Today’s Lecture

- Interface controls
  - Toyota – Drive by Joystick
- Mixture
  - Surface Controls
  - XML Parsing
- Exams
  - Back on Wednesday
  - Will also do a mid-point feedback survey

Reminders
- Blog Post (Week)
- Project 2
- Homework 5
No more steering wheels - drive by joystick!

http://www.youtube.com/watch?v=d9_xAdZ2KxY
Surface – Here on out

- Homework 5 – 2 weeks
  - Groups of 2-4 students
  - Test it via the Surface SDK
  - Needs to run successfully on the Surface
  - Cookie v2.0
    - Surface Canvas
  - FlowDocument
    - Put on a surface item
    - Image + files from a directory
  - Library control
    - Group of videos – from a directory
Project 3
- Start next Monday
- Groups of 2-4 (recommend more)
- Two choices
  - Research viewer – CSE
  - Alumni landmark viewer
- Blend Homework 5 + Homework 6
Surface Controls

- Multiple resources
  - Wiki
    - Reference API link
    - Class listing w/WPF equivalents
      - [http://netscale.cse.nd.edu/twiki/bin/view/Edu/SysIF09RefAPISurfaceClasses](http://netscale.cse.nd.edu/twiki/bin/view/Edu/SysIF09RefAPISurfaceClasses)
  - Surface Community site
    - Home page on Surface Developer Unit
  - Surface SDK
    - Controls example

12/13/2009
Create a Surface project

- Two choices
  - Surface WPF
  - Surface XNA

- Could modify an existing project
  - using modifier
  - xmlns modifiers
Surface Controls

☐ XML Namespace
  - s namespace
  - All controls are prefixed with s:

☐ Subtle points
  - Normal controls do not respond to touch
    - Touch -> Surface sense
  - Normal controls prefixed with s:
    - s:SurfaceButton
    - s:SurfaceListBox
    - s:SurfaceTextBox
## Surface Controls

### Control list

<table>
<thead>
<tr>
<th>Control</th>
<th>Surface Control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Button</td>
<td>SurfaceButton</td>
<td>Button that responds to surface input</td>
</tr>
<tr>
<td>CheckBox</td>
<td>SurfaceCheckBox</td>
<td>Check box that responds to surface input</td>
</tr>
<tr>
<td>InkCanvas?</td>
<td>SurfaceInkCanvas</td>
<td>Ink canvas that accepts Surface input</td>
</tr>
<tr>
<td>ListBox?</td>
<td>SurfaceListBox</td>
<td>List box that responds to surface input</td>
</tr>
<tr>
<td>ListBoxItem?</td>
<td>SurfaceListBoxItem</td>
<td>An item in a surface list box that responds to surface input</td>
</tr>
<tr>
<td>PasswordBox?</td>
<td>SurfacePasswordBox</td>
<td>Text box that has input obscured such as with a password</td>
</tr>
<tr>
<td>RadioButton?</td>
<td>SurfaceRadioButton</td>
<td>Radio button for the Surface UI</td>
</tr>
<tr>
<td>ScrollViewer?</td>
<td>SurfaceScrollView</td>
<td>Slider interface for the Surface UI</td>
</tr>
<tr>
<td>Slider</td>
<td>SurfaceSlider</td>
<td>Slider interface for the Surface UI</td>
</tr>
<tr>
<td>TextBox?</td>
<td>SurfaceTextBox</td>
<td>Text box for the Surface UI that brings up a keyboard for entering text</td>
</tr>
<tr>
<td>ToggleButton?</td>
<td>SurfaceToggleButton</td>
<td>Toggle button for the Surface UI</td>
</tr>
</tbody>
</table>

Various others as well that you can instantiate

Some we have not covered as of yet but you should be able to figure them out
Surface Controls

- Normal WPF rules apply
  - Tweak visuals
    - Style
    - ControlTemplate
  - Trigger-based operations
    - Gain additional events
    - React to a finger touch
    - React to a blob touch
    - React to certain gestures
    - React to byte tags
Example

- How would we put in a Surface button?
Discussion

☐ Is there a s:Grid?
New Surface Controls

- ScatterView
  - Equivalent of a canvas + Surface
  - Container – like a Canvas
- ScatterViewItem
  - Object that we can manipulate
    - Spin
    - Zoom
    - Flick
    - Can also enable / disable properties
Example - ScatterView
Example Code

- How could we code up floating image plus text?
  - ControlTemplate
  - TargetType
  - Image
  - TextBlock
Library Controls

- LibraryStack
  - Condensed version of LibraryBar
  - LibraryStackItem, LibraryBarItem

- LibraryBar

Combine with ScatterView

ScatterView is just a container
ElementMenu

- ElementMenu
  - Circles with choices
  - ElementMenuItem
    - Header to identify

Diagram:
- CSE
- EE
- CBE
- Engineering
Tag Visualization

- Will cover in more detail later

Read Reacting to Physical Objects link on Wiki for Wed
Surface Control Summary

- Just like normal WPF controls
  - ScatterViewItem
    - Get rotation, sizing, etc. for free
  - Tag Visualizer
    - React to tags placed on the table
  - Library
    - Stack or bar of items
XML Parsing

- Lots of XAML writing
- How does it get parsed?
- Can you parse XML in your code?
  - Cover XAML parsing on Wednesday
  - Today: C# parsing
    - SAX: Simple API for XML Parsing
      - Cover today
    - DOM: Document Object Model
      - Heavier weight parsing
XML File Structure

<MyInfo>
  <Info> John Doe
    <Class Name="CSE40416" />
    <Class Name="CSE40422" />
  </Info>
  ...
</MyInfo>
<MyInfo>
  <Info> John Doe
  <Class Name="CSE40416" />
  <Class Name="CSE40422" />
  </Info>
  ...

Think of it like levels
Level 0 -> MyInfo
Level 1 -> Info
Level 2 -> Class
XMLTextReader

- Instantiate
  - System.Xml.XmlTextReader
  - Constructor -> specify file

- Read
  - MoveToContent
  - React to type of node
    - Element
    - EndElement
Example Code

```csharp
System.Xml.XmlTextReader theReader;

theReader = new XmlTextReader("test.xml");

while (theReader.Read())
{
    theReader.MoveToContent();

    if (theReader.NodeType == XmlNodeType.Element)
    {
        if (theReader.Name == "Class")
        {
            String sTemp = theReader.GetAttribute("Name");
        }
    }
}
```

Attempts to find that attribute
XML

- eXtensible
  - Can have attributes we understand
  - Can have attributes we ignore

- Map class to XML
  - static function that creates a new object from XML
  - Function to save at a point in a XML file
  - `<ListMyStuff>`
    - `<MyItem ... />`
    - `<MyItem ... />`
Questions?

- Weekly Blog
- Project 2
- Homework 5 (Posted Wed)