Black Box Software Testing

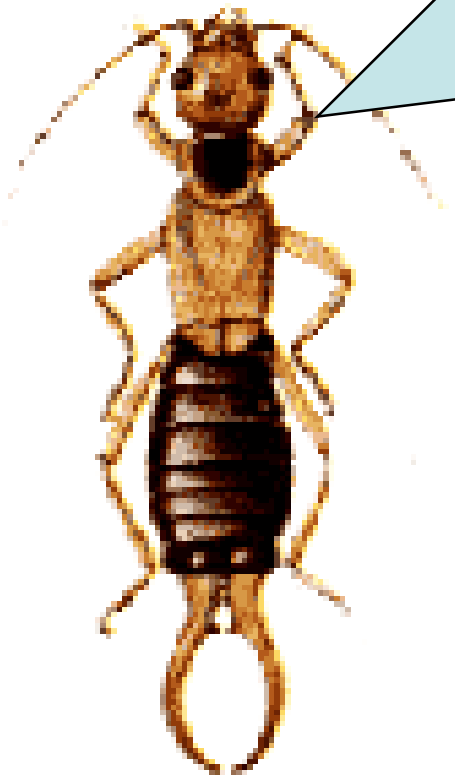
Fall 2004

Part 31 -- Exercises

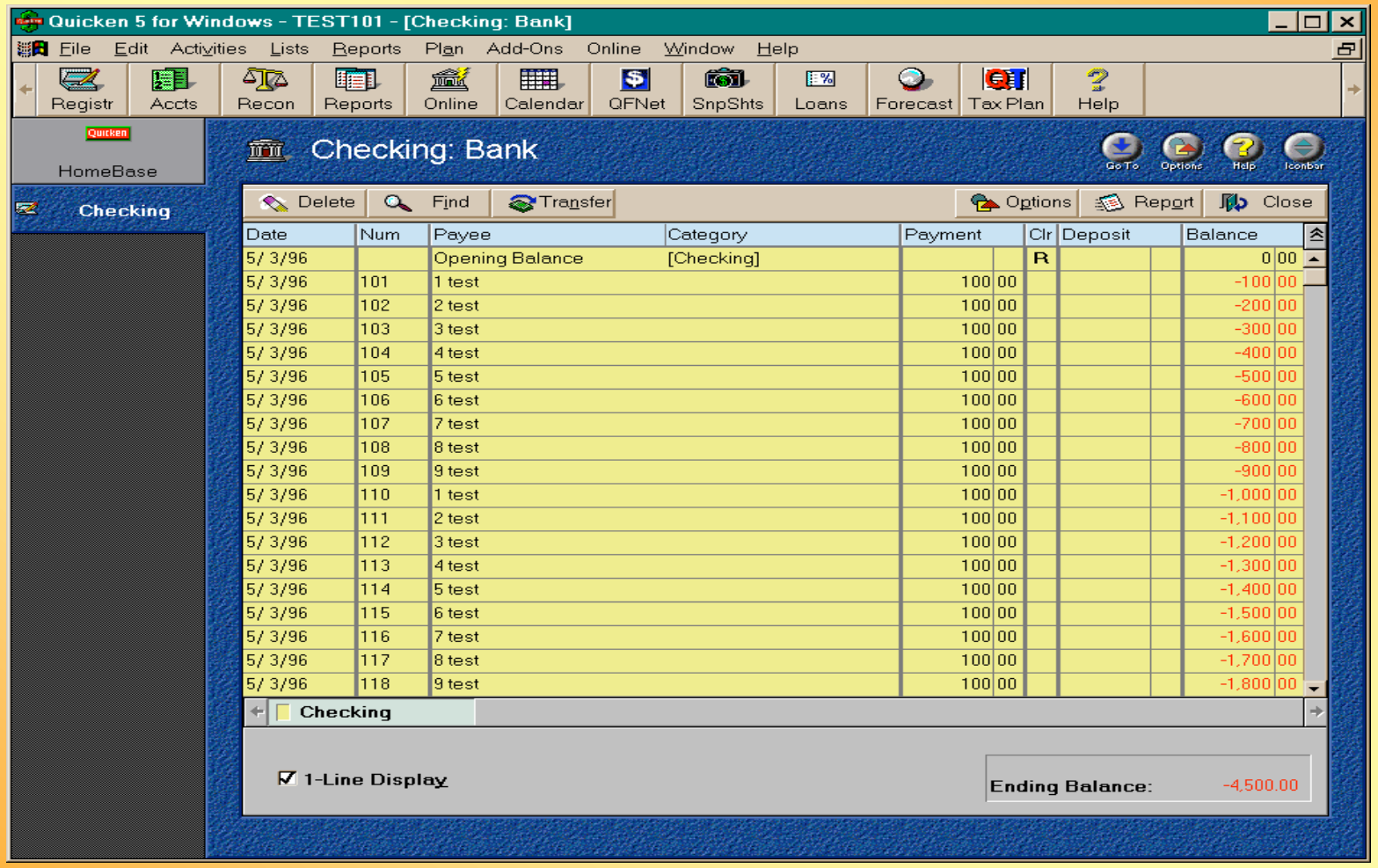
by
Cem Kaner, J.D., Ph.D.
Professor of Software Engineering
Florida Institute of Technology
and
James Bach
Principal, Satisfice Inc.

These notes are partially based on research that was supported by NSF Grant EIA-0113539 ITR/SY+PE: "Improving the Education of Software Testers." Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Bug reporting exercises

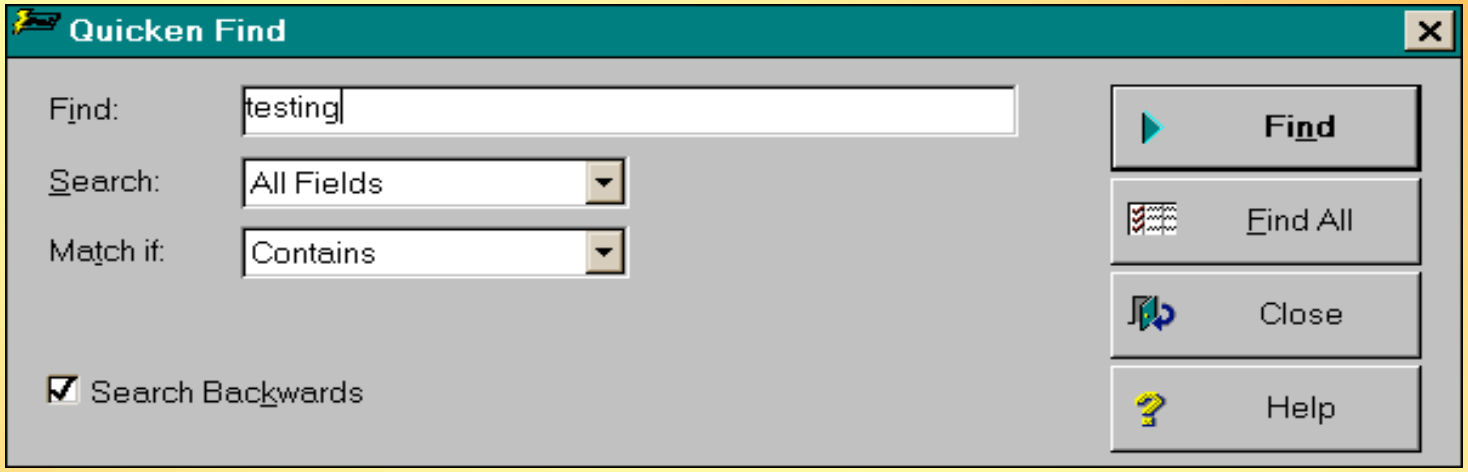


Bug Reporting Exercise 1 (1)

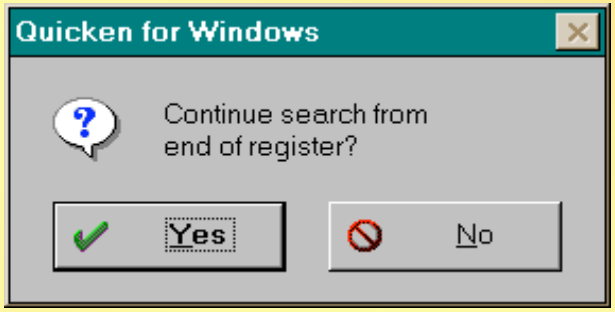


Create a sample database of cheques. Enter many new cheques.

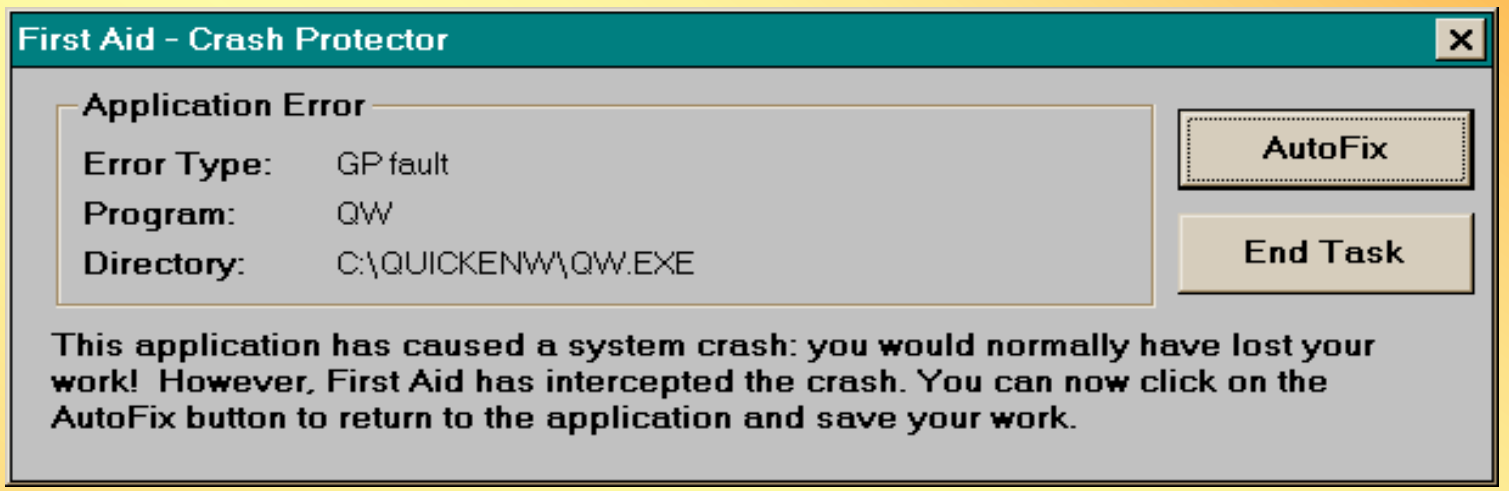
Bug Reporting Exercise 1 (2)



Now search the cheques to find one. Here, I searched for the word “testing”. The program searches backwards, from the currently selected cheque to the start of the register. It doesn’t find any instances of “testing” so it asks whether it should keep searching from the end of the register backwards.



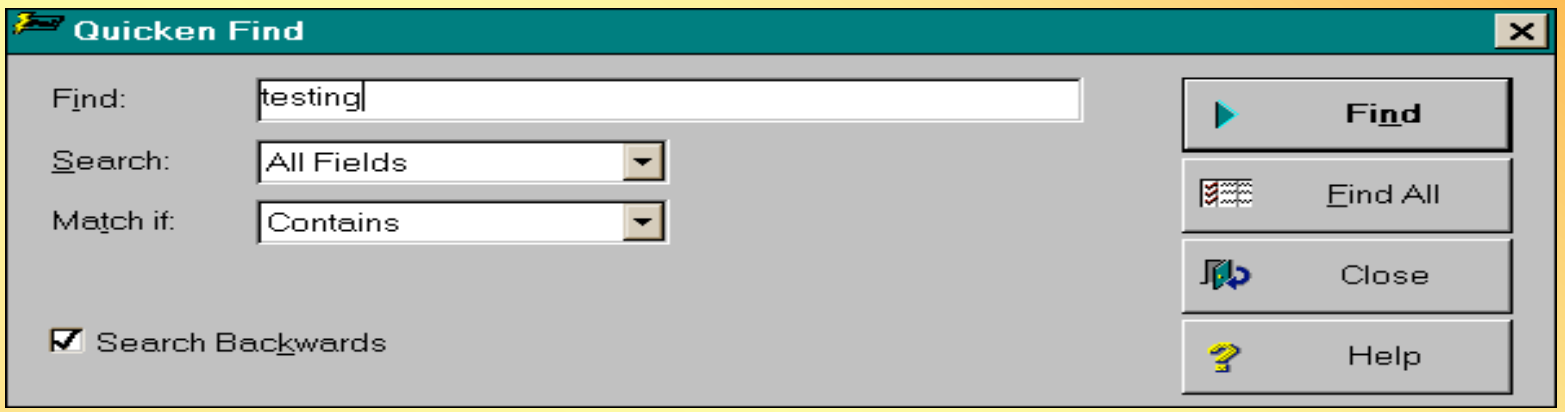
Bug Reporting Exercise 1 (3)



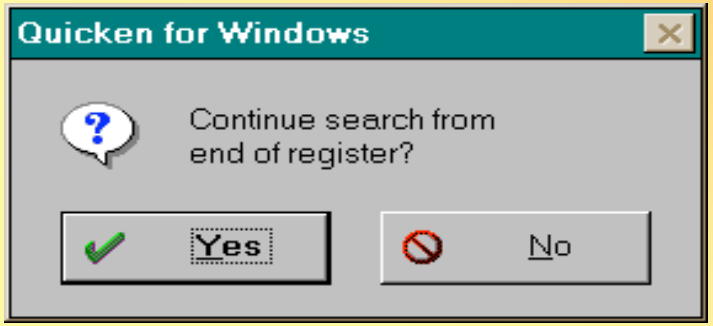
Kaboom! A General Protection Fault!

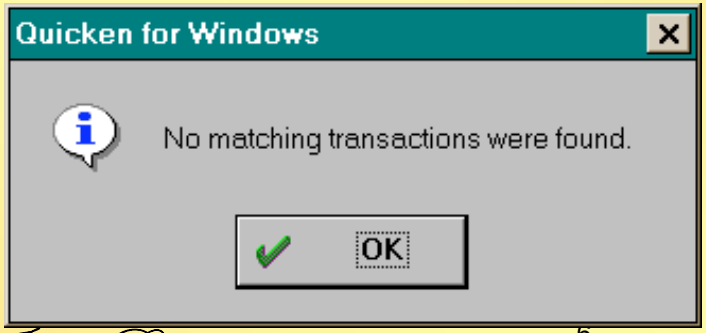
- The "First Aid" application tries to protect the customer from losing data when there is a GP fault. It's always *possible* that the crash was *caused* by an interaction between Quicken and First Aid, so try the test again after turning off First Aid.
- When I re-ran the test, Quicken crashed again, with a Win 95 system window that identifies a GP Fault. (These are harder to screen shoot, so it's not here.) Therefore the bug was not due to First Aid.

Bug Reporting Exercise 1 (4)



When analyzing a bug, it's wise to try to recreate it on another computer. I did that. This time, the search didn't crash. The crashing computer is a Pentium with 32 megs RAM, a Logitech trackball, the MS keyboard, a 1.6 gig hard drive, no disk compression, a 4 meg high res MPEG video card and a big monitor. The other is an 8 meg 486 with an MS Mouse, an old standard keyboard, a 540 meg hard drive (compressed) and basic SVGA video.





Bug Reporting Exercise 1 (5)

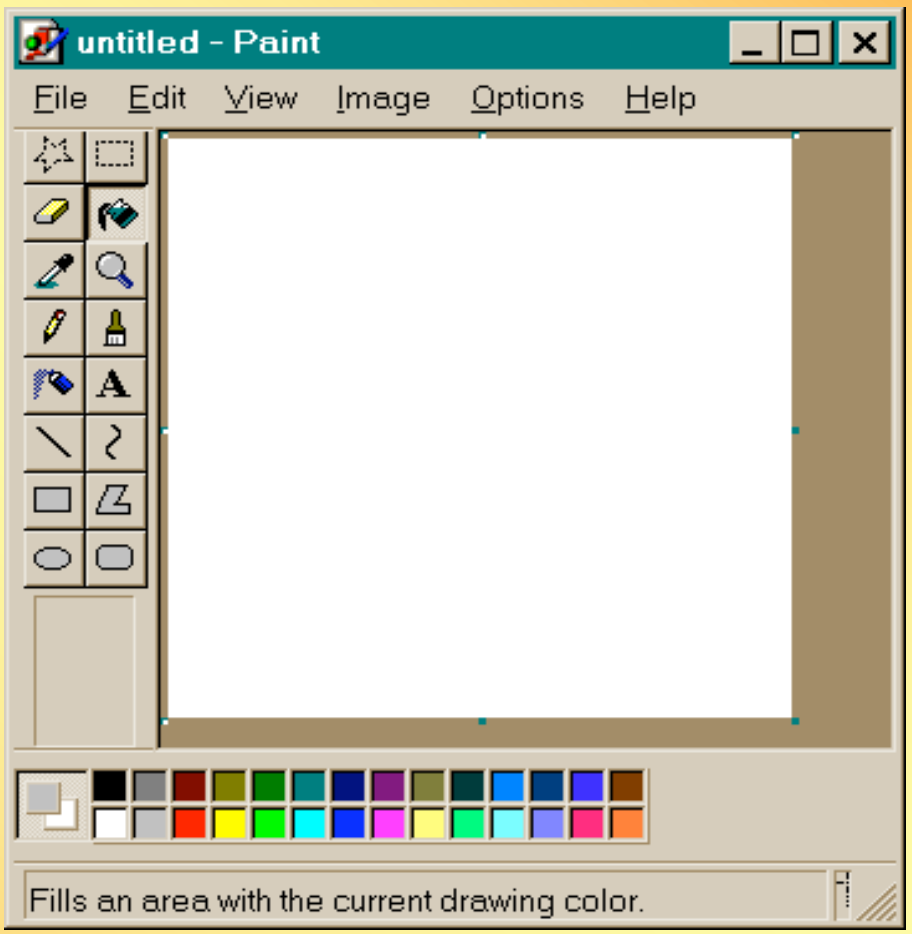
- Because this is a crash, you decide to get it into the tracking system right away. You'll do more troubleshooting later. So here is your assignment.
  1. Write these two sections of the bug report:
     - Problem Summary
     - Problem Description
  2. What other tests should you run? Why? Write down your list.
  3. Meet with your group to read each other's reports.
     - How good is the summary?
     - How clear is the description?
     - How complete is the description?
     - How accurate is the description?
     - How promising is your list of ideas?

Bug Reporting Exercise 2 (1)

- The following group of slides are from Windows Paint 95. Please don't spend your time replicating the steps or the bug. (You're welcome to do so if you are curious, but it is not necessary for analysis of this exercise.)
- Treat the steps that follow as fully reproducible. If you go back to ANY step, you can reproduce it.
- In case you aren't familiar with paint programs, the key idea is that you lay down dots. For example, when you draw a circle, the result is a set of dots, not an object. If you were using a draw program, you could draw the circle and then later select the circle, move it, cut it, etc. In a paint program, you cannot select the circle once you've drawn it. You can select an area that includes the dots that make up the circle, but that area is simply a bitmap and none of the dots in it have any relationship to any of the others.

Bug Exercise 2 Continued

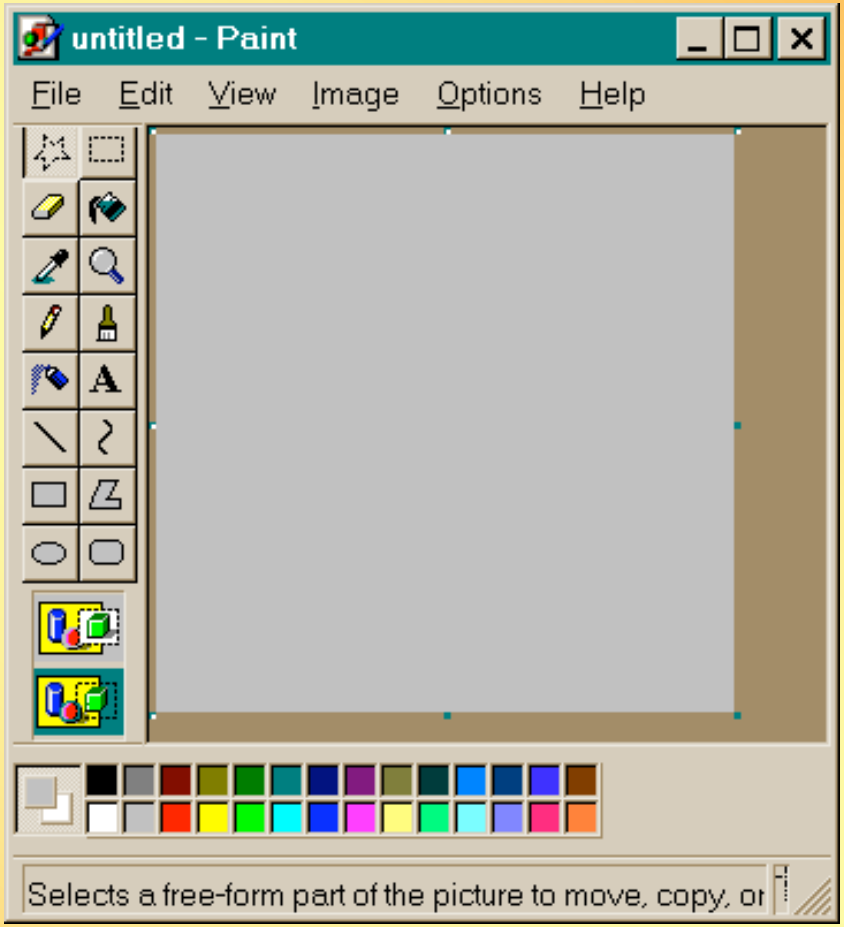
Here's the opening screen. The background is white. The first thing that we'll do is select the Pain  an
We'll use this to lay down a layer of grey paint on top of the background. Then, when we cut or move an area, we'll see the white background behind what was moved.



Bug Exercise 2 Continued

Here's the screen again, but the background has been painted gray.

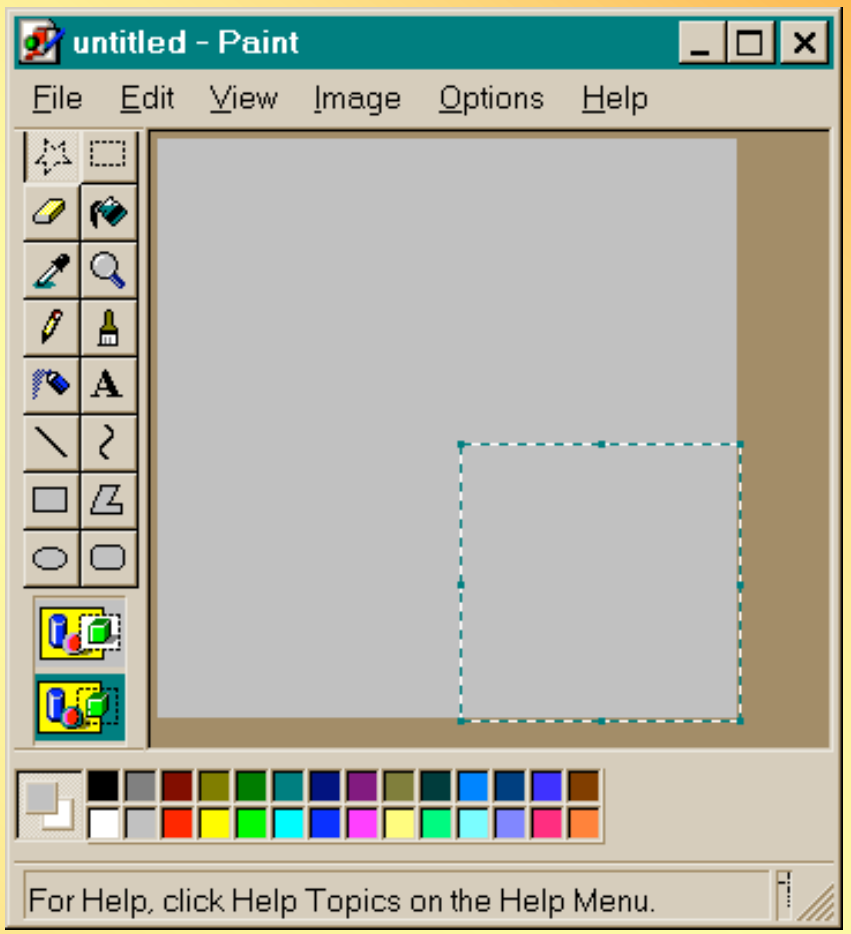
The star in the upper left corner is a freehand selection tool. After you click on it, you can trace around any part of the picture. The tracing selects that part of the picture. Then you can cut it, copy it, move it, etc.



Bug Exercise 2 Continued

This shows an area selected with the freehand selection tool. The bottom right corner is selected. (The dashed line surrounds the selected area.)

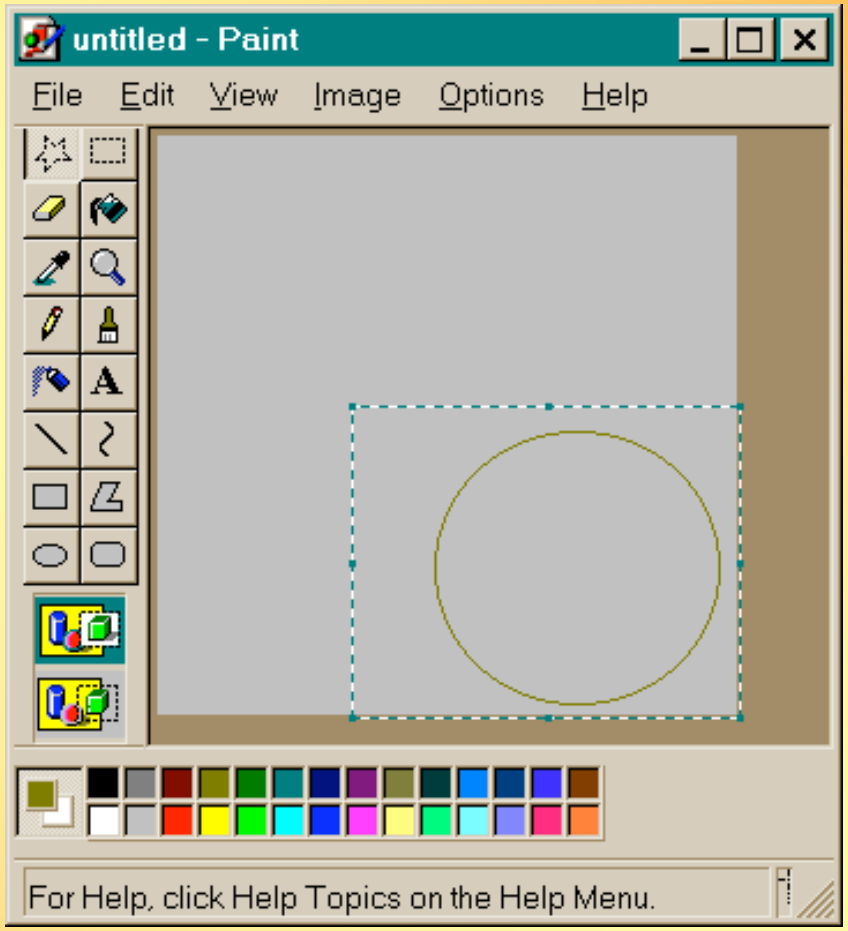
NOTE: The actual area selected might not be perfectly rectangular. The freehand tool shows a rectangle that is just big enough to enclose the selected area. For our purposes, this is not a bug. This is a design decision by Microsoft.



Bug Exercise 2 Continued

Next, we'll draw a circle (so you can see what's selected), then use the freehand select tool to select the area around it.

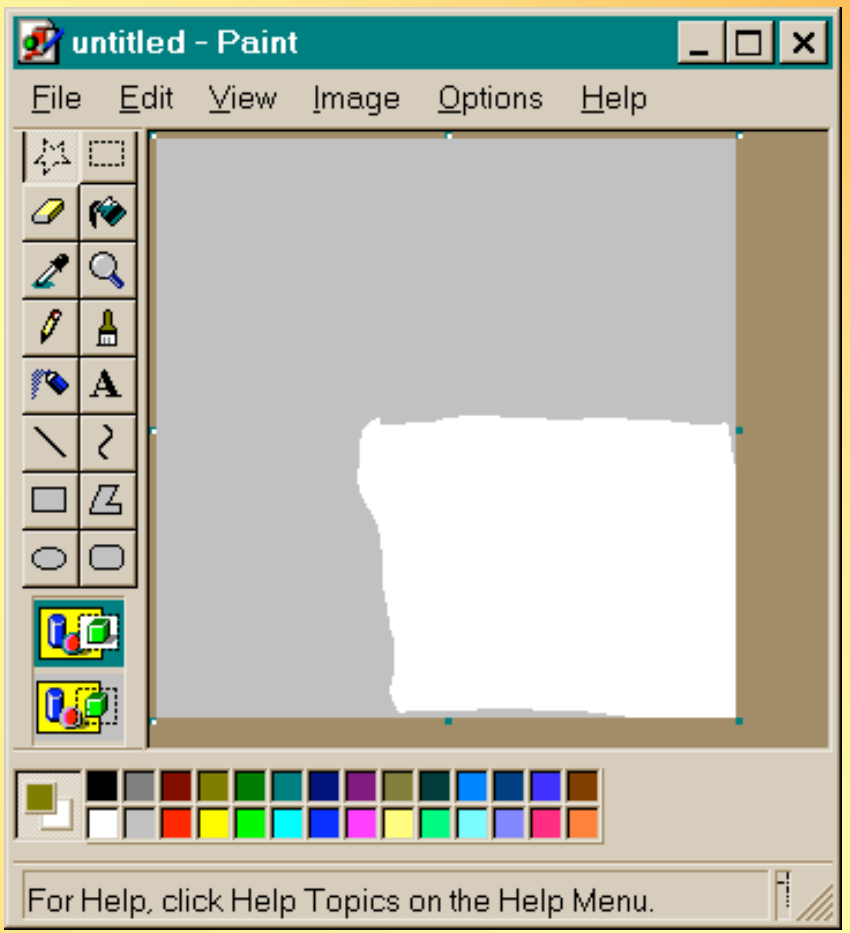
When you use the freehand selection tool, you select an area by moving the mouse. The real area selected is not a perfect rectangle. The rectangle just shows us where the selected area is.



Bug Exercise 2 Continued

Now we cut the selection. (To do this, press Ctrl-X.)

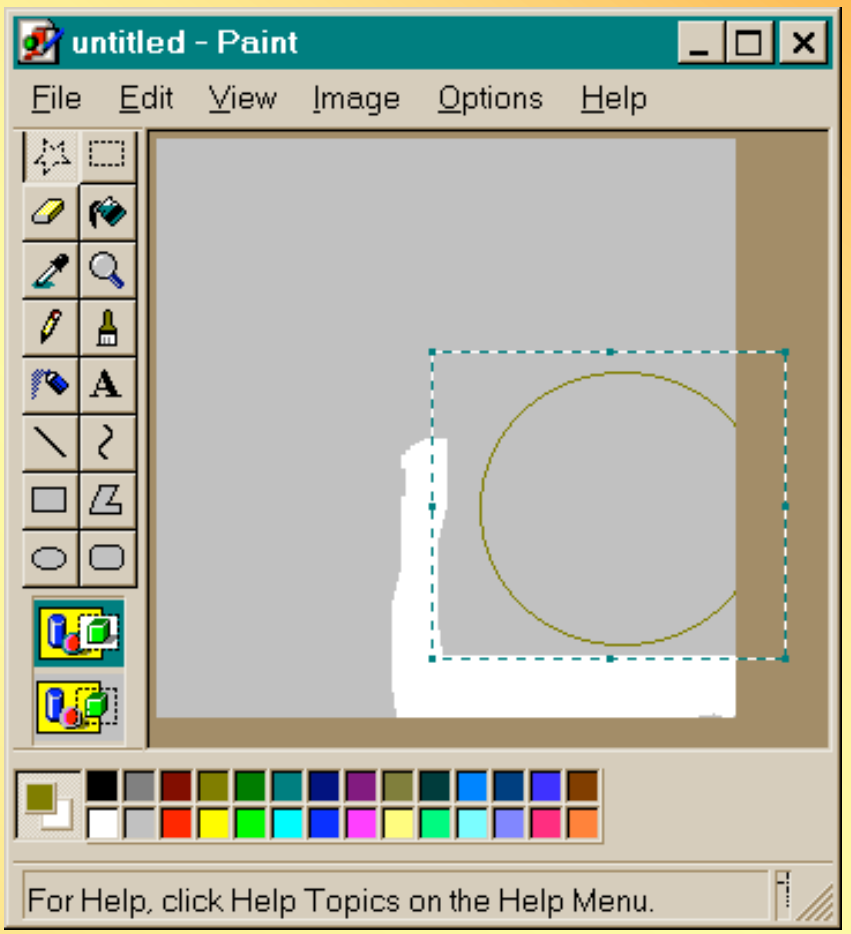
The jagged border shows exactly the area that was selected.



Bug Exercise 2 Continued

Next, select the area around the circle and drag it up and to the right.

This works.

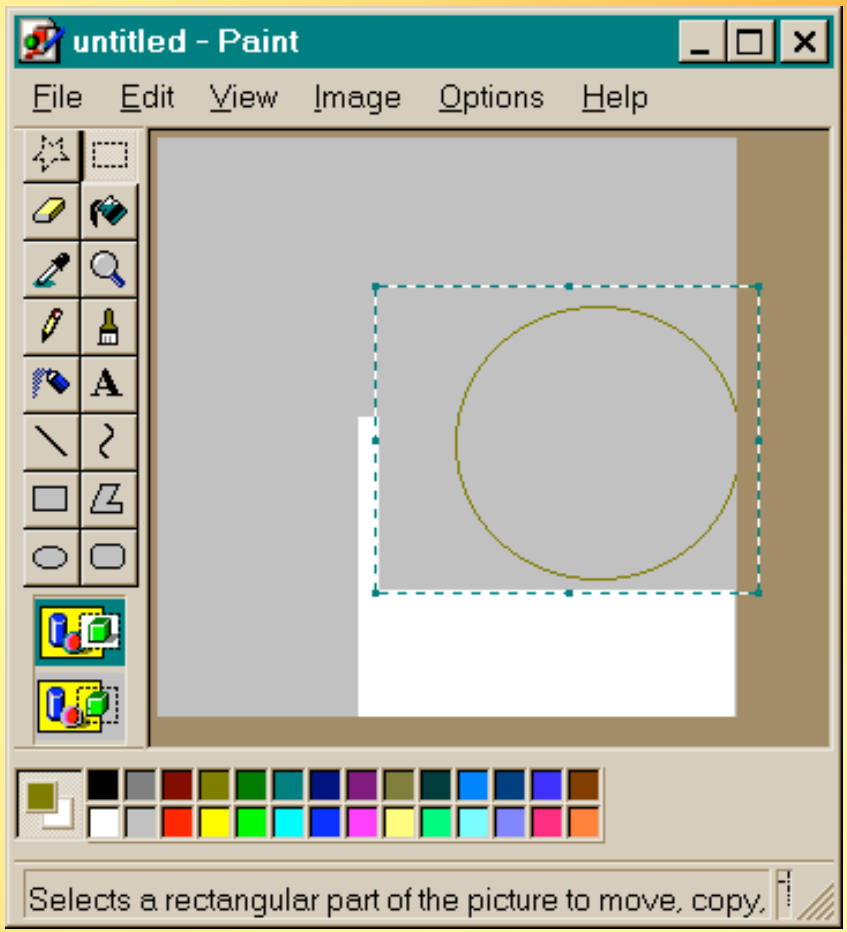


Bug Exercise 2 Continued

This time, we'll try the Rectangular Selection tool.

With this one, if you move the mouse to select an area, the area that is actually selected is the smallest rectangle that encloses the path that your mouse drew.

So, draw a circle, click the Rectangular Selection tool, select the area around the circle and move it up. It works.

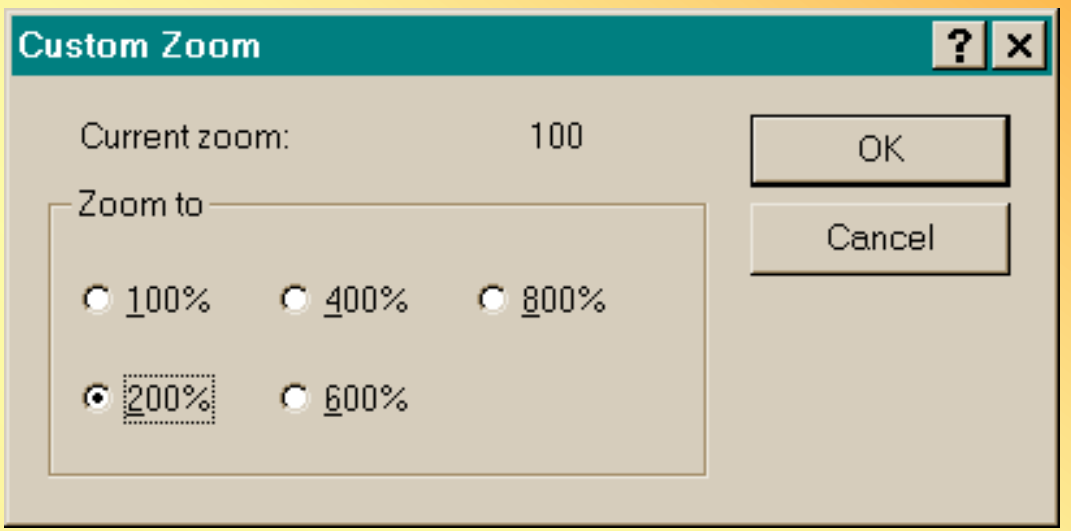


Bug Exercise 2 Continued

- Well, this was just too boring, because everything is working. When you don't find a bug while testing a feature, one tactic is to keep testing the feature but combine it with some other test.
- In this case, we'll try Zooming the image. When you zoom 200%, the picture itself doesn't change size, but the display doubles in size. Every dot is displayed as twice as tall and twice as wide.

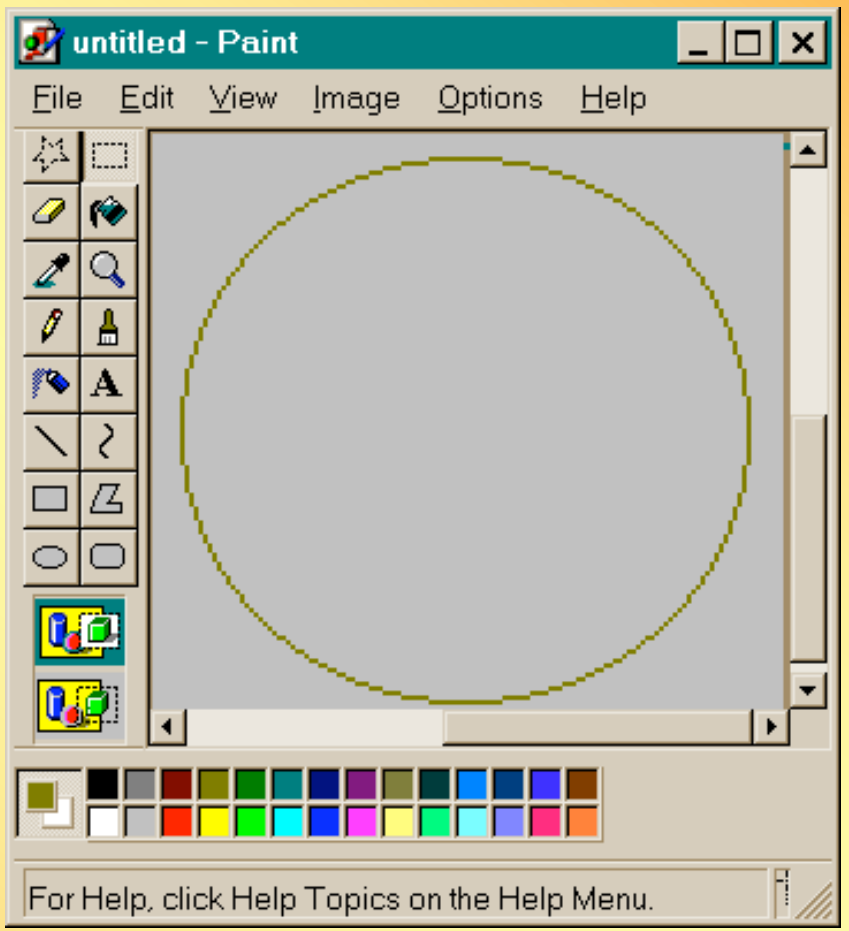
Bug Exercise 2 Continued

Bring up the Custom Zoom dialog, and select 200% zoom, click OK.



Bug Exercise 2 Continued

It worked. The paint area is displayed twice as tall and twice as wide. We're looking at the bottom right corner. To see the rest, we could move the scroll bars up or left.

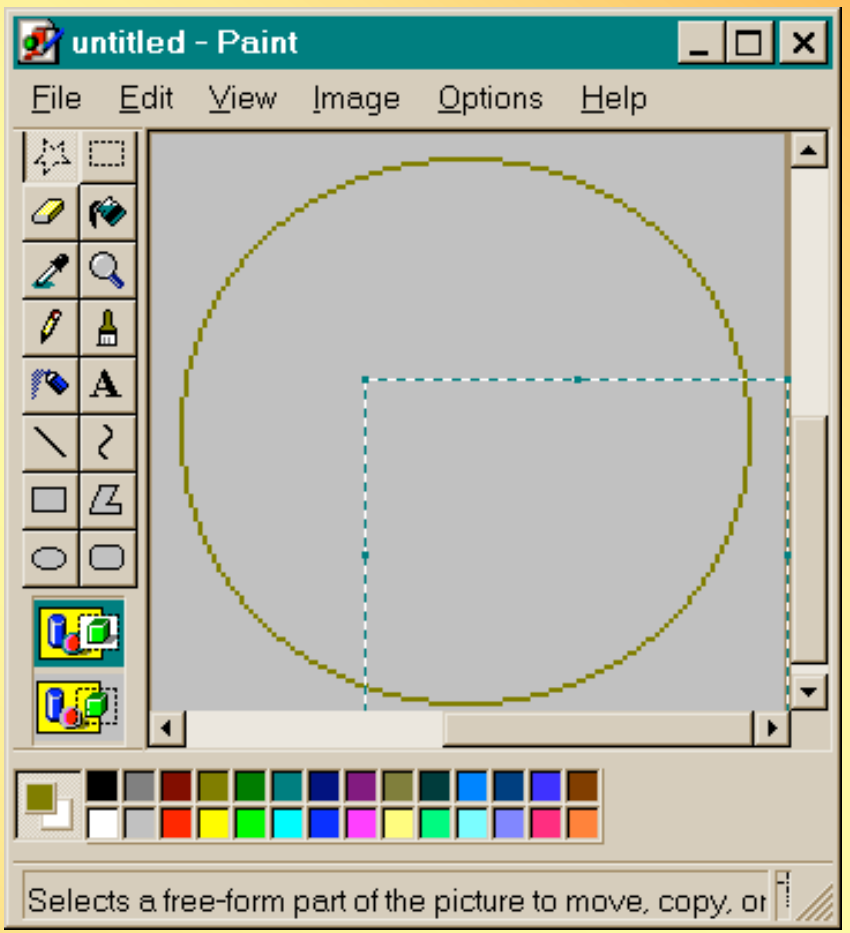


Bug Exercise 2 Continued

So, we select part of the circle using the freehand selection tool. We'll try the move and cut features.

Cutting fails.

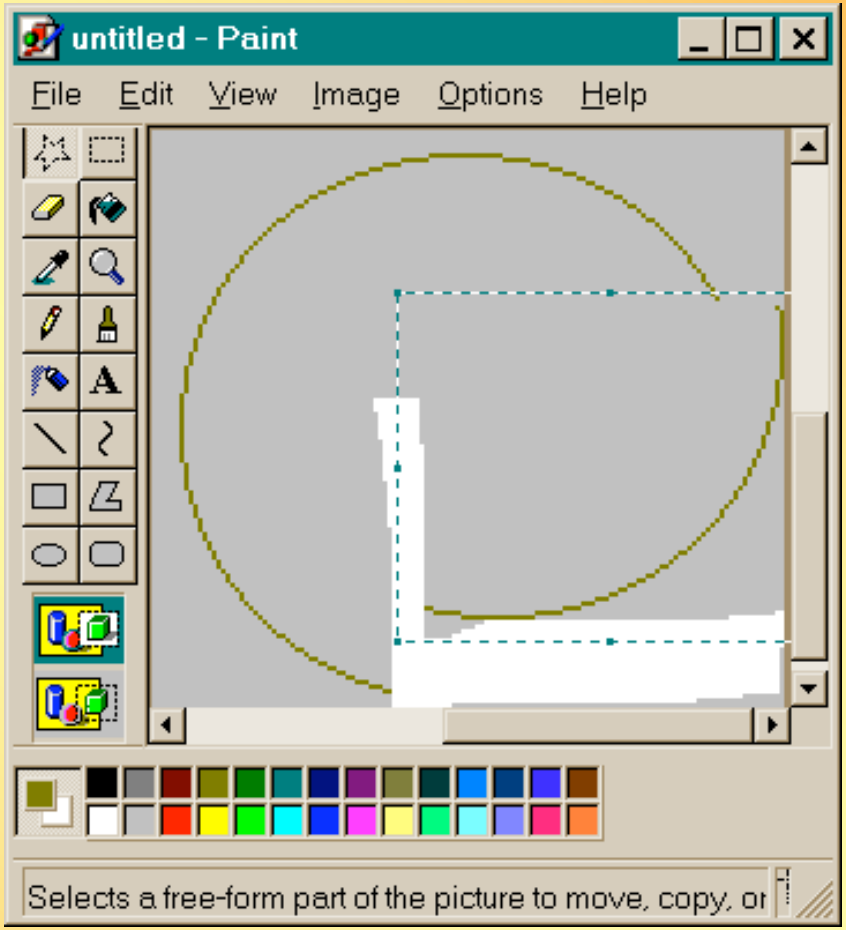
When we try to cut the selection, the dashed line disappears, but nothing goes away.



Bug Exercise 2 Continued

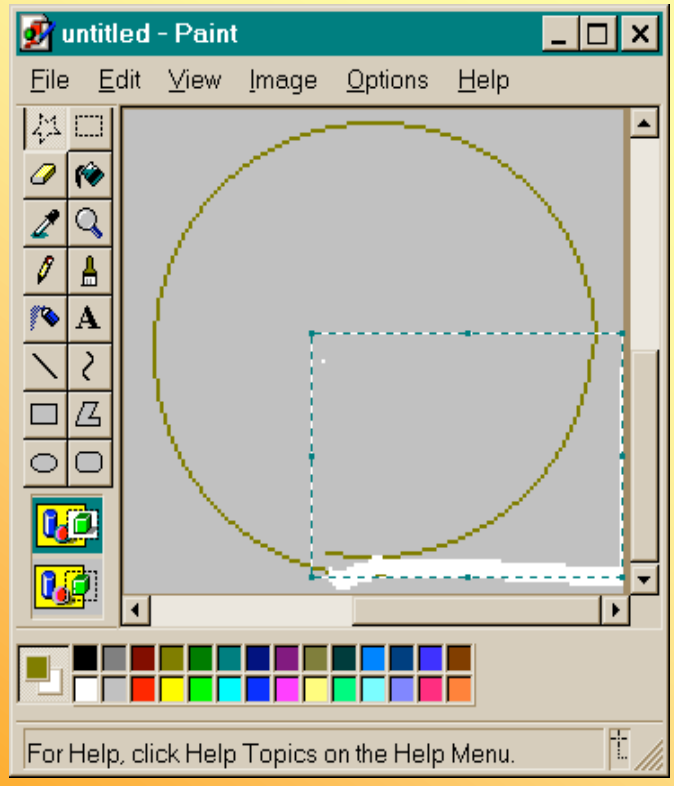
Draw the circle, zoom to 200%, select the area.

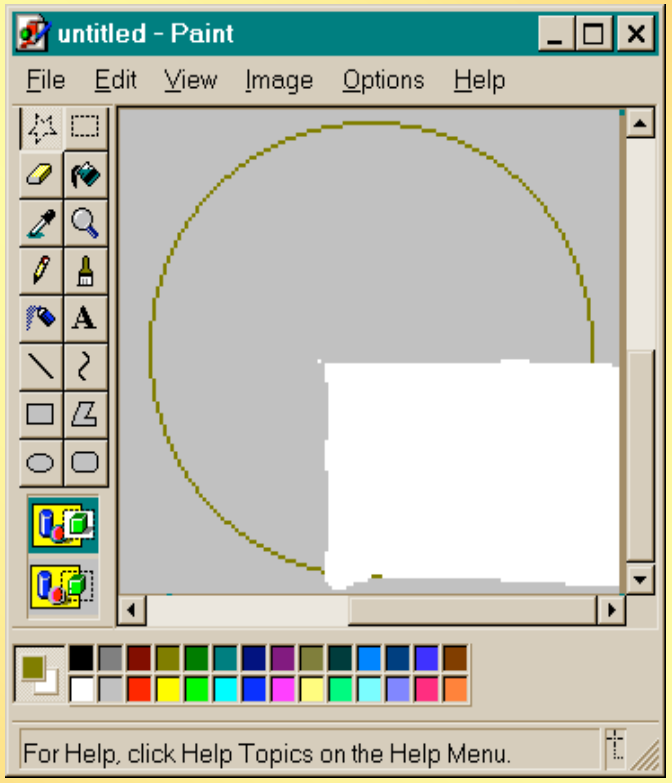
Drag the area up and to the right. It works.



Bug Exercise 2 Continued

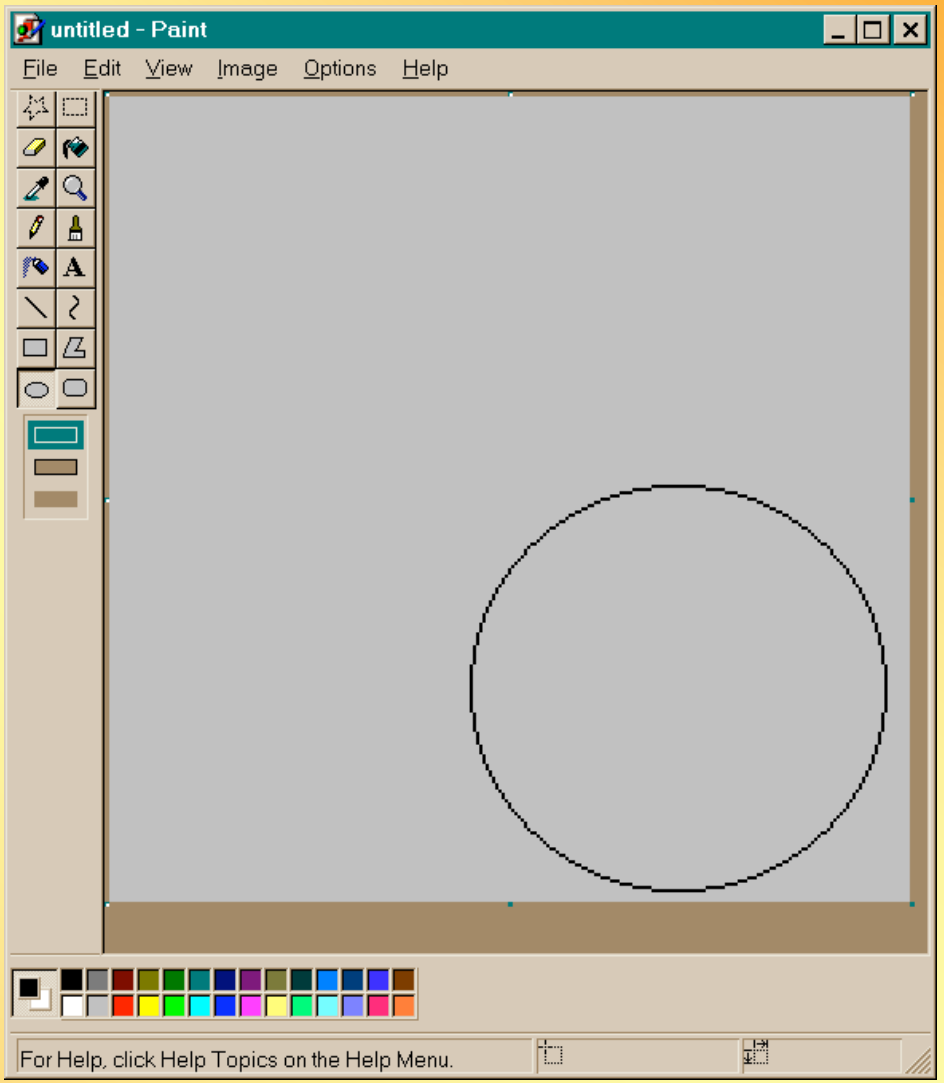
Draw the circle, zoom to 200%, select the area.
Now try this. Select the area and move it a bit. THEN press Ctrl-X to cut. This time, cutting works.





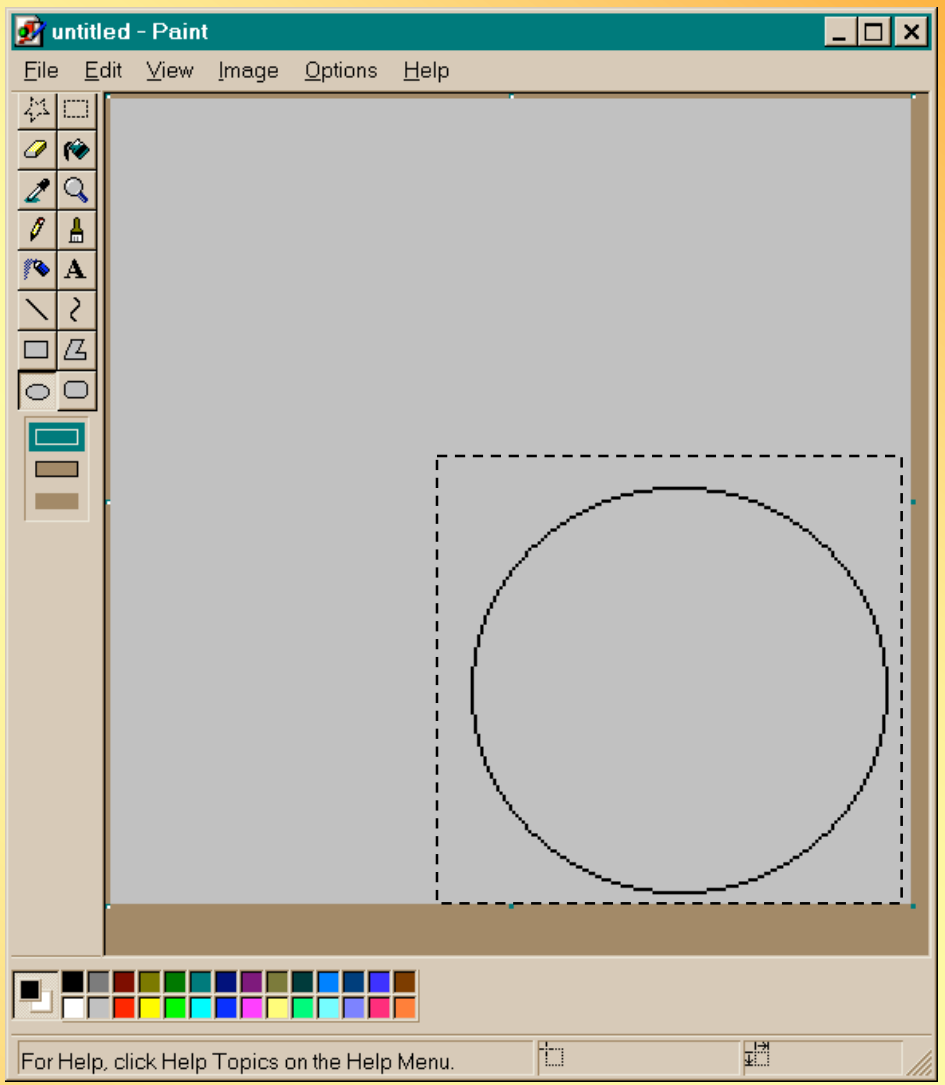
Bug Exercise 2 Continued

Draw the circle, zoom to 200%, and this time, *grow the window* so you can see the whole drawing area.



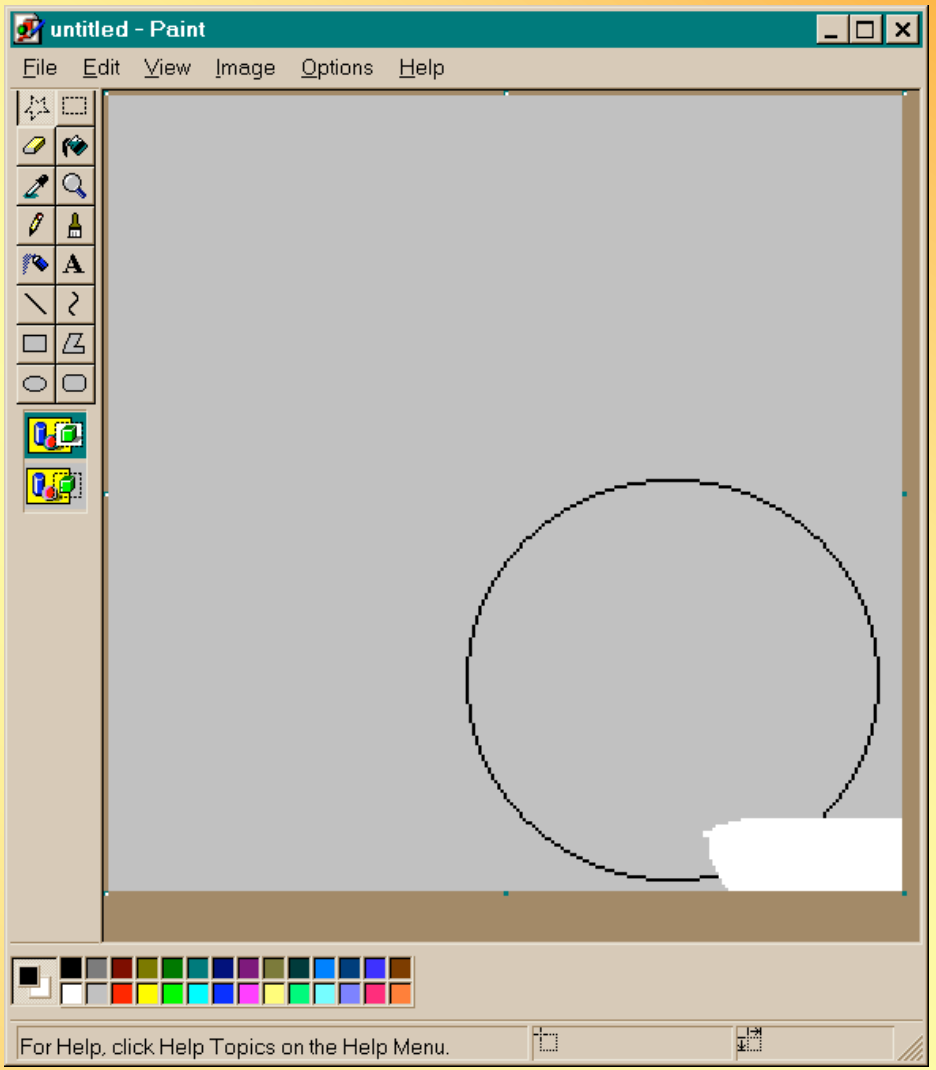
Bug Exercise 2 Continued

Now, select the circle.
That seems to work.



Bug Exercise 2 Continued

But when you press Ctrl-X to cut the circle, the program cuts the wrong area.



Bug Exercise 2 Continued

Now, write a bug report. I want two sections:

- The Problem summary (or title)
- The Problem Description (how to reproduce the problem)

Additionally, please describe three follow-up tests that you would run with this bug

Notes on Exercise #1

I do some analysis before writing. Here's a structure for making your notes:

OBSERVED FAILURES • *General protection fault*	CONDITIONS • *search for non-existent text* • *search backwards* • *Yes to query, search from end of register*
OTHER CONDITIONS (maybe irrelevant) • *Configurations (list them all)*	NOTES

Notes on Exercise #1

MY SUMMARY

GPF on search for non-existent text. (Configuration dependent.)

MY PROBLEM DESCRIPTION

1. *Start the program*
2. *Open a database (I used the TESTING file)*
3. *Search (backwards) for a string that doesn't appear in the database*
4. *When the program asks whether to search from end of register, click YES*
5. *Result = GPF*

NOTES: This bug is configuration dependent. The two machines involved are the two at my desk, if you need to replicate while I'm gone. I'll do further analysis later, but I put this into the database now in order to give you an early warning of a serious bug. The configurations of the two machines are:

Replicates	*Fails to Replicate*
Pentium	*486*
etc	*etc*