

Construct an ActiveX control that displays the following window. The border is required and the horizontal and vertical lines should bisect the window.



Use the following steps to create your control.

1. Create a new control. (Don't modify any of your previous efforts, start from scratch!)
2. Complete steps 2 through 7 of project 3. Cut and paste is **STRONGLY RECOMMENDED**.
3. Instead of fattening the lines, we will create an event that tells which small square the mouse was clicked in.
4. Go to the *View* Menu and select *Class Wizard*. Go to the fourth tab, *ActiveX Events*.
5. Make sure your class CxCtrl is selected in the Class Name box, where *x* is the name of your project.
6. Click on *Add Event*.
7. In the *External Name* box, type the word *Box*.
8. In the Parameters box, type the word *Position* in the *Name* column and *LPCTSTR* in the *Type* column. (Typing two consecutive *l*'s will make *LPCTSTR* appear.)
9. Click OK.
10. Now go to the first tab, *Message Maps*.
11. Make sure your class CxCtrl is selected in the Class Name box, where *x* is the name of your project.
12. In the *Messages:* box, scroll down and find the line that reads `WM_LBUTTONDOWN`. (This box contains lots of stuff!) Click on this line to select it.
13. Click on the *Add Function* button.
14. In the Member Functions box, click on the line that begins "OnLButtonDown" to make sure it is selected, then click on *Edit Code*.
15. Replace the TODO line with your code. (Do not delete the other line!) We must do several things here. First we must get the rectangle defining the screen-coordinates of our control's window. Then we must determine where the mouse click occurred, then we must fire an event identifying the square that was clicked.
16. The following lines of code will extract the window's rectangle.

```
RECT rcBounds;  
CWnd::GetClientRect(&rcBounds);
```
17. The following lines of code will determine the mouse X and Y coordinates.

```
long MouseX = point.x;  
long MouseY = point.y; // use point.x, point.y directly if you wish
```
18. Next, compute `VertMid` and `HorzMid` as in Program 3.

19. The following code will fire the event.

```
if (MouseX >= rcBounds.left && MouseX < VertMid
    && MouseY >= rcBounds.top && MouseY < HorzMid)
{
    FireBox("Upper Left");
}
else if ( ... )
{
    FireBox("Lower Left");
}
... (Test for Upper Right and Lower Right)
else
{
    FireBox("Unknown Location");
}
```

20. Compile the control, and test it in Visual Basic. Have Visual Basic display the message returned by the event. Make sure to click on the edges of the window.

21. Turn in "The Usual"