Published: 8/24/2011

Windows Phone

XNA for Windows Phone

Rob S. Miles | Microsoft MVP | University of Hull, UK
Andy Wigley | Microsoft MVP | Appa Mundi

Session 11.0

NOKIA

Course Schedule

- Session 1 – Tuesday, August 23, 2011
  - Building Windows Phone Apps with Visual Studio 2010
  - Silverlight on Windows Phone—Introduction
  - Silverlight on Windows Phone—Advanced
  - Using Expression to Build Windows Phone Interfaces
  - Windows Phone Fast Application Switching
  - Windows Phone Multi-tasking & Background Tasks
  - Using Windows Phone Resources (Bing Maps, Camera, etc.)
- Session 2 – Wednesday, August 24, 2011
  - Application Data Storage on Windows Phone
  - Using Networks with Windows Phone
  - Tiles & Notifications on Windows Phone
  - **XNA for Windows Phone**
  - Selling a Windows Phone Application

NOKIA Microsoft

Published: 8/24/2011

Topics

- XNA on Windows Phone
- How XNA Games Work
- Creating an XNA game
- Images and Sprites
- Using the Touch Panel
- Using the Accelerometer and Motion sensor
- Sound in XNA Games
- Combining XNA and Silverlight
- XNA games and Fast Application Switching

NOKIA Microsoft

Published: 8/24/2011

XNA Introduction

Windows Phone as an XNA Platform

- Windows Phone is a great platform for games
- Performance is impressive, especially in 3D
  - Hardware based graphics acceleration
- There are some very interesting input options
- You can use all the hardware and sensors in your Windows Phone games
- Potential for Xbox Live integration
  - Support for Avatars and Achievements

NOKIA Microsoft

Published: 8/24/2011

Quick Overview of XNA

- The XNA Framework provides everything you need to get started writing games:
- Full Content Management (integrated into Visual Studio)
- Support for 2D Sprite-based gameplay
- Support for 3D games
- Common behaviours across the Windows PC, Xbox 360 and Windows Phone
  - One game engine can run on all platforms
- Well factored object model

NOKIA Microsoft

How Games Work

Every game that has ever been written has these fundamental behaviours:

- Initialise all the resources at the start
  - fetch all textures, models, scripts etc
- Repeatedly run the game loop:
  - Update the game world
    - read the controllers, update the state and position of game elements
  - Draw the game world
    - render the game elements on the viewing device

NOKIA Microsoft

Methods in an XNA Game

- The XNA Game class contains methods that will provide these behaviours
- Initialise all the resources at the start
  - The `Initialize` and `LoadContent` methods are used to start the game
- Repeatedly run the game loop:
  - Update the game world
    - The `Update` method
  - Draw the game world
    - The `Draw` method

NOKIA Microsoft

Getting Started with XNA

- When you create a new XNA game project you are provided with empty versions of the game methods
- Creating an XNA game is a matter of filling in these methods to get the game behaviours that are required
- We are going to start by getting some cheese moving around the display
- Then we are going to start rolling it around to score points
- This is the latest cheese game in a long running series:
- Behold CHEESE ROLLER!

NOKIA Microsoft

Creating a Game World

- An XNA game uses a number of variables to represent the state of the game itself
  - The `Update` method will update their values
  - The `Draw` method will produce a display that reflects their value
- In the case of Cheese Roller the game must manage the texture and draw position of the cheese on the screen

NOKIA Microsoft

Published: 8/24/2011

Creating a Cheese Sprite

```
Texture2D cheeseTexture;
Rectangle cheeseDrawPosition;
```

- The cheese is actually a sprite
  - A sprite is a texture design and some position information
- The design is held in an XNA `Texture2D` object
- The position is held in an XNA `Rectangle`
  - It holds X and Y position and Width and Height

NOKIA Microsoft

Loading the Cheese Texture

```
protected override void LoadContent()
{
    // Create a SpriteBatch, used to draw textures
    spriteBatch = new SpriteBatch(GraphicsDevice);

    cheeseTexture = Content.Load<Texture2D>("Edam");
}
```

- `LoadContent` method called when th game starts running
- It fetches all the assets and loads them into game elements

NOKIA Microsoft

Creating the Draw Rectangle

```
protected override void Initialize()
{
    // Create the draw rectangle
    // Cheese should be a 12th the size of the screen
    cheeseDrawPosition = new Rectangle(
        0, 0, // top left hand corner of the screen
        GraphicsDevice.Viewport.Width / 12, // width
        GraphicsDevice.Viewport.Width / 12  // height
        );
    base.Initialize();
}
```

- The draw position rectangle determines the size of the sprite and where on the screen it is drawn

NOKIA Microsoft

Drawing the Cheese

```
protected override void Draw(GameTime gameTime)
{
   GraphicsDevice.Clear(Color.CornflowerBlue);

    spriteBatch.Begin();
    spriteBatch.Draw(cheeseTexture, cheeseDrawPosition,
                                                  Color.White);
    spriteBatch.End();

    base.Draw(gameTime);
}
```

- The `Draw` method draws the texture at the position described by the rectangle

NOKIA Microsoft

Published: 8/24/2011

Windows Phone

Demo

Demo 1: Drawing Cheese

Published: 8/24/2011

Windows Phone and Orientation

- By default a Windows Phone XNA game assumes it is running in "landscape" mode with the screen on the left of the controls
- I want to change this, because I want to send the cheese down the long axis of the phone when it is rolled
- I can select orientation when the game starts

NOKIA Microsoft

Published: 8/24/2011

Orientation Management

- An XNA game can specify the orientations it can support
- The game can bind a method to the event which is fired when the orientation changes
  - Games can animate the transition to the new orientation if they wish
- When orientation changes the origin for the draw position is moved to the top left hand corner of the viewport
- The frame of reference for accelerometer readings (of which more later) is always as for the phone held in portrait mode

NOKIA Microsoft

Size and the Scaler

- An XNA game can also request a specific size of back buffer for the display
- The game will be given this resolution irrespective of the actual device
  - The Graphics Processor (GPU) performs this scaling automatically and interpolates to remove any jagged edges
- This is a great way to get performance boost by rendering to a lower resolution than the screen itself
  - In a fast moving game the lower resolution is not noticeable

NOKIA Microsoft

Published: 8/24/2011

Orientation and Display Size

- We can set the orientation and the desired screen size when the game starts up

```
// Tell XNA we can only support Portrait orientation
// Can support a range of orientations by ORing
// together orientation values
graphics.SupportedOrientations = DisplayOrientation.Portrait;

// Set up hardware scaling of the display area
graphics.PreferredBackBufferWidth = 480;
graphics.PreferredBackBufferHeight = 800;
```

NOKIA Microsoft

Full Screen Mode

- By default an XNA game will leave a border at the top of the screen for signal strength and battery status displays
  - This is very obvious if the "Light" colour scheme is used
- By setting this option the XNA game will use the whole of the screen

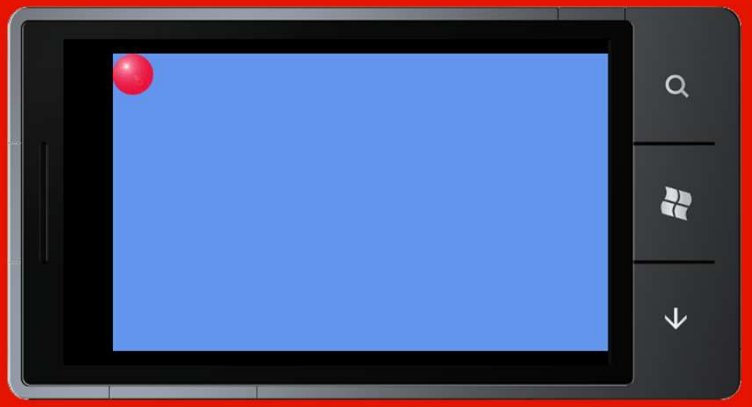
```
// Want a full screen display
graphics.IsFullScreen = true;
```

NOKIA Microsoft

Published: 8/24/2011

Windows Phone

Demo



Demo 2: Orientation and Scale

Adding Movement

Storing the Cheese Position

```
Vector2 cheesePosition;
Vector2 cheeseSpeed;
float friction = 0.99f;
```

- We manage the rolling movement of the cheese by using a vector to give the position and another to hold the speed
- Each time the game updates we also apply a friction value to slow the cheese down

NOKIA Microsoft

Updating the Cheese Position

```
cheesePosition = cheesePosition + cheeseSpeed ;
cheeseSpeed = cheeseSpeed * friction;
```

- The `Update` method is called 30 times a second by XNA to update the game objects
- This moves the cheese and reduces the speed according to the friction
- This is a very simple Physics Model

NOKIA Microsoft

Bouncing the Cheese

```
if (cheeseDrawPosition.X < 0)
    cheeseSpeed.X = Math.Abs(cheeseSpeed.X);
if (cheeseDrawPosition.Right >

GraphicsDevice.Viewport.Width)
    cheeseSpeed.X = -Math.Abs(cheeseSpeed.X);
```

- This code bounces the cheese off the edges of the screen
- If the cheese moves off the sides the speed in the given direction is reversed
- This is a “perfect” bounce

NOKIA Microsoft

Setting the Cheese Draw Position

```
cheeseDrawPosition.X = (int)(cheesePosition.X + 0.5f);
cheeseDrawPosition.Y = (int)(cheesePosition.Y + 0.5f);
```

- The cheese position vector is used to set the draw rectangle position of the cheese on the display
- The code needs to convert the floating point values in the vector into integers for the rectangle

NOKIA Microsoft

The story so far...

- We now have a cheese that will roll around the screen
- It will bounce off the edges when it hits them
- It has a simple physics model that simulates friction and causes the movement to slow down over time
- Now we have to get the cheese to move in the first place
- The Windows Phone does not have any provision for a gamepad or any physical buttons
- But it does have a touch panel with built in gesture support

NOKIA Microsoft

Gesture Support

Windows Phone Gesture Support

- For the ultimate in control you can get direct access to touch events from the panel
  - Up to four events can be tracked at one time
  - Each event is uniquely identified throughout its lifetime
- However, XNA games can also use built in gesture recognition
- Register an interest in a particular gesture type and then get notification when one has been performed

NOKIA Microsoft

Supported Destures

- The touch panel can detect a number of different gestures including
  - Tap
  - DoubleTap
  - Hold
  - HorizontalDrag, VerticalDrag and FreeDrag
  - Pinch
  - Flick
- The Cheese Roller game is going to use the Flick gesture

NOKIA Microsoft

Published: 8/24/2011

Registering an Interest in a Gesture

```
TouchPanel.EnabledGestures = GestureType.Flick;
```

- The game must select those gestures that it wants to use
- This can be done once at the start of the game
- I do it in the XNA `Initialise` method
- A game can also query the TouchPanel about the number of points it can track simultaneously

NOKIA Microsoft

Published: 8/24/2011

Retrieving Gestures

- The `Flick` gesture returns a `Delta` value which gives the speed of the flick
  - The divide factor of 100 is something I came up with by playtesting
  - The smaller the value, the more powerful the effect of the flick

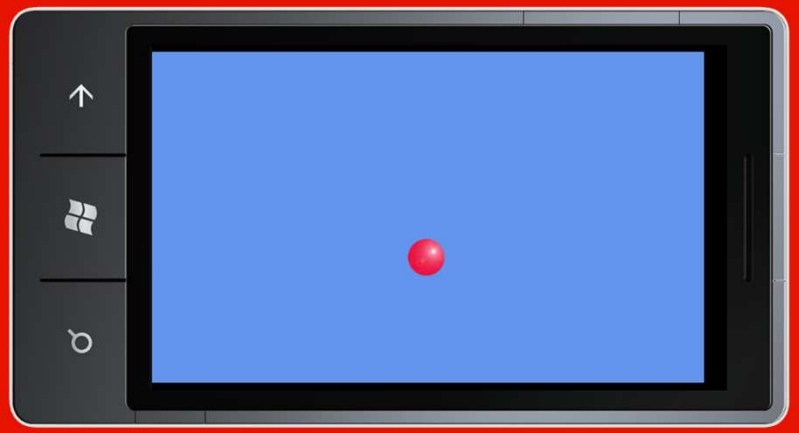
```
while (TouchPanel.IsGestureAvailable)
{
    GestureSample gesture = TouchPanel.ReadGesture();

    if (gesture.GestureType == GestureType.Flick)
        cheeseSpeed = gesture.Delta / 100;
}
```

NOKIA Microsoft

Published: 8/24/2011

Windows Phone

Demo



3: Flicking Cheese

Improving on Cheese Roller

- The obvious improvement to Cheese Roller is to add more cheese
- If we do this the physics becomes more complicated
- Fortunately there are a lot of engines out there that can help
- One of these is the Farseer engine
- It is a 2D physics engine that is very easy to use and works well on Windows Phone
- The FarSeer Physics Engine is a Codeplex based project you can download from here:

http://farseerphysics.codeplex.com/

NOKIA Microsoft

Published: 8/24/2011

Accelerometer and Motion Sensor

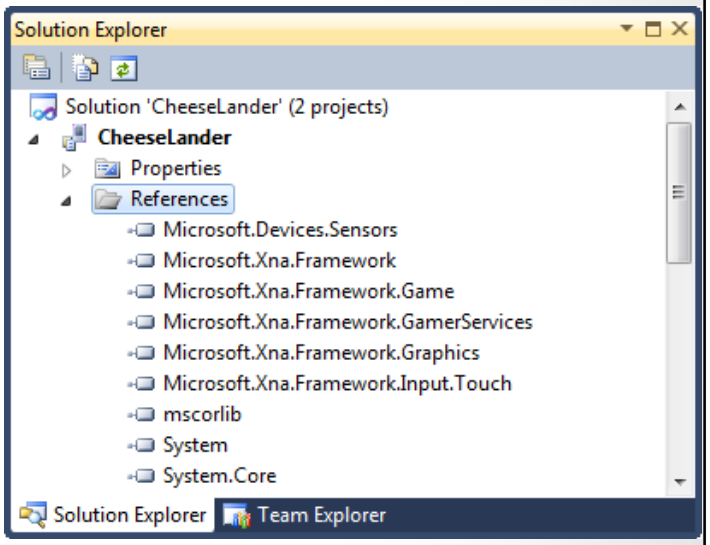
Using the Phone Accelerometer

- You can also use the accelerometer to control the behaviour of objects in your game
- The Accelerometer can measure acceleration in X, Y and Z
- You can use just the X and Y values to turn it into a replacement for a gamepad
- The values that are returned are in the same range
  - -1 to +1 in each axis

NOKIA Microsoft

Adding the Sensors Library

- The reason why the accelerometer is event driven is that XNA actually uses the same sensor interface as Silverlight
- This means that you need to include the appropriate sensor library into your program to obtain accelerometer readings
- You need to add Microsoft.Devices.Sensors to your solution to bring in the library



NOKIA Microsoft

Adding the Sensors Namespace

```
using Microsoft.Devices.Sensors;
```

- Once you have added the library you can use the Sensors objects
- Adding the namespace to your program makes the code a bit cleaner
- Note that you only have to do this for the accelerometer

NOKIA Microsoft

XNA 4.0 Accelerometer

- Unlike other XNA input devices the accelerometer in XNA 4.0 is event driven
- The accelerometer generates events when new readings are available
- You must bind a method to the event
- The method can store the settings for later use

NOKIA Microsoft

Creating an Accelerometer

```
Accelerometer accel = new Accelerometer();
accel.ReadingChanged +=
        new EventHandler<AccelerometerReadingEventArgs>
                                    (accel_ReadingChanged);
accel.Start();
```

- The above code runs in the `Initialise` method to set up the accelerometer
- It creates a new `Accelerometer`, binds an event handler to it and then starts it running

NOKIA Microsoft

Reading the Accelerometer

```
object accelLock = new object();
Vector3 accelReading;

void accel_ReadingChanged(object sender,
                          AccelerometerReadingEventArgs e)
{
    lock (accelLock)
    {
        accelReading.X = (float)e.X;
        accelReading.Y = (float)e.Y;
        accelReading.Z = (float)e.Z;
    }
}
```

NOKIA Microsoft

Using the Accelerometer

```
lock (accelLock)
{
  Vector2 cheeseAcceleration = new
                Vector2(accelReading.X, -accelReading.Y);
  cheeseSpeed = cheeseSpeed + (cheeseAcceleration / 5f);
}
```

- This code uses the accelerometer values stored by the event handler to manage the cheese movement
- Note that I use a lock to stop the event handler and the Update method from fighting over the values

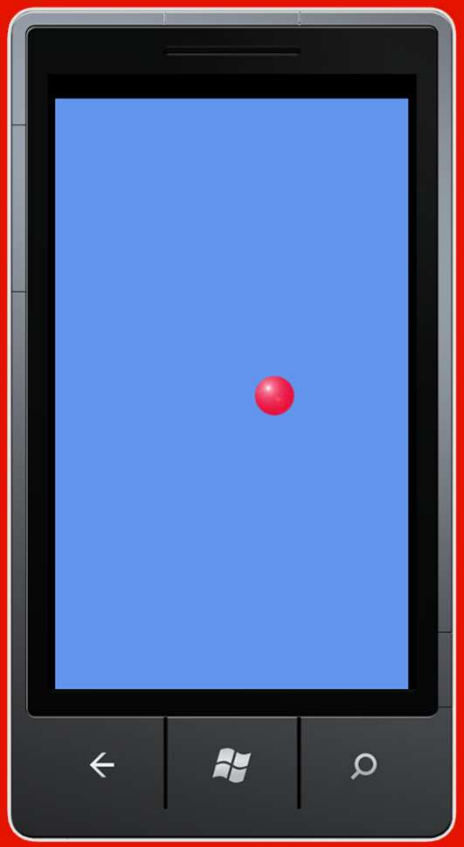
NOKIA Microsoft

Published: 8/24/2011

Windows Phone

Demo

4: Tipping Cheese



The Motion Object

- Windows Phone devices may contain a gyroscope which provides better detection of phone movement
- To make it easier to use this in conjunction with the accelerometer in the phone there is a Motion class which can be used to perform sensor fusion and combine the readings of the various devices
- It is used in the same way as the Accelerometer, but it provides a wider range of data, including the pitch and yaw of the device
- This feature is only available in Mango phones

NOKIA Microsoft

Motion Data

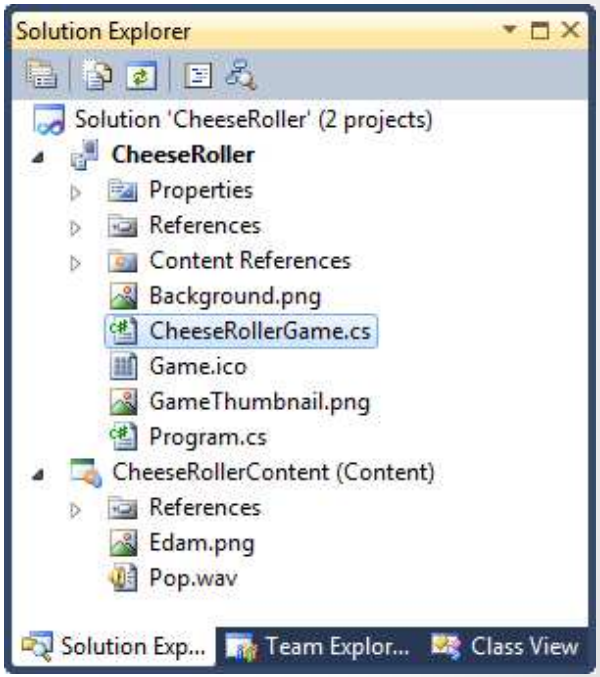
- The Motion sensor can provide:
  - Attitude
  - Pitch, Yaw and Roll
  - Rotation rate
  - Direction of Gravity
- It will use whatever sensors are available
- It can make it easier to create augmented reality applications and games

NOKIA Microsoft

Sound Effects

Adding Sound Effects

- Sound effects can be added to games as additional resources
- They are loaded into the Content project and provided to the game by the Content Manager



NOKIA Microsoft

Loading Sound Content

```
Texture2D cheeseTexture;
SoundEffect popSound;
protected override void LoadContent()
{
    ...
    cheeseTexture = Content.Load<Texture2D>("Edam");
    popSound = Content.Load<SoundEffect>("Pop");
}
```

- The LoadContent method is called to fetch game assets
- In this case it loads the cheese texture and the collision sound

NOKIA Microsoft

Playing a Sound Effect

```
if (cheeseDrawPosition.X < 0)
{
    cheeseSpeed.X = Math.Abs(cheeseSpeed.X);
    popSound.Play();
}
```

- When the cheese hits the edges the game now plays the pop sound
- Sound effects are held in memory and are played instantaneously
- The Windows Phone can play many sounds at the same time

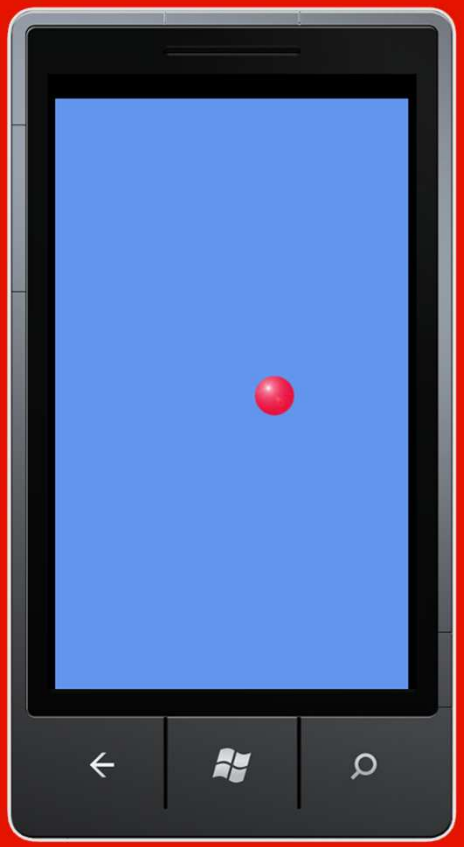
NOKIA Microsoft

Published: 8/24/2011

Windows Phone

Demo

5: Sound Effects



Music Playback

- An XNA game can also play back music files
- These can be loaded from content which is part of the game assets or from the media collection on the phone
- It is also possible to load the album artwork and use this in games
- Music is played by the `MediaPlayer` object, and is not managed in the same way as sound effects

NOKIA Microsoft

3D Games

- XNA provides full support for 3D games including the use of 3D models in gameplay
- The hardware acceleration provides very impressive, high performance graphics
- You can make use of a number of pre-build shaders that have been specifically optimised for phone use

NOKIA Microsoft

Published: 8/24/2011

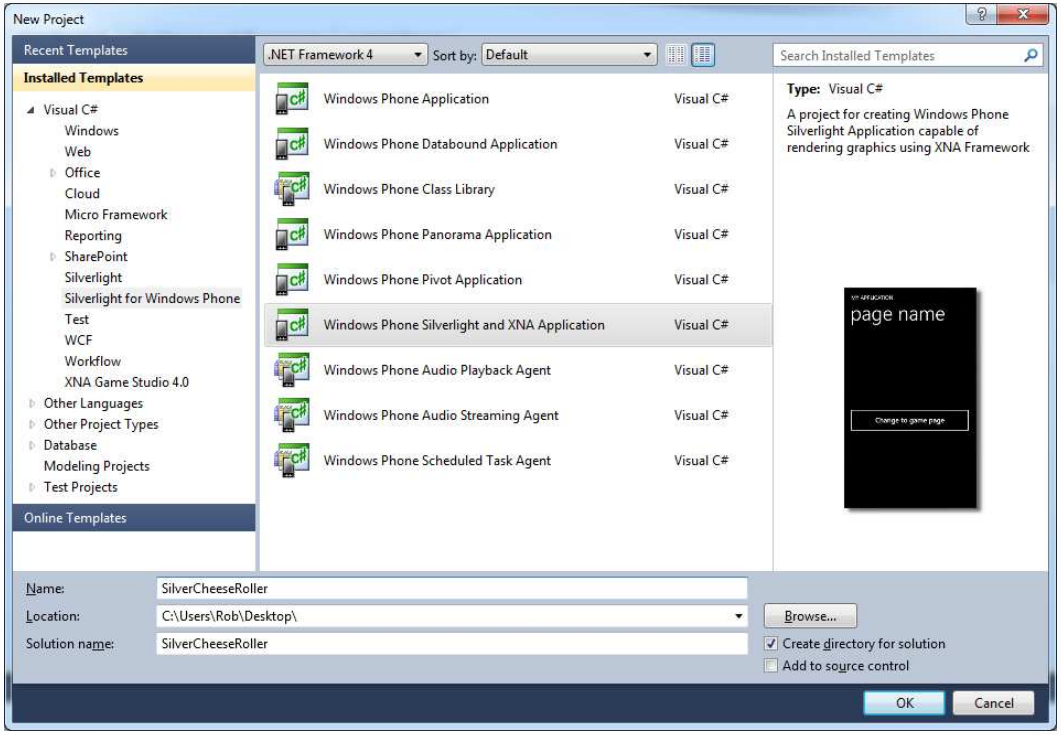
Combining XNA and Silverlight

XNA and Silverlight Combined

- It is possible to create an application that combines XNA and Silverlight
- The XNA runs within a Silverlight page
- This makes it easy to use XNA to create gameplay and Silverlight to build the user interface
- Possible to put Silverlight elements onto an XNA game screen so the user can interact with both elements at the same time

NOKIA Microsoft

Creating a Combined Project

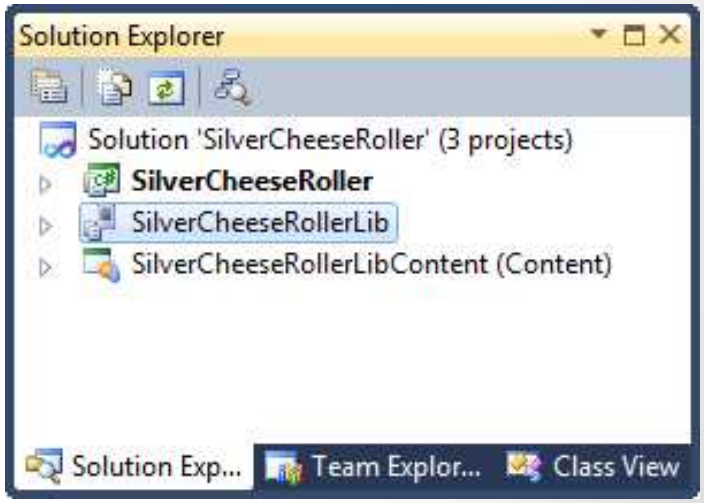


- This project contains both Silverlight and XNA elements

NOKIA Microsoft

A Combined Solution

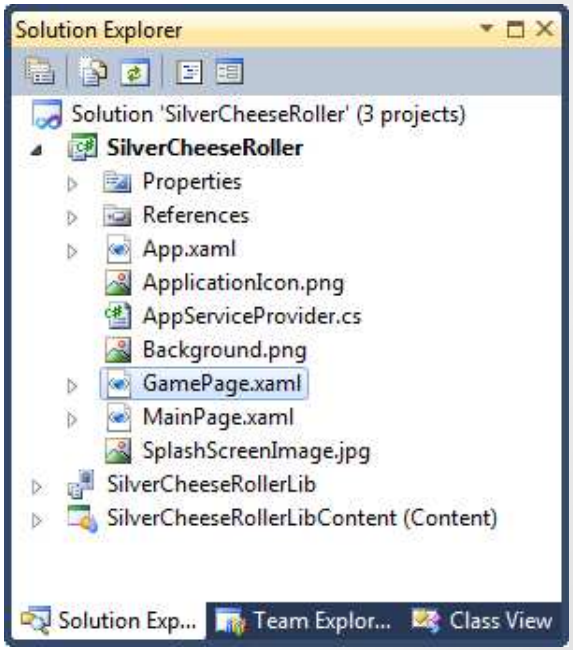
- A combined solution contains three projects
  - The Silverlight project
  - An XNA library project
  - The content project
- These are created automatically by Visual Studio



NOKIA Microsoft

The XNA GamePage

- The XNA gameplay is displayed on a specific Silverlight page that is added to the project when it is created
- When this page is navigated to the XNA game behind it will start running
- However, the Silverlight system is still active around it



NOKIA Microsoft

Published: 8/24/2011

Starting a Combined Application

- When the combined application starts the MainPage is displayed
- It contains a button that can be used to navigate to the game page
- You can build your own game menu and start screen here

NOKIA

Microsoft

Published: 8/24/2011

Navigating to the game page

```
private void Button_Click(object sender,
                          RoutedEventArgs e)
{
    NavigationService.Navigate(new Uri("/GamePage.xaml",
                               UriKind.Relative));
}
```

- This is the button event hander in MainPage.xaml.cs
- Navigating to the page will trigger the XNA gameplay to start

NOKIA Microsoft

The game page

```
protected override void OnNavigatedTo(NavigationEventArgs e)
protected override void OnNavigatedFrom(NavigationEventArgs e)
private void OnUpdate(object sender, GameTimerEventArgs e)
private void OnDraw(object sender, GameTimerEventArgs e)
```

- These are the methods on the game page that contorl gameplay
  - OnNavigatedTo will do the job of the Initialise and LoadContent methods
  - OnNavigatedFrom is used to suspend the game
  - OnUpdate is the Update method
  - OnDraw is the Draw method

NOKIA Microsoft

Published: 8/24/2011

Combining XNA and Silverlight

```
<!--No XAML content is required as the page is rendered
entirely with the XNA Framework-->
```

- The default combined project does not contain any XAML content for the XNA page
- If XAML elements are added to this page they will not be displayed without some extra code
  - The Silverlight page is rendered to a texture which is drawn in the XNA game
- We have to add the code to do this

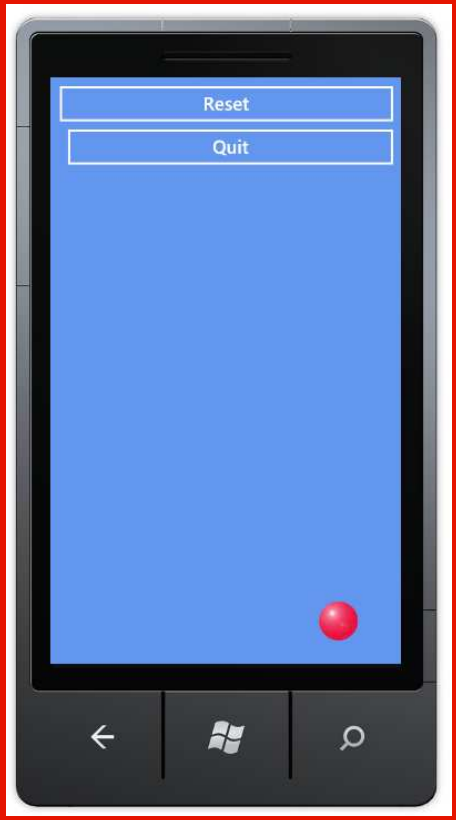
NOKIA Microsoft

Published: 8/24/2011

Windows Phone

Demo

6: Silverlight and XNA



Published: 8/24/2011

XNA and Fast Application Switching

XNA and Silverlight

- The XNA framework behaves differently with respect to Fast Application Switching
  - The messages sent to the application have different names and behave slightly differently
- The behaviour behind the application is exactly the same, it is just that the way a game responds is different
  - Games have slightly different needs, for example a game should pause when it is switched out of memory so that the player doesn't get "dumped" back into the game on resumption

NOKIA Microsoft

XNA Application Switching

- There is a useful MSDN document that explains how to persist and recover XNA games :

http://create.msdn.com/downloads/?id=634&filename=Tombstoning_WP7_Games.docx

NOKIA Microsoft

Game Design Issues

- Remember that when a user moves away from an application there is no guarantee that it will be resumed in the future
- Game players will be come very frustrated if they lose all their progress because of an incoming phone call
- For this reason the game design should contain "checkpoints" which limit the amount of progress that a player could lose when they interrupt a game
- This also means that the game has less to store when it move from one part to the next

NOKIA Microsoft

Review

- The Windows Phone platform provides a mix of resources that can be used to create compelling games
- The Touch sensor and accelerometer inputs are easy to use in XNA programs
- It is easy to add sound and music to gameplay
- It is easy to combine Silverlight and XNA games in a single application

NOKIA Microsoft