Lab 2: Introduction to WPF and programming with the MS Surface

Out: Thursday, 3 February 2011
Due: Thursday, 10 February 2011

Handouts: Introduction to WPF, WPF Quick Start, "How do I...?" assignments

Getting Help: Eric McCann is available to help you with Surface and programming questions. His office hours (drop-in) and email are on the syllabus.

Overview: In this lab, you will start to program the Surface using WPF. (There are also XNA exercises, but these are optional.)

What to do in this lab:

1. Read the WPF handout accompanying this lab. Pay most attention to the concept of code-behind with XAML, as you will be using this concept for this exercise.

2. Complete the WPF Quick Start accompanying this lab. (The Quick Start can also be found in the MS Surface documentation.)

3. Complete the ‘How Do I…?’ examples that are attached (they can be found in the Surface help documentation under Start > All Programs > Microsoft Surface SDK > Surface SDK Documentation – in the documentation it is under “Samples > How Do I Examples?”). Examples 3 and 4 are not required because they can only be tested on the actual Surface (they both rely on the raw camera feed which is not simulated on the Surface Simulator). Example 14 uses XNA and is optional—if you think that you want to use XNA to program your final project, it would be a good idea to complete this example. Otherwise it is not required.

After completing these examples, create a single application that demonstrates the concepts from examples 1-17. There are several ways to do this. One way to complete the assignment would be to create a ‘Controls Box’ style application which had multiple tabs or sections, each of which corresponded to a general concept as outlined below:

* ScatterView
  o Adding items programmatically
  o Displaying video
  o Hosting a user control
  o Customizing the shape of the ScatterViewItem
  o Data binding
  o Displaying and manipulating shapes
* Tag visualizations
- Creating a custom visualization
- Implementing a custom removal behavior

* LibraryBar control
  - Customizing the disabled effect
  - Grouping data

* Other
  - Continuously panning list
  - Touch-enabled user control
  - Moving UI elements with touch
  - (optional) Moving images with XNA

(Number 17, creating custom styles, can be applied to any or all of the above items.)

Another, more creative way of completing the assignment would be to think of some application in which all (or most – your application may be too busy if you were to incorporate all of them) of the above concepts could be displayed for a useful purpose. ‘Photo Paint’ is a good example of an application like this. The features of the application can be broken down as follows:

* SVI Control
  - Item 1
    - Video
    - Stop, pause, record, play buttons
    - SurfaceInkCanvas with record and playback features
  - Item 2
    - SurfaceInkCanvas with one-shot undo button and eraser button
    - Custom color selection widget
  - Item 3
    - Image
    - Draw, clear canvas, and move mode buttons
    - SurfaceInkCanvas

Wrap your control logic as suggested in example 18 to reduce CPU strain when your application is deactivated.

You do not need to incorporate the result of example 19. In that example you create a Windows Forms program that triggers random surface events when you press a button; it can be used to test a program without touching it physically. It’s a good test program to have around, but you do not need to turn in anything for it.

**What to turn in:**

Your final application will be presented in class on Thursday, 10 February. Your presentation can either be on the Surface Simulator on your own computer, or you can demo your application on the Surface in the lab, provided that you have your project copied onto the Surface before class starts. You should hand in a short write up describing the application that you will present in class.