

# P3.24

Frank Mock

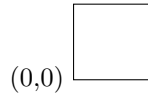
September 26, 2011

## 1 Specification

Exercise P3.24. Write a program that draws a square with corner points  $(0, 0)$  and  $(1, 1)$ . Prompt the user for a mouse click. If the user clicked inside the square, then show a message Congratulations. Otherwise, show a message You missed.

## 2 Design

Assume that the points forming the square have dimensions:



and that the user's point is in variable  $p$ . Display the message to the user someplace outside the square. For example at some point  $(2,2)$ .

## 3 Implementation

"p3\_24.cpp" 1≡

```
< Include Files 3c >
int ccc_win_main()//authors entry point that calls name
{
< Draw a Square 2a >
< Get the point where the user clicked 2b >
< Determine if that point is in the square 3a >
return 0;
}
◇
```

The authors graphics library includes a `Point` class and a `Line` class which allow points and lines to be drawn on the screen. A `Point` has an `x-` and a

y-coordinate. For example, `Point(1,3)` is a `Point` with x-coordinate 1 and y-coordinate 3. Two points can be joined by a line, represented by a `Line` object that is constructed from two `Point` objects, it's start point and end points.

```
Point p(1, 3);
Point q(4, 7);
Line s(p, q);
```

Both the `Point` and `Line` class implement the member function `move(x, y)` which changes position of an object, moving the entire object by the x and y units specified.

```
s.move(1, 0)
```

This moves the line `s` 1 unit in the x direction (to the right). `cwin` is a window object used to display graphic objects such as `Line` and `Point` to the screen.

```
cwin << s;
```

*⟨ Draw a Square 2a ⟩*  $\equiv$

```
//draw a 1x1 unit square
Point top_left(0, 1);
Point top_right(1, 1);
Point bottom_left(0, 0);

Line horizontal(top_left, top_right);
Line vertical(top_left, bottom_left);

cwin << horizontal << vertical;

horizontal.move(0, -1);
vertical.move(1, 0);

cwin << horizontal << vertical;
◇
```

Fragment referenced in 1.

In the author's library, there is a function named `get_mouse` that returns the point where the left-button of the mouse was clicked. Invoke it using the `cwin` object. Store the point in variable `p`

*⟨ Get the point where the user clicked 2b ⟩*  $\equiv$

```
Point p = cwin.get_mouse("Try to click inside the square.");
◇
```

Fragment referenced in 1.

There is a member function named `verb—getx()` | and `gety()` | to retrieve the coordinates from a point object. (e.g. `p`)

$\langle \text{Determine if that point is in the square 3a} \rangle \equiv$

```
//assign x and y coordinates to double variable
double x = p.get_x();
double y = p.get_y();
//determine if the user cliciked inside the square
if(x <= 1 && x >= 0 && y <=1 && y >= 0)
{
    cwin << p << Message(p, "Congratulations, You did it!");
}
else
{
    cwin << p << Message(p, "Sorry, You missed!");
}
◇
```

Fragment referenced in 1.

"p3\_24.bat" 3b $\equiv$

```
g++ -mwindows -I C:\C++_Programs\bigc2_sourcecode\cccfiles -o p3_24 p3_24.cpp ^
C:\C++_Programs\bigc2_sourcecode\cccfiles\ccc_msw.cpp ^
C:\C++_Programs\bigc2_sourcecode\cccfiles\ccc_shap.cpp ^
-lgdi32
◇
```

Theses are the include files for this program

$\langle \text{Include Files 3c} \rangle \equiv$

```
#include "ccc_win.h"
◇
```

Fragment referenced in 1.

## 4 Test