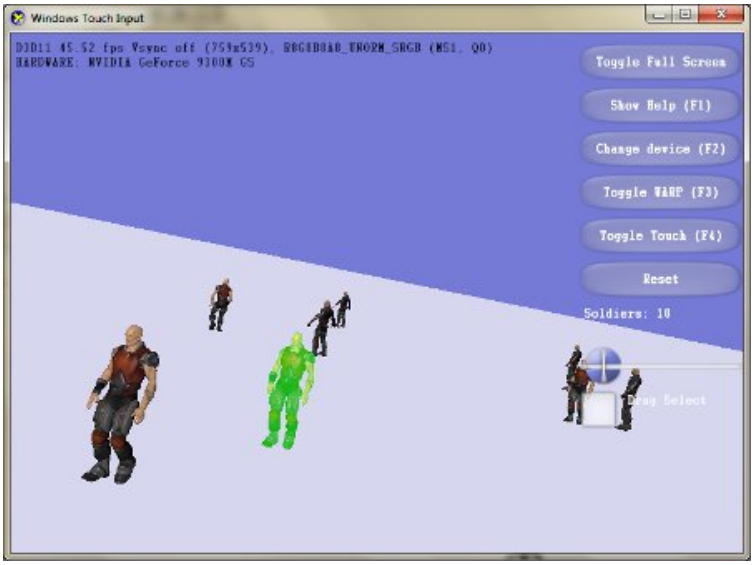
## WindowsTouch11 Sample

Collapse All

This sampls demonstrates how to use the Windows Touch APIs in a game-like scenario.




## Path

| | |
|---|---|
| **Source** | SDK root\Samples\C++\Direct3D11\WindowsTouch11 |
| **Executable** | SDK root\Samples\C++\Direct3D11\Bin\x86 or x64\WindowsTouch11.exe |

## Sample Requirements

This sample requires the Windows 7 SDK and windows multi-touch device. Use the touch screen to control the sample:

- Touch a unit to select.
- Touch a spot on the ground plane to issue commands.
- Rotate the camera by placing 2 fingers on the screen and dragging.
- Pinch the screen to zoom in.
- If the **Drag Select** box is checked, multiple units can be selected by dragging a finger across them.

## Details

Game developers will want to use the touch messages to create their own gestures for the added flexibility. Writing a touch application is as easy as registering a touch window and then acting based on the touch messages

```
rt = RegisterTouchWindow( hWnd, 0 );
```

WindowsTouch11 listens for WM_TOUCH messages in the **MsgProc** function. When a touch is received, the number of touch inputs and their values are queried with **GetTouchInputInfo**.

```
if ( uMsg == WM_TOUCH ) {
  INT nInputs = wParam & 0xffff;
  UINT handle = LOWORD( wParam );
  if ( nInputs > g_iTouchInputArraySize ) {
    g_iTouchInputArraySize = nInputs;
    SAFE_DELETE_ARRAY( g_pInputs );
    g_pInputs = new TOUCHINPUT[nInputs];
  }

  if ( GetTouchInputInfo( ( HTOUCHINPUT )lParam, nInputs, g_pInputs, sizeof( TOUCHINPUT ) ) )
  {
```

**GetTouchInputInfo** fills an array of **TOUCHINPUT** structures.

```
    TOUCHINPUT ti = g_pInputs[0];
```

To determine if the touch input is down or up, the *dwFlags* are tested against the TOUCHEVENTF flags.

```
    if ( m_UIstate == UI_NO_TOUCH_CONTROL ||
         m_UIstate == UI_UNIT_SELECTION_TOUCH_CONTROL ) {
```

```
  if ( nInputs ==  1) {

    if ( ti.dwFlags & TOUCHEVENTF_DOWN )
    {
      TouchDown ( ti.x*g_fPixelScaler - g_Window.left, ti.y*g_fPixelScaler - g_Window.top );
    }
    else if ( ti.dwFlags & TOUCHEVENTF_MOVE )
    {
      TouchMove ( ti.x*g_fPixelScaler - g_Window.left, ti.y*g_fPixelScaler - g_Window.top );
    }
    else if ( ti.dwFlags & TOUCHEVENTF_UP )
    {
      TouchUp ( ti.x*g_fPixelScaler - g_Window.left, ti.y*g_fPixelScaler - g_Window.top );
    }
  }
}
```