Win32 Windows

Jim Fawcett

CSE681 – SW Modeling & Analysis

Fall 2011

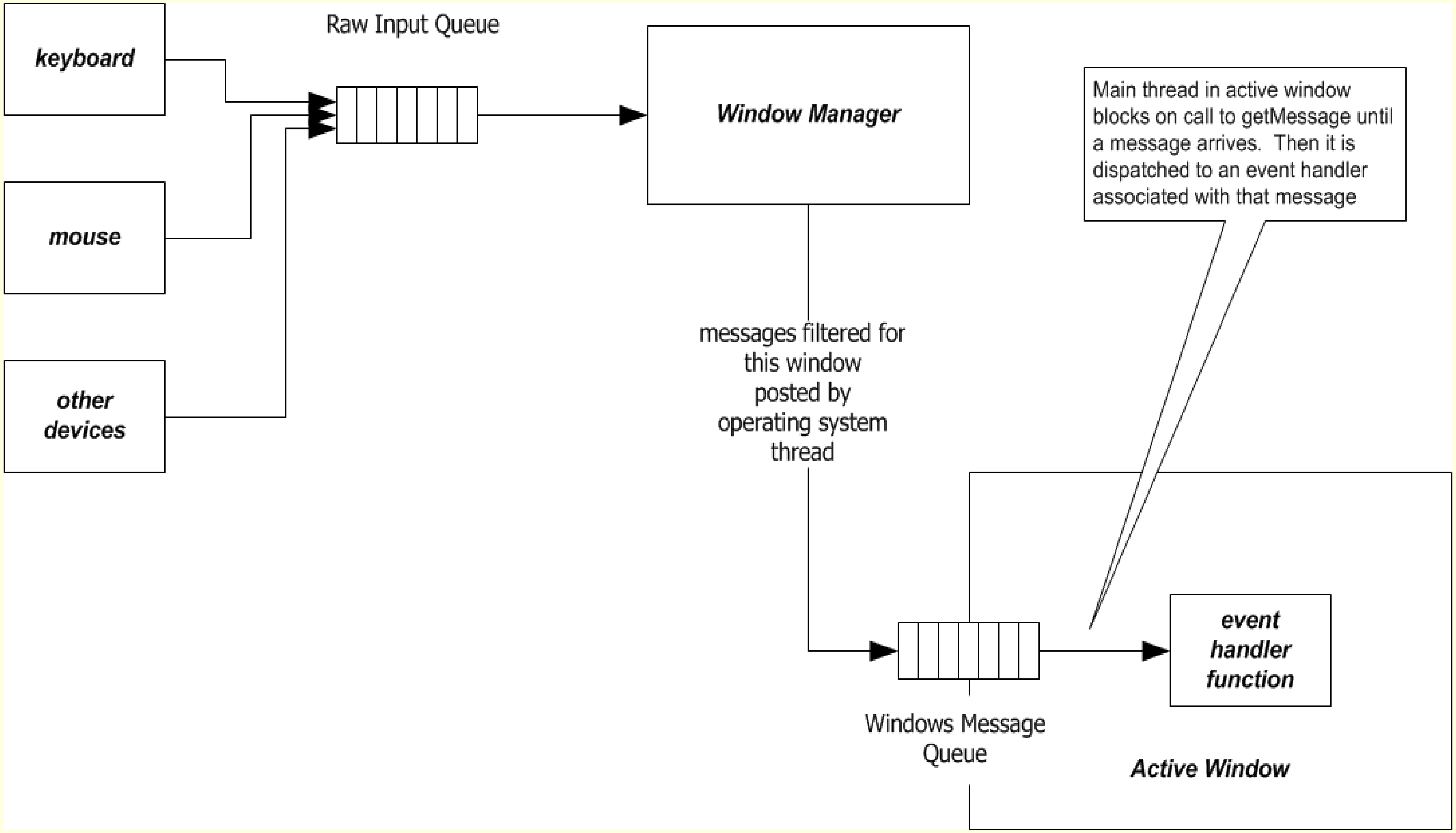
Topics

- References
- Introduction
- Simplest Win32 Windows Program
- Structure of Windows Application
- Controls
- Dialogs

References

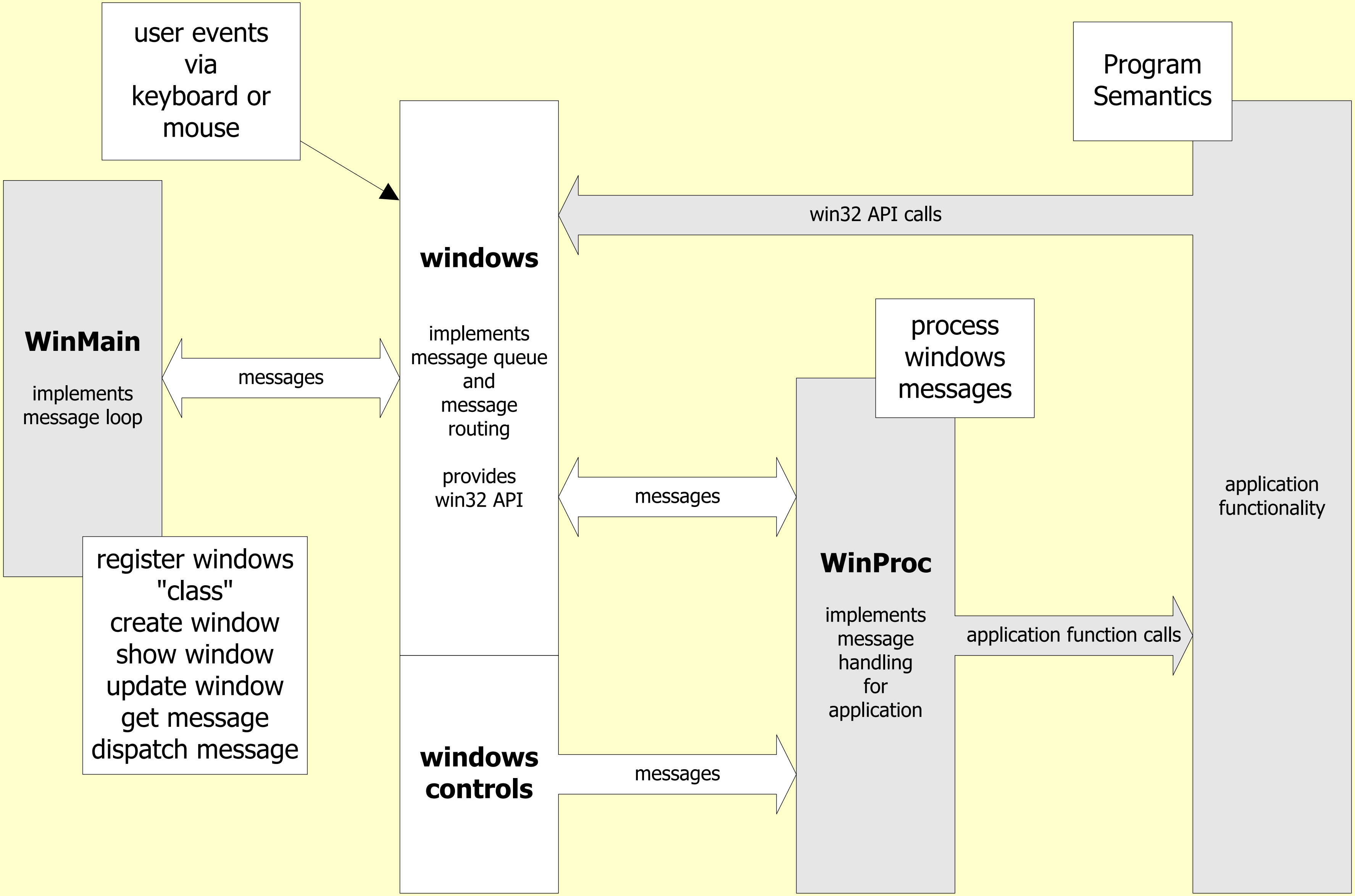
- Catch22 Tutorials
- WinProg
- Windows_Programming
- Orphan Tutorial
- Simple Windows Code Fragments
- Subclassing
- Windows and Messages
- Dialogs
- Controls
  - User Controls
  - Common Controls
- Windows Shell
  - Browser Control

Windows Event Processing



Main thread in active window blocks on call to getMessage until a message arrives. Then it is dispatched to an event handler associated with that message

Classic Windows Processing



user events via keyboard or mouse

Program Semantics

WinMain

implements message loop

messages

windows

implements message queue and message routing

provides win32 API

win32 API calls

process windows messages

application functionality

messages

WinProc

implements message handling for application

application function calls

register windows "class"
create window
show window
update window
get message
dispatch message

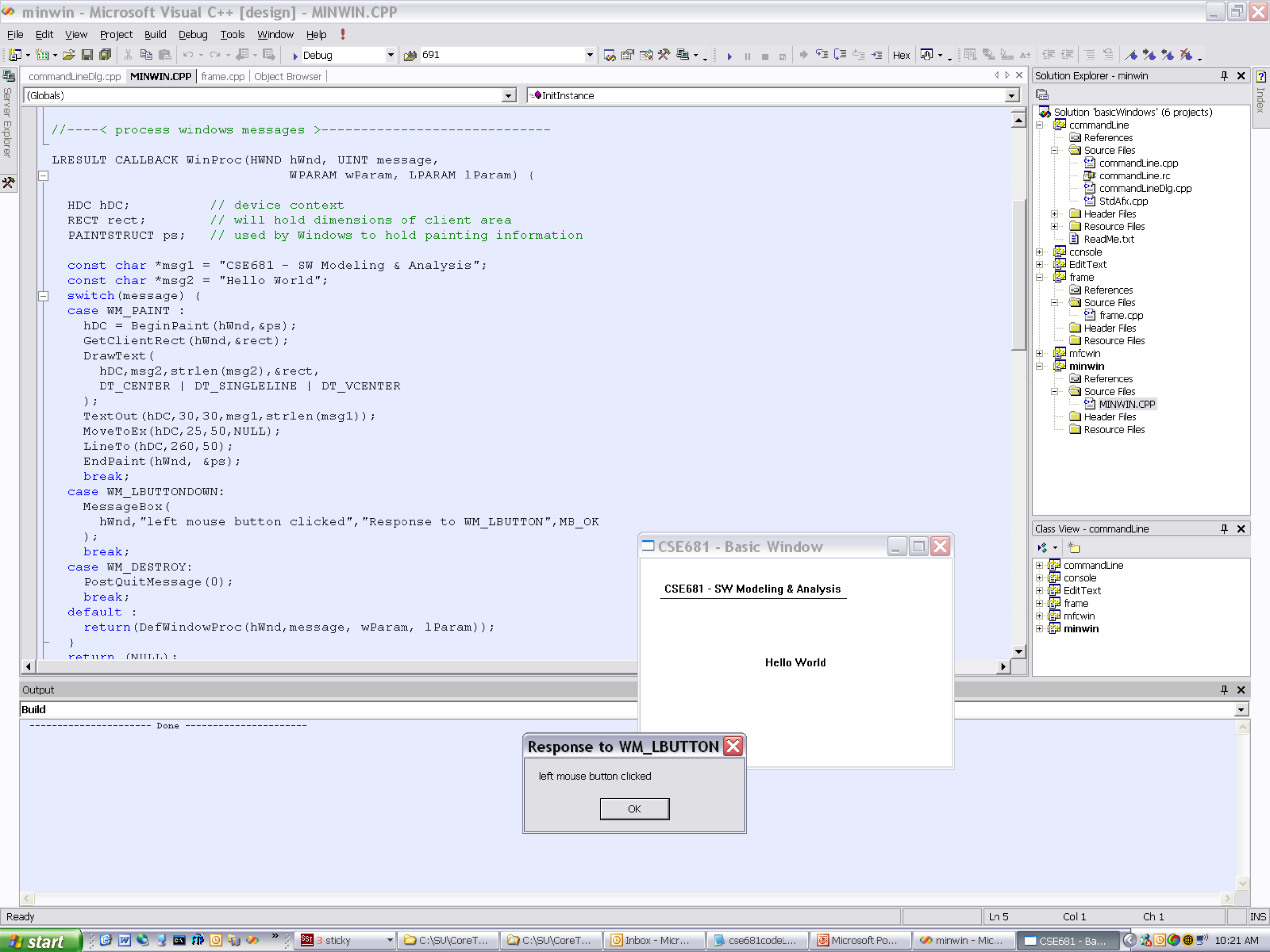
windows controls

messages

The Simplest Classical Windows Program

- Output from the MINWIN program is shown on the next slide. The program:
  - Creates a frame window
  - Paints a fixed text message in the window
  - Reacts to left mouse button clicks by creating a MessageBox
- You will find the program code, minwin.cpp, and project solution, minwin.sln in the folder MINWIN.

  www.ecs.syr.edu/faculty/fawcett/handouts/CoreTechnologies/WindowsProgramming/code/basicWindows/Minwin



minwin - Microsoft Visual C++ [design] - MINWIN.CPP

```
//----< process windows messages >-----------------------------

LRESULT CALLBACK WinProc(HWND hWnd, UINT message,
                              WPARAM wParam, LPARAM lParam) {

  HDC hDC;            // device context
  RECT rect;          // will hold dimensions of client area
  PAINTSTRUCT ps;     // used by Windows to hold painting information

  const char *msg1 = "CSE681 - SW Modeling & Analysis";
  const char *msg2 = "Hello World";
  switch(message) {
  case WM_PAINT :
    hDC = BeginPaint(hWnd,&ps);
    GetClientRect(hWnd,&rect);
    DrawText(
      hDC,msg2,strlen(msg2),&rect,
      DT_CENTER | DT_SINGLELINE | DT_VCENTER
    );
    TextOut(hDC,30,30,msg1,strlen(msg1));
    MoveToEx(hDC,25,50,NULL);
    LineTo(hDC,260,50);
    EndPaint(hWnd, &ps);
    break;
  case WM_LBUTTONDOWN:
    MessageBox(
      hWnd,"left mouse button clicked","Response to WM_LBUTTON",MB_OK
    );
    break;
  case WM_DESTROY:
    PostQuitMessage(0);
    break;
  default :
    return(DefWindowProc(hWnd,message, wParam, lParam));
  }
  return (NULL);
```

CSE681 - Basic Window

CSE681 - SW Modeling & Analysis

Hello World

Response to WM_LBUTTON

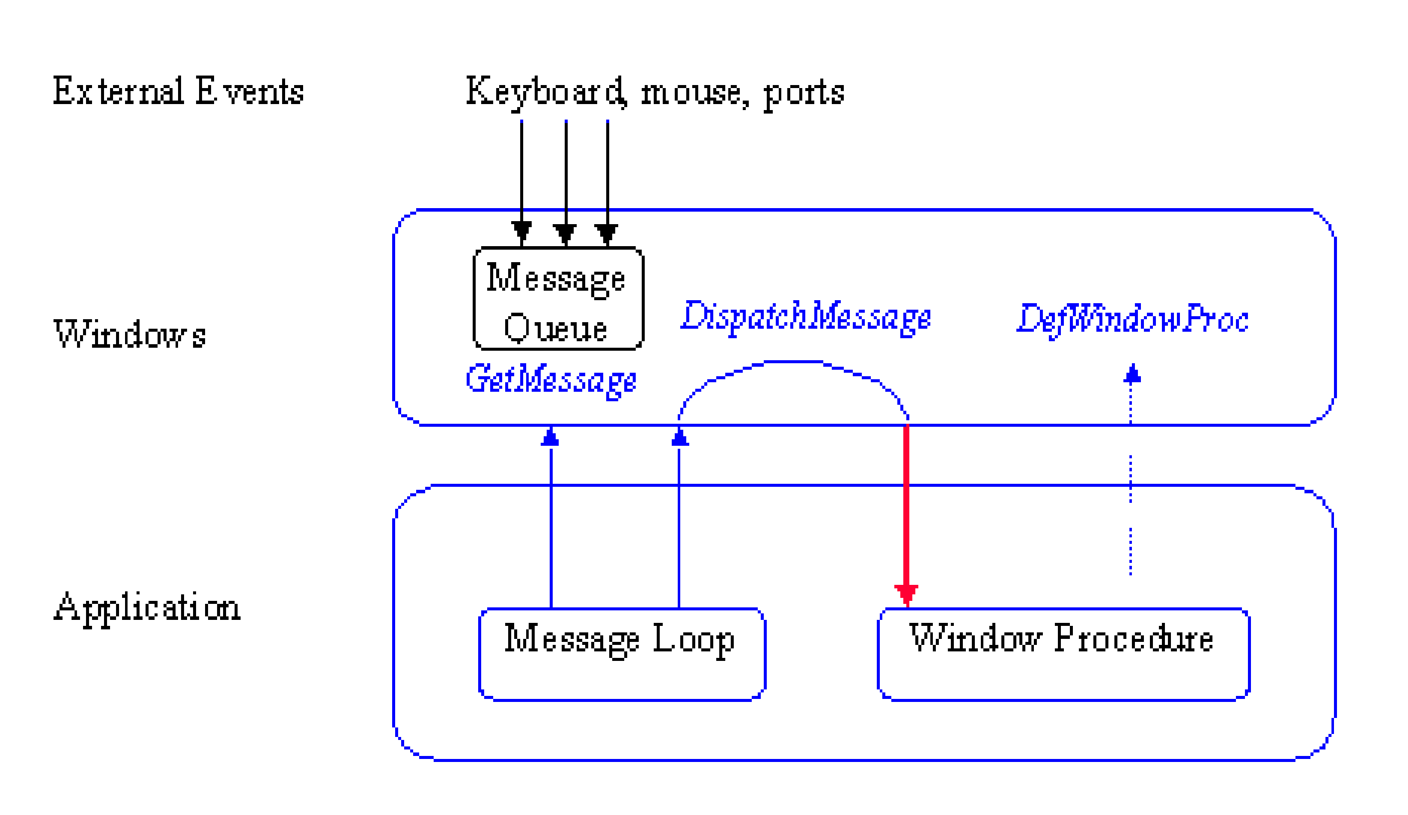
left mouse button clicked

OK

Output — Build

```
--------------------- Done ---------------------
```

Structure of a Win32 Windows Program



Data Types

Windows Type	Equivalent
LPVOID	void*
LPSTR	wchar_t*
TCHAR	char or wchar_t
LPTSTR	char* or wchar_t*
DWORD	unsigned 32 bit integer
WORD	unsigned 16 bit integer
HANDLE	Windows handle
HINSTANCE	Handle to program instance
HMENU	Menu handle
WPARAM	16 bit parameter
LPARAM	32 bit parameter

Controls

Control	Library	Comments
User Controls	Windows.h User32.dll	Child window
Common Controls	Windows.h CommCtl.dll	COM based
Browser Control	MSHTML.h SHDOCVW.dll MSHTML.dll	COM based
Common Dialogs	Commdlg.h Comdlg32.dll	COM based

User Controls

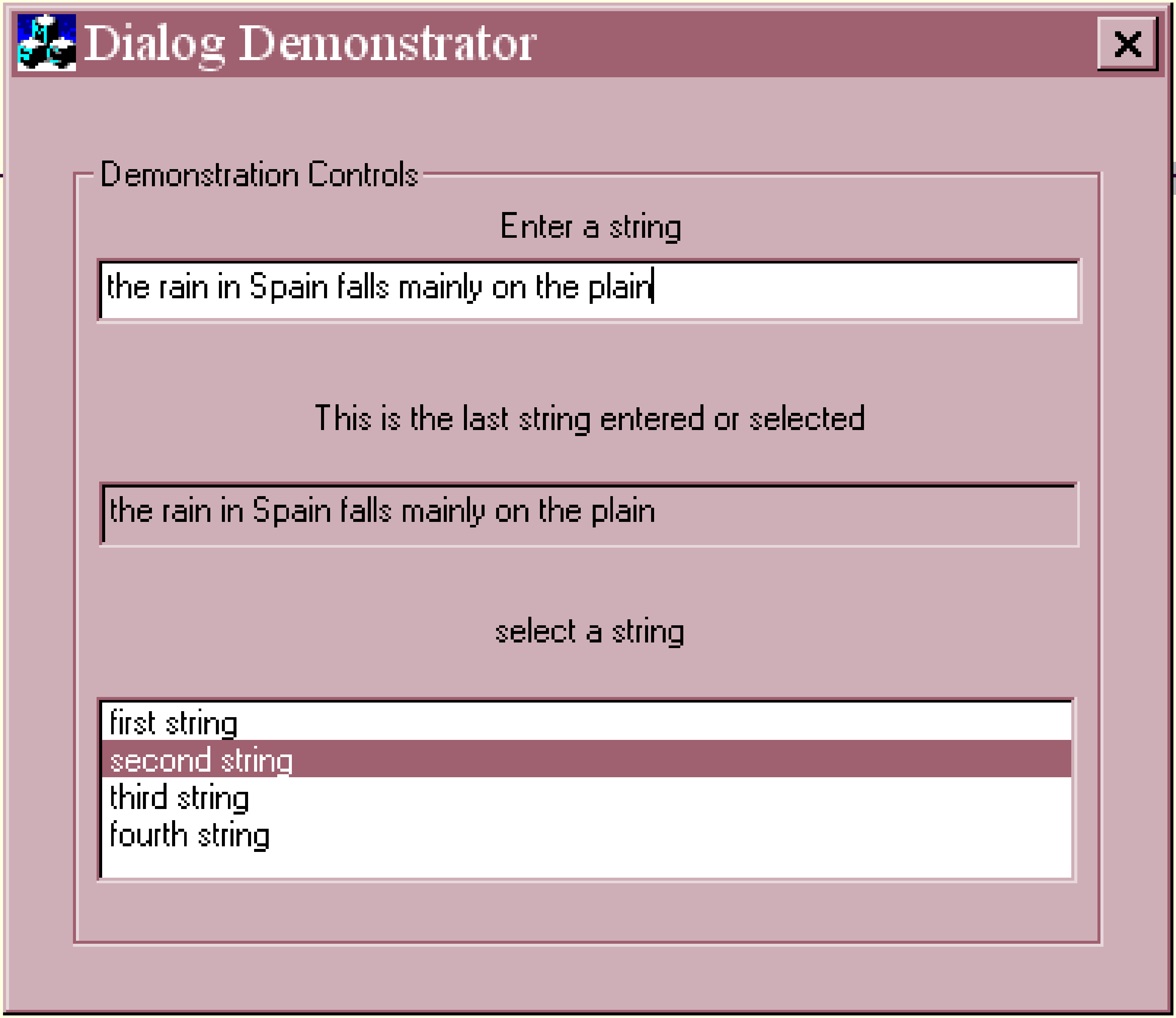
- User controls are:
  - Button, ComboBox, Edit, ListBox, RichEdit, RichEdit_Class, Scrollbar, Static
- Most used messages sent to control:
  - WM_GETTEXT, WM_GETTEXTLENGTH, WM_SETTEXT, WM_SETFONT, WM_SETFOCUS, WM_KILLFOCUS
- Notifications – messages sent by control to parent:
  - WM_COMMAND
    - LPARAM handle to control window
    - WPARAM
      - HIWORD(wParam): control-defined notification code
      - LOWORD(wParam): control Identifier

Common Controls

- Common Controls are:
  - Animation, Buttons, Calendar, ComboBoxes, Date and Time Picker, Drag List Boxes, Flat Scroll Bars, Image List, IP Address, List Boxes, ListView, Progress Bar, Property Sheets, Rebar, Rich Edit, Scroll Bars, Status Bars, Toolbar, ToolTip, Trackbar, TreeView, Up-Down
  - Messages: WM_NOTIFY, …
  - Notifications: NM_CHAR, NM_CLICK, …

Dialogs

- Dialogs come in two flavors:
  - Modal Dialogs
    - Has own message loop and service thread
    - Does not relinquish focus until closed
    - Results are retrieved at closure
  - Non-Modal Dialogs
    - Runs on parent's thread
    - Provides callback function for interaction with parent
    - Does not hold onto focus
    - Results may be retrieved while dialog is open



Dialog Demonstrator

Demonstration Controls

Enter a string

the rain in Spain falls mainly on the plain

This is the last string entered or selected

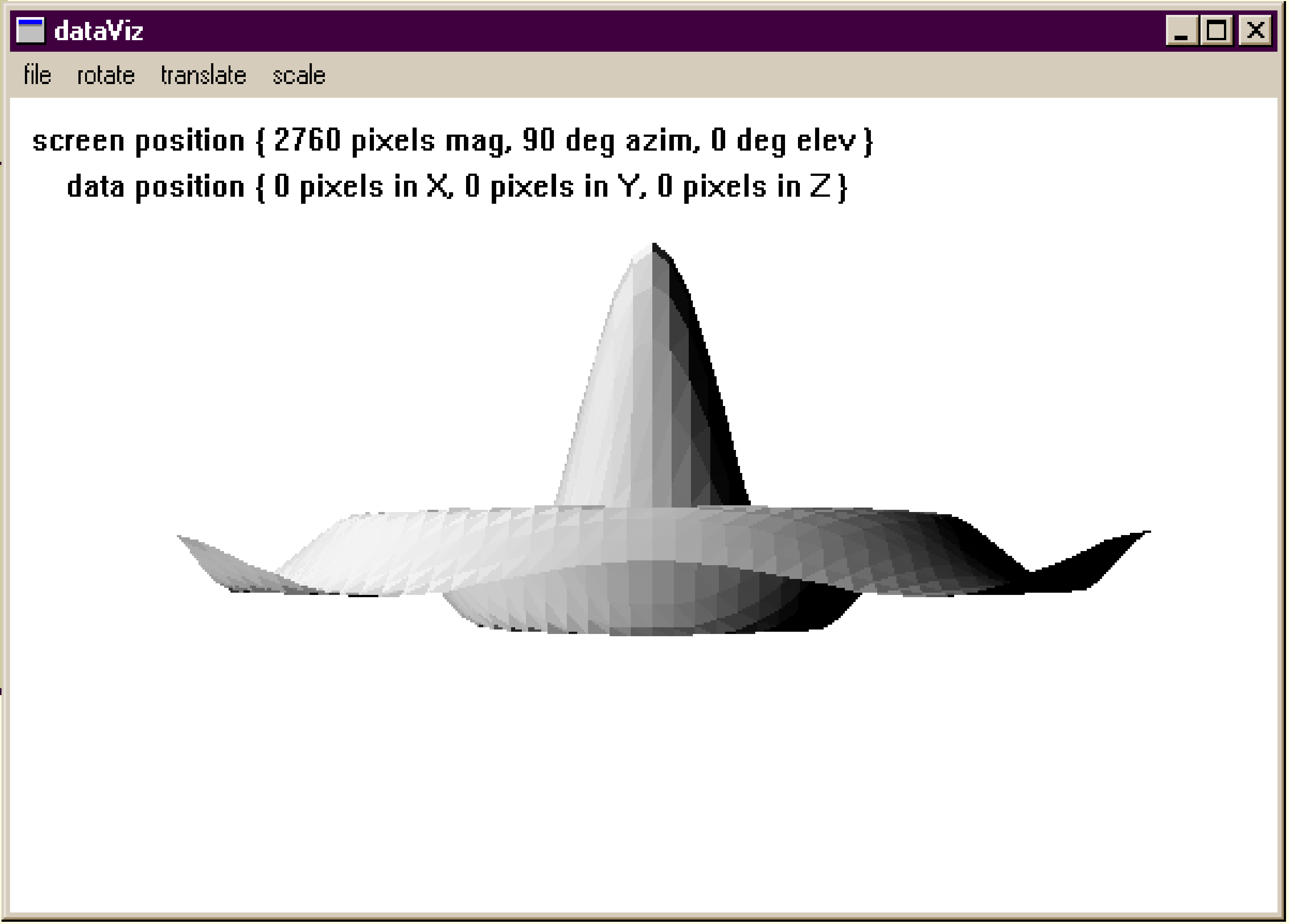
the rain in Spain falls mainly on the plain

select a string

- first string
- second string
- third string
- fourth string

Frames

- Frame Window:
  - Has canvas that covers most of the window, called the client area:
    - Canvas typically used for drawing and text, but can host controls as well.
    - Designer has to provide all hosting infrastructure at each control site on canvas.
  - Frame supports menus, toolbars, and status bars.
  - Can host Dialogs, created by menu clicks.
  - Visual Studio provides resource editors for menus and dialogs, even for Win32 applications.
- Windows Functions

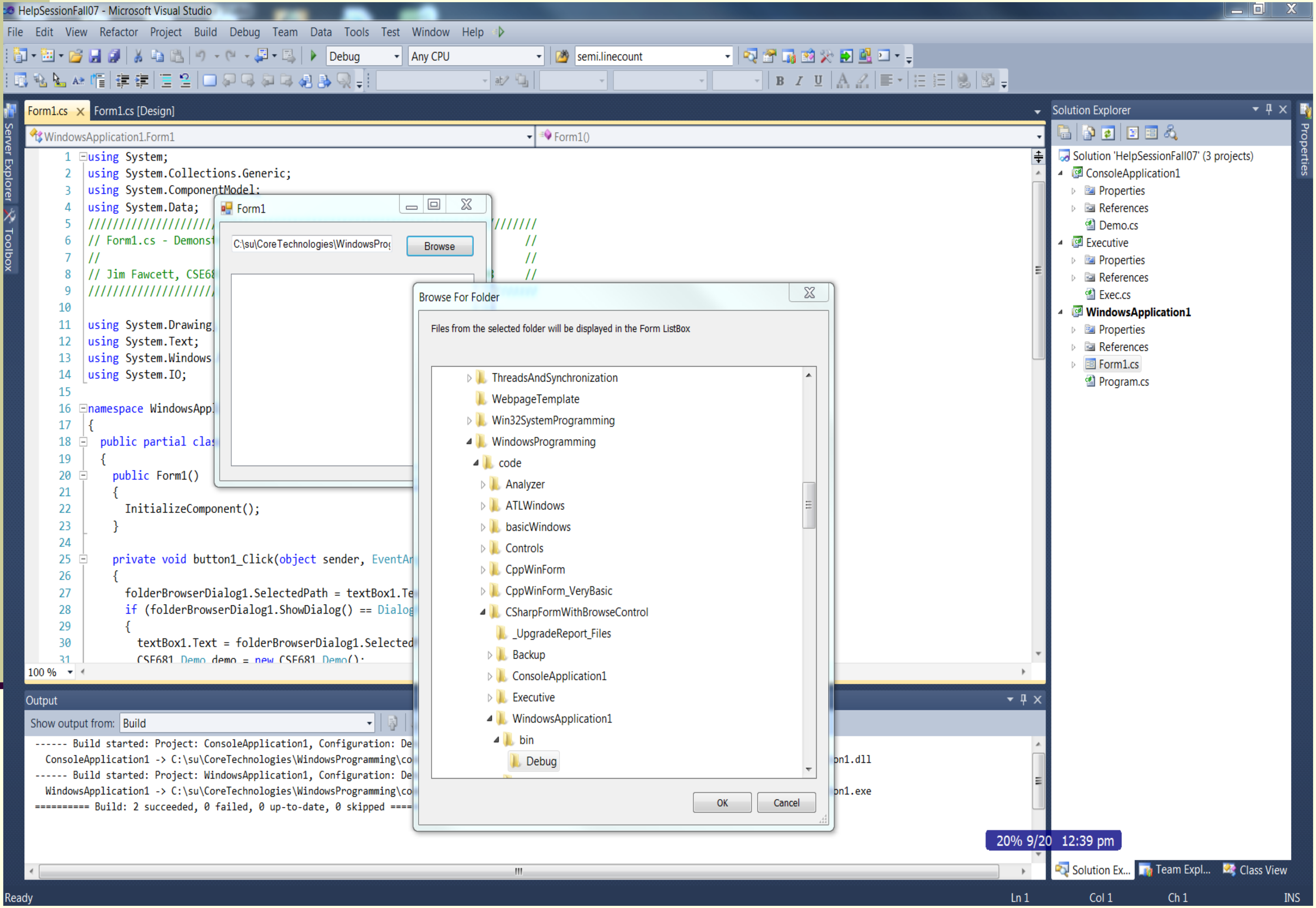


screen position { 2760 pixels mag, 90 deg azim, 0 deg elev }

data position { 0 pixels in X, 0 pixels in Y, 0 pixels in Z }

Common Dialogs

- Common Dialogs are:
  - Color, Find, Font, OpenFile, PageSetup, Print, PrintPropertySheet, Replace, SaveFile
  - Messages
  - Notifications



```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
////////////////////
// Form1.cs - Demons
//
// Jim Fawcett, CSE68
////////////////////

using System.Drawing
using System.Text;
using System.Windows
using System.IO;

namespace WindowsAppl
{
  public partial clas
  {
    public Form1()
    {
      InitializeComponent();
    }

    private void button1_Click(object sender, EventAr
    {
      folderBrowserDialog1.SelectedPath = textBox1.Te
      if (folderBrowserDialog1.ShowDialog() == Dialog
      {
        textBox1.Text = folderBrowserDialog1.Selected
        CSE681.Demo demo = new CSE681.Demo();
```

C:\su\CoreTechnologies\WindowsProg

Files from the selected folder will be displayed in the Form ListBox

```
------ Build started: Project: ConsoleApplication1, Configuration: De
  ConsoleApplication1 -> C:\su\CoreTechnologies\WindowsProgramming\co                    on1.dll
------ Build started: Project: WindowsApplication1, Configuration: De
  WindowsApplication1 -> C:\su\CoreTechnologies\WindowsProgramming\co                    on1.exe
========== Build: 2 succeeded, 0 failed, 0 up-to-date, 0 skipped ====
```

20% 9/20 12:39 pm

End of Presentation