

Windows
Phone

Touch Input in
XNA



7

Rob Miles & Andy Wigley
Windows Phone 7 Jump Start
Microsoft Corporation

Agenda

Touch input on Windows Phone


Getting Touch inputs

Iterating through touch events

Creating flicks

2

Windows Phone Jump Start

Windows
Phone

Touch Input on
Windows Phone

Touch Input

- Touch Input in XNA is really easy to use
- It is managed in terms of “touch events” which describe a particular touch action
- Each event is uniquely identified over its lifetime
- This makes tracking events really easy to do

Windows Phone Jump Start

Touch Input and Zune HD

- The touch events on Windows Phone work in exactly the same way as the Zune HD ones
- Code that works on one device will work fine on the other
- This makes the Zune HD a great place to test out your touch input routines

Windows Phone Jump Start



Getting Touch Events

Touch Events

- An XNA game can get a list of currently active touch points and their state:
 - Pressed
 - Released
 - Dragged
- Up to four input points are tracked
- Your game can request a list of currently active points

7

Windows Phone Jump Start

Getting the Touch Status

```
TouchCollection touchState = TouchPanel.GetState();
```

- The `TouchPanel1` class in an XNA provides a `GetState` method that is used to read the state of the touch panel
- This is exactly how other input devices (keyboard, gamepad, mouse) work
- The `GetState` method returns a list of events that are currently active

8

Windows Phone Jump Start

Detecting Touch Events

```
1. TouchCollection touchState = TouchPanel.GetState();  
2. if (touchState.Count > 0)  
3. {  
4.     // the screen has been touched  
5. }
```

- If the collection contains any elements it means that the screen is being touched
- For very simple interaction (e.g. start a new game) this may be sufficient

9

Windows Phone Jump Start

Iterating Touch Events

```
1. TouchCollection touchState = TouchPanel.GetState();
2. foreach (TouchLocation touch in touchState)
3. {
4.     // look at each touch event in turn
5. }
```

- Your game can look through the touch events and deal with each one in turn
- You can also access individual elements in the collection using a subscript

10

Windows Phone Jump Start

Touch Position

```
1. foreach (TouchLocation touch in touchState)
2. {
3.     Point touchPoint = new Point((int) touch.Position.X,
4.                                   (int) touch.Position.Y);
5. }
```

- The TouchLocation object exposes a Position property that gives the location on the screen
- This is given in the same pixel coordinates used to position objects

11

Windows Phone Jump Start

Touch State

```
1. foreach (TouchLocation touch in touchState) {
2.     Point touchPoint = new Point((int) touch.Position.X,
3.                                   (int) touch.Position.Y);
4.     if (touch.State == TouchLocationState.Pressed) {
5.         if (cheeseRectangle.Contains(touchPoint)) {
6.             // cheese has been pressed
7.         }
8.     }
9. }
```

- The TouchLocaton also exposes a State property that gives the state of that touch event

12

Windows Phone Jump Start

Phone Piano

Demo



Demo 1: Windows Phone Piano

13

Windows Phone Jump Start

 Windows Phone

Creating Flicks

Flicking

- You can use the touch interface to flick things around the screen
- To do this you need to record the start of a “flick” event and the end
- You can do this by using the TouchEvent ID and the state of each event

Windows Phone Jump Start

Cheese Chuck

- The idea of this game is to flick the cheese onto the bread
- Players get extra points for the more sides that the cheese bounces off
- The game will use the touch interface to flick the cheese around



16

Windows Phone Jump Start

Recording Flick State

```
1. bool cheeseTouched = false;
2. TouchLocation cheeseTouch;
3. DateTime cheeseTouchTime;
```

- The game needs to hold data about a flick
- cheeseTouched is true if we are in the middle of a flick action
- cheeseTouch is the location of the start of the flick
- cheeseTouchTime is the time the flick started

17

Windows Phone Jump Start

Starting a Flick

```
1. switch (touch.State)
2. {
3.     case TouchLocationState.Pressed:
4.         Point touchPoint = new Point((int)touch.Position.X,
5.                                     (int)touch.Position.Y);
6.         if (cheeseDrawPos.Contains(touchPoint)) {
7.             // Have got a touch event down on the cheese
8.             cheeseTouch = touch;
9.             cheeseTouched = true;
10.            cheeseTouchTime = DateTime.Now;
11.        }
12.        break;
```

- Record the start of the flick event

18

Windows Phone Jump Start

Flick Direction

```
1. case TouchLocationState.Released:
2.     if (touch.Id == cheeseTouch.Id) {
3.         Vector2 direction;
4.         Vector2 start = cheeseTouch.Position;
5.         Vector2 end = touch.Position;
6.         Vector2.Subtract(ref end, ref start, out direction);
```

- This calculates a vector that gives the direction and magnitude of the flick
- The previously stored touch event is used as the starting point

Flick Power

```
1. long magnitude = DateTime.Now.Subtract(cheeseTouchTime).Ticks;
2. Vector2 force =
3.     Vector2.Multiply(direction, 300000f / magnitude);
4. cheeseXSpeed = force.X;
5. cheeseYSpeed = force.Y;
6. cheeseTouched = false;
```

- The faster the flick the shorter the time between it starting and it ending
- We use this to calculate the flick magnitude
 - Note the “fiddle factor” of 300000f

Hot Cheese

```
1. if (cheeseTouched) {
2.     spriteBatch.Draw(cheeseTexture, cheeseDrawPos,
3.         cheeseActiveColor);
4. }
5. else {
6.     spriteBatch.Draw(cheeseTexture, cheeseDrawPos,
7.         cheeseInactiveColor);
8. }
```

- When the cheese is being flicked I change the draw colour to red to highlight it

Cheese Friction

1. // Add some friction to slow the cheese down
2. cheeseXSpeed *= friction;
3. cheeseYSpeed *= friction;

- The cheese needs to slow down once it has been flicked
- The value of friction is 0.9f which gives a reasonable slowdown
- By changing the value I can make the game easier or harder

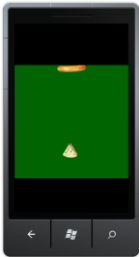
22

Windows Phone Jump Start

Cheese Flicker

Demo

Demo 2: Cheese Flick



23

Windows Phone Jump Start

Caching Touch Events

- Each time you read the touch panel status you get the current state
- This means that if several different game components want to use the touch panel your game will need to hold a local copy of the event list
- This copy can then be used during that game update

24

Windows Phone Jump Start

© 2010 Microsoft Corporation. All rights reserved. Microsoft, Windows, Windows Vista and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries. The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.

8

Review

Windows Phone and the Zune HD have the same touch interface

Up to 4 touch events can be tracked at once

Each touch event has a unique ID and a status

Your game can detect when a touch position has moved

The XNA touch panel only works on Fridays

25

Windows Phone Jump Start

Review

Windows Phone and the Zune HD have the same touch interface

Up to 4 touch events can be tracked at once

Each touch event has a unique ID and a status

Your game can detect when a touch position has moved

The XNA touch panel only works on Fridays

26

Windows Phone Jump Start

Review

Windows Phone and the Zune HD have the same touch interface

Up to 4 touch events can be tracked at once

Each touch event has a unique ID and a status

Your game can detect when a touch position has moved

The XNA touch panel only works on Fridays

27

Windows Phone Jump Start

Review

Windows Phone and the Zune HD have the same touch interface

Up to 4 touch events can be tracked at once

Each touch event has a unique ID and a status

Your game can detect when a touch position has moved

The XNA touch panel only works on Fridays

28

Windows Phone Jump Start

Review

Windows Phone and the Zune HD have the same touch interface

Up to 4 touch events can be tracked at once

Each touch event has a unique ID and a status

Your game can detect when a touch position has moved

The XNA touch panel only works on Fridays

29

Windows Phone Jump Start

Review

Windows Phone and the Zune HD have the same touch interface

Up to 4 touch events can be tracked at once

Each touch event has a unique ID and a status

Your game can detect when a touch position has moved

The XNA touch panel only works on Fridays

30

Windows Phone Jump Start

Coming Up Next...

Using the Guide for text input

Controlling media playback with XNA

31

Windows Phone Jump Start



Windows

Phone

© 2010 Microsoft Corporation. All rights reserved. Microsoft, Windows, Windows Vista and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries.

The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation.

MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.

© 2010 Microsoft Corporation. All rights reserved. Microsoft, Windows, Windows Vista and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries. The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.

11