

Windows Phone 7 - Tutoriel 9 : Tic-tac-toe

Morpion ou Tic-Tac-Toe, un jeu très simple avec les règles Gagnant/
Nul appliquées, en utilisant Silverlight sur Windows Phone 7

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

Date de publication : 27/03/2012

Dernière mise à jour : 27/03/2012

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 *Intermédiaire*

Commentez ce tutoriel :

Traduction.....	3
Introduction.....	3
Morpion ou Tic-Tac-Toe, un jeu très simple avec les règles Gagnant/Nul appliquées, en utilisant Silverlight sur Windows Phone 7.....	3
Étape 1.....	3
Étape 2.....	4
Étape 3.....	5
Étape 4.....	6
Étape 5.....	7
Étape 6.....	8
Étape 7.....	9
Étape 8.....	10
Étape 9.....	11
Étape 10.....	12
Étape 11.....	13
Étape 12.....	14
Étape 13.....	15
Étape 14.....	16
Conclusion.....	16
Liens.....	16
Remerciements.....	17

Traduction

Ce tutoriel est la traduction la plus fidèle possible du tutoriel original de **Peter Bull**, **Windows Phone 7 - Noughts and Crosses or Tic-Tac-Toe with Winner / Draw rules applied using Silverlight on Windows Phone 7**.

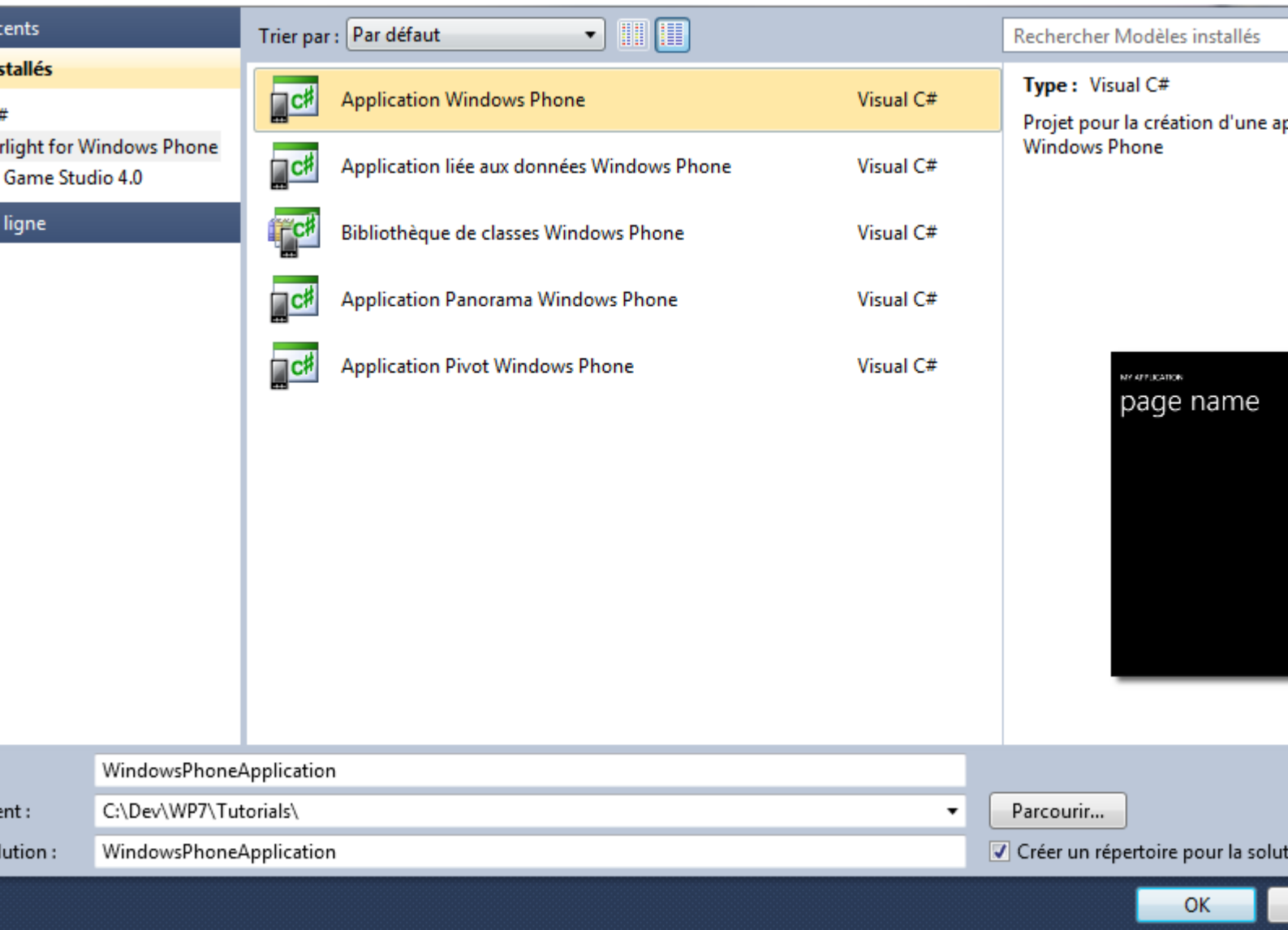
Introduction

Morpion ou Tic-Tac-Toe est un jeu très simple à jouer et à créer avec ce tutoriel en utilisant Silverlight sur Windows Phone 7.

Morpion ou Tic-Tac-Toe, un jeu très simple avec les règles Gagnant/Nul appliquées, 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="*" />
    <RowDefinition Height="Auto" />
  </Grid.RowDefinitions>
  <Button Grid.Row="1" Width="160" Content="new" Click="New_Click" />
</Grid>
```

XAML :



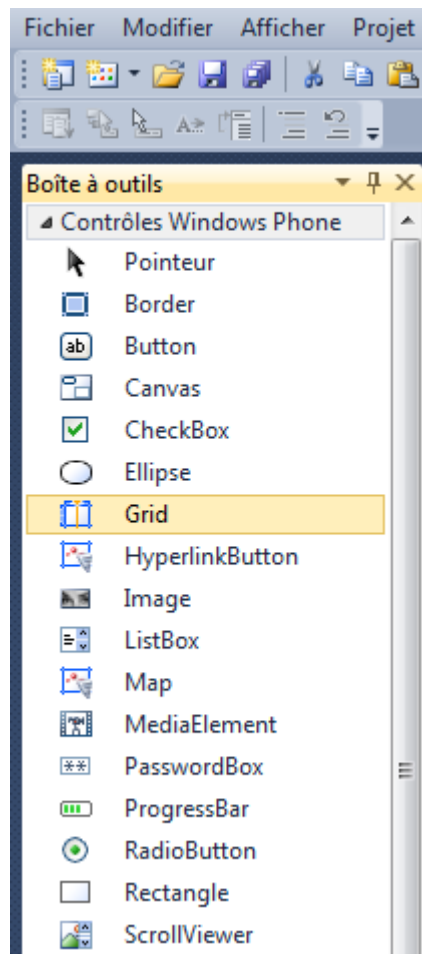
```
<Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0">
  <Grid x:Name="ContentMain">
    <Grid.RowDefinitions>
      <RowDefinition Height="*" />
      <RowDefinition Height="Auto" />
    </Grid.RowDefinitions>
    <Button Grid.Row="1" Width="160" Content="new" Click="New_Click" />
  </Grid>
</Grid>
```

Design :



Étape 4

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



Étape 5

Dessinez un *Grid* sur la page en glissant un *Grid* de la *Boîte à outils* sur la partie intermédiaire du *Grid* sur la page, ensuite dans le volet XAML modifiez la ligne *grid1* comme ceci :

```
<Grid Height="400" Width="400" Name="Display"/>
```

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 les *déclarations* suivantes :

```
bool _hasWon = false;
string _piece = "";
string _nought = "O";
string _cross = "X";
string[,] _board = new string[3,3];
```



```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Animation;
using System.Windows.Shapes;
using Microsoft.Phone.Controls;

namespace NoughtsAndCrosses
{
    public partial class MainPage : PhoneApplicationPage
    {
        bool _hasWon = false;
        string _piece = "";
        string _nought = "O";
        string _cross = "X";
        string[,] _board = new string[3, 3];

        // Constructor
        public MainPage()
        {
            InitializeComponent();
        }
    }
}

```

Étape 7

Puisque nous en sommes à la *vue de code* pour *MainPage.xaml.cs* au-dessus de `public MainPage()` tapez les *méthodes* suivantes :

```

private bool Winner()
{
    return
        (_board[0, 0] == _piece && _board[0, 1] == _piece && _board[0, 2] == _piece) ||
        (_board[1, 0] == _piece && _board[1, 1] == _piece && _board[1, 2] == _piece) ||
        (_board[2, 0] == _piece && _board[2, 1] == _piece && _board[2, 2] == _piece) ||
        (_board[0, 0] == _piece && _board[1, 0] == _piece && _board[2, 0] == _piece) ||
        (_board[0, 1] == _piece && _board[1, 1] == _piece && _board[2, 1] == _piece) ||
        (_board[0, 2] == _piece && _board[1, 2] == _piece && _board[2, 2] == _piece) ||
        (_board[0, 0] == _piece && _board[1, 1] == _piece && _board[2, 2] == _piece) ||
        (_board[0, 2] == _piece && _board[1, 1] == _piece && _board[2, 0] == _piece);
}

private bool Drawn()
{
    return
        _board[0, 0] != "" && _board[0, 1] != "" && _board[0, 2] != "" &&
        _board[1, 0] != "" && _board[1, 1] != "" && _board[1, 2] != "" &&
        _board[2, 0] != "" && _board[2, 1] != "" && _board[2, 2] != "";
}

```

```
private bool Winner()
{
    return
        (_board[0, 0] == _piece && _board[0, 1] == _piece && _board[0, 2] == _piece) |
        (_board[1, 0] == _piece && _board[1, 1] == _piece && _board[1, 2] == _piece) |
        (_board[2, 0] == _piece && _board[2, 1] == _piece && _board[2, 2] == _piece) |
        (_board[0, 0] == _piece && _board[1, 0] == _piece && _board[2, 0] == _piece) |
        (_board[0, 1] == _piece && _board[1, 1] == _piece && _board[2, 1] == _piece) |
        (_board[0, 2] == _piece && _board[1, 2] == _piece && _board[2, 2] == _piece) |
        (_board[0, 0] == _piece && _board[1, 1] == _piece && _board[2, 2] == _piece) |
        (_board[0, 2] == _piece && _board[1, 1] == _piece && _board[2, 0] == _piece);
}

private bool Drawn()
{
    return
        _board[0, 0] != "" && _board[0, 1] != "" && _board[0, 2] != "" &&
        _board[1, 0] != "" && _board[1, 1] != "" && _board[1, 2] != "" &&
        _board[2, 0] != "" && _board[2, 1] != "" && _board[2, 2] != "";
}
```

Étape 8

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

```
private Path GetPiece()
{
    if ((_piece == _cross)) // Draw X
    {
        Path _lines = new Path();
        LineGeometry _line1 = new LineGeometry();
        LineGeometry _line2 = new LineGeometry();
        GeometryGroup _linegroup = new GeometryGroup();
        _line1.StartPoint = new Point(0, 0);
        _line1.EndPoint = new Point(64, 64);
        _line2.StartPoint = new Point(64, 0);
        _line2.EndPoint = new Point(0, 64);
        _linegroup.Children.Add(_line1);
        _linegroup.Children.Add(_line2);
        _lines.Data = _linegroup;
        _lines.Stroke = new SolidColorBrush(Colors.Red);
        _lines.StrokeThickness = 6;
        _lines.Margin = new Thickness(5);
        return _lines;
    }
    else // Draw O
    {
        EllipseGeometry _ellipse = new EllipseGeometry();
        Path _circle = new Path();
        _ellipse.Center = new Point(32, 32);
        _ellipse.RadiusX = 32;
        _ellipse.RadiusY = 32;
        _circle.Data = _ellipse;
        _circle.Stroke = new SolidColorBrush(Colors.Blue);
        _circle.StrokeThickness = 6;
        _circle.Margin = new Thickness(5);
        return _circle;
    }
}
```

```

private Path GetPiece()
{
    if ((_piece == _cross)) // Draw X
    {
        Path _lines = new Path();
        LineGeometry _line1 = new LineGeometry();
        LineGeometry _line2 = new LineGeometry();
        GeometryGroup _linegroup = new GeometryGroup();
        _line1.StartPoint = new Point(0, 0);
        _line1.EndPoint = new Point(64, 64);
        _line2.StartPoint = new Point(64, 0);
        _line2.EndPoint = new Point(0, 64);
        _linegroup.Children.Add(_line1);
        _linegroup.Children.Add(_line2);
        _lines.Data = _linegroup;
        _lines.Stroke = new SolidColorBrush(Colors.Red);
        _lines.StrokeThickness = 6;
        _lines.Margin = new Thickness(5);
        return _lines;
    }
    else // Draw O
    {
        EllipseGeometry _ellipse = new EllipseGeometry();
        Path _circle = new Path();
        _ellipse.Center = new Point(32, 32);
        _ellipse.RadiusX = 32;
        _ellipse.RadiusY = 32;
        _circle.Data = _ellipse;
        _circle.Stroke = new SolidColorBrush(Colors.Blue);
        _circle.StrokeThickness = 6;
        _circle.Margin = new Thickness(5);
        return _circle;
    }
}

```

Étape 9

En restant toujours dans la *vue de code* pour *MainPage.xaml.cs*, en dessous du « } » de la méthode `private Path GetPiece()` tapez le *gestionnaire d'évènement Button* suivant :

```

private void Button_Click(object sender, System.Windows.RoutedEventArgs e)
{
    if (!_hasWon)
    {
        Button _btn = new Button();
        _btn = ((Button) sender);
        if ((_btn.Content == null))
        {
            _btn.Content = GetPiece();
            _board[(int)_btn.GetValue(Grid.RowProperty),
                (int)_btn.GetValue(Grid.ColumnProperty)] = _piece;
        }
        if (Winner())
        {
            _hasWon = true;
            MessageBox.Show((_piece + " wins!"), "Noughts and Crosses", MessageBoxButton.OK);
        }
        else if (Drawn())
        {
            MessageBox.Show("Draw!", "Noughts and Crosses", MessageBoxButton.OK);
        }
    }
}

```

```
else
{
    _piece = ((_piece == _cross) ? _nought : _cross); // Swap Players
}
}
else
{
    MessageBox.Show("Game Over!", "Noughts and Crosses", MessageBoxButton.OK);
}
}
```

```
private void Button_Click(object sender, System.Windows.RoutedEventArgs e)
{
    if (!_hasWon)
    {
        Button _btn = new Button();
        _btn = ((Button)sender);
        if (_btn.Content == null)
        {
            _btn.Content = GetPiece();
            _board[(int)_btn.GetValue(Grid.RowProperty),
                (int)_btn.GetValue(Grid.ColumnProperty)] = _piece;
        }
        if (Winner())
        {
            _hasWon = true;
            MessageBox.Show(_piece + " wins!", "Noughts and Crosses", MessageBoxButton.OK);
        }
        else if (Drawn())
        {
            MessageBox.Show("Draw!", "Noughts and Crosses", MessageBoxButton.OK);
        }
        else
        {
            _piece = ((_piece == _cross) ? _nought : _cross); // Swap Players
        }
    }
    else
    {
        MessageBox.Show("Game Over!", "Noughts and Crosses", MessageBoxButton.OK);
    }
}
```

Étape 10

Toujours dans le *vue de code* pour *MainPage.xaml.cs*, en dessous du « } » de la méthode `private void Button_Click(...)` tapez les *méthodes* suivantes :

```
private void Add(ref Grid Grid, int Row, int Column)
{
    Button _btn = new Button();
    _btn.Click += Button_Click;
    _btn.Content = null;
    _btn.Margin = new Thickness(2);
    _btn.SetValue(Grid.ColumnProperty, Column);
    _btn.SetValue(Grid.RowProperty, Row);
    Grid.Children.Add(_btn);
}

private void Layout(ref Grid Grid)
{
}
```

```

Grid.Children.Clear();
Grid.ColumnDefinitions.Clear();
Grid.RowDefinitions.Clear();
// Setup 3x3 Grid
for (int Index = 0; (Index <= 2); Index++)
{
    Grid.RowDefinitions.Add(new RowDefinition());
    Grid.ColumnDefinitions.Add(new ColumnDefinition());
}
Add(ref Grid, 0, 0); // Top Left
Add(ref Grid, 0, 1); // Top Middle
Add(ref Grid, 0, 2); // Top Right
Add(ref Grid, 1, 0); // Middle Left
Add(ref Grid, 1, 1); // Centre
Add(ref Grid, 1, 2); // Middle Right
Add(ref Grid, 2, 0); // Bottom Left
Add(ref Grid, 2, 1); // Bottom Middle
Add(ref Grid, 2, 2); // Bottom Right
}

```

```

private void Add(ref Grid Grid, int Row, int Column)
{
    Button _btn = new Button();
    _btn.Click += Button_Click;
    _btn.Content = null;
    _btn.Margin = new Thickness(2);
    _btn.SetValue(Grid.ColumnProperty, Column);
    _btn.SetValue(Grid.RowProperty, Row);
    Grid.Children.Add(_btn);
}

private void Layout(ref Grid Grid)
{
    Grid.Children.Clear();
    Grid.ColumnDefinitions.Clear();
    Grid.RowDefinitions.Clear();
    // Setup 3x3 Grid
    for (int Index = 0; (Index <= 2); Index++)
    {
        Grid.RowDefinitions.Add(new RowDefinition());
        Grid.ColumnDefinitions.Add(new ColumnDefinition());
    }
    Add(ref Grid, 0, 0); // Top Left
    Add(ref Grid, 0, 1); // Top Middle
    Add(ref Grid, 0, 2); // Top Right
    Add(ref Grid, 1, 0); // Middle Left
    Add(ref Grid, 1, 1); // Centre
    Add(ref Grid, 1, 2); // Middle Right
    Add(ref Grid, 2, 0); // Bottom Left
    Add(ref Grid, 2, 1); // Bottom Middle
    Add(ref Grid, 2, 2); // Bottom Right
}

```

Étape 11

Toujours dans la *vue de code* pour *MainPage.xaml.cs*, en dessous du « } » de la méthode `public MainPage()` tapez le *gestionnaire d'évènement* suivant :

```

private void New_Click(object sender, RoutedEventArgs e)
{
    Layout(ref Display);
}

```

```

_board[0, 0] = "";
_board[0, 1] = "";
_board[0, 2] = "";
_board[1, 0] = "";
_board[1, 1] = "";
_board[1, 2] = "";
_board[2, 0] = "";
_board[2, 1] = "";
_board[2, 2] = "";
_hasWon = false;
if ((MessageBox.Show((_cross + " to go first?"), "Noughts and Crosses",
MessageBoxButton.OKCancel) == MessageBoxResult.OK))
{
    _piece = _cross;
}
else
{
    _piece = _nought;
}
}

```

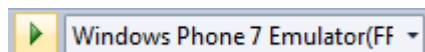
```

private void New_Click(object sender, RoutedEventArgs e)
{
    Layout(ref Display);
    _board[0, 0] = "";
    _board[0, 1] = "";
    _board[0, 2] = "";
    _board[1, 0] = "";
    _board[1, 1] = "";
    _board[1, 2] = "";
    _board[2, 0] = "";
    _board[2, 1] = "";
    _board[2, 2] = "";
    _hasWon = false;
    if ((MessageBox.Show((_cross + " to go first?"), "Noughts and Crosses",
    MessageBoxButton.OKCancel) == MessageBoxResult.OK))
    {
        _piece = _cross;
    }
    else
    {
        _piece = _nought;
    }
}

```

Étape 12

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* :

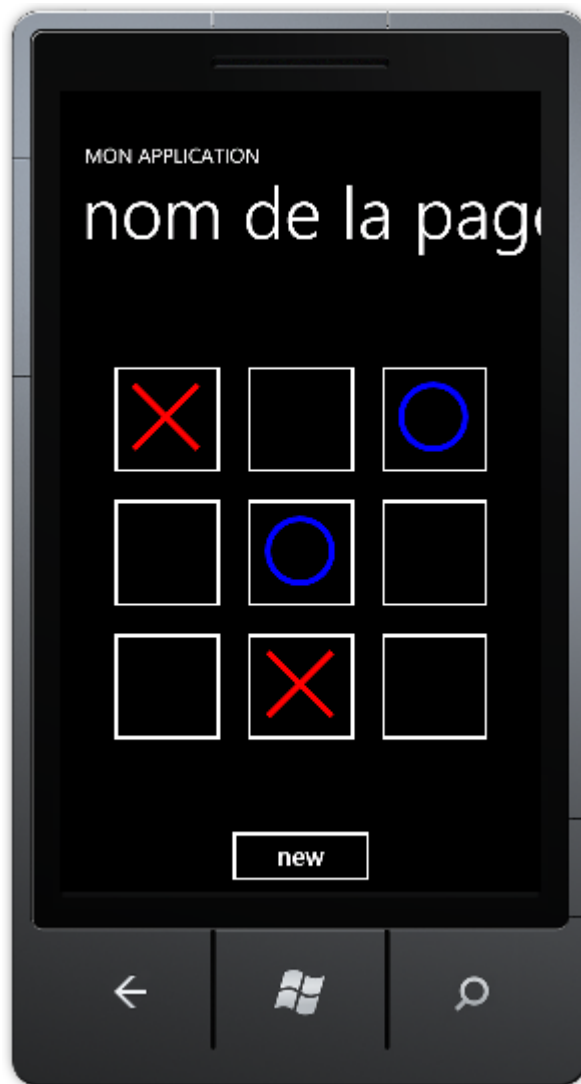


Une fois que vous l'aurez fait, ce qui suit apparaîtra dans l'*émulateur Windows Phone* après qu'il aura été chargé :



Étape 13

Appuyez sur le bouton *new*, ensuite on vous demandera « *X to go first ?* », choisissez *OK* pour *X* ou *Cancel* pour que ce soit *O* qui commence en premier, et ceci apparaîtra :



Étape 14

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

Tic-tac-toe est un jeu très simple. Essayez d'ajouter quelques fonctionnalités plus avancées telles qu'un Joueur 2 (l'ordinateur) en utilisant des nombres aléatoires ou essayez tout simplement de changer les couleurs utilisées. Personnalisez le jeu !

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.