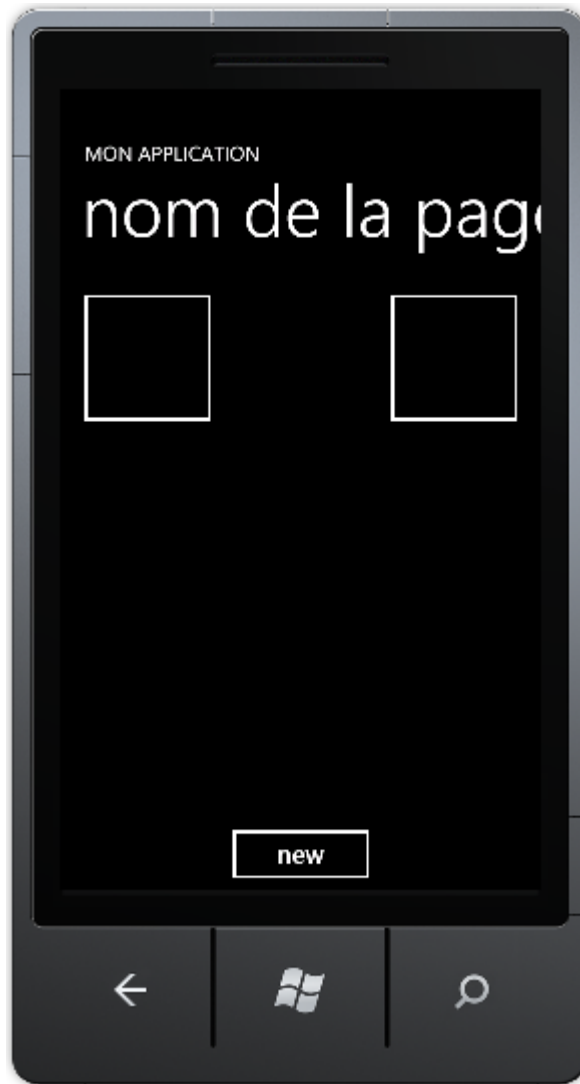
private void DiceTwo_Click(object sender, RoutedEventArgs e)
{
    DiceTwo.Content = Dice(Roll());
}
```

Étape 10

Enregistrez le *Projet* maintenant que vous avez terminé l'application *Windows Phone Silverlight*. Sélectionnez l'option *Windows Phone Emulator* ensuite sélectionnez *Debug* puis *Démarrer le débogage* ou cliquez sur *Démarrer le débogage* :

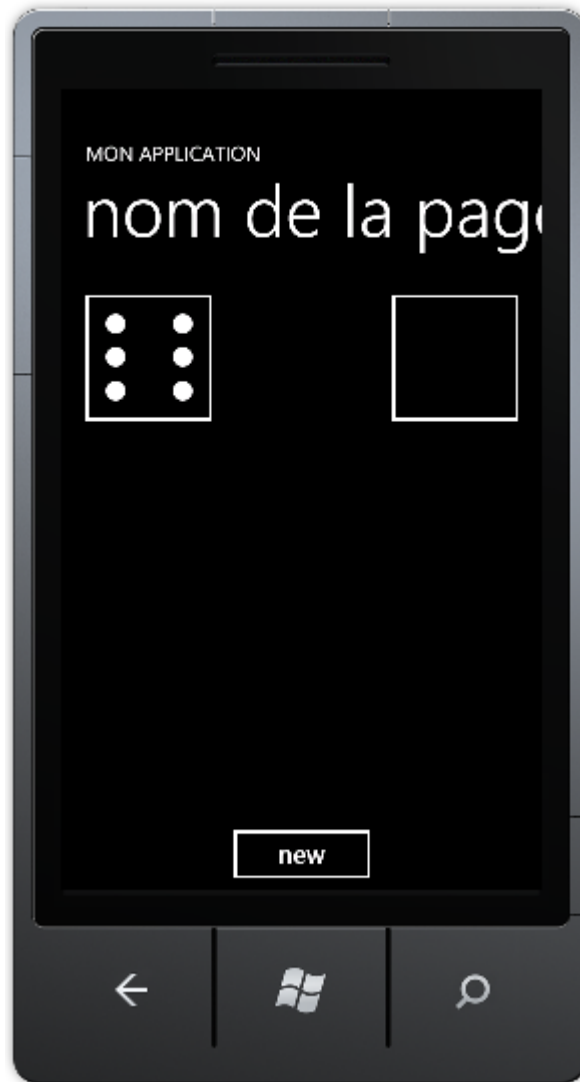


Une fois que vous l'aurez fait, ce qui suit apparaîtra dans *l'émulateur Windows Phone* après qu'il aura été chargé :



Étape 11

Appuyez sur le grand bouton sur la gauche (*DiceOne*) pour le Joueur Un :



Étape 12

Vous pouvez ensuite *Arrêter l'application* en sélectionnant la fenêtre d'application *Visual Studio 2010* et en cliquant sur le bouton *Arrêter le débogage* :



Conclusion

En appuyant soit sur le premier grand bouton vous avez le Joueur Un soit sur le second et vous avez le Joueur Deux (à gauche et à droite respectivement). Ceci est un jeu de dés très simple, vous pouvez ajouter un système de score pour voir qui a fait un meilleur score et attribuez à ce joueur des points ou voir ce que vous pouvez ajouter d'autre à ce jeu simple - personnalisez-la !

Liens

Télécharger le code source du tutoriel.

Remerciements

Je tiens ici à remercier **Peter Bull** de m'avoir autorisé à traduire son tutoriel.
Je remercie **xxx** pour sa relecture technique et ses propositions.
Je remercie également **yyy** pour sa relecture orthographique et ses propositions.