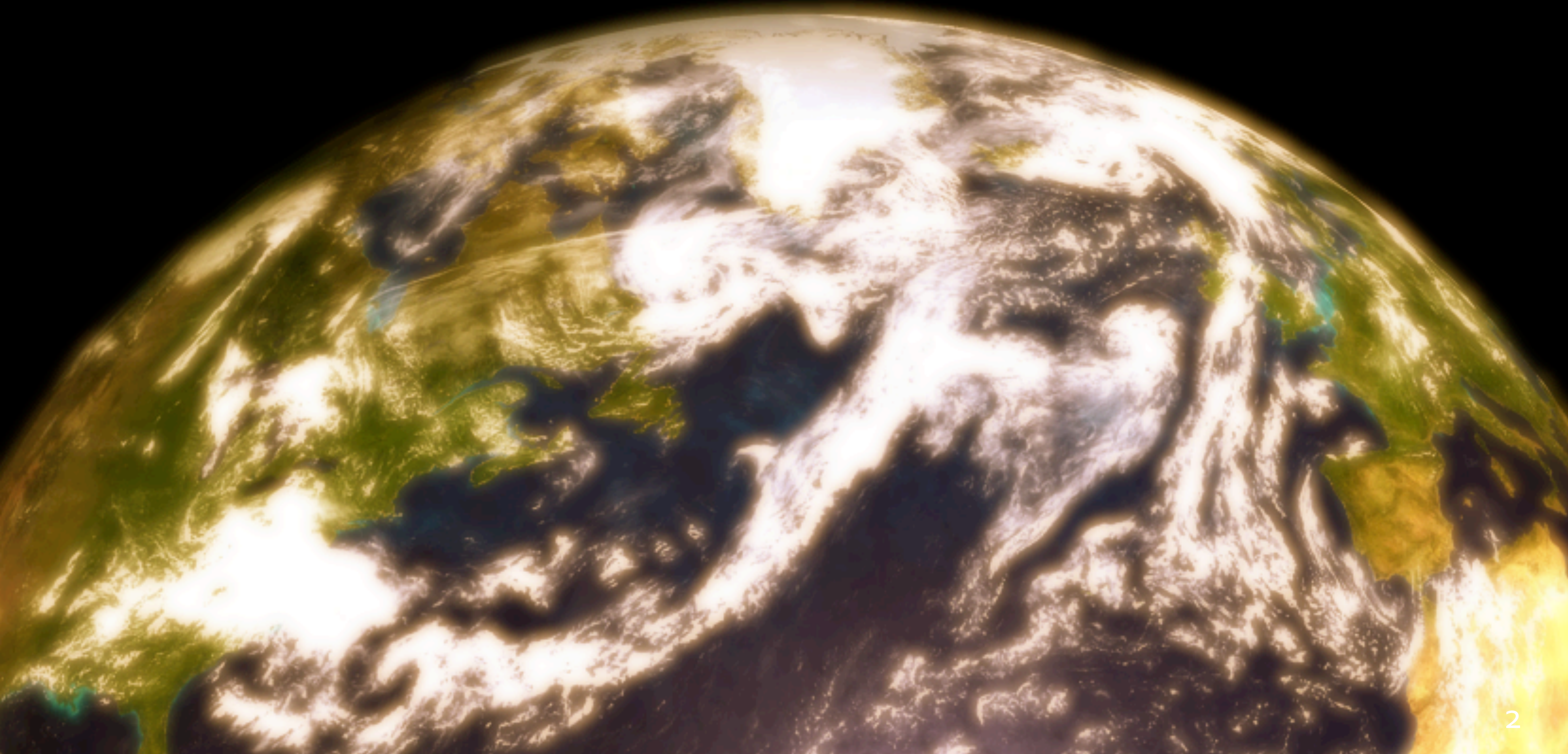
Asking and Answering Questions about the Causes of Software Behavior

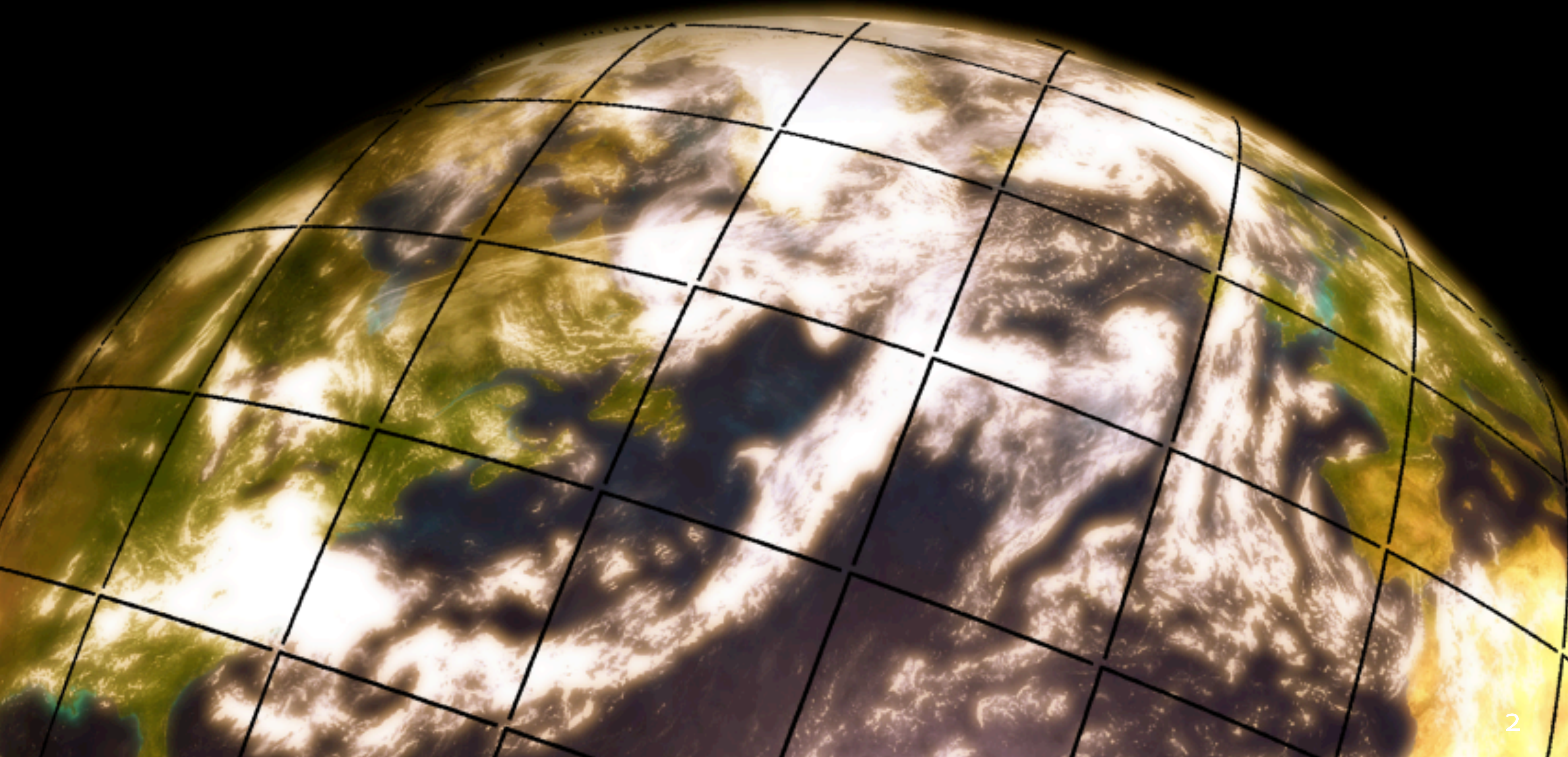
Andrew Ko

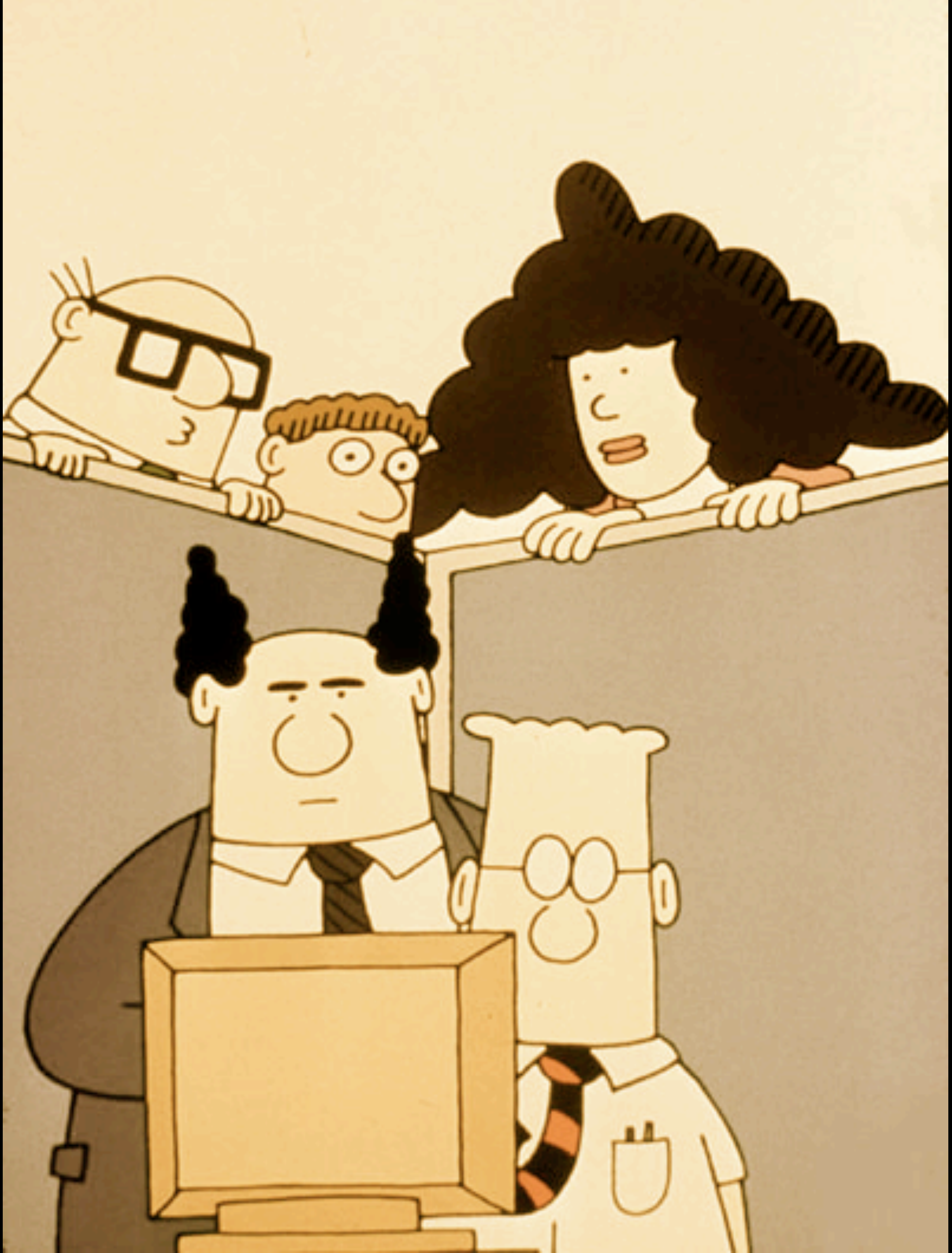
Human-Computer Interaction Institute

Carnegie Mellon



software is everywhere





program understanding

an essential and fundamental part of

fixing bugs…

adding features…

maintaining legacy code…

adapting code for new purposes…

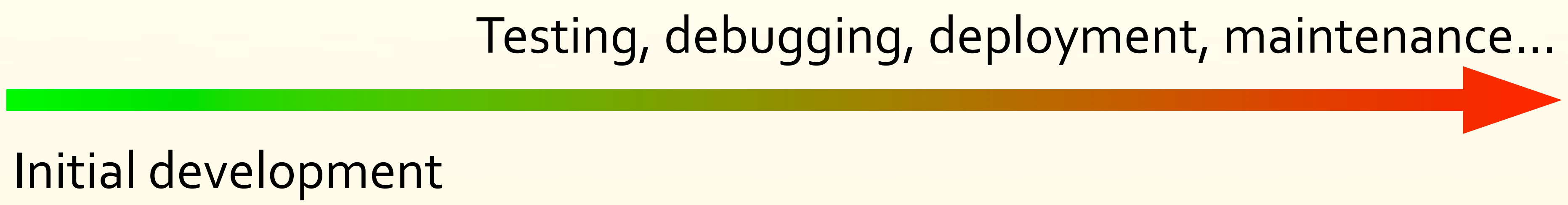
reusing components…

… **identifying and correcting defects** during the software development process represents over **half** of development **costs** … and accounts for **30 to 90 percent of labor** expended to produce a working program."

National Institute of Standards and Technology, 2002

… **identifying and correcting defects** during the software development process represents over **half** of development **costs** … and accounts for **30 to 90 percent of labor** expended to produce a working program.”

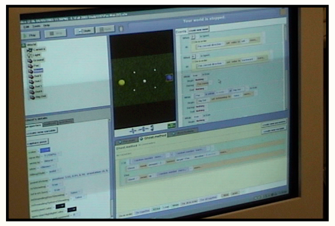
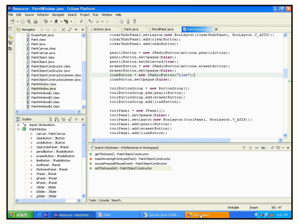
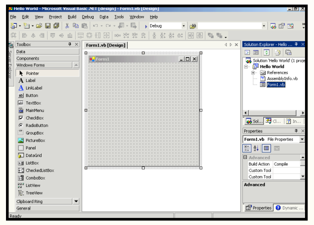
National Institute of Standards and Technology, 2002

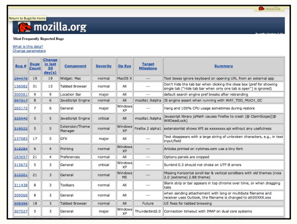


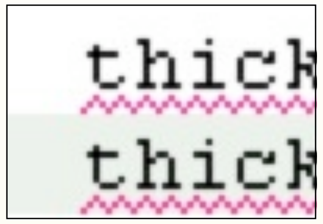
why is program understanding **difficult**?

what could make understanding **easier**?

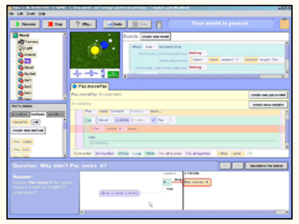
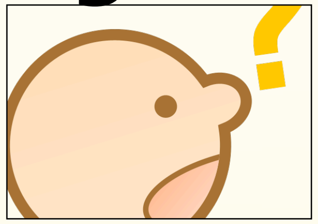
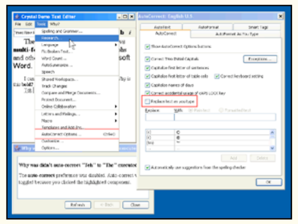
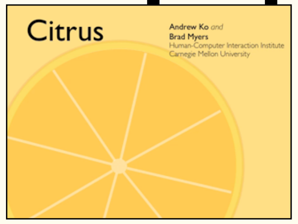
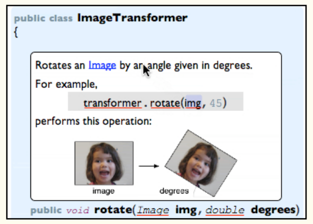
studies of program understanding in **multiple contexts**



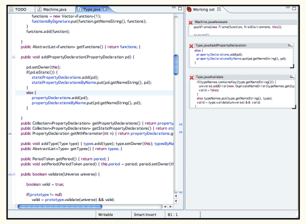


technologies for **different populations** of users

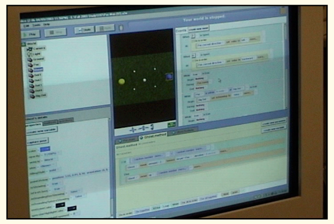
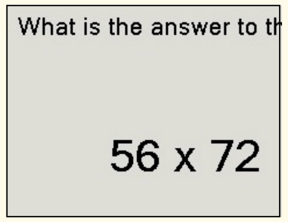




Citrus
Andrew Ko and
Brad Myers
Human-Computer Interaction Institute
Carnegie Mellon University


public class ImageTransformer
{
Rotates an Image by an angle given in degrees.
For example,
transformer.rotate(img, 45)
performs this operation:

public void rotate(Image img, double degrees)

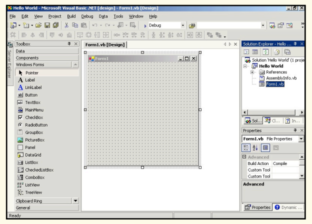


evaluations of these technologies

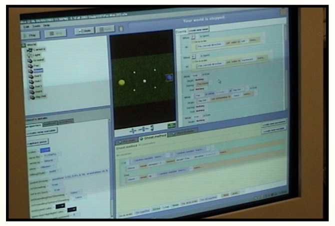
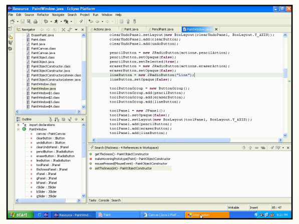
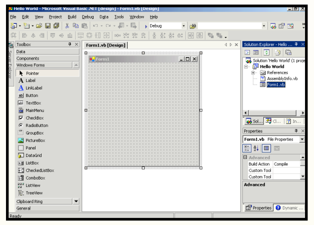



What is the answer to th

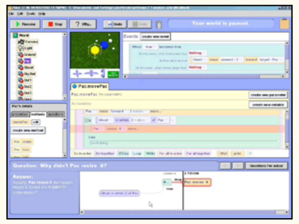
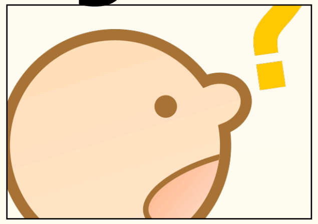
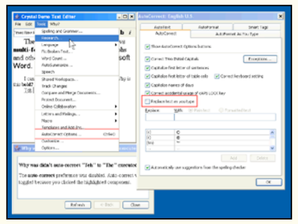
56 x 72



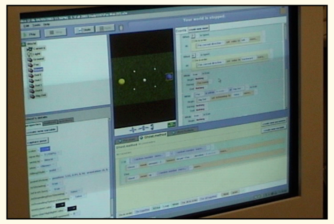
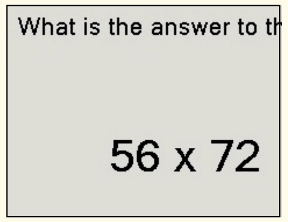
studies of program understanding in **multiple contexts**

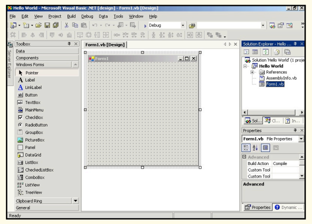




technologies for **different populations** of users

evaluations of these technologies



outline

programming **languages**

computer science **ed**

psychology of programming

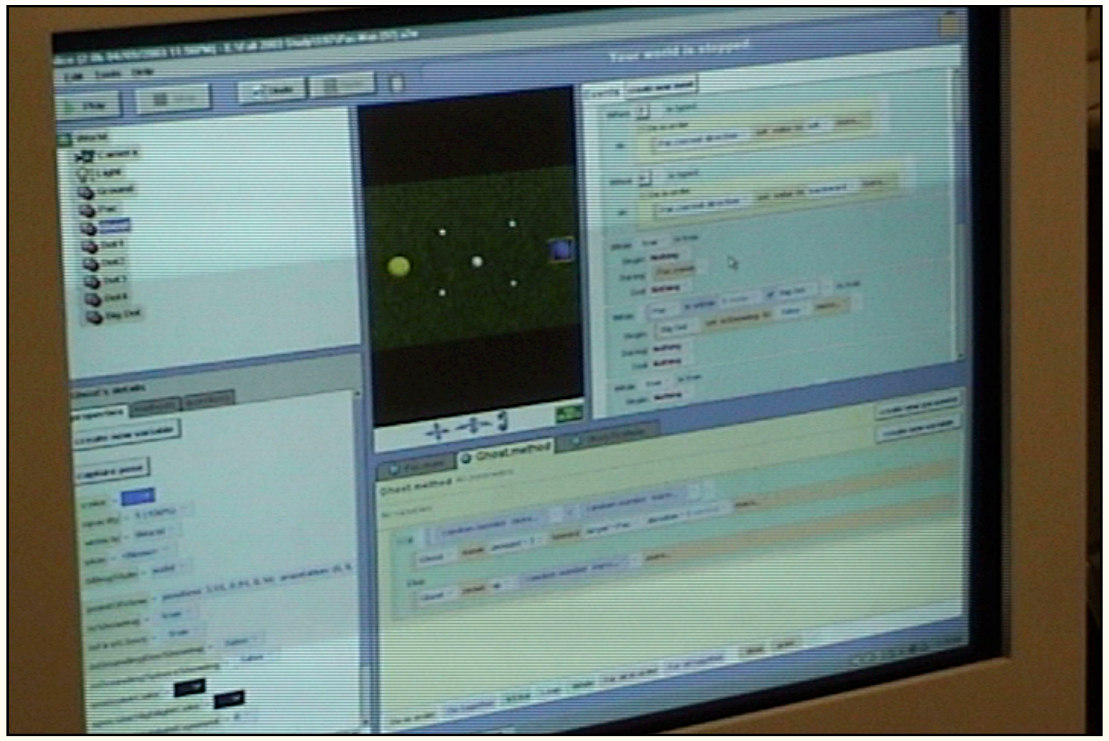
related work

human-computer **interaction**

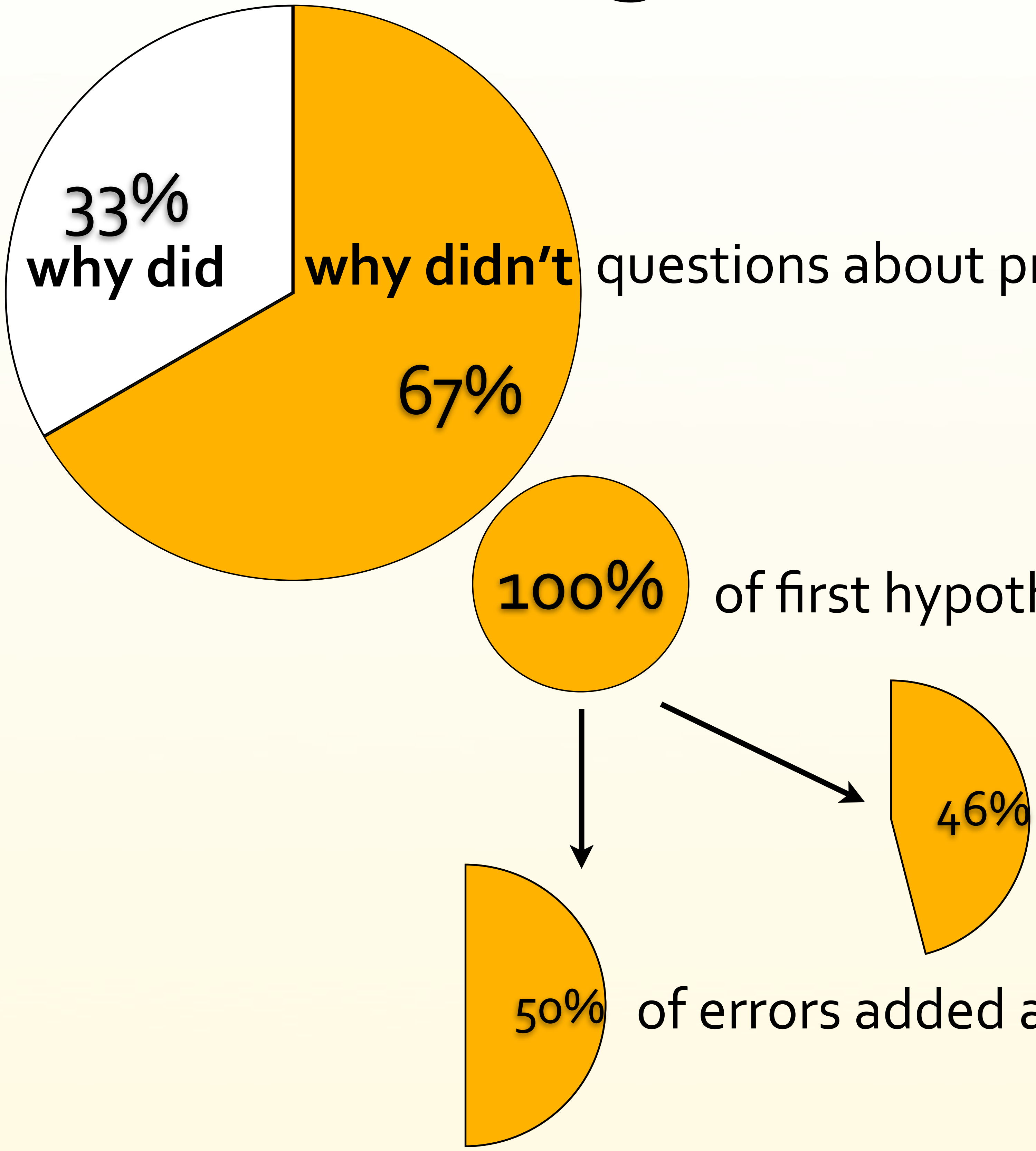
software **engineering**

novices using Alice

- **6** participants
- varying programming experience
- **created** a simple Pac Man **game**
- asked to **think aloud**
- **2 hour** session
- **videotaped** from behind



novices using Alice



why didn't questions about program output

100% of first hypotheses were **wrong**

46% time spent debugging

50% of errors added as a result of false hypotheses

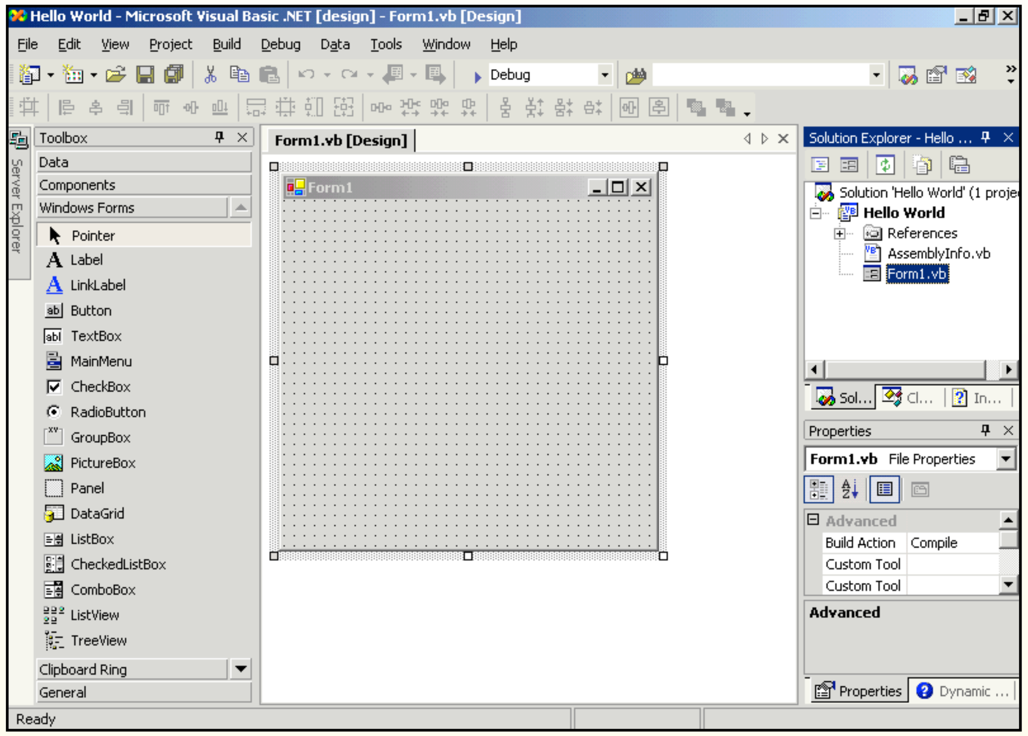
students learning Visual Basic

- **30 students** learning VB.NET.
- 4 programming **assignments**
- 2 **TAs** available in computer lab
- **when asked** for help, TAs recorded

  what student was "stuck" on

  how they became stuck

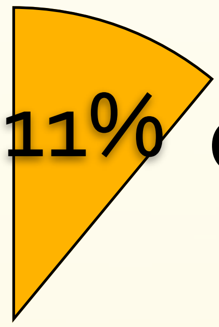
  what student did to become "**unstuck**"



students learning Visual Basic

struggled to **form** hypotheses

consulted **peers** for hypotheses



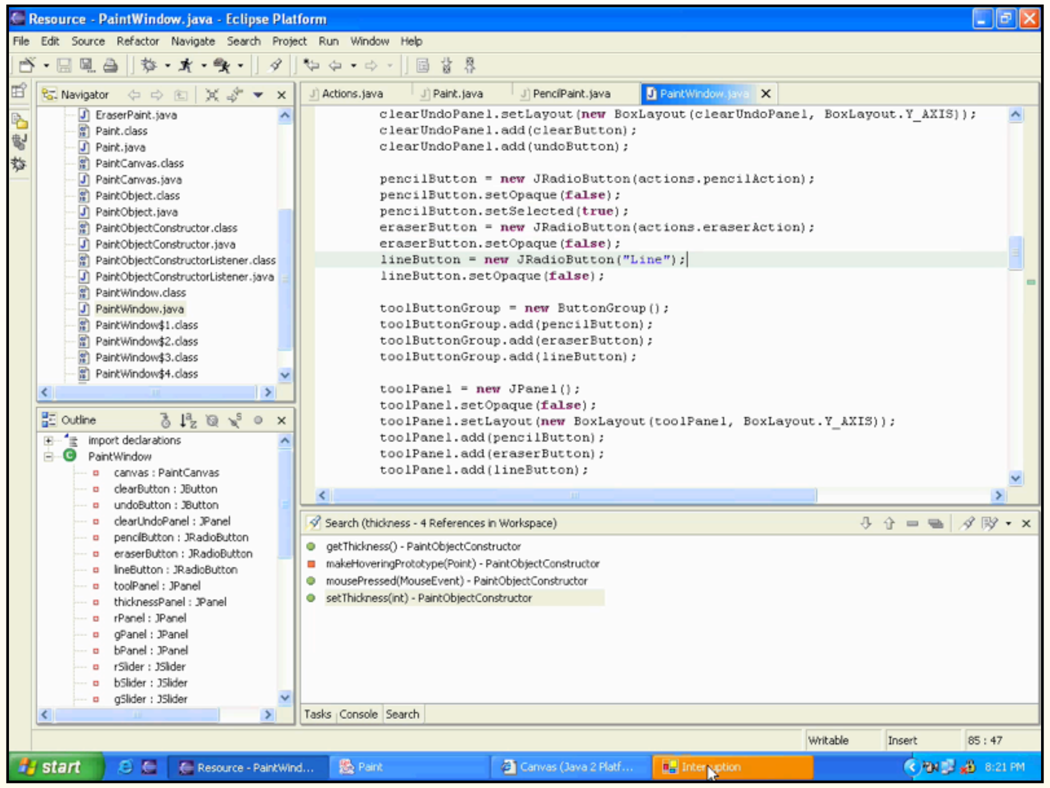
in 11% of cases, couldn't think way to **test** hypothesis

students **misperceived** program output,

investigating **non-issues**

debugging in Eclipse

- **31 Java** programmers
- 3 debugging **tasks**
- 2 enhancement **tasks**
- worked on a **painting** program
- used **Eclipse 2.0** and the web
- screen captured



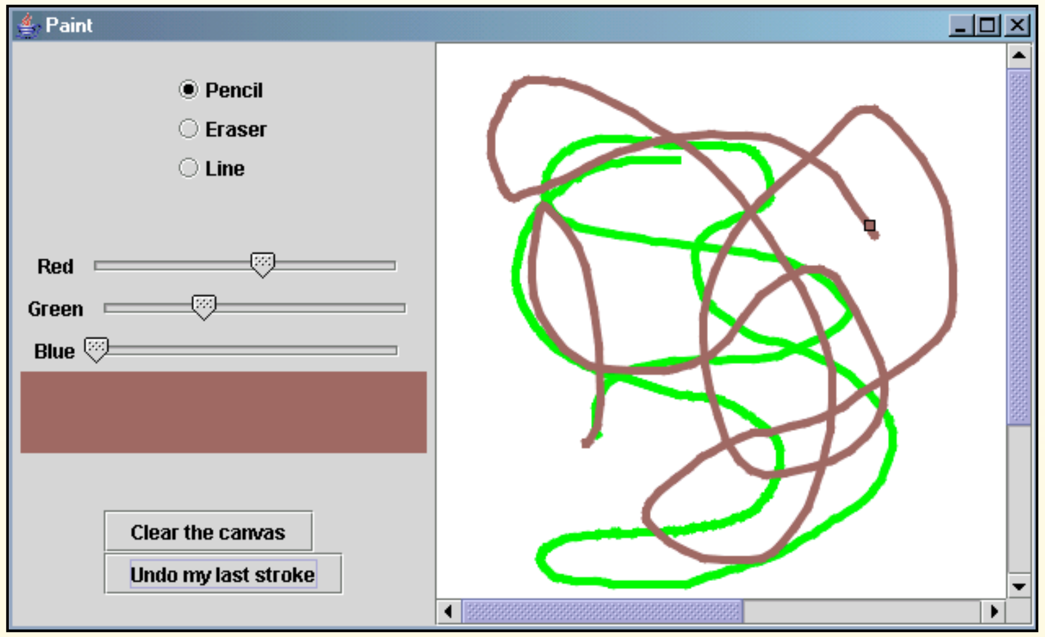
EraserPaint.java
Paint.class
Paint.java
PaintCanvas.class
PaintCanvas.java
PaintObject.class
PaintObject.java
PaintObjectConstructor.class
PaintObjectConstructor.java
PaintObjectConstructorListener.class
PaintObjectConstructorListener.java
PaintWindow.class
PaintWindow.java
PaintWindow$1.class
PaintWindow$2.class
PaintWindow$3.class
PaintWindow$4.class

import declarations
PaintWindow
canvas : PaintCanvas
clearButton : JButton
undoButton : JButton
clearUndoPanel : JPanel
pencilButton : JRadioButton
eraserButton : JRadioButton
lineButton : JRadioButton
toolPanel : JPanel
thicknessPanel : JPanel
rPanel : JPanel
gPanel : JPanel
bPanel : JPanel
rSlider : JSlider
bSlider : JSlider
gSlider : JSlider

```
        clearUndoPanel.setLayout(new BoxLayout(clearUndoPanel, BoxLayout.Y_AXIS));
        clearUndoPanel.add(clearButton);
        clearUndoPanel.add(undoButton);

        pencilButton = new JRadioButton(actions.pencilAction);
        pencilButton.setOpaque(false);
        pencilButton.setSelected(true);
        eraserButton = new JRadioButton(actions.eraserAction);
        eraserButton.setOpaque(false);
        lineButton = new JRadioButton("Line");
        lineButton.setOpaque(false);

        toolButtonGroup = new ButtonGroup();
        toolButtonGroup.add(pencilButton);
        toolButtonGroup.add(eraserButton);
        toolButtonGroup.add(lineButton);

        toolPanel = new JPanel();
        toolPanel.setOpaque(false);
        toolPanel.setLayout(new BoxLayout(toolPanel, BoxLayout.Y_AXIS));
        toolPanel.add(pencilButton);
        toolPanel.add(eraserButton);
        toolPanel.add(lineButton);
```

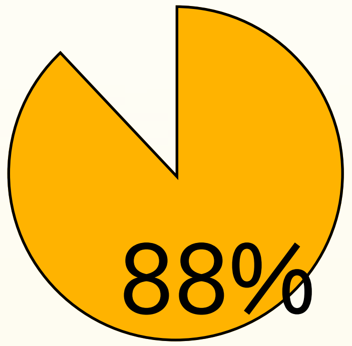
Search (thickness - 4 References in Workspace)

getThickness() - PaintObjectConstructor
makeHoveringPrototype(Point) - PaintObjectConstructor
mousePressed(MouseEvent) - PaintObjectConstructor
setThickness(int) - PaintObjectConstructor

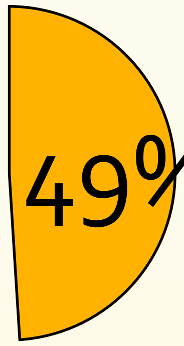


debugging in Eclipse

hypotheses were based on program output



88% of hypotheses were **false**

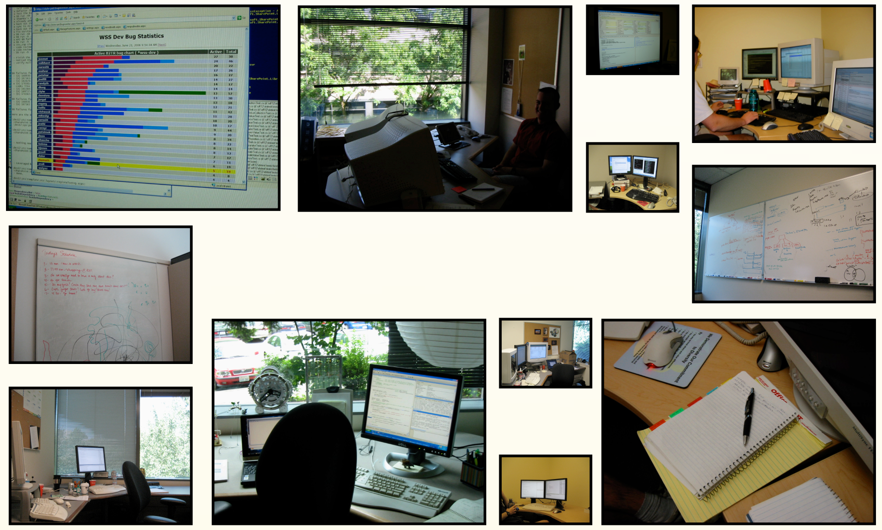


49% of time spent checking **irrelevant code**

many **hypotheses** went **untested**, leading to **misunderstandings** in later tasks

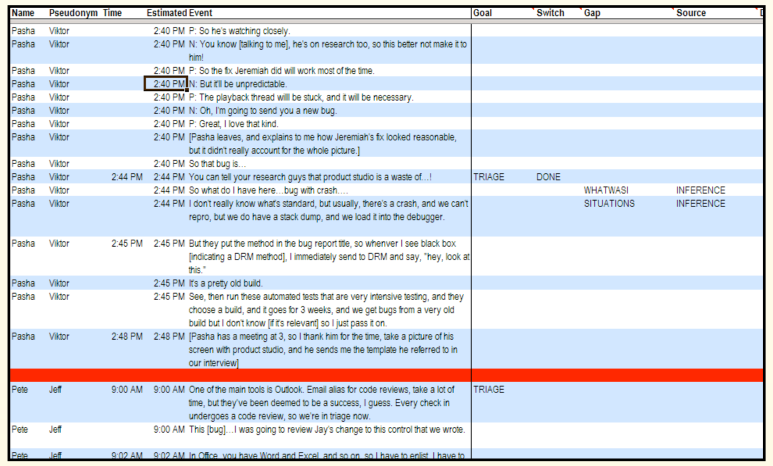
information needs at Microsoft

observed **25 hours** of coding and bug fixing, in the role of “new hires”



357 pages of handwritten notes

4,231 events in an spreadsheet



Name	Pseudonym	Time	Estimated	Event	Goal	Switch	Gap	Source
Pasha	Viktor		2:40 PM	P: So he's watching closely.				
Pasha	Viktor		2:40 PM	N: You know [talking to me], he's on research too, so this better not make it to him!				
Pasha	Viktor		2:40 PM	P: So the fix Jeremiah did will work most of the time.				
Pasha	Viktor		2:40 PM	N: But it'll be unpredictable.				
Pasha	Viktor		2:40 PM	P: The playback thread willl be stuck, and it will be necessary.				
Pasha	Viktor		2:40 PM	N: Oh, I'm going to send you a new bug.				
Pasha	Viktor		2:40 PM	P: Great, I love that kind.				
Pasha	Viktor		2:40 PM	[Pasha leaves, and explains to me how Jeremiah's fix looked reasonable, but it didn't really account for the whole picture.]				
Pasha	Viktor		2:40 PM	So that bug is…				
Pasha	Viktor	2:44 PM	2:44 PM	You can tell your research guys that product studio is a waste of…!	TRIAGE	DONE		
Pasha	Viktor		2:44 PM	So what do I have here…bug with crash….			WHATWASI	INFERENCE
Pasha	Viktor		2:44 PM	I don't really know what's standard, but usually, there's a crash, and we can't repro, but we do have a stack dump, and we load it into the debugger.			SITUATIONS	INFERENCE
Pasha	Viktor	2:45 PM	2:45 PM	But they put the method in the bug report title, so whenver I see black box [indicating a DRM method], I immediately send to DRM and say, "hey, look at this."				
Pasha	Viktor		2:45 PM	It's a pretty old build.				
Pasha	Viktor		2:45 PM	See, then run these automated tests that are very intensive testing, and they choose a build, and it goes for 3 weeks, and we get bugs from a very old build but I don't know [if it's relevant] so I just pass it on.				
Pasha	Viktor	2:48 PM	2:48 PM	[Pasha has a meeting at 3, so I thank him for the time, take a picture of his screen with product studio, and he sends me the template he referred to in our interview]				
Pete	Jeff	9:00 AM	9:00 AM	One of the main tools is Outlook. Email alias for code reviews, take a lot of time, but they've been deemed to be a success, I guess. Every check in undergoes a code review, so we're in triage now.	TRIAGE			
Pete	Jeff		9:00 AM	This [bug]…I was going to review Jay's change to this control that we wrote.				
Pete	Jeff	9:02 AM	9:02 AM	In Office, you have Word and Excel, and so on, so I have to enlist. I have to				

information needs at Microsoft

information needs at Microsoft

what code caused this program state?
why was this code implemented this way?
what code could have caused this behavior?
in what situations does this failure occur?
have resources I depend on changed?
what is the program supposed to do?
what have my coworkers been doing?
how do I use this data structure or function?
did I make any mistakes?
is this problem worth fixing?
what's statically related to this code?
what are the implications of this change?

most common unsatisfied needs

	% unsatisfied	max time
what code caused this program state?	**61%**	**21 min**
why was this code implemented this way?	**44%**	**21 min**
what code could have caused this behavior?	**36%**	**17 min**

- relied heavily on **coworkers** to answer questions
- long periods of hypothesis **refinement**
- experts explored many hypotheses in parallel

summary

- program understanding is **hypothesis-driven**…

    people ask **'why' questions** about **program output**

    most **initial hypotheses** are **incorrect**

    incorrect hypotheses can lead to **new bugs**, **misunderstandings** about program **execution**

- true for novices, end-users, Java programmers, industry developers

the problem

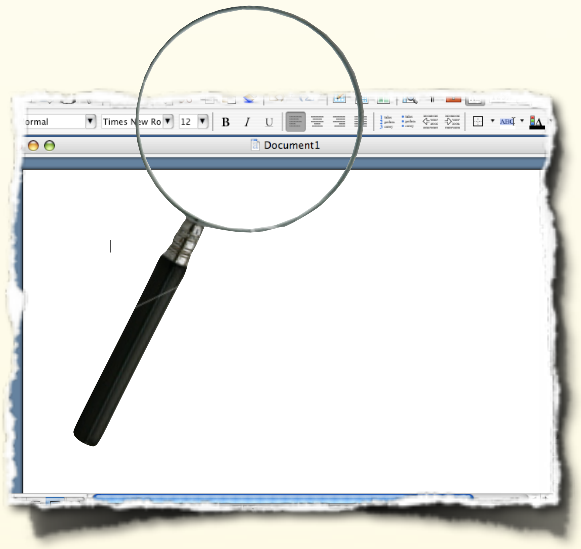
today's tools **require** people to ***guess***
what **code** is responsible

the problem

today's tools **require** people to ***guess*** what **code** is responsible

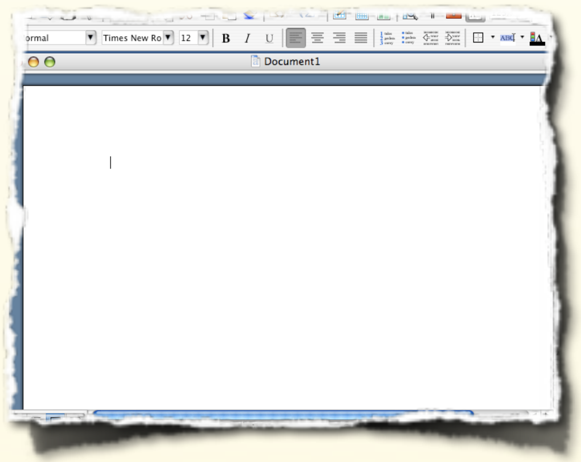
the idea

what if people could **point** to **output** and see the **code** responsible?



the idea

what if people could **point** to **output** and see the **code** responsible?



outline

problem

studies

the whyline

implementation

evaluation

conclusions

outline

problem

studies

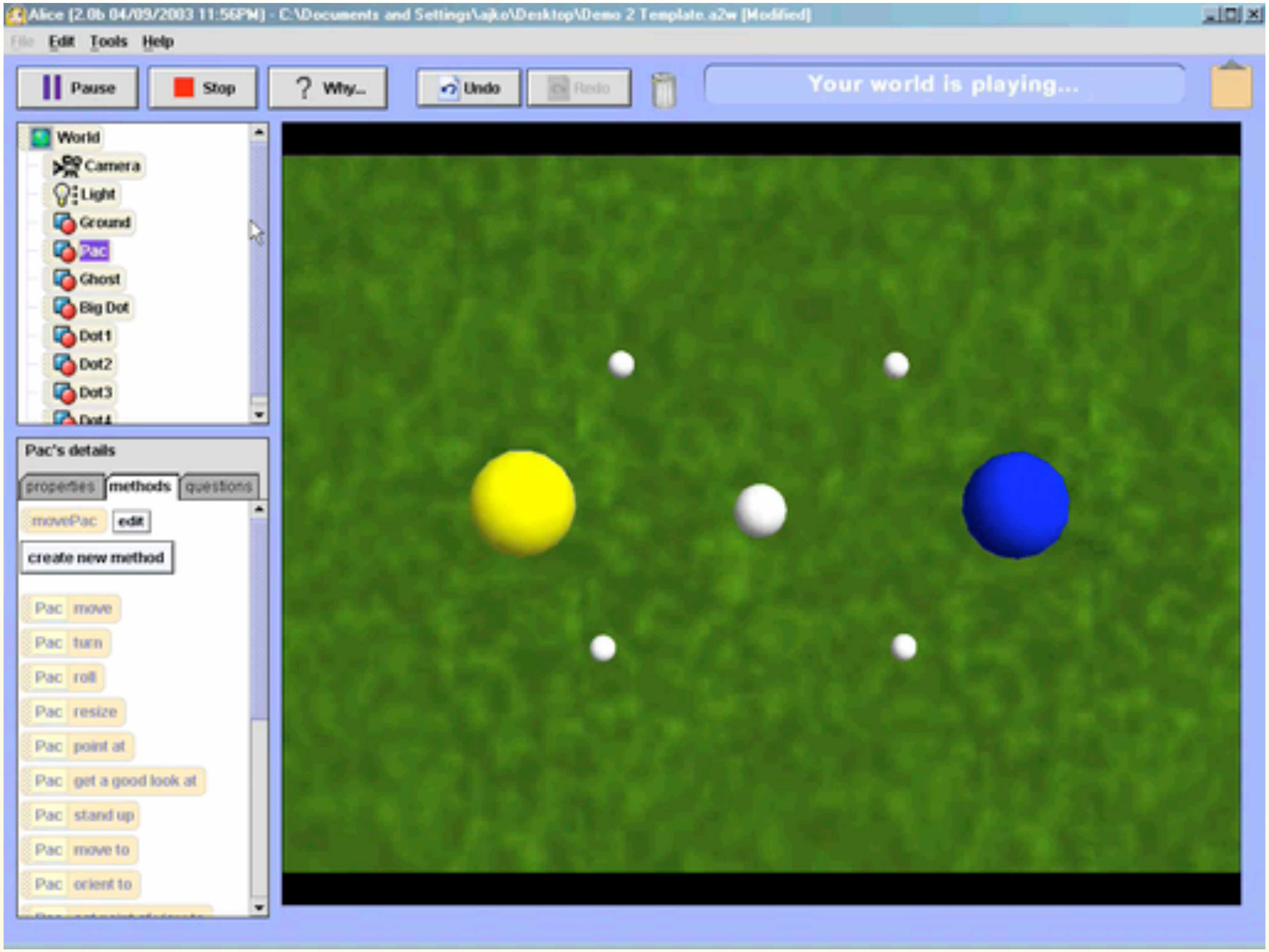
the whyline

implementation

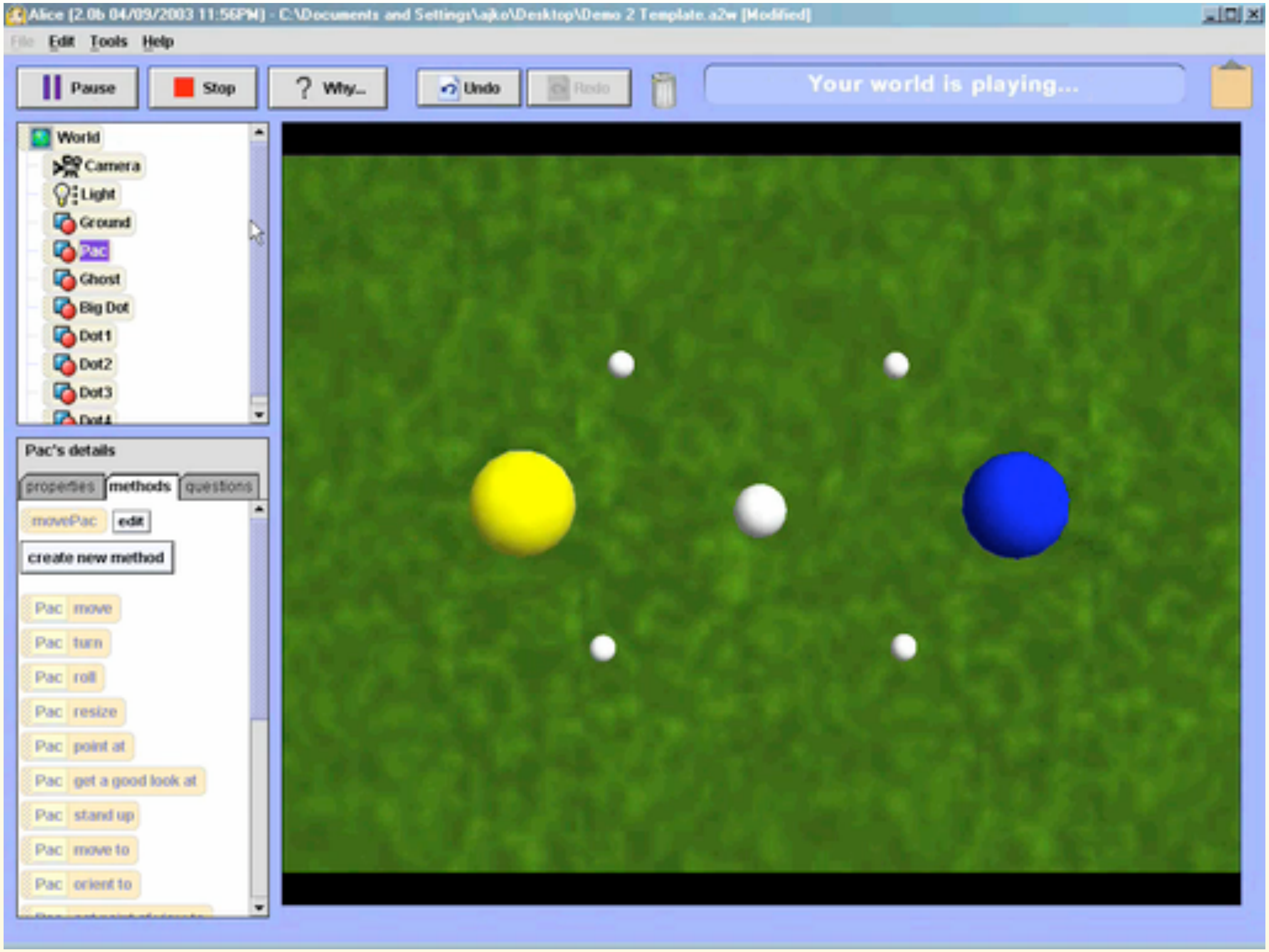
evaluation

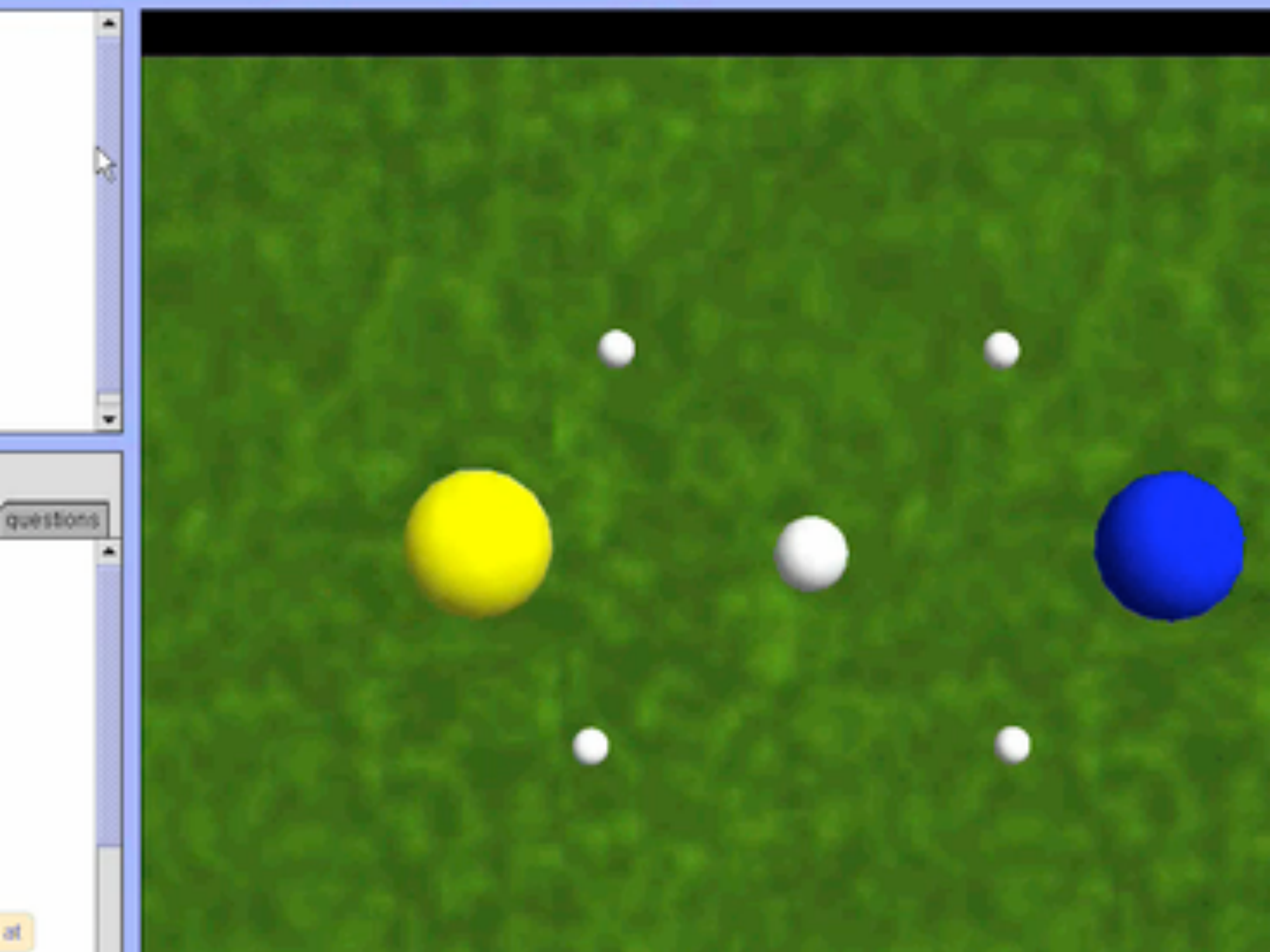
conclusions

a whyline for Alice

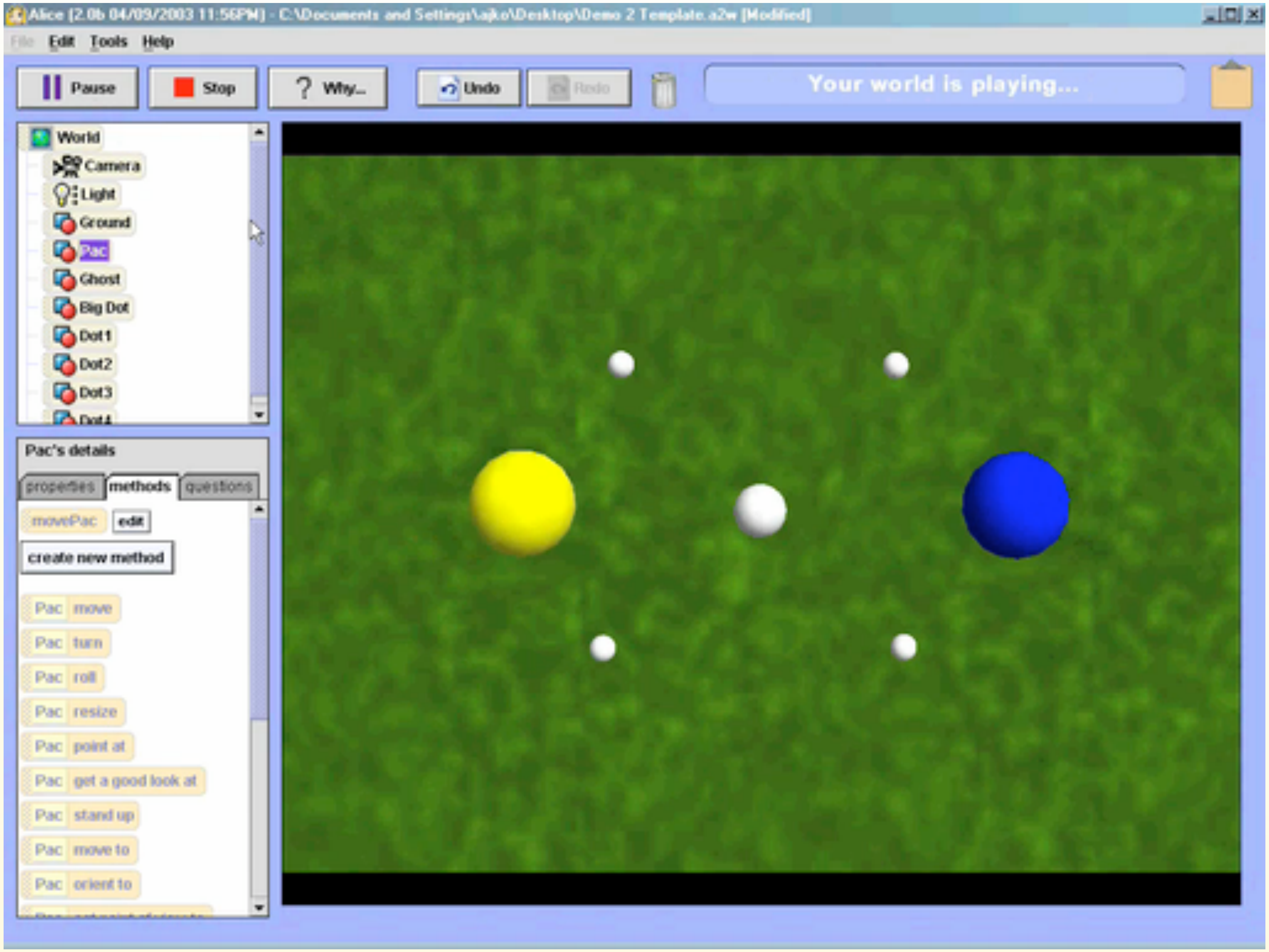


a whyline for Alice

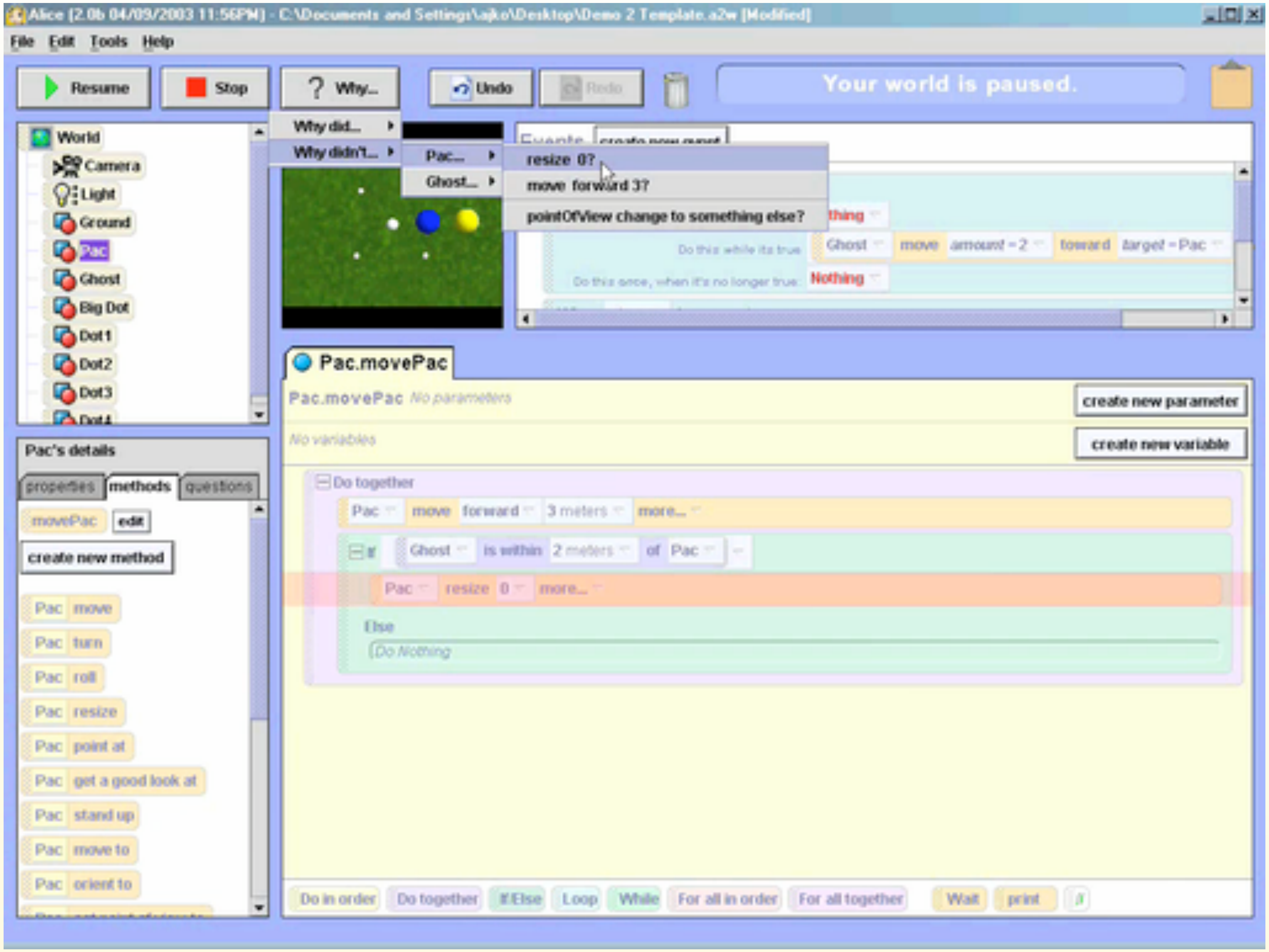



questions
at

a whyline for Alice



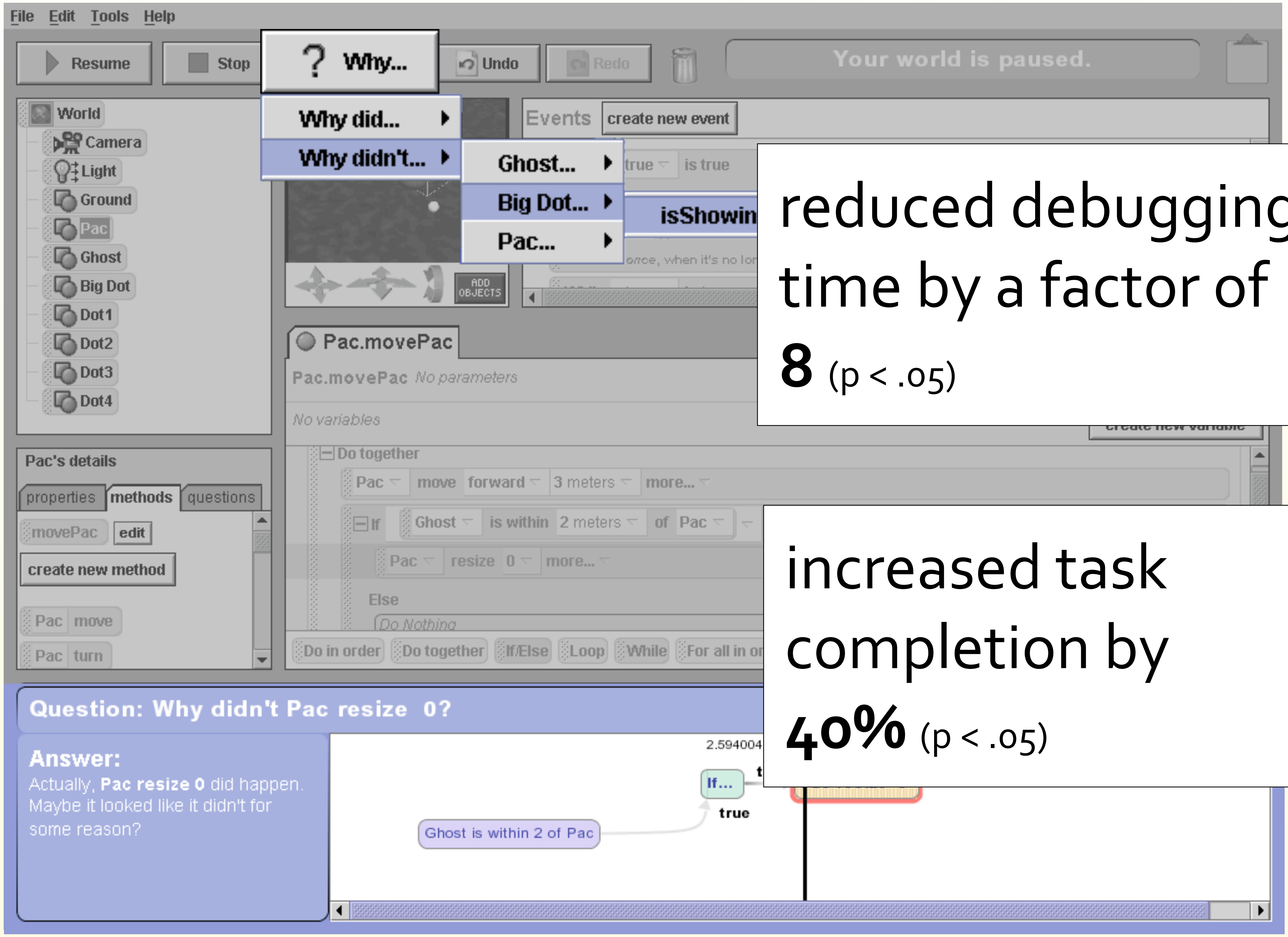
a whyline for Alice



```
Do this while its true   Ghost  move  amount = 2  toward  target = Pac
Do this once, when it's no longer true:  Nothing
```

```
Pac.movePac
Pac.movePac  No parameters
No variables
Do together
    Pac  move  forward  3 meters  more...
    If  Ghost  is within  2 meters  of  Pac
        Pac  resize  0  more...
    Else
        Do Nothing
```

a whyline for Alice



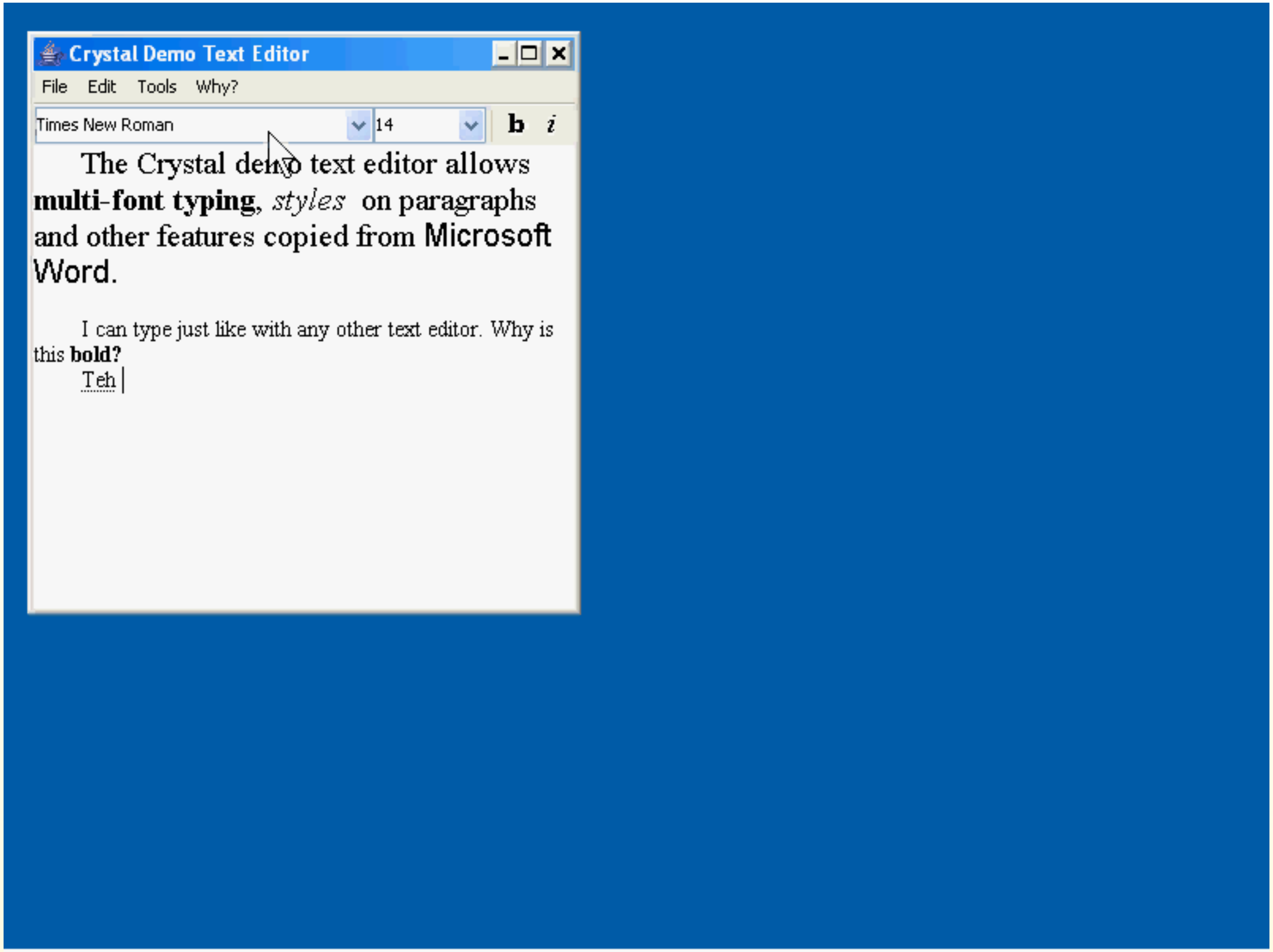
Question: Why didn't Pac resize 0?

Answer:
Actually, **Pac resize 0** did happen. Maybe it looked like it didn't for some reason?

reduced debugging time by a factor of **8** (p < .05)

increased task completion by **40%** (p < .05)

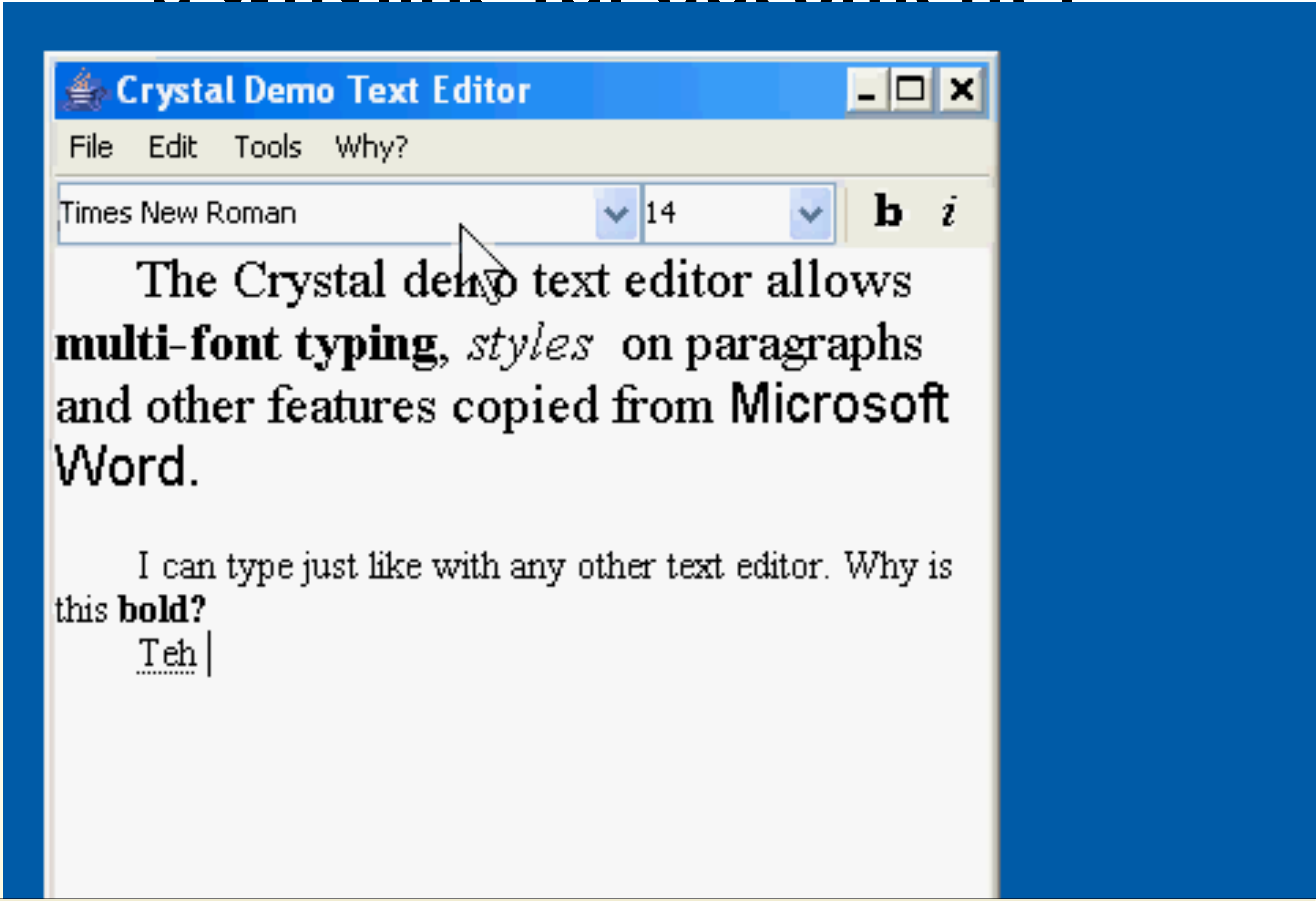
a whyline for documents



The Crystal demo text editor allows **multi-font typing**, *styles* on paragraphs and other features copied from Microsoft Word.

I can type just like with any other text editor. Why is this **bold?**
Teh

a whyline for documents



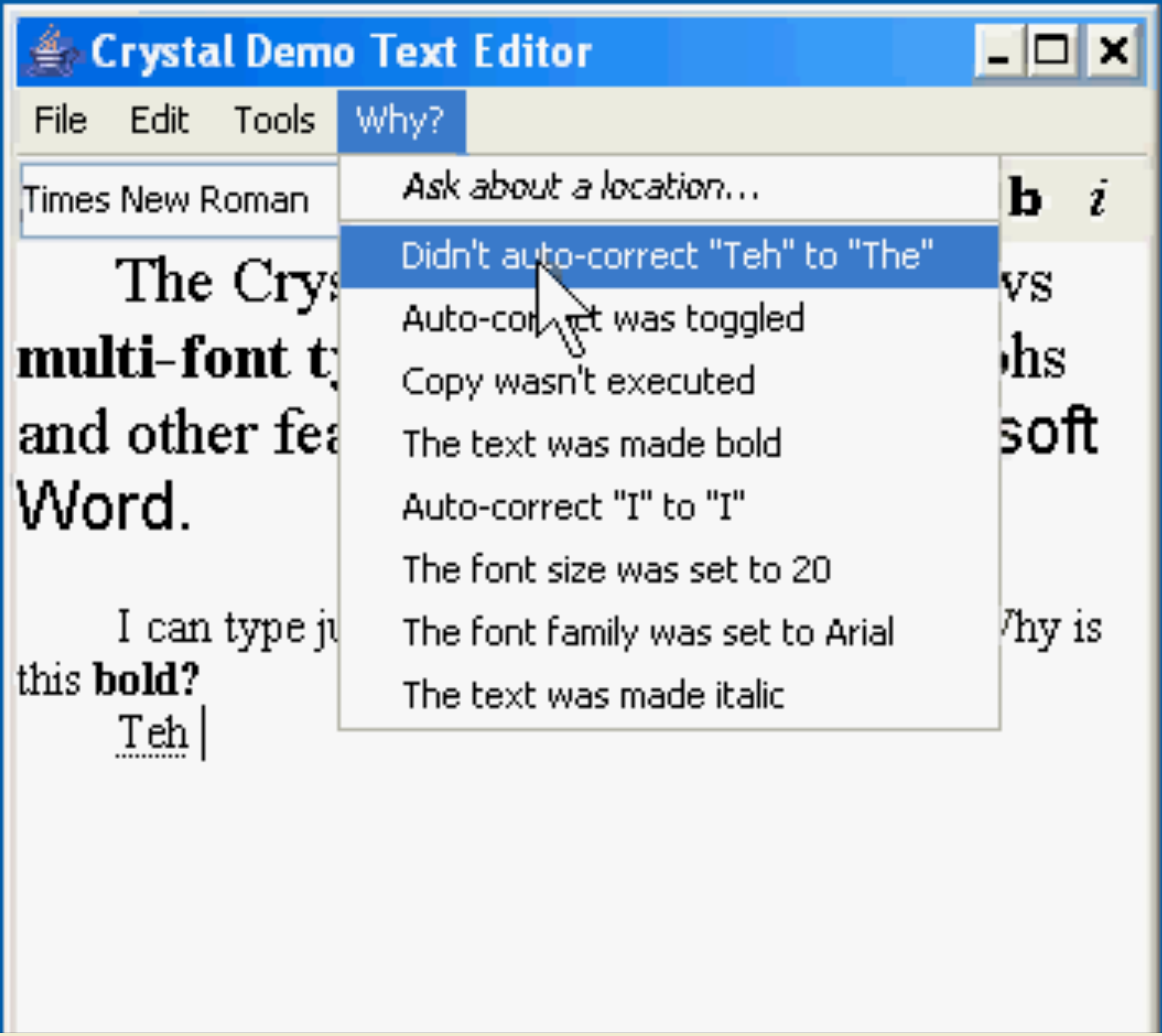
Times New Roman 14

The Crystal demo text editor allows **multi-font typing**, *styles* on paragraphs and other features copied from Microsoft Word.

I can type just like with any other text editor. Why is this **bold?**

Teh

a whyline for documents



Ask about a location...

Didn't auto-correct "Teh" to "The"

Auto-correct was toggled

Copy wasn't executed

The text was made bold

Auto-correct "I" to "I"

The font size was set to 20

The font family was set to Arial

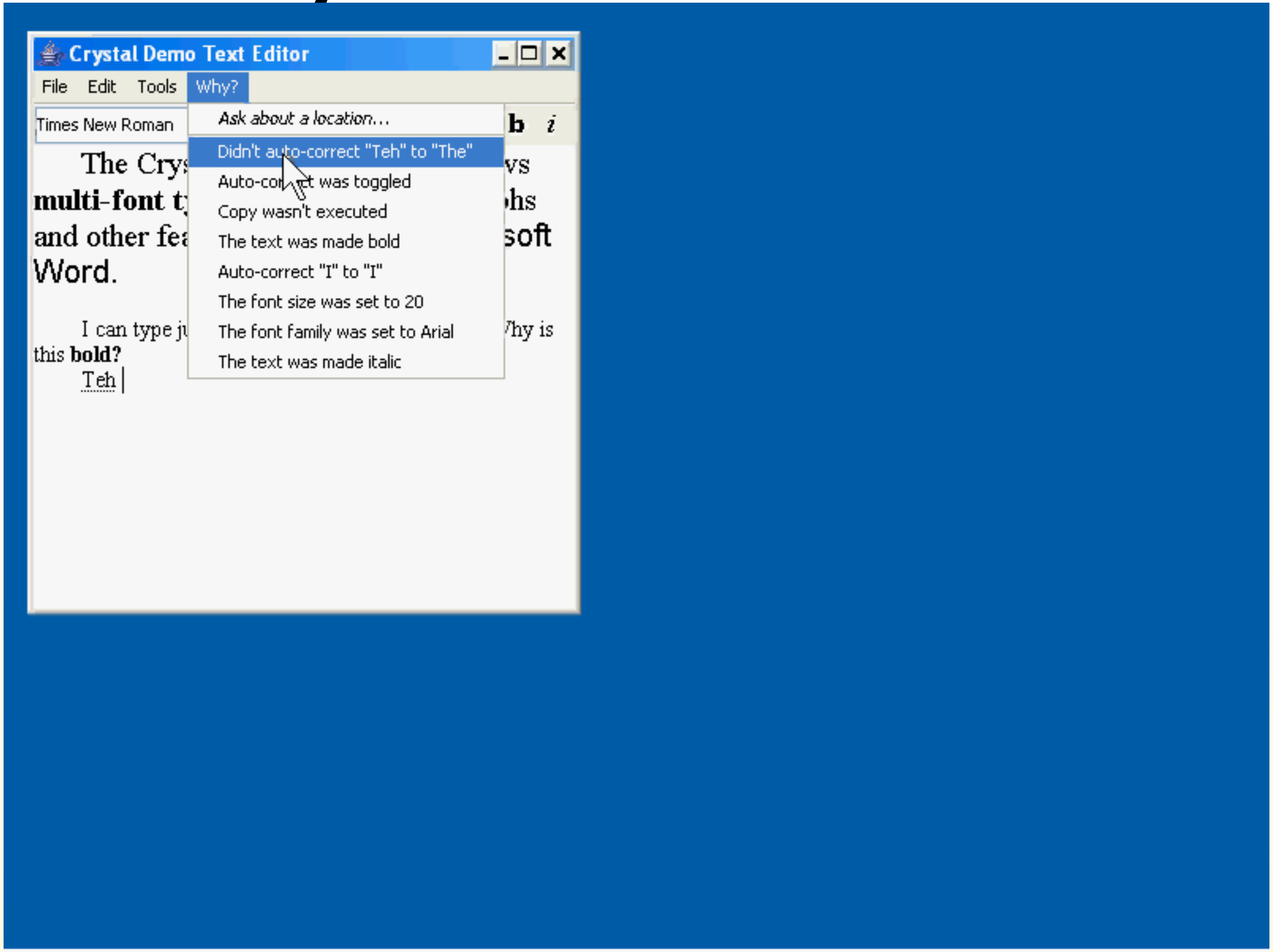
The text was made italic

The Crys … vs **multi-font t** … hs and other fea … soft Word.

I can type ju … Why is this **bold?**

Teh

a whyline for documents



Ask about a location...
Didn't auto-correct "Teh" to "The"
Auto-correct was toggled
Copy wasn't executed
The text was made bold
Auto-correct "I" to "I"
The font size was set to 20
The font family was set to Arial
The text was made italic

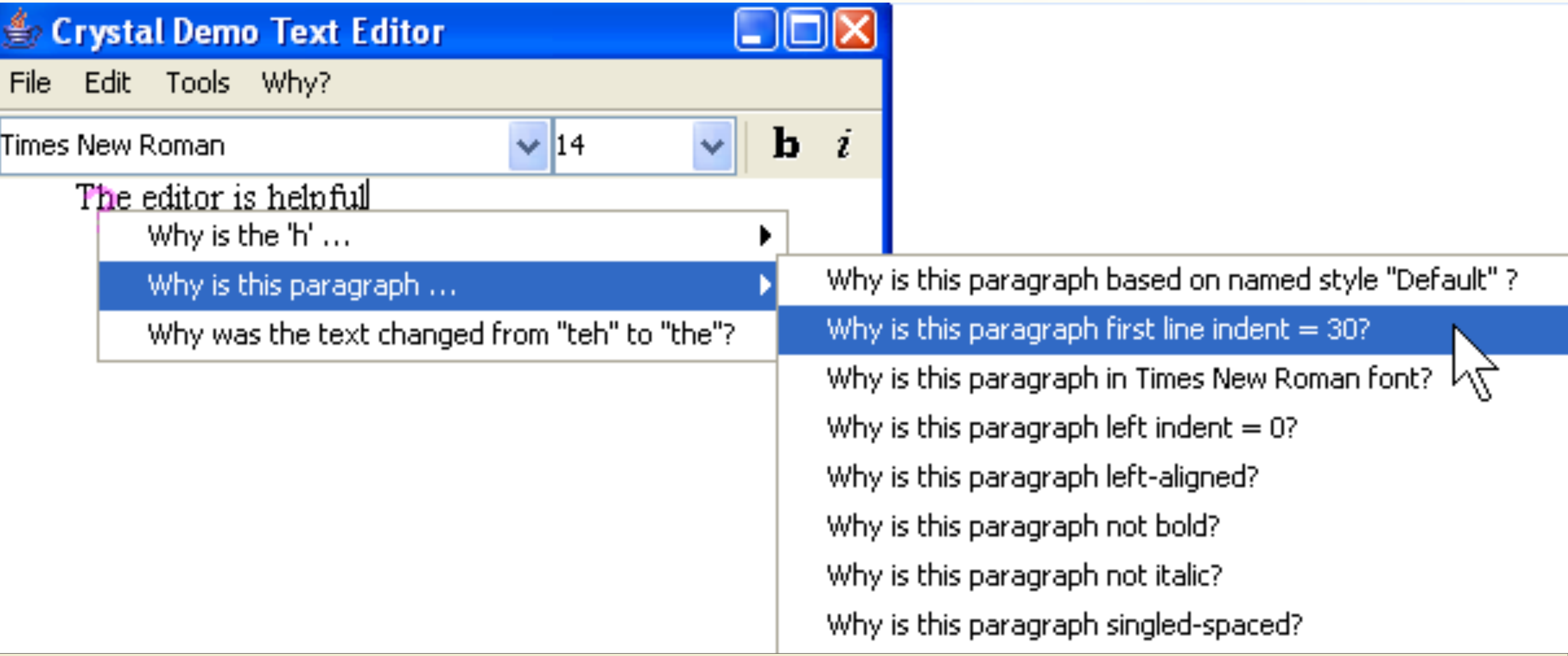
The Crys vs
multi-font t hs
and other fea soft
Word.

I can type ju Why is
this **bold?**
Teh

a whyline for documents

users completed tasks
20% faster (p < .05)

users completed
30% more tasks (p < .05)



Crystal Demo Text Editor

File Edit Tools Why?

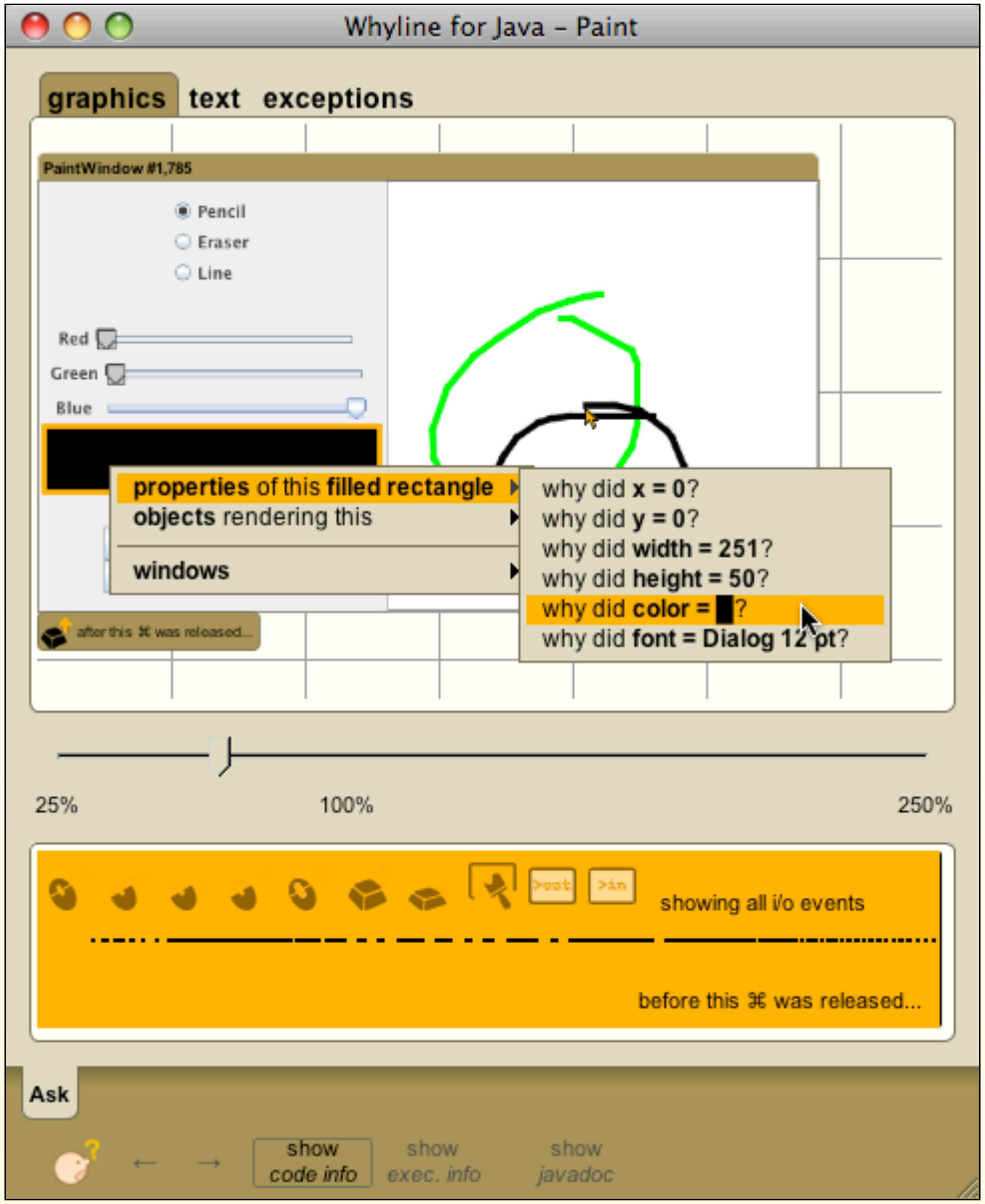
Times New Roman 14 b i

The editor is helpful

Why is the 'h' …
Why is this paragraph …
Why was the text changed from "teh" to "the"?

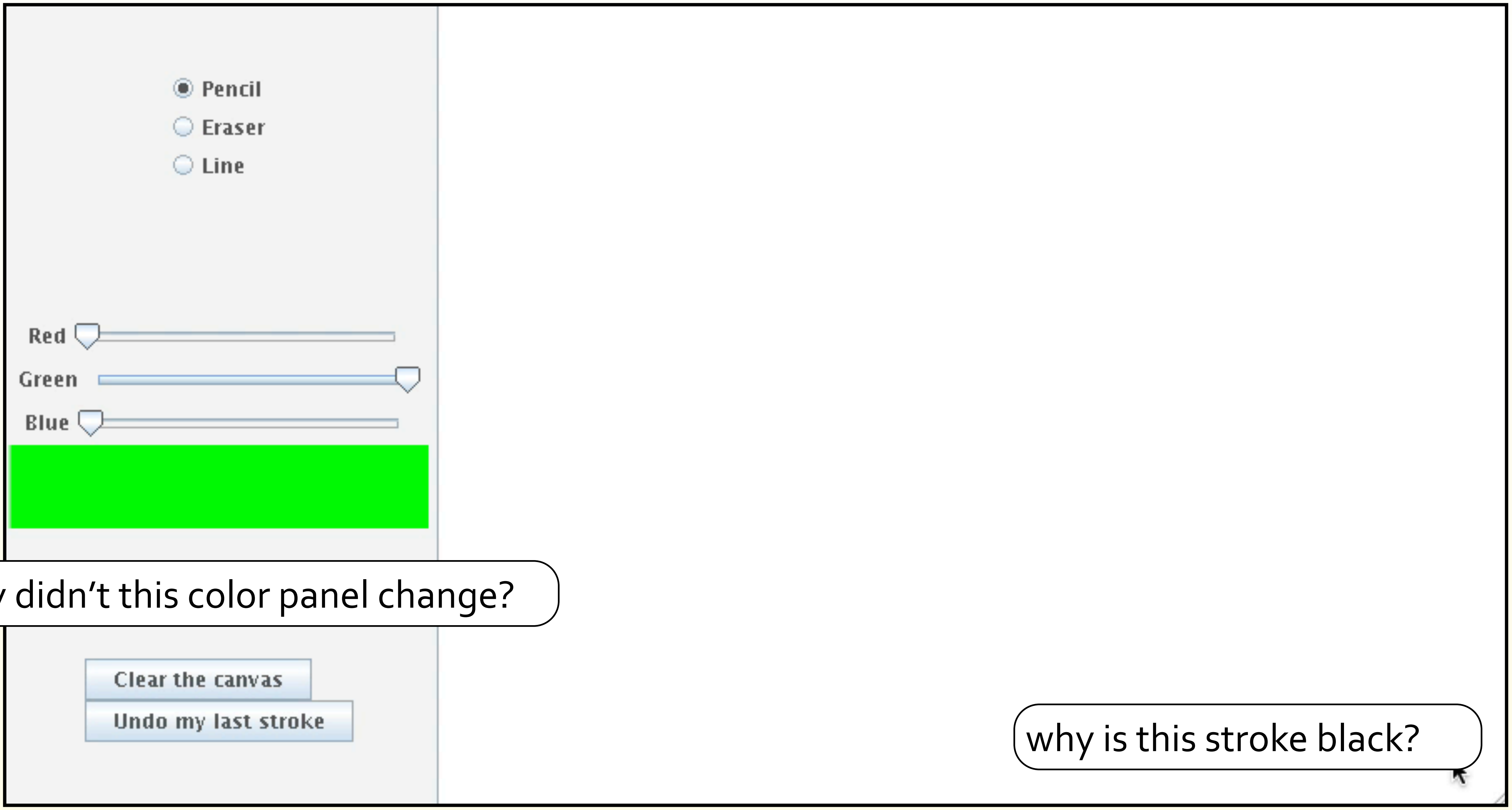
Why is this paragraph based on named style "Default" ?
Why is this paragraph first line indent = 30?
Why is this paragraph in Times New Roman font?
Why is this paragraph left indent = 0?
Why is this paragraph left-aligned?
Why is this paragraph not bold?
Why is this paragraph not italic?
Why is this paragraph singled-spaced?

a whyline for Java



a bug

a bug



why didn’t this color panel change?

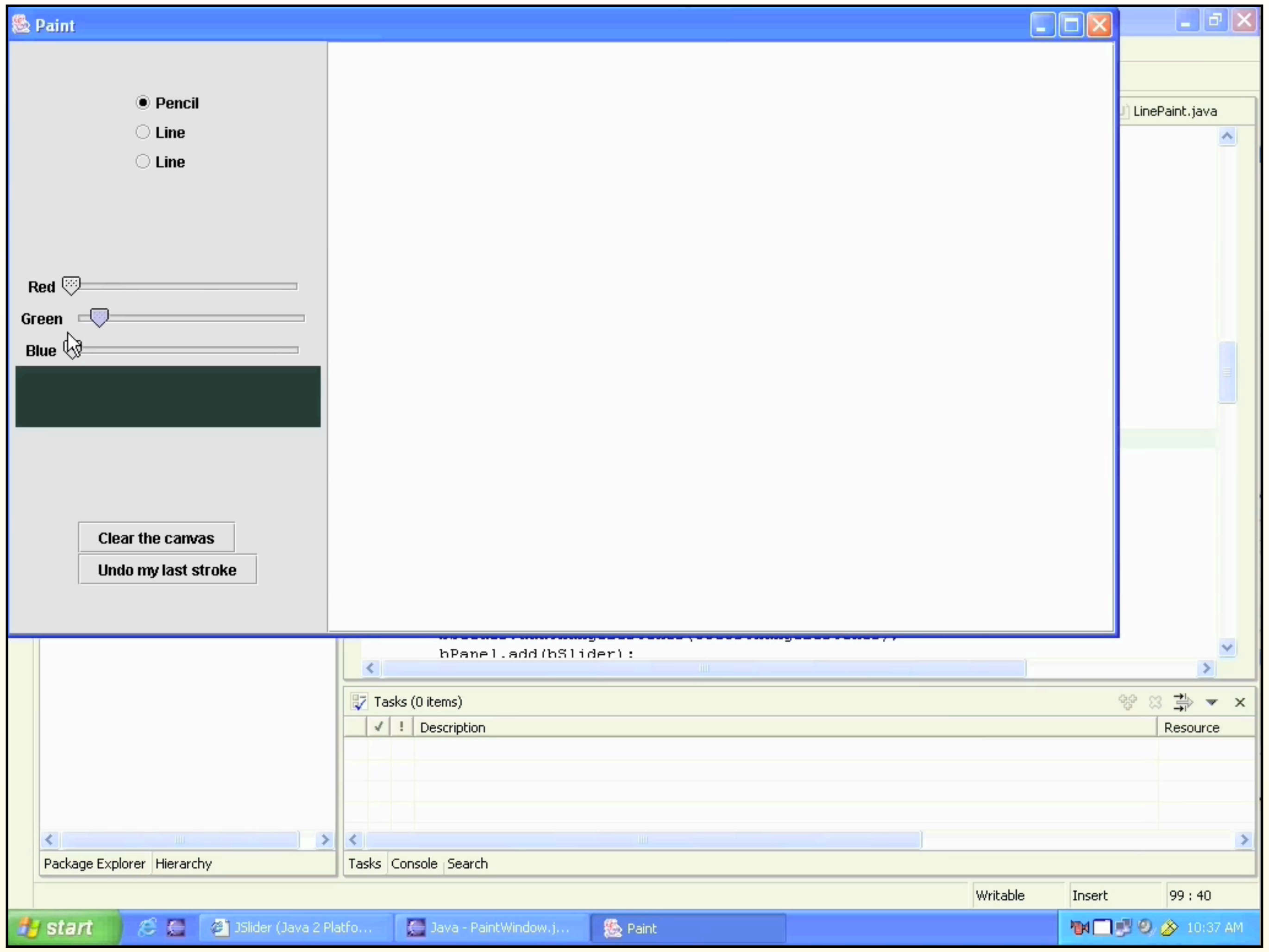
why is this stroke black?

what normally happens

why is the stroke black?

what normally happens

why is the stroke black?



```
bPanel.add(bSlider);
```

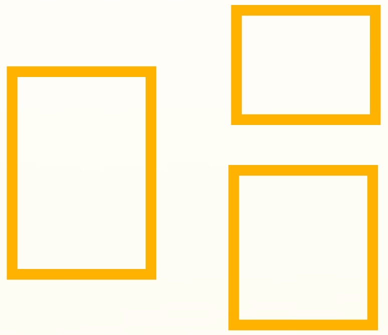
~~maybe it's a slider initialization problem…~~

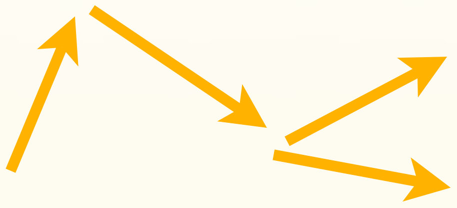
~~maybe the slider isn't connected to anything…~~

~~is the JSlider argument incorrect?~~

maybe the color isn't computed properly…

stumbled onto bug accidentally





breakpoint ●

println()

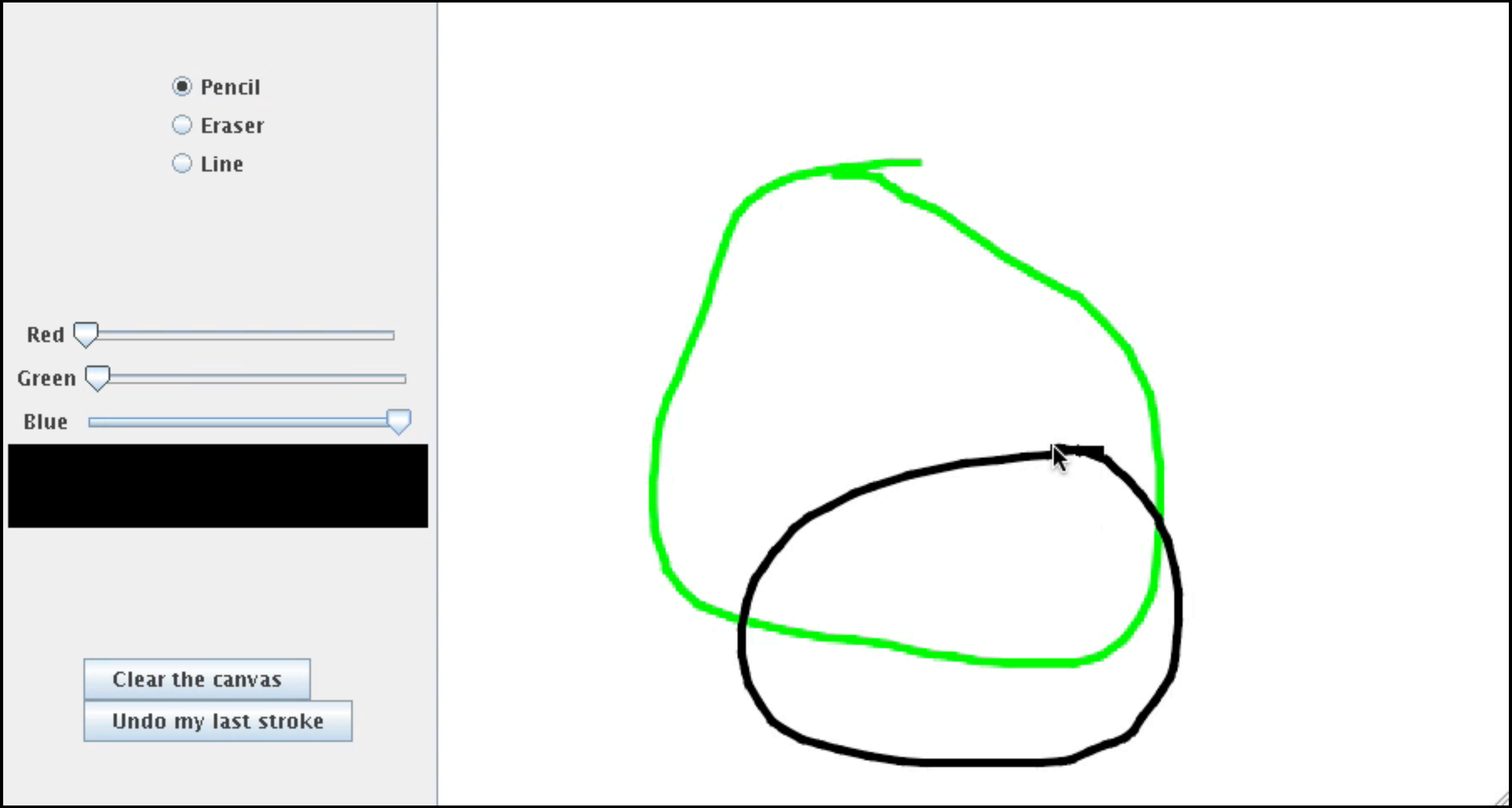
(10 minutes, 27× speed)

whyline demo

at least two ways to ask this question …

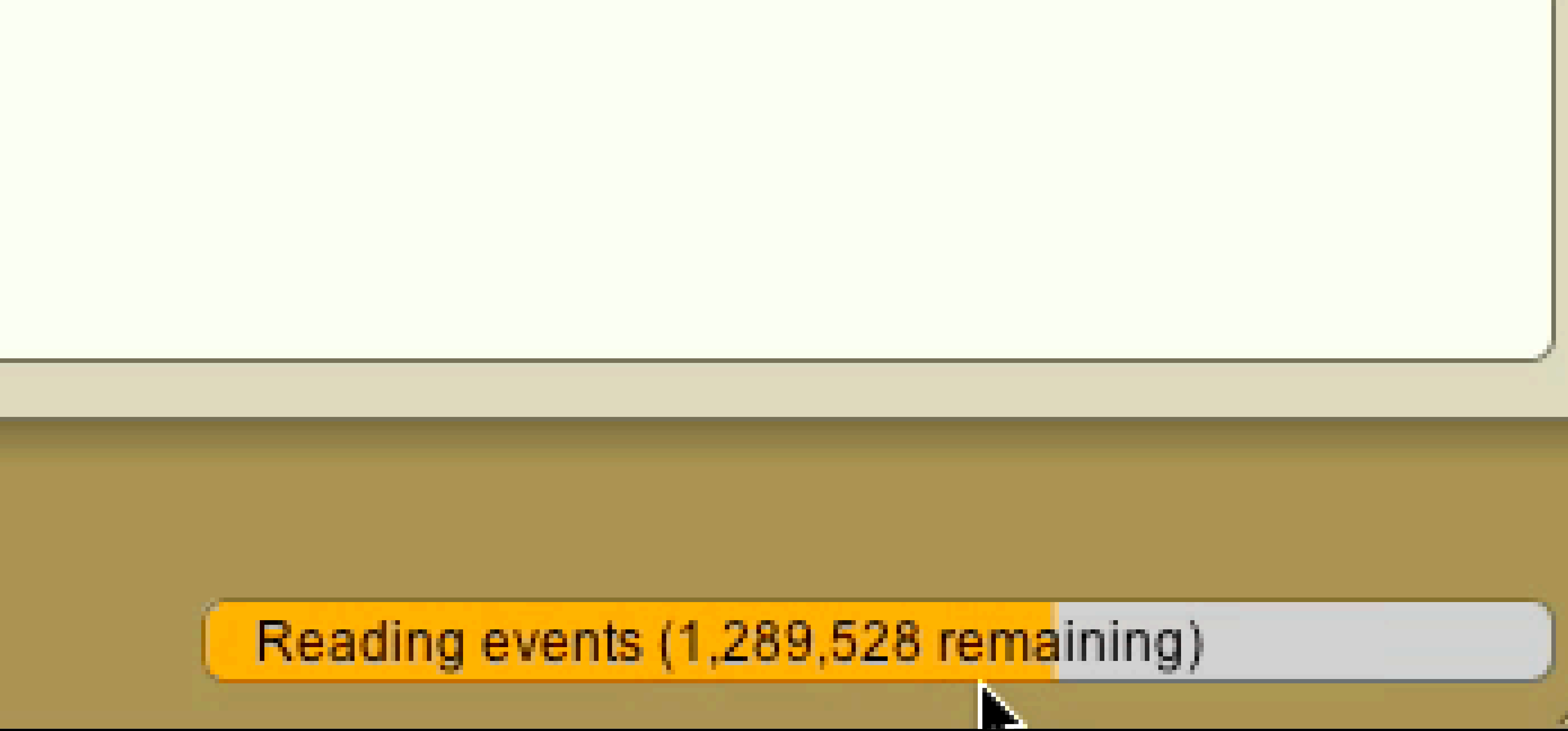
why was the **`line`** color black?

why didn't the **`color panel`** repaint?



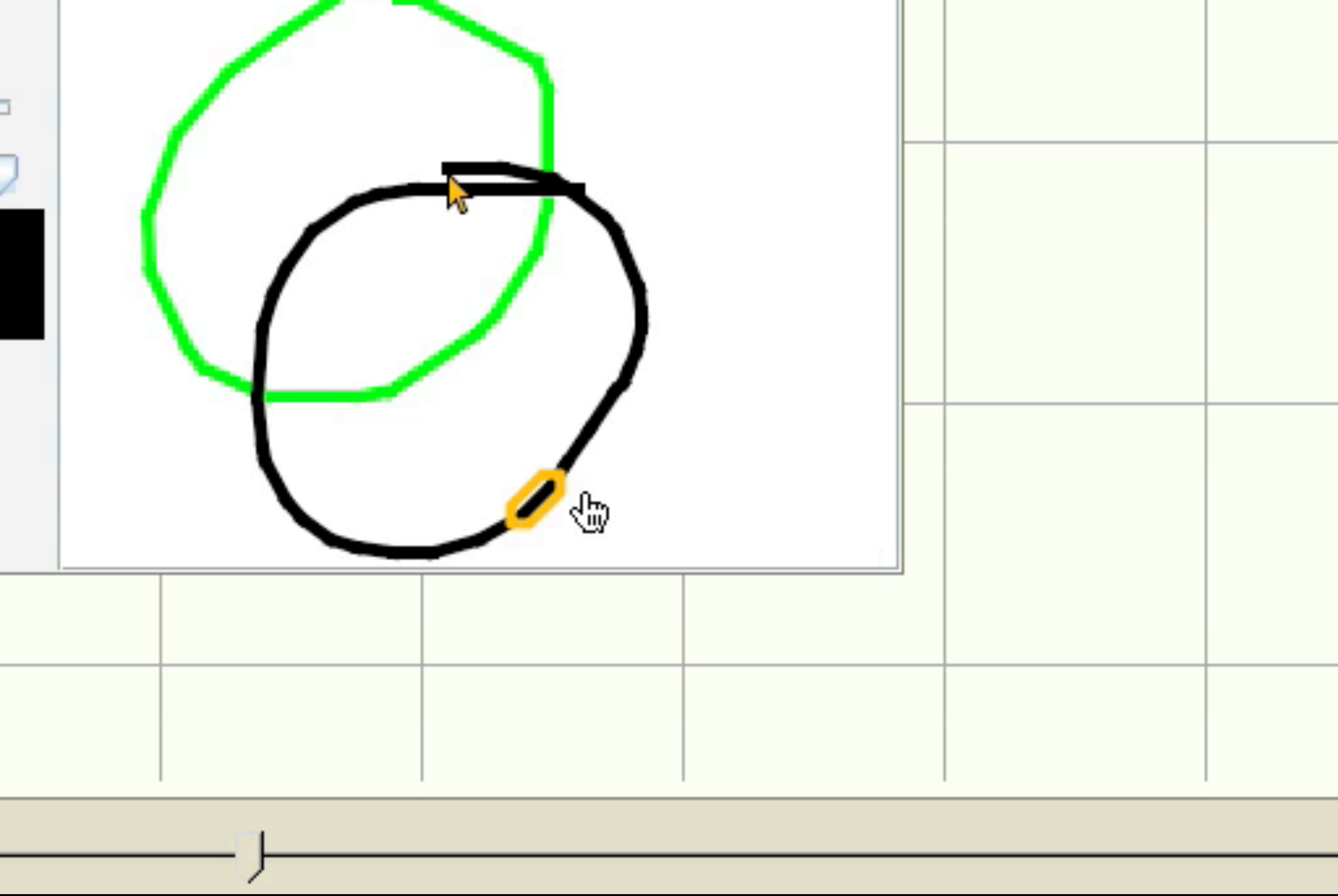
Selected tool: Pencil [✓]

record the problem


Reading events (1,289,528 remaining)

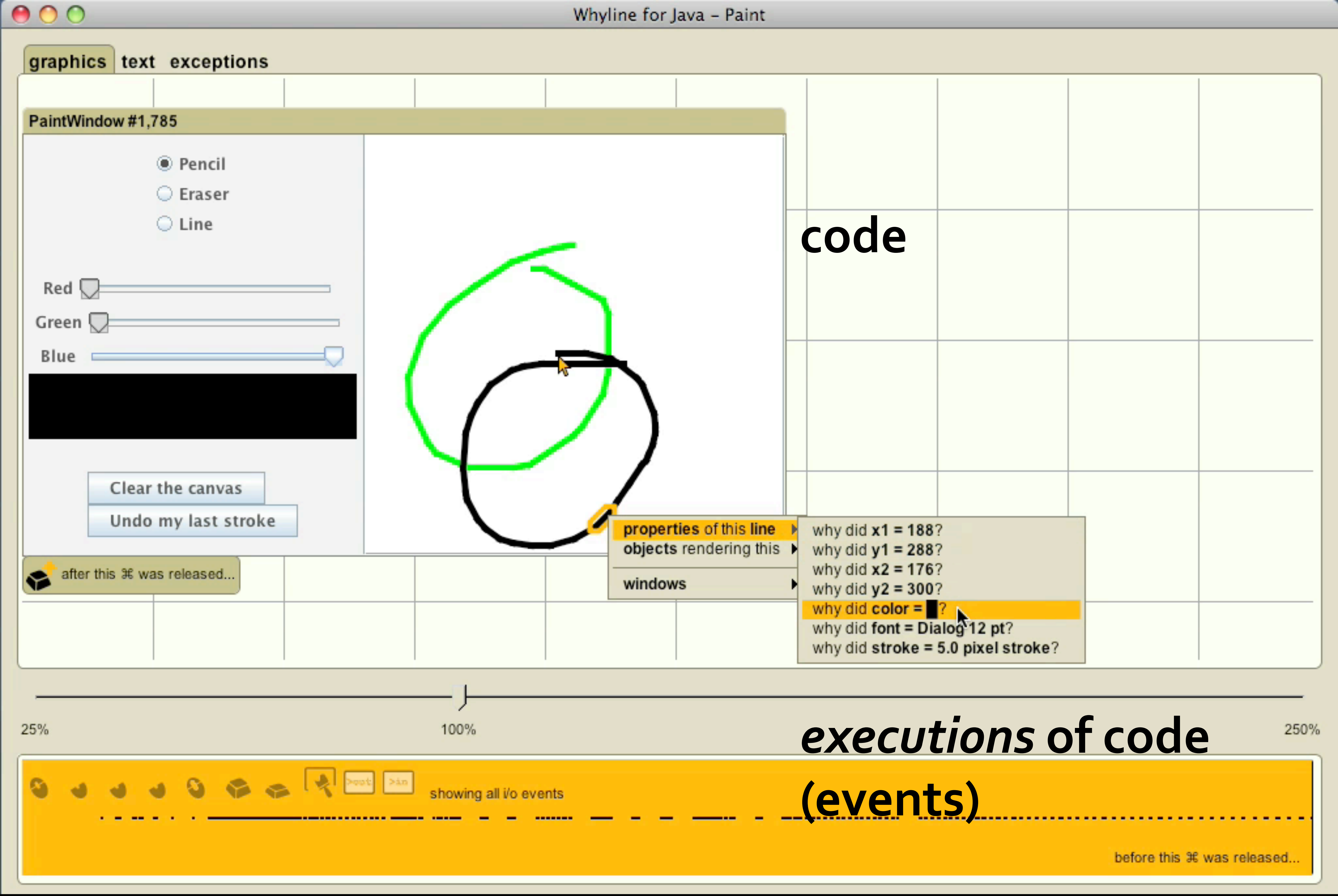
load the recording

why was the line color black?



why was the line color black?

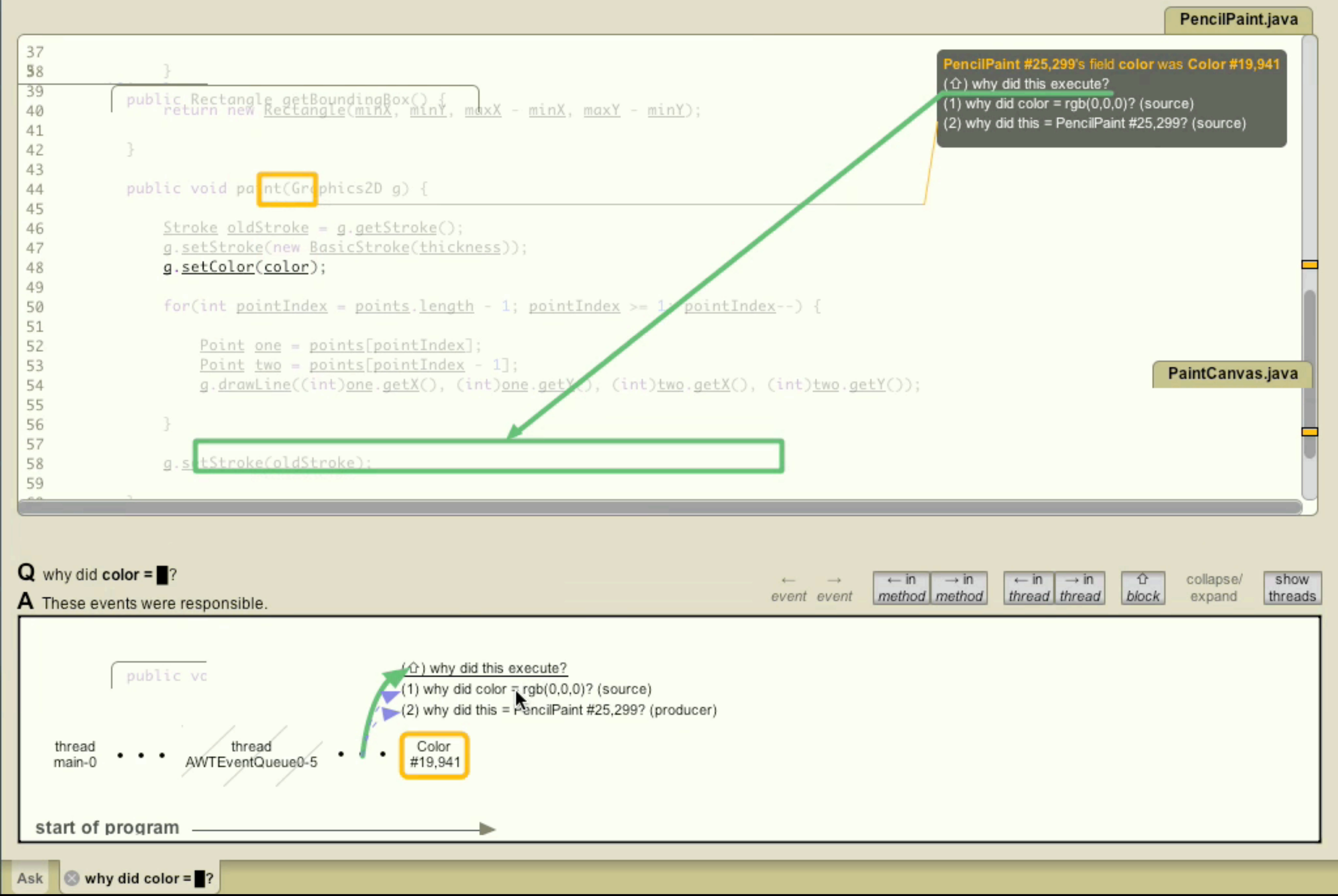
why was the line color black?



code

executions of code (events)

why was the line color black?



PencilPaint.java

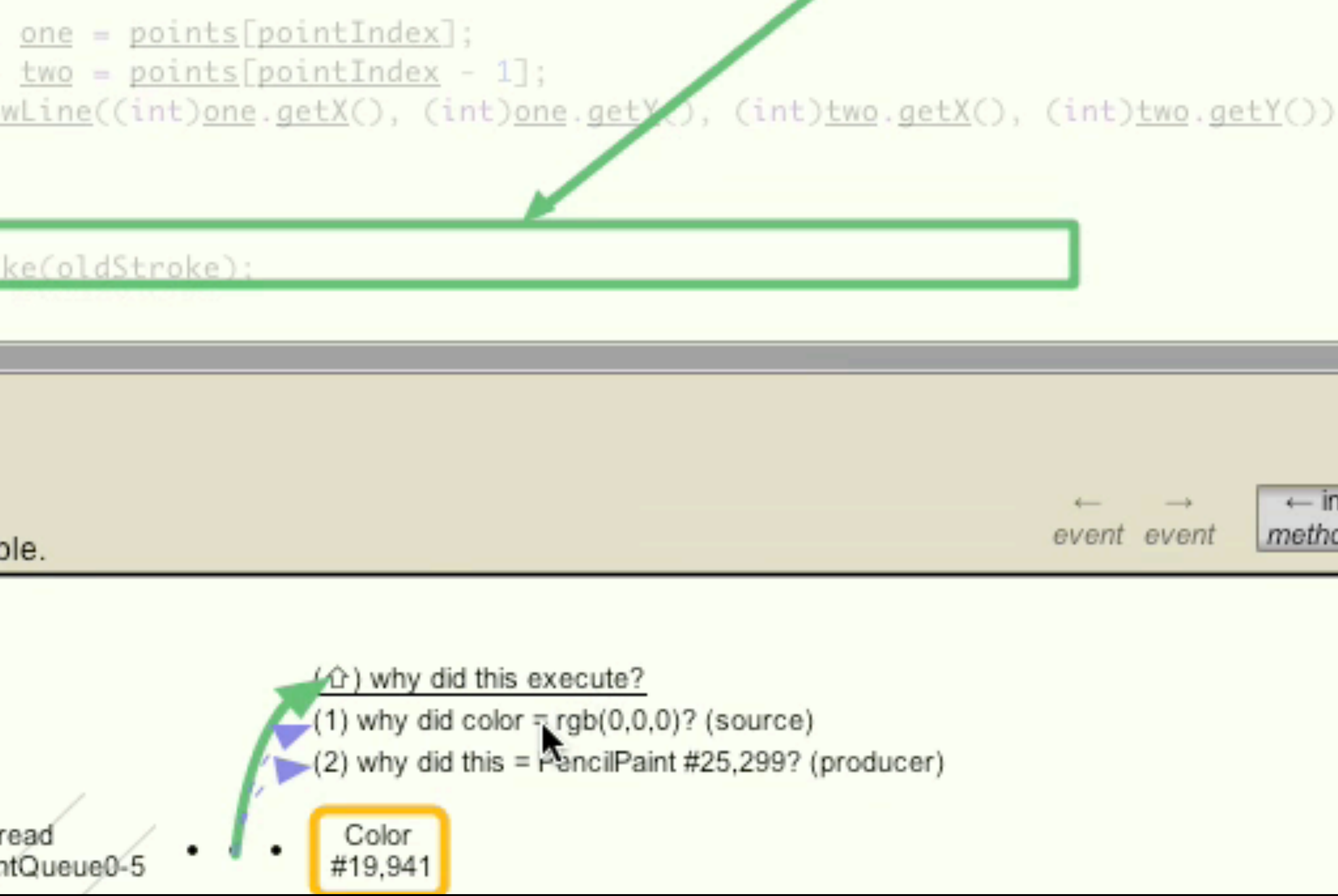
```
        }

    public Rectangle getBoundingBox() {
        return new Rectangle(minX, minY, maxX - minX, maxY - minY);

    }

    public void paint(Graphics2D g) {

        Stroke oldStroke = g.getStroke();
        g.setStroke(new BasicStroke(thickness));
        g.setColor(color);

        for(int pointIndex = points.length - 1; pointIndex >= 1; pointIndex--) {

            Point one = points[pointIndex];
            Point two = points[pointIndex - 1];
            g.drawLine((int)one.getX(), (int)one.getY(), (int)two.getX(), (int)two.getY());

        }

        g.setStroke(oldStroke);
```

PencilPaint #25,299's field color was Color #19,941
(⇧) why did this execute?
(1) why did color = rgb(0,0,0)? (source)
(2) why did this = PencilPaint #25,299? (source)

PaintCanvas.java

Q why did color = █?
A These events were responsible.

why was the line color black?

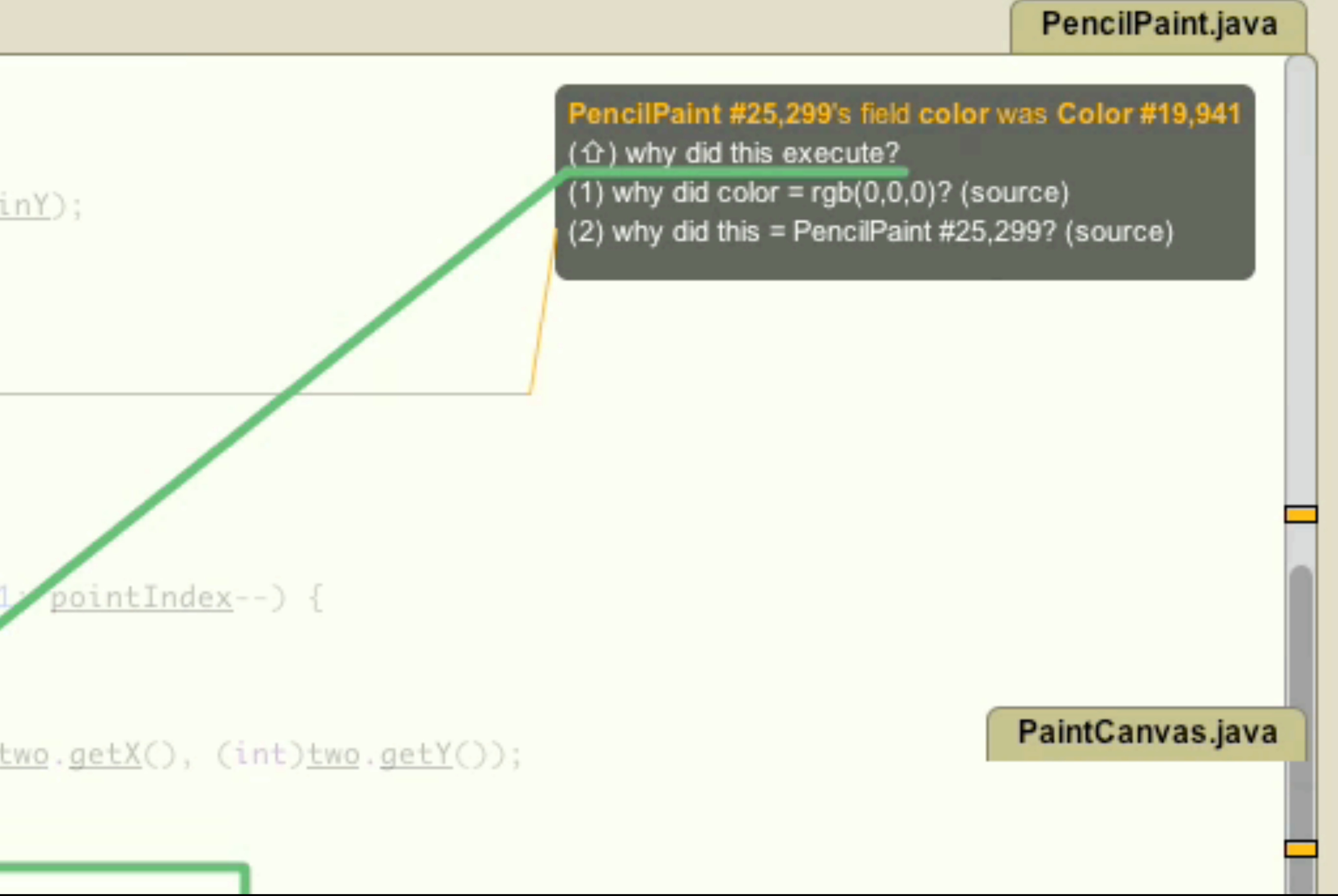


```
 one = points[pointIndex];
 two = points[pointIndex - 1];
wLine((int)one.getX(), (int)one.getY(), (int)two.getX(), (int)two.getY())

ke(oldStroke);
```

ole.

(⇧) why did this execute?
(1) why did color = rgb(0,0,0)? (source)
(2) why did this = PencilPaint #25,299? (producer)

why was the line color black?

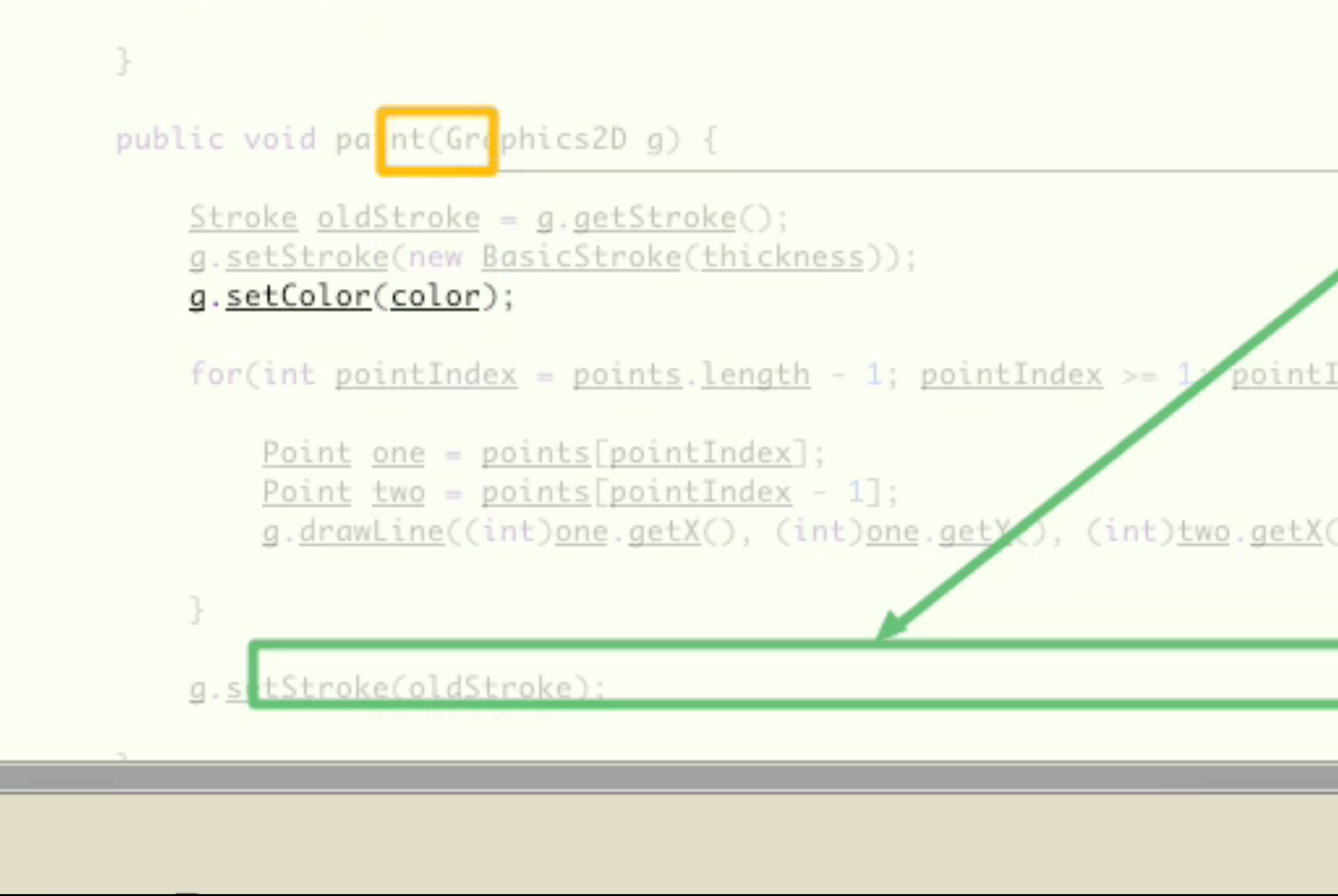

PencilPaint.java

PencilPaint #25,299's field color was Color #19,941
(⇧) why did this execute?
(1) why did color = rgb(0,0,0)? (source)
(2) why did this = PencilPaint #25,299? (source)

```
inY);

1; pointIndex--) {

two.getX(), (int)two.getY());
```

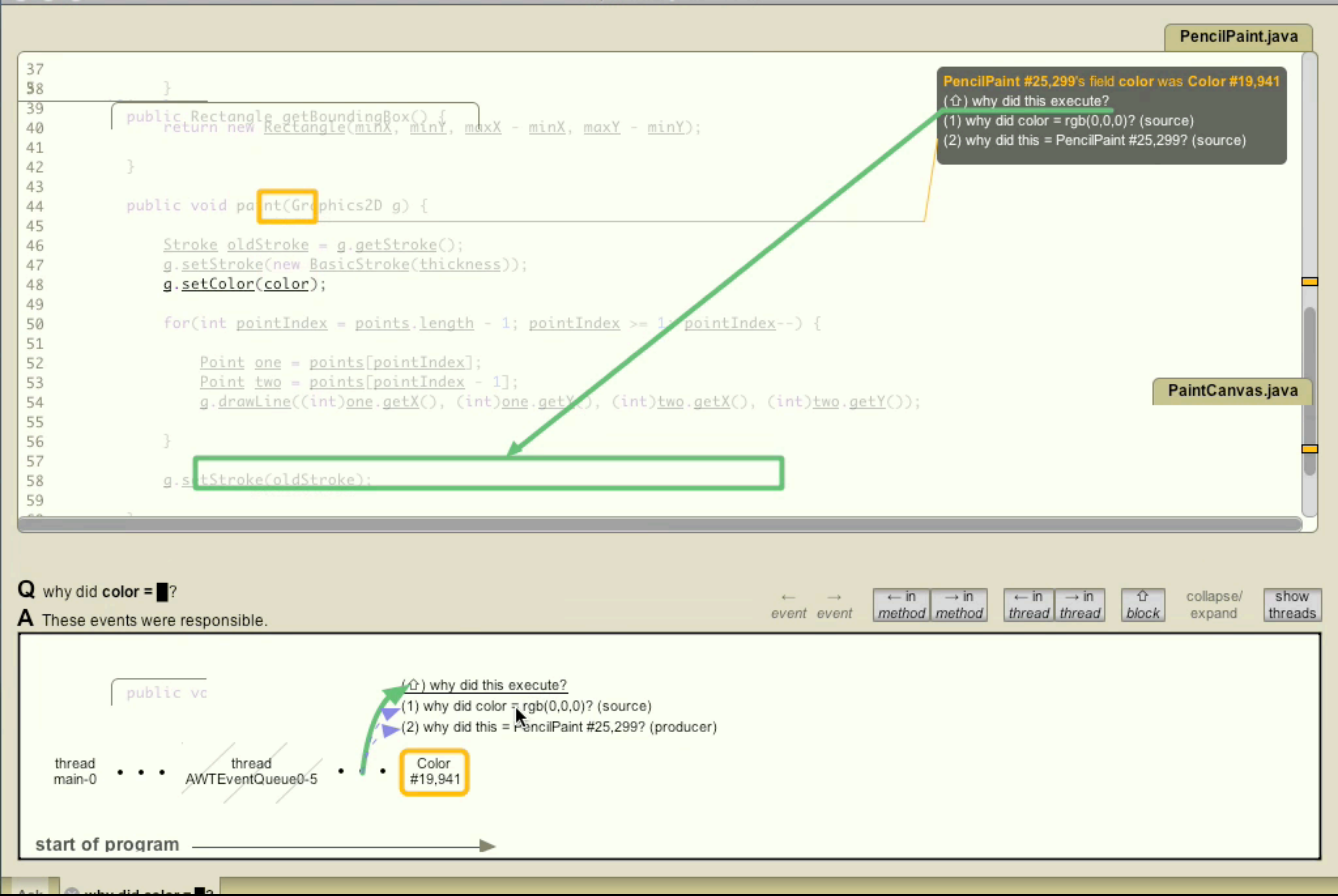
PaintCanvas.java

why was the line color black?



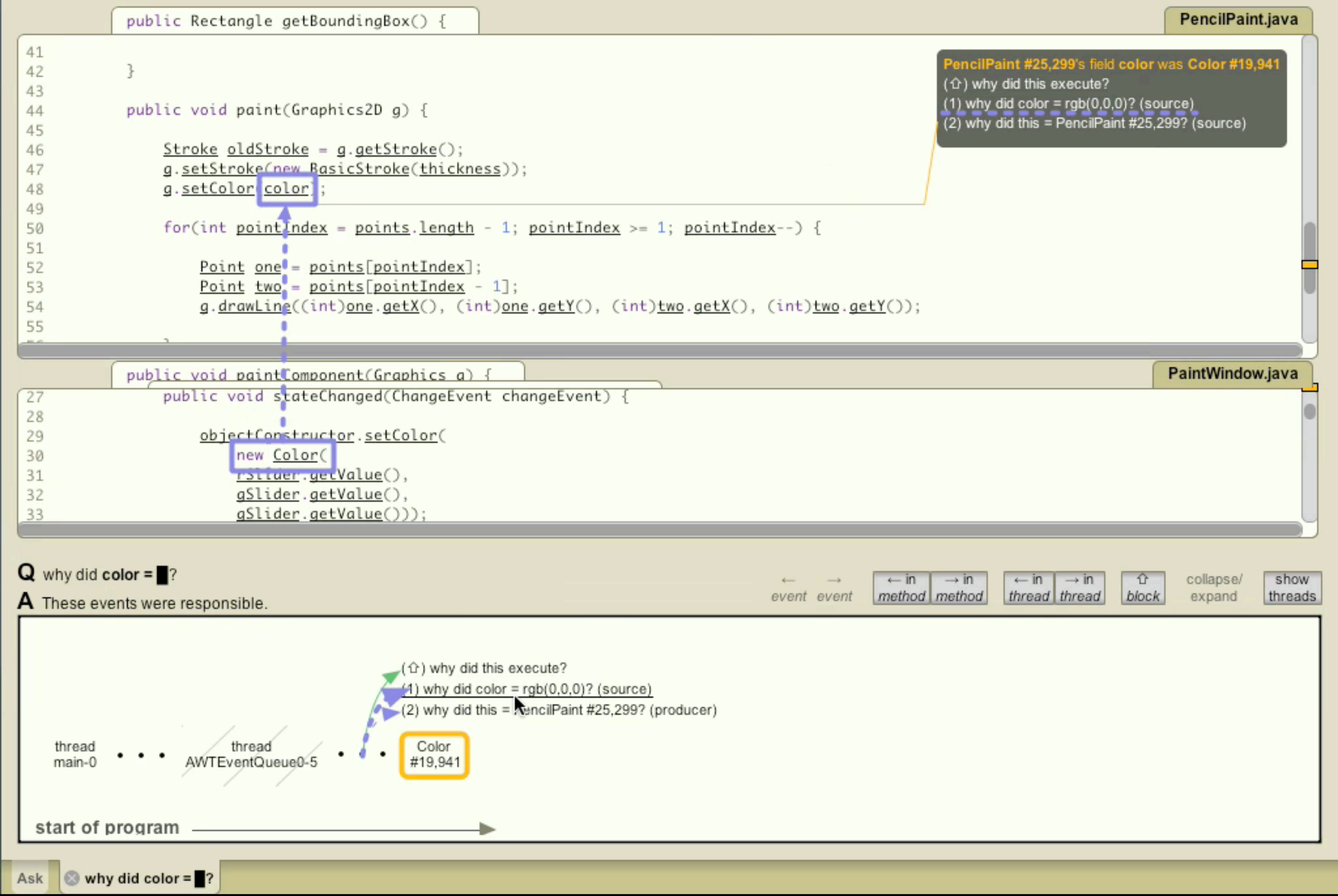
```
}

public void paint(Graphics2D g) {

    Stroke oldStroke = g.getStroke();
    g.setStroke(new BasicStroke(thickness));
    g.setColor(color);

    for(int pointIndex = points.length - 1; pointIndex >= 1; pointI

        Point one = points[pointIndex];
        Point two = points[pointIndex - 1];
        g.drawLine((int)one.getX(), (int)one.getY(), (int)two.getX(

    }

    g.setStroke(oldStroke);
```

why was the line color black?


PencilPaint.java
PaintCanvas.java
PencilPaint #25,299's field color was Color #19,941
(⇧) why did this execute?
(1) why did color = rgb(0,0,0)? (source)
(2) why did this = PencilPaint #25,299? (source)
Q why did color = █?
A These events were responsible.
← event
→ event
← in method
→ in method
← in thread
→ in thread
⇧ block
collapse/ expand
show threads
(⇧) why did this execute?
(1) why did color = rgb(0,0,0)? (source)
(2) why did this = PencilPaint #25,299? (producer)
thread main-0
thread AWTEventQueue0-5
Color #19,941
start of program

```
        }
    public Rectangle getBoundingBox() {
        return new Rectangle(minX, minY, maxX - minX, maxY - minY);

    }

    public void paint(Graphics2D g) {

        Stroke oldStroke = g.getStroke();
        g.setStroke(new BasicStroke(thickness));
        g.setColor(color);

        for(int pointIndex = points.length - 1; pointIndex >= 1; pointIndex--) {

            Point one = points[pointIndex];
            Point two = points[pointIndex - 1];
            g.drawLine((int)one.getX(), (int)one.getY(), (int)two.getX(), (int)two.getY());

        }

        g.setStroke(oldStroke);
```

why was the line color black?



```
public Rectangle getBoundingBox() {

    }

    public void paint(Graphics2D g) {

        Stroke oldStroke = g.getStroke();
        g.setStroke(new BasicStroke(thickness));
        g.setColor(color);

        for(int pointIndex = points.length - 1; pointIndex >= 1; pointIndex--) {

            Point one = points[pointIndex];
            Point two = points[pointIndex - 1];
            g.drawLine((int)one.getX(), (int)one.getY(), (int)two.getX(), (int)two.getY());
```

PencilPaint #25,299's field color was Color #19,941
(⇧) why did this execute?
(1) why did color = rgb(0,0,0)? (source)
(2) why did this = PencilPaint #25,299? (source)

```
public void paintComponent(Graphics g) {
    public void stateChanged(ChangeEvent changeEvent) {

        objectConstructor.setColor(
            new Color(
            rSlider.getValue(),
            gSlider.getValue(),
            gSlider.getValue()));
```

Q why did **color = █**?
A These events were responsible.

(⇧) why did this execute?
(1) why did color = rgb(0,0,0)? (source)
(2) why did this = PencilPaint #25,299? (producer)

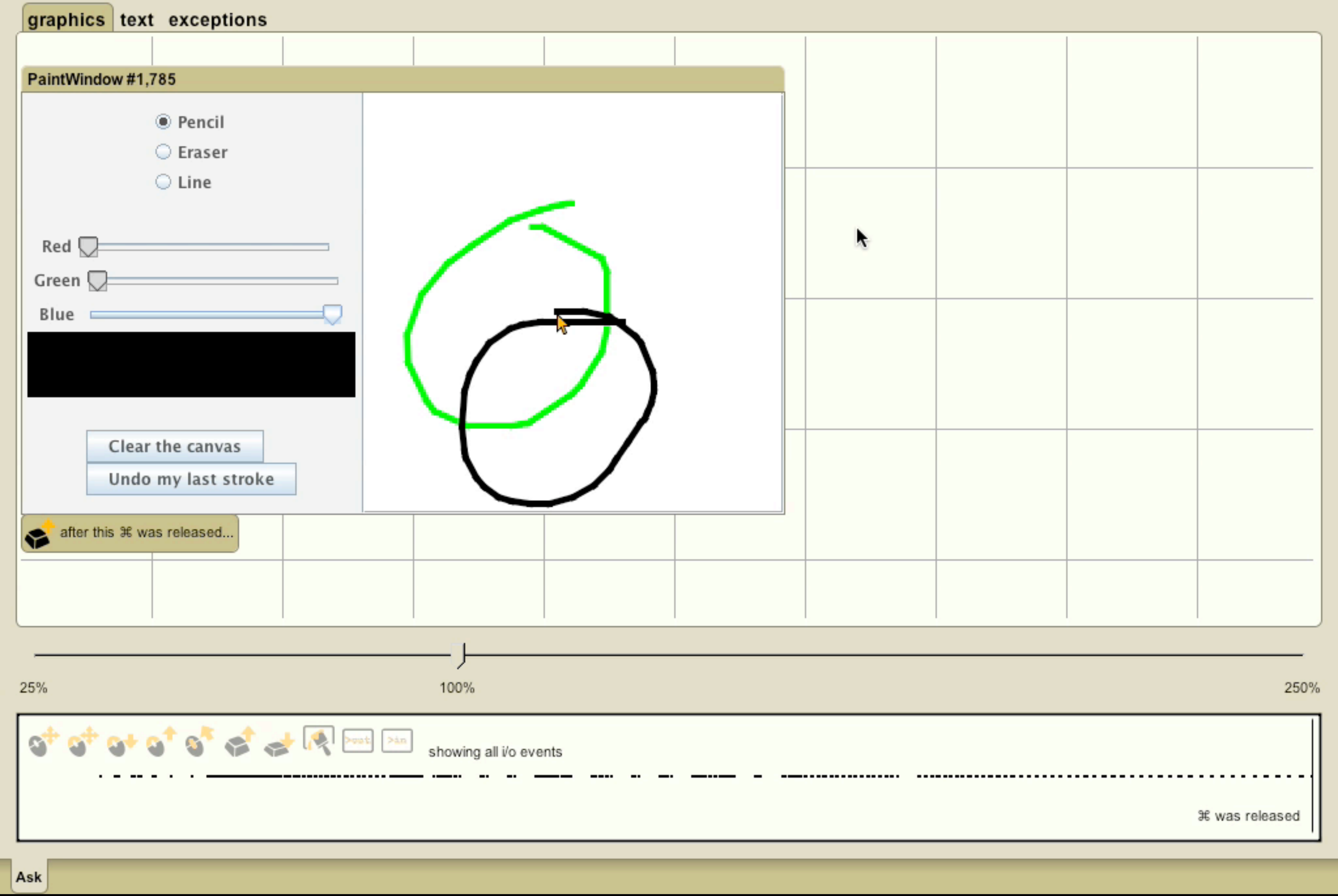
why was the line color black?

whyline demo

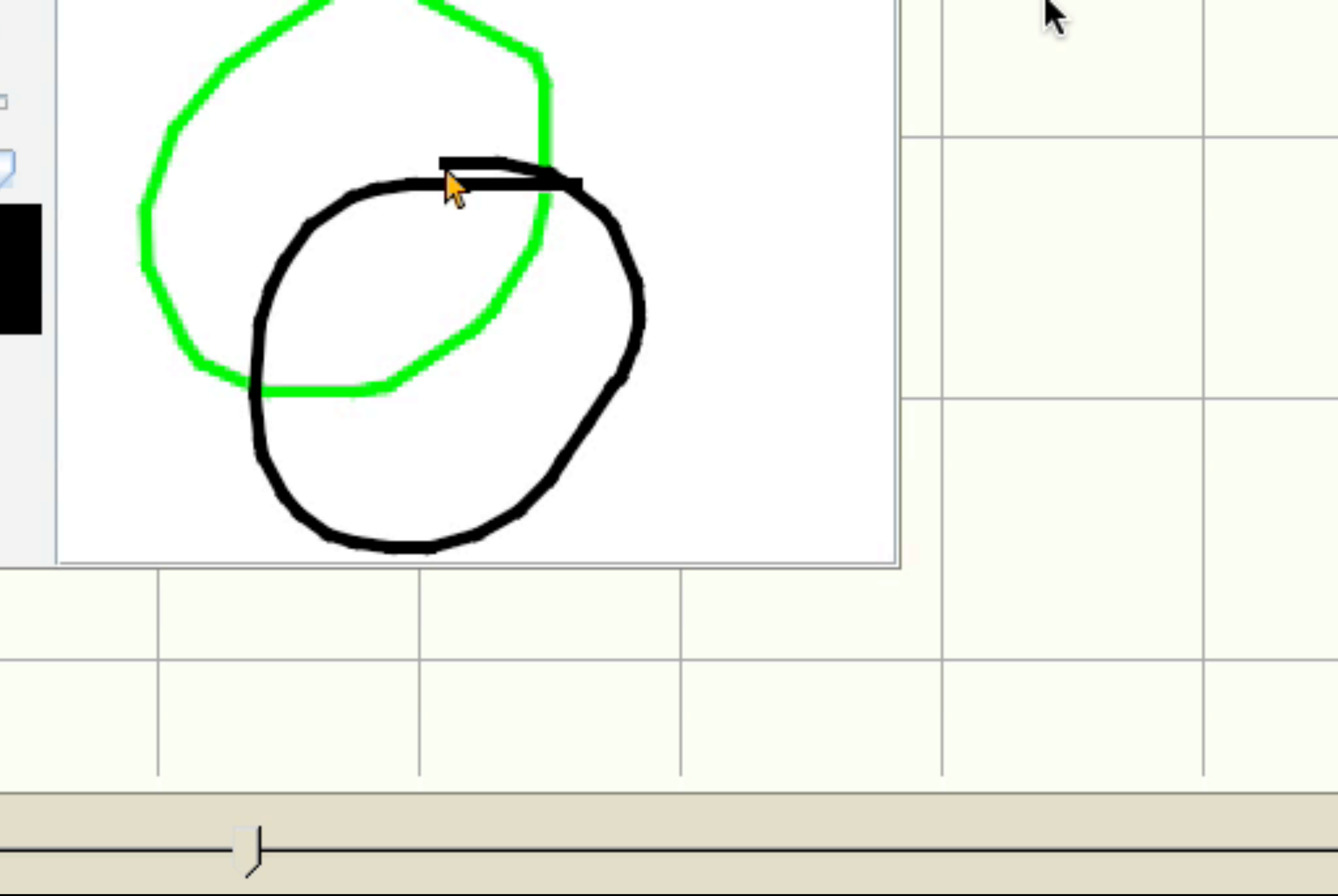
at least two ways to ask this question …

why was the **`line`** color black?

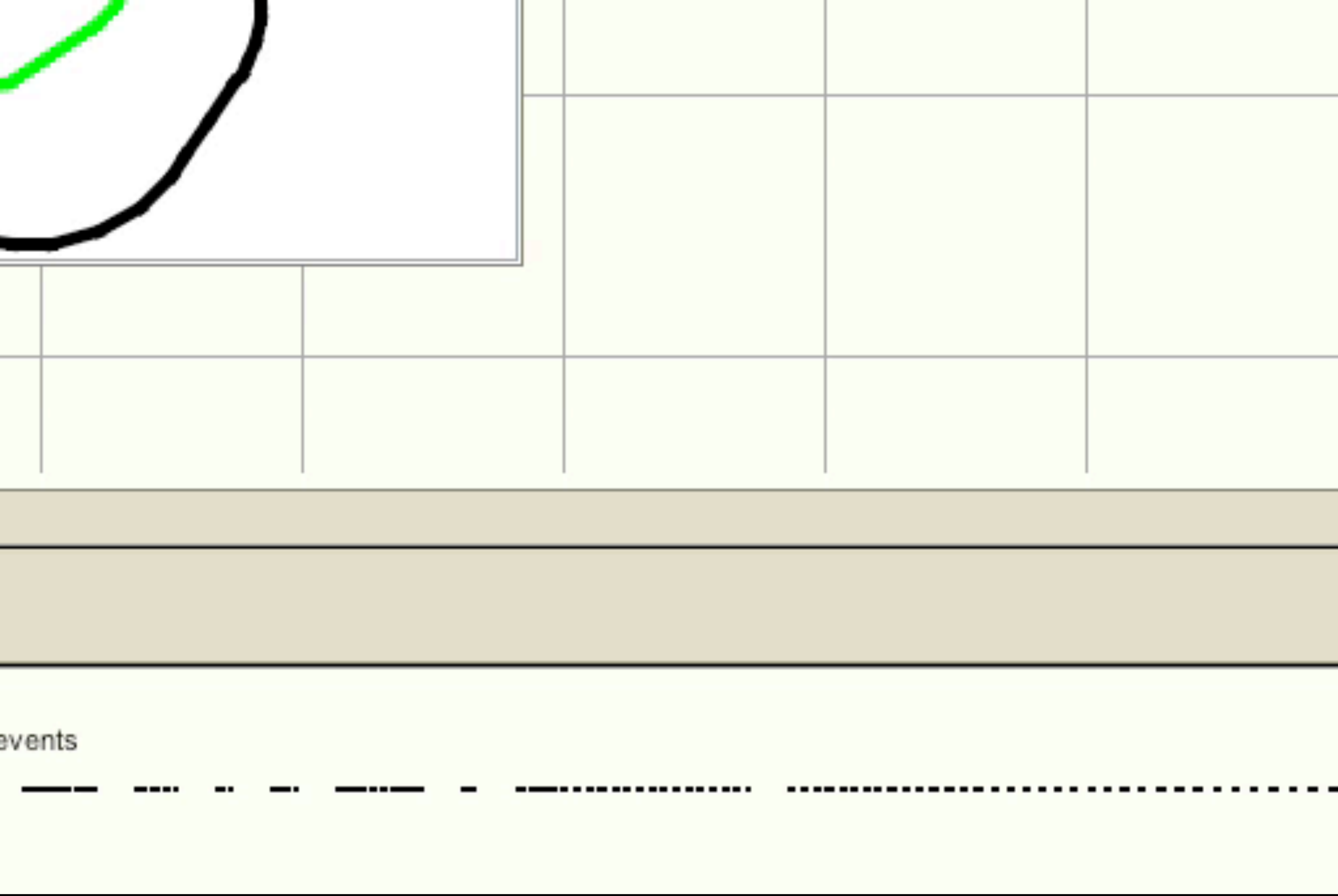
why didn't the **`color panel`** paint?



why didn’t the panel paint?

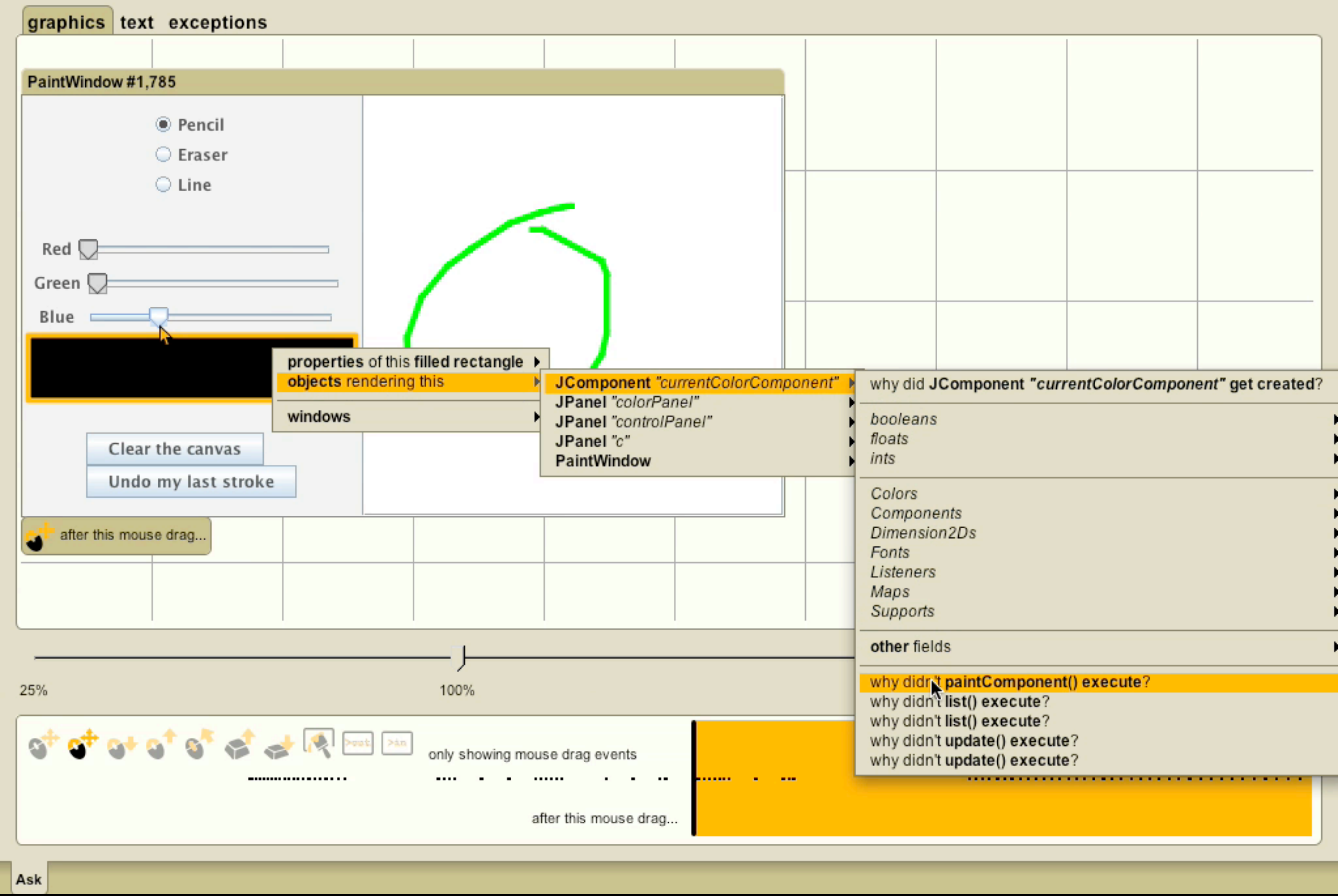


why didn’t the panel paint?

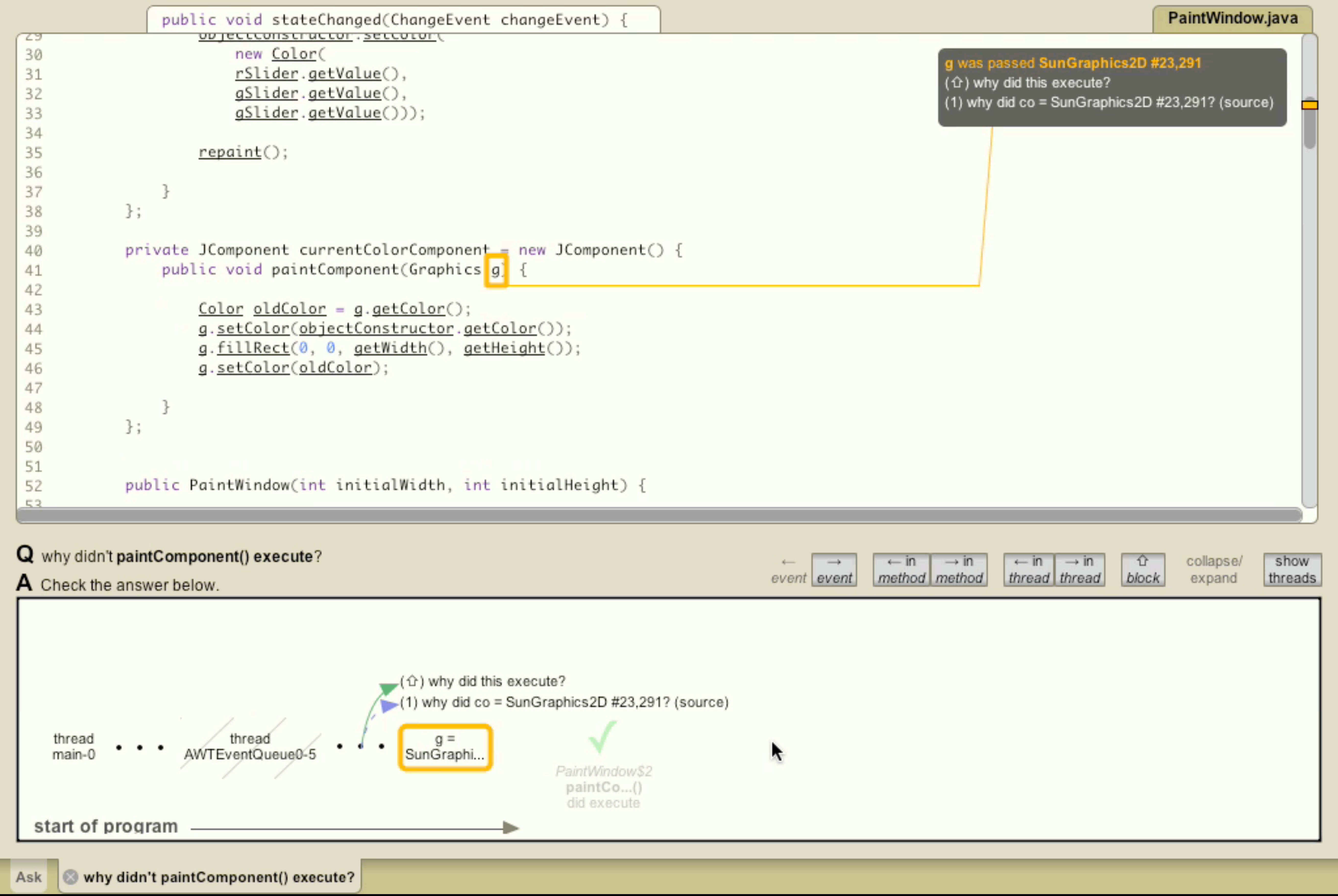


why didn’t the panel paint?

it did paint…


graphics
text
exceptions
PaintWindow #1,785
Pencil
Eraser
Line
Red
Green
Blue
Clear the canvas
Undo my last stroke
after this mouse drag...
properties of this filled rectangle
objects rendering this
windows
JComponent "currentColorComponent"
JPanel "colorPanel"
JPanel "controlPanel"
JPanel "c"
PaintWindow
why did JComponent "currentColorComponent" get created?
booleans
floats
ints
Colors
Components
Dimension2Ds
Fonts
Listeners
Maps
Supports
other fields
why didn't paintComponent() execute?
why didn't list() execute?
why didn't list() execute?
why didn't update() execute?
why didn't update() execute?
25%
100%
only showing mouse drag events
after this mouse drag...
Ask

it did paint…


public void stateChanged(ChangeEvent changeEvent) {

PaintWindow.java

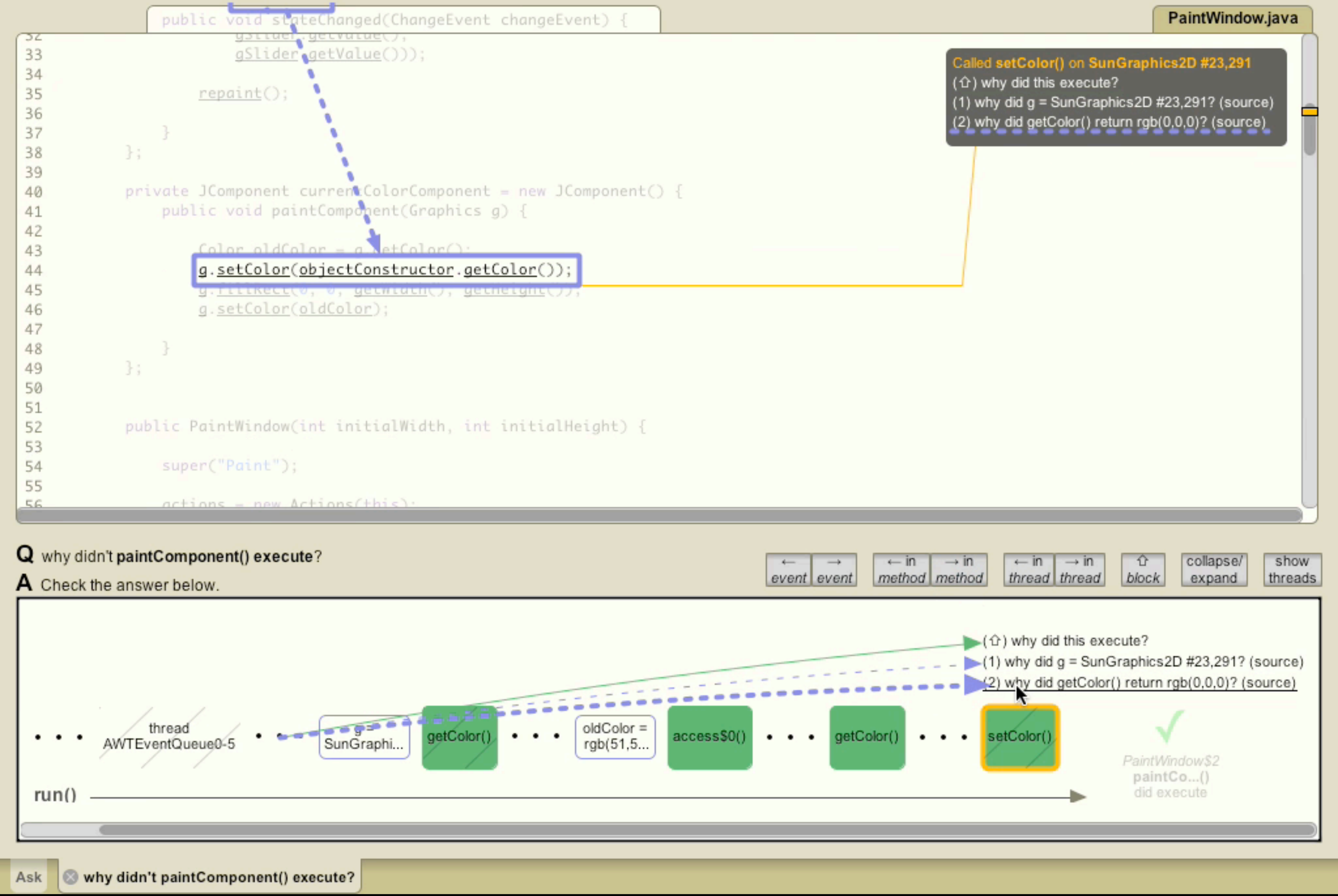
```
                objectConstructor.setColor(
                    new Color(
                    rSlider.getValue(),
                    gSlider.getValue(),
                    gSlider.getValue()));

                repaint();

            }
        };

        private JComponent currentColorComponent = new JComponent() {
            public void paintComponent(Graphics g) {

                Color oldColor = g.getColor();
                g.setColor(objectConstructor.getColor());
                g.fillRect(0, 0, getWidth(), getHeight());
                g.setColor(oldColor);

            }
        };


        public PaintWindow(int initialWidth, int initialHeight) {
```

g was passed SunGraphics2D #23,291
(⇧) why did this execute?
(1) why did co = SunGraphics2D #23,291? (source)

Q why didn't **paintComponent()** execute?
A Check the answer below.

(⇧) why did this execute?
(1) why did co = SunGraphics2D #23,291? (source)
thread main-0
thread AWTEventQueue0-5
g = SunGraphi...
PaintWindow$2 paintCo...() did execute
start of program

step forward to the color used...



```
public void stateChanged(ChangeEvent changeEvent) {
        gSlider.getValue()));

    repaint();

  }
};

private JComponent currentColorComponent = new JComponent() {
    public void paintComponent(Graphics g) {

        Color oldColor = g.getColor();
        g.setColor(objectConstructor.getColor());
        g.fillRect(0, 0, getWidth(), getHeight());
        g.setColor(oldColor);

    }
};

public PaintWindow(int initialWidth, int initialHeight) {

    super("Paint");

    actions = new Actions(this);
```

Called setColor() on SunGraphics2D #23,291
(⇧) why did this execute?
(1) why did g = SunGraphics2D #23,291? (source)
(2) why did getColor() return rgb(0,0,0)? (source)

Q why didn't **paintComponent()** execute?
A Check the answer below.

(⇧) why did this execute?
(1) why did g = SunGraphics2D #23,291? (source)
(2) why did getColor() return rgb(0,0,0)? (source)

. . . thread AWTEventQueue0-5 . . g = SunGraphi... getColor() . . . oldColor = rgb(51,5... access$0() . . . getColor() . . . setColor()

PaintWindow$2 paintCo...() did execute

run()

why didn't paintComponent() execute?

find the bug

outline

outline

problem

studies

the whyline

implementation

evaluation

conclusions

a typical cycle

developer…

edit compile **debug** fix …

the whyline cycle

developer…

edit compile **record** **load** **ask** fix …

① ② ③

system…

① instrument bytecode
record thread history

② convert serial history to
random access history

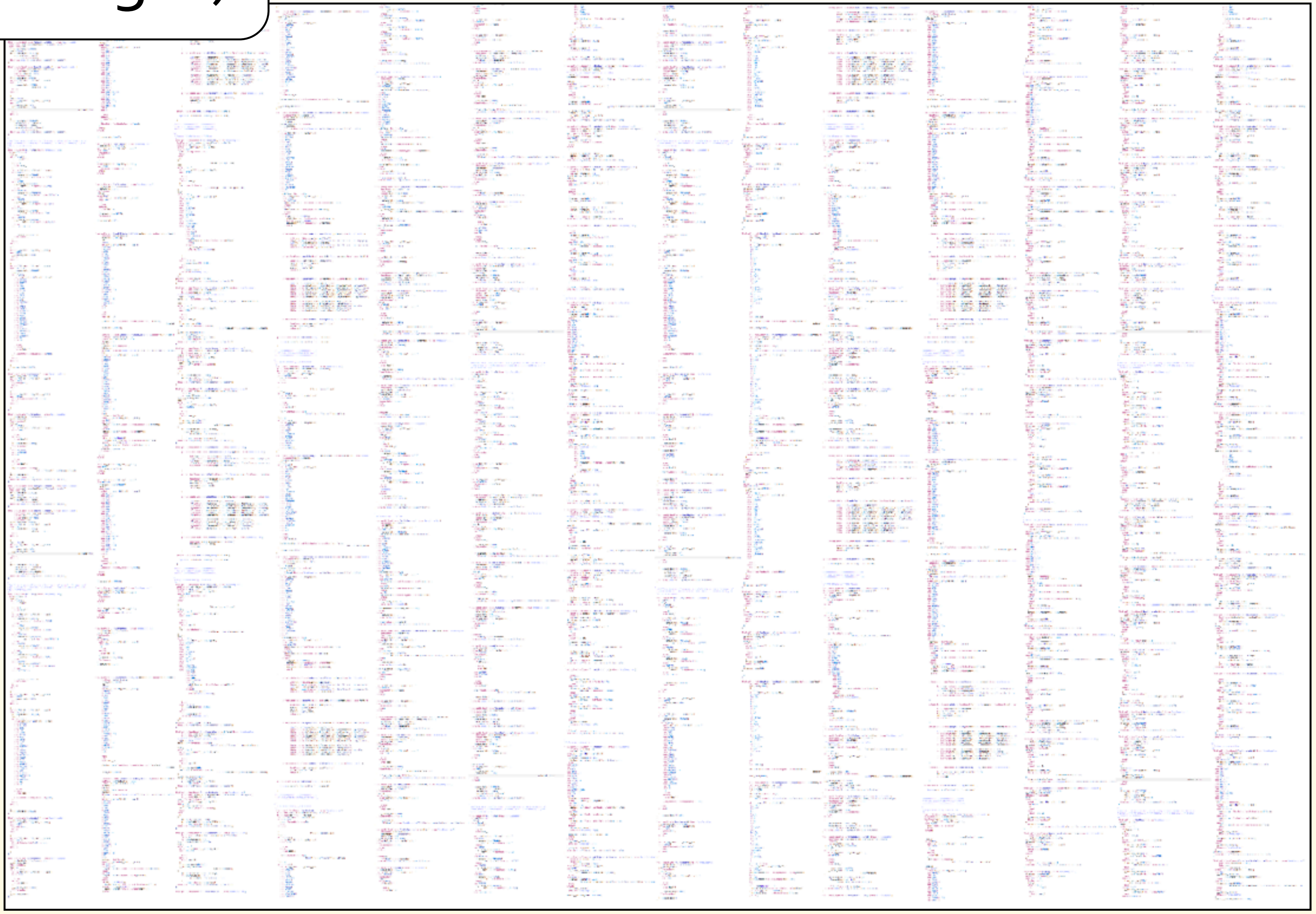
③ extract questions
from code

find primitive **output statements**

```
drawString(x, y, string)
```

```
fillRect(x, y, width, height)
```

```
setFont(font)
```



find primitive output statements

drawString(x, y, string)

fillRect(x, y, width, height)

setFont(font)

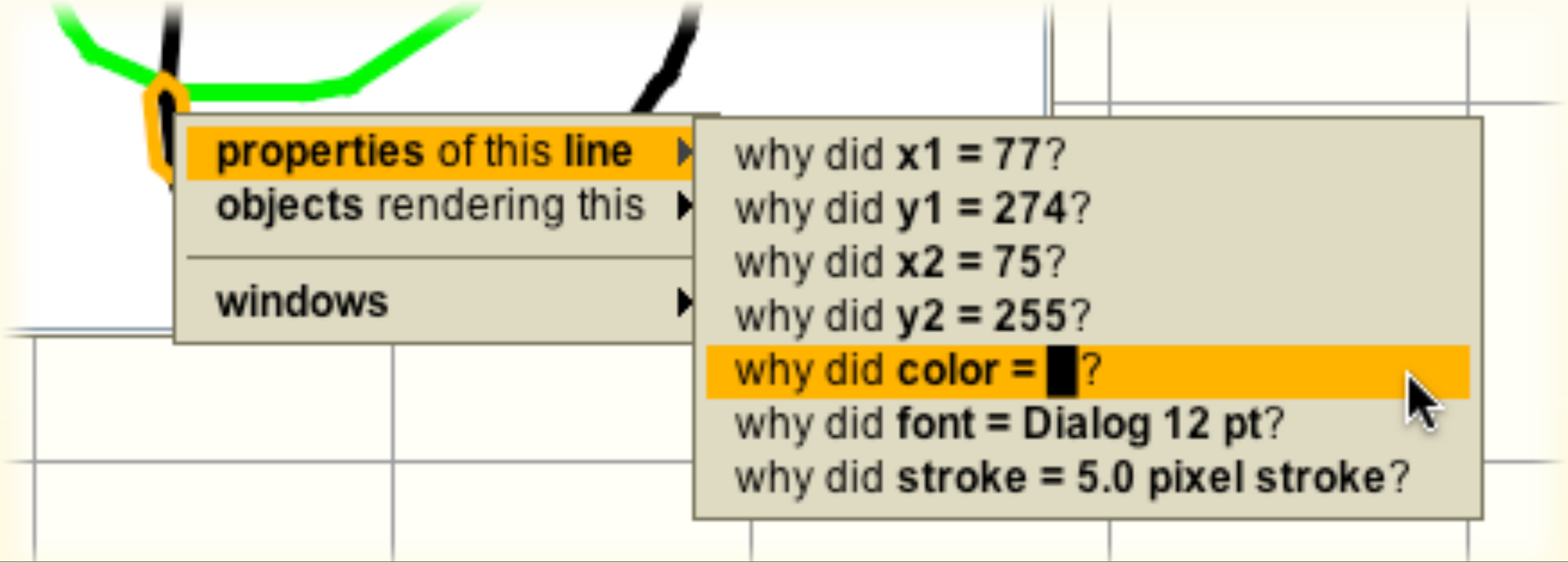
ask primitive **questions**

drawString(x, y, string)

fillRect(x, y, width, height)

setFont(font)

why did *argument* = *value*?



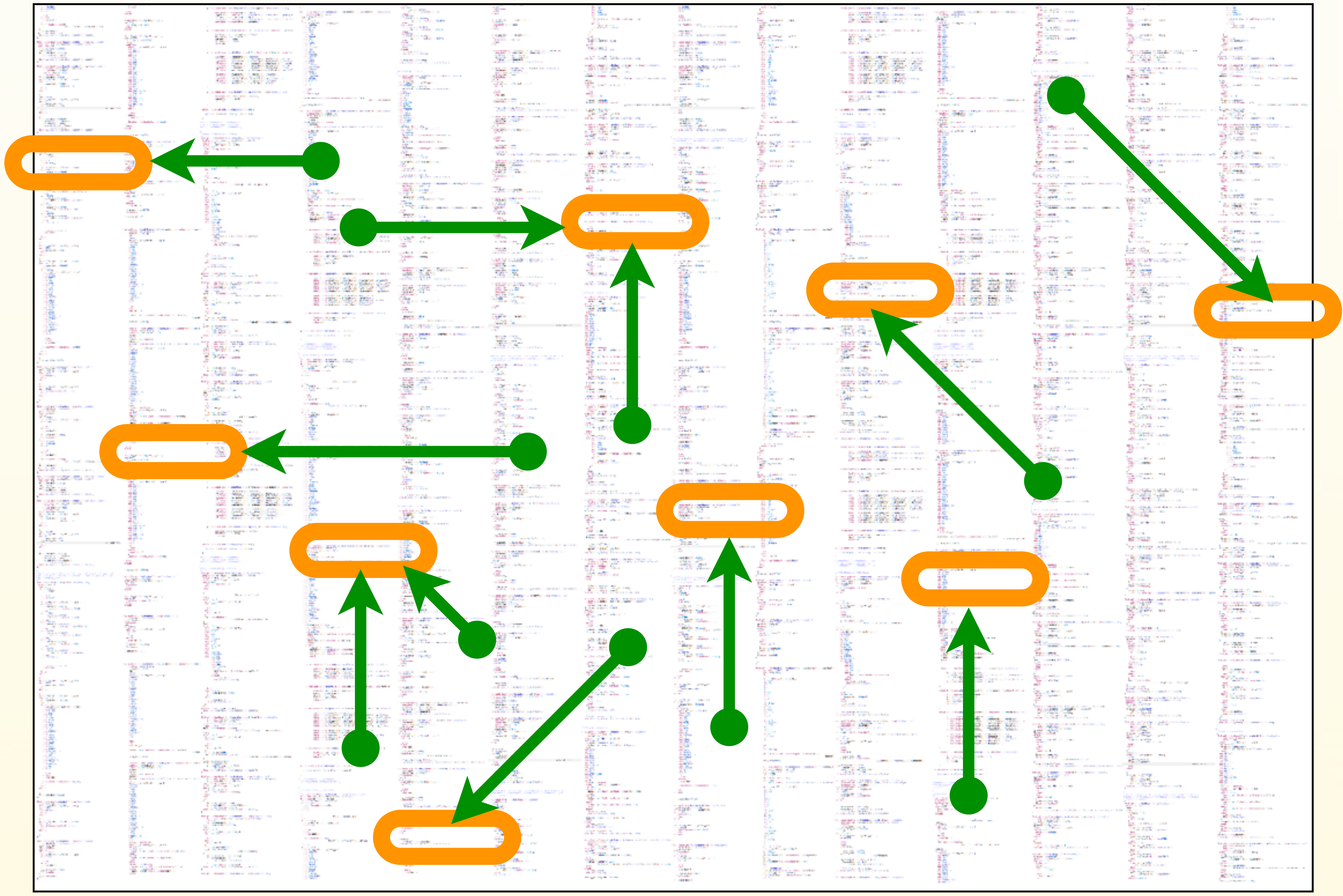
why did **x1 = 77**?
why did **y1 = 274**?
why did **x2 = 75**?
why did **y2 = 255**?
why did **color = ■**?
why did **font = Dialog 12 pt**?
why did **stroke = 5.0 pixel stroke**?

find output-invoking **data**

```
class PencilPaint
    draw() {
        ...
        drawLine(
            x1, y1,
            x2, y2)
    }
```

find output-invoking **data**

```
class PencilPaint
    draw() {
        ...
        drawLine(
            x1, y1,
            x2, y2)
    }
```



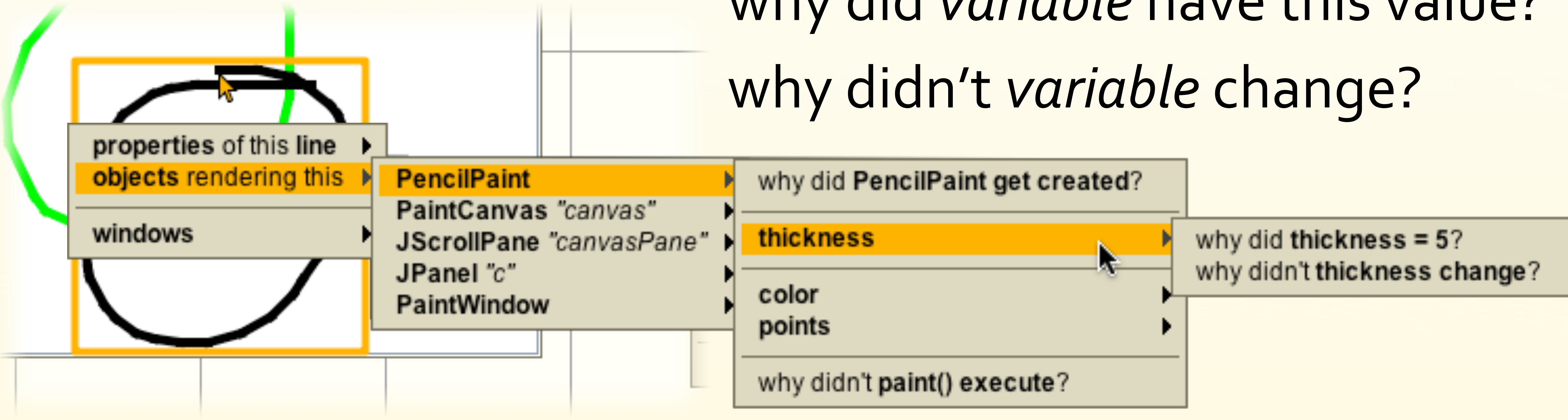
ask output-invoking **questions**

```
class PencilPaint
    draw() {
        ...
        drawLine(
            x1, y1,
            x2, y2)
    }
```

why did *subject* get created?

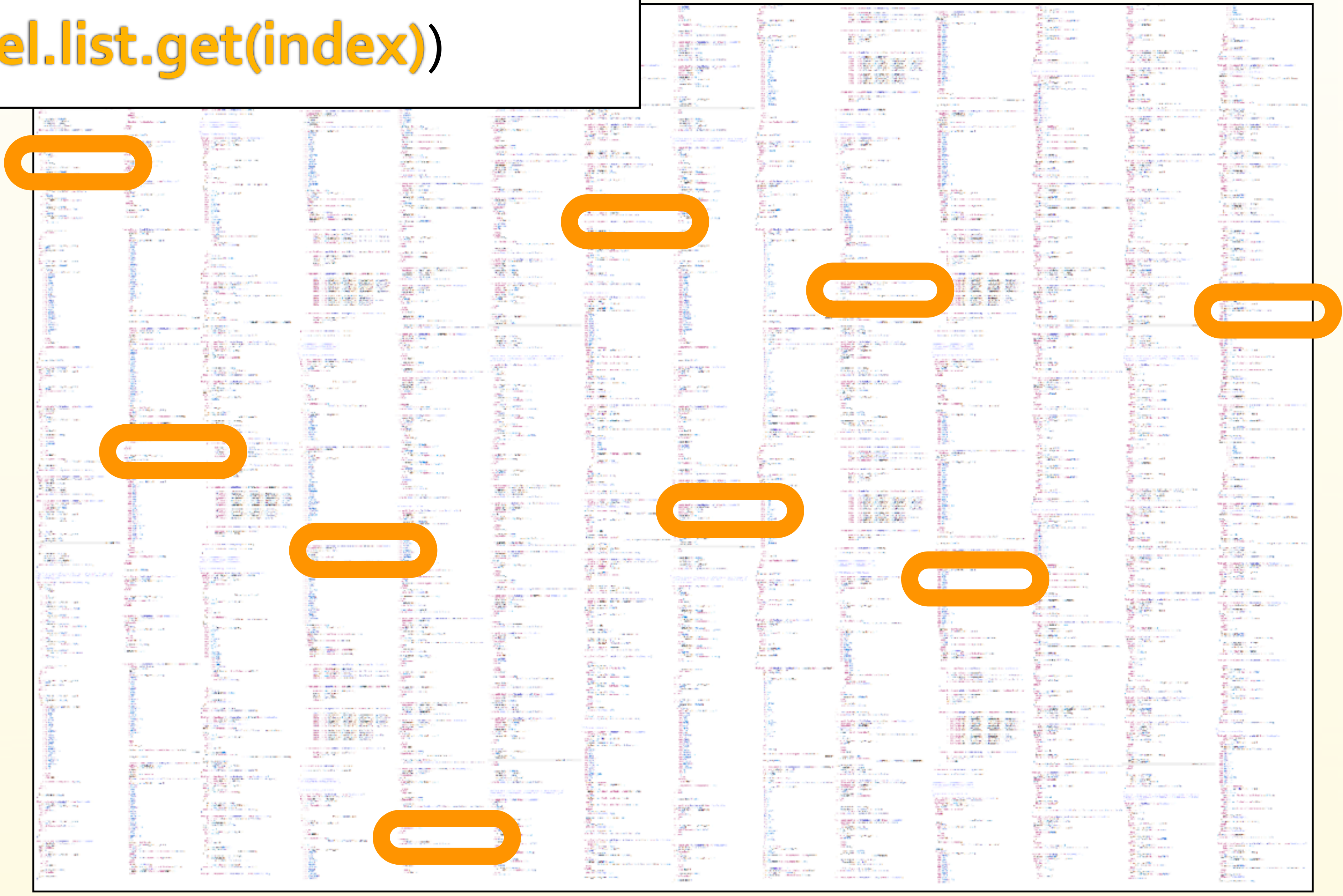
why did *variable* have this value?

why didn't *variable* change?



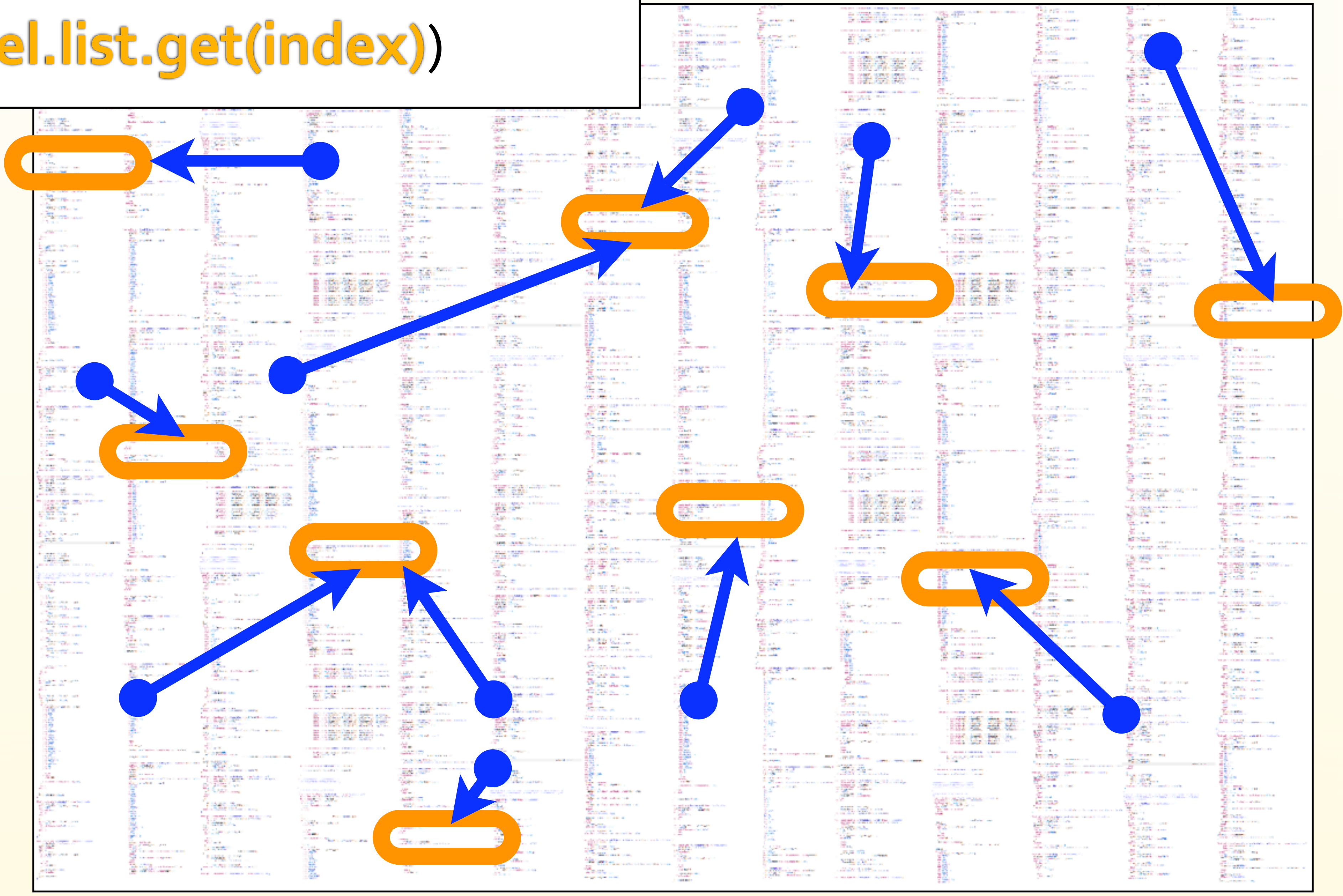
find output-affecting **data**

```
ComboBox combo = new ComboBox(model)
...
paint() {
    drawString(model.list.get(index))
```



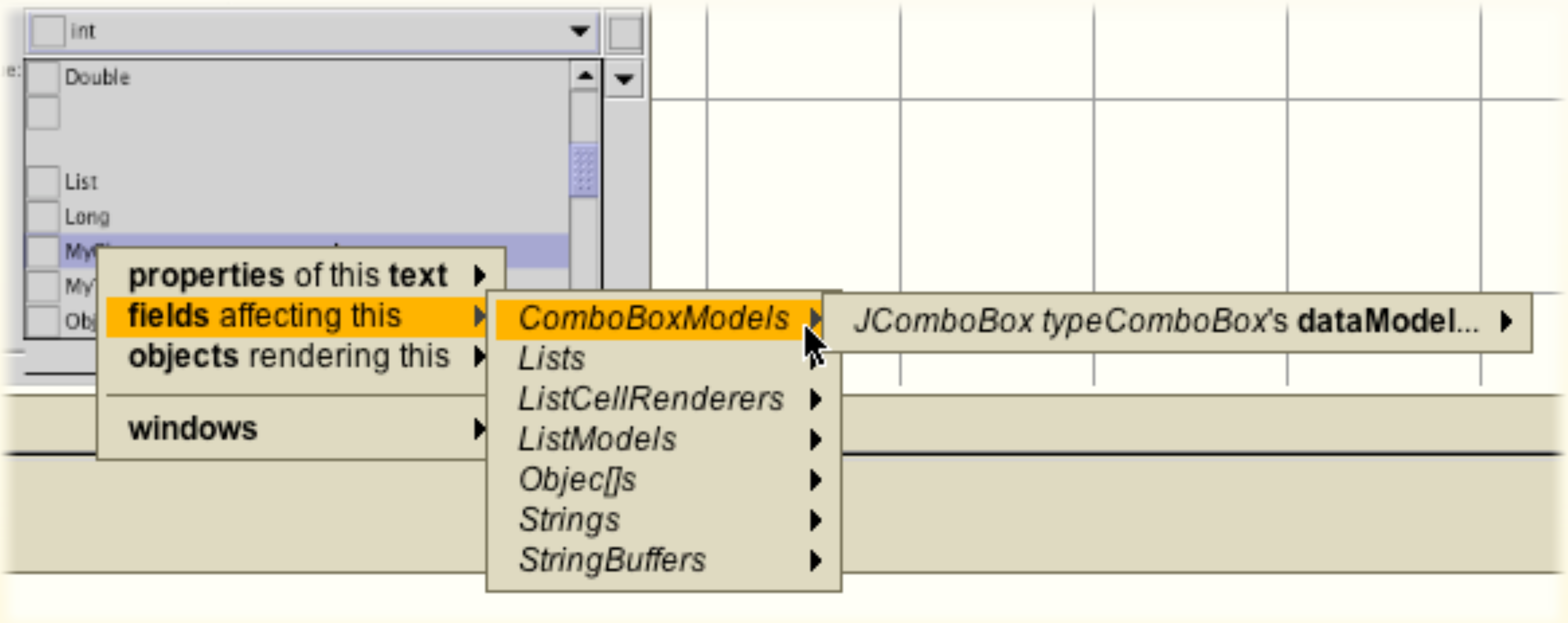
find output-affecting **data**

```
ComboBox combo = new ComboBox(model)
...
paint() {
    drawString(model.list.get(index))
```



ask output-affecting **data** questions

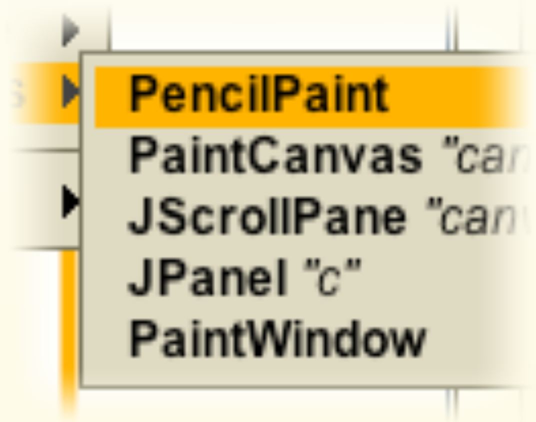
```
ComboBox combo = new ComboBox(model)
...
paint() {
    drawString(model.list.get(index))
```



filtering questions by 'familiarity'

```
class Button
    paint() {
        lookandfeel.paint()
```

PencilPaint
~~ComponentUI~~
PaintCanvas
~~ScrollPaneUI~~
JScrollPane
~~ComponentUI~~
JPanel
~~WindowUI~~
PaintWindow

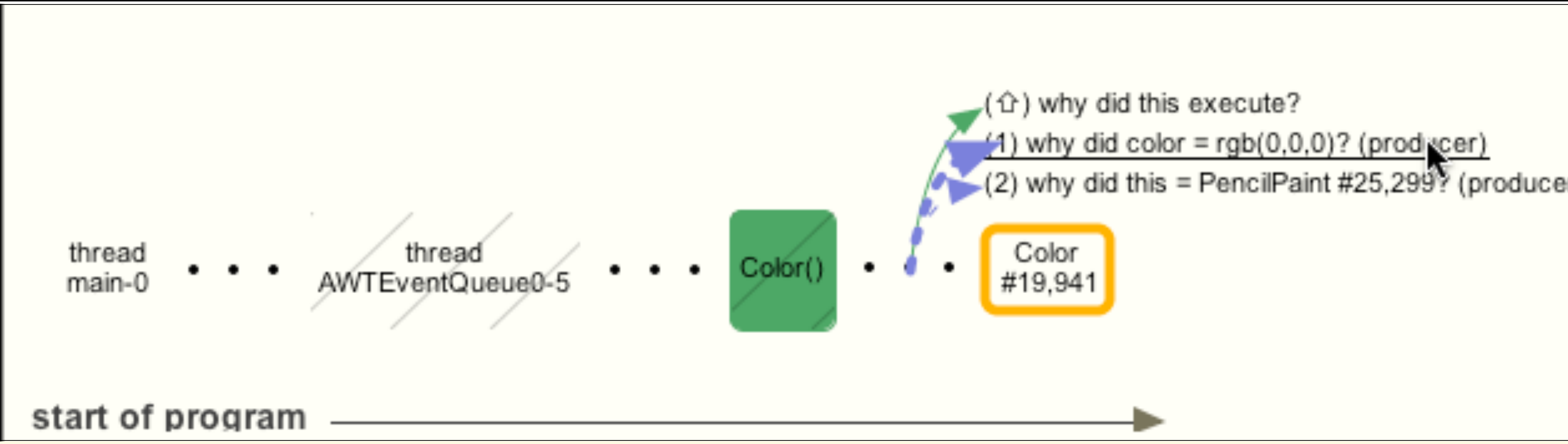


PencilPaint
PaintCanvas "ca
JScrollPane "can
JPanel "c"
PaintWindow

- **intermediaries**
  - look and feel delegates
  - proxies
- **familiarity** = classes
  - **declared** in **editable** code
  - **referenced** in **editable** code
- only include questions about **familiar code entities**

presenting ‘why did’ answers

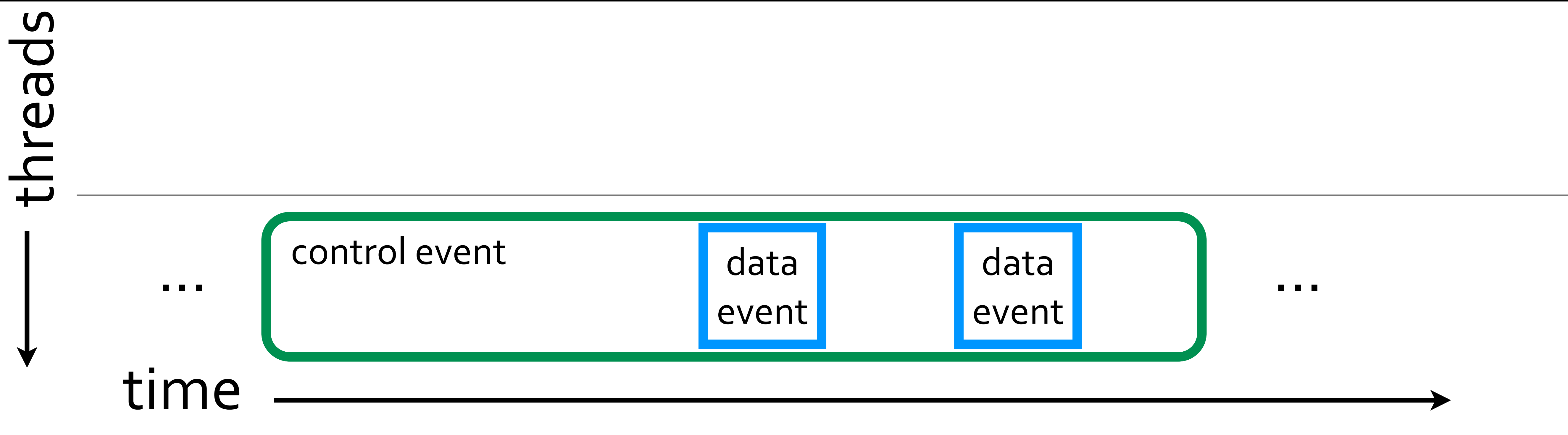
- answer derived with **precise dynamic slicing**
- **a** timeline (left to right)
- control dependencies as **nested blocks**
- data dependencies **inside** of blocks



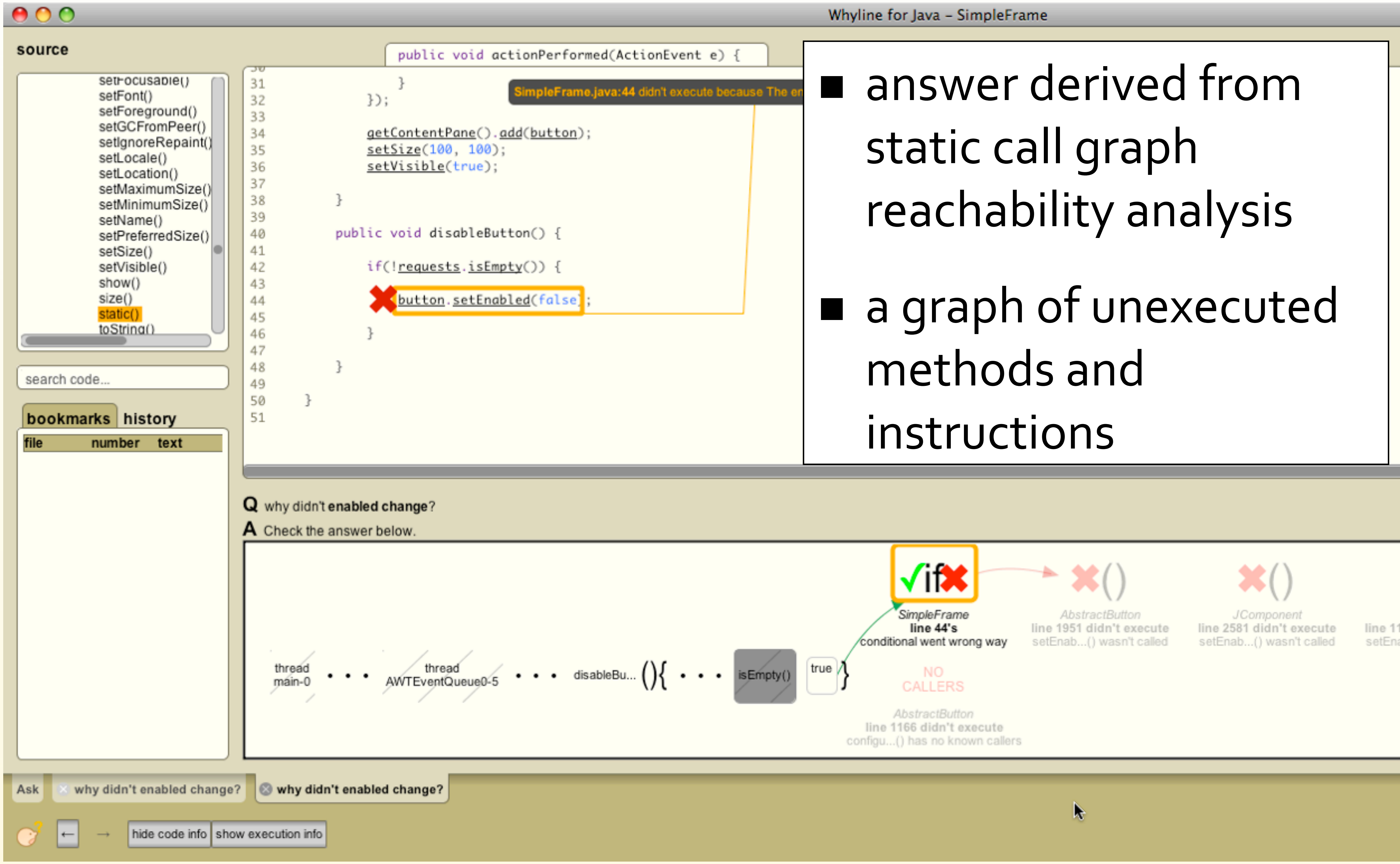
(⇧) why did this execute?
(1) why did color = rgb(0,0,0)? (producer)
(2) why did this = PencilPaint #25,299? (produce
thread main-0
thread AWTEventQueue0-5
Color()
Color #19,941
start of program

presenting ‘why did’ answers

- answer derived with **precise dynamic slicing**
- **a** timeline (left to right)
- control dependencies as **nested blocks**
- data dependencies **inside** of blocks



presenting ‘why didn’t’ answers



```
public void actionPerformed(ActionEvent e) {
31                  }
32              });
33
34              getContentPane().add(button);
35              setSize(100, 100);
36              setVisible(true);
37
38          }
39
40          public void disableButton() {
41
42              if(!requests.isEmpty()) {
43
44                  button.setEnabled(false);
45
46              }
47
48          }
49
50      }
51
```

SimpleFrame.java:44 didn't execute because The e

Q why didn't **enabled change**?

A Check the answer below.

thread main-0
thread AWTEventQueue0-5
disableBu...
isEmpty()
true
if
SimpleFrame line 44's conditional went wrong way
NO CALLERS
AbstractButton line 1166 didn't execute configu...() has no known callers
AbstractButton line 1951 didn't execute setEnab...() wasn't called
JComponent line 2581 didn't execute setEnab...() wasn't called
line 11
setEna

- answer derived from static call graph reachability analysis
- a graph of unexecuted methods and instructions

outline

problem

studies

the whyline

implementation

evaluation

conclusions

outline

problem

studies

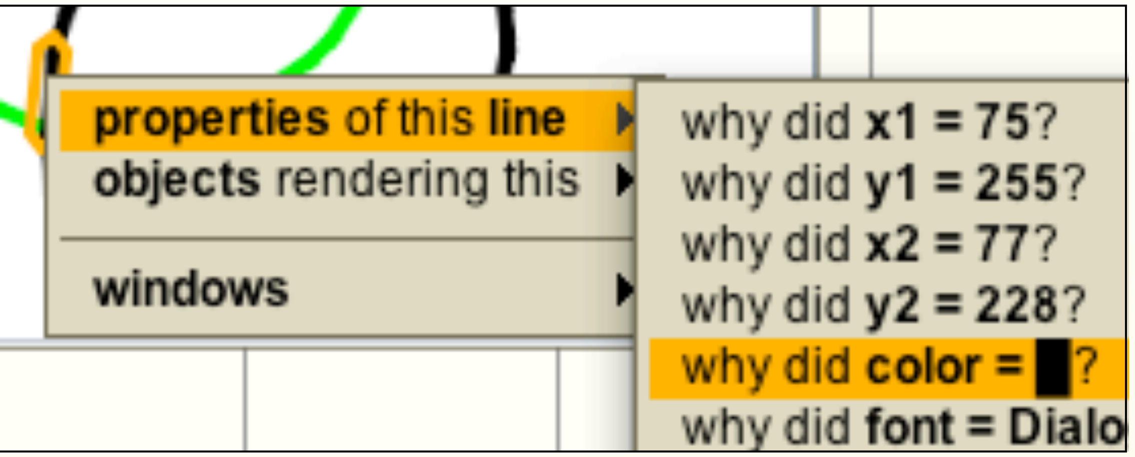
the whyline

implementation

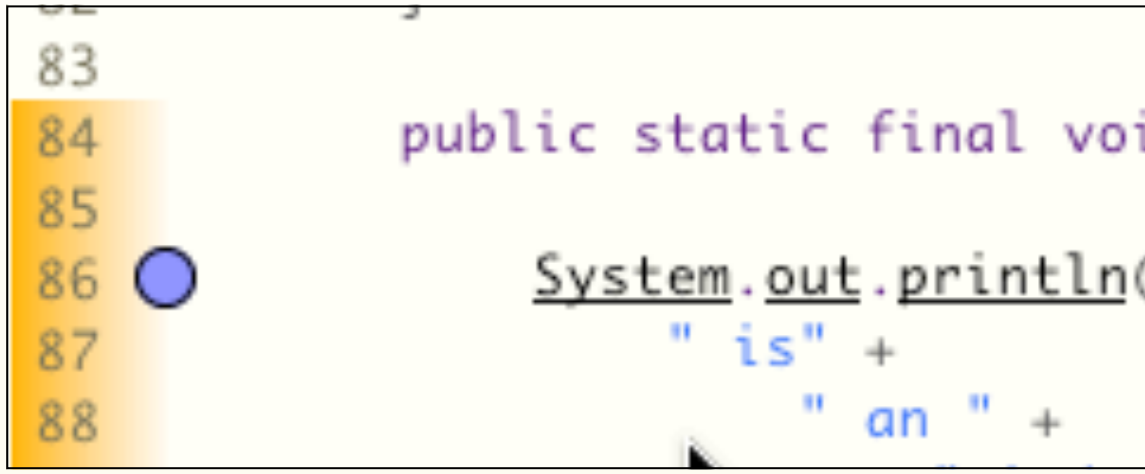
evaluation

conclusions

a comparison study

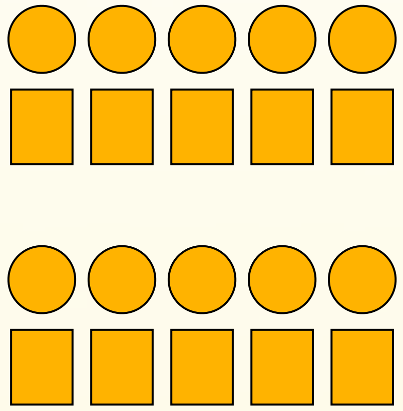


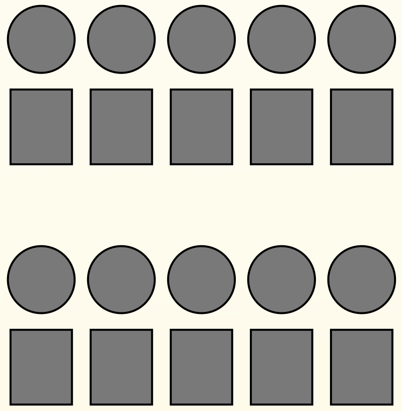
vs



```
public static final voi

    System.out.println(
        " is" +
            " an " +
```

Whyline group

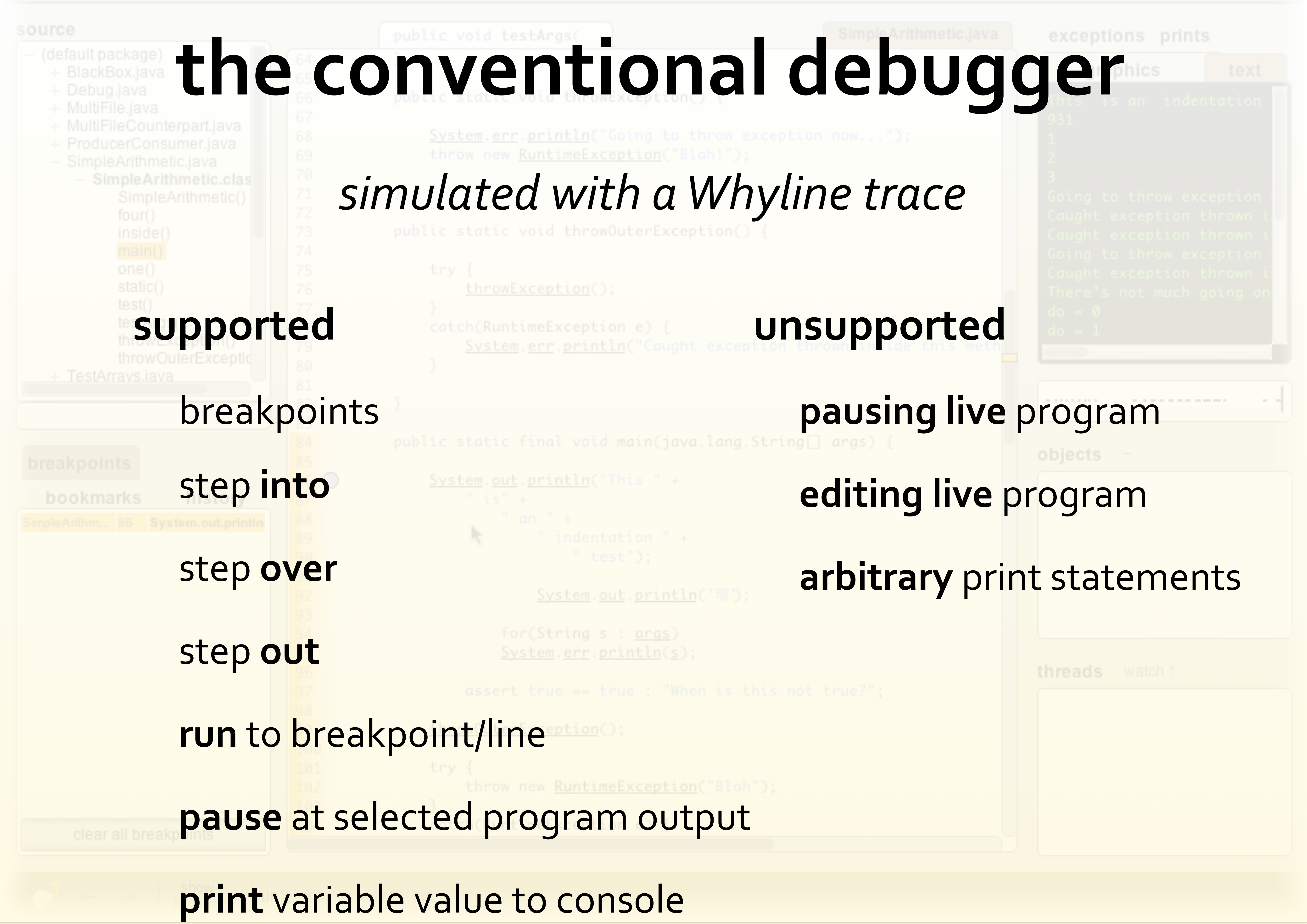




control group

both groups had modern IDE features

show declaration, show callers, show references, etc.



the conventional debugger

simulated with a Whyline trace

supported

breakpoints

step **into**

step **over**

step **out**

run to breakpoint/line

pause at selected program output

print variable value to console

unsupported

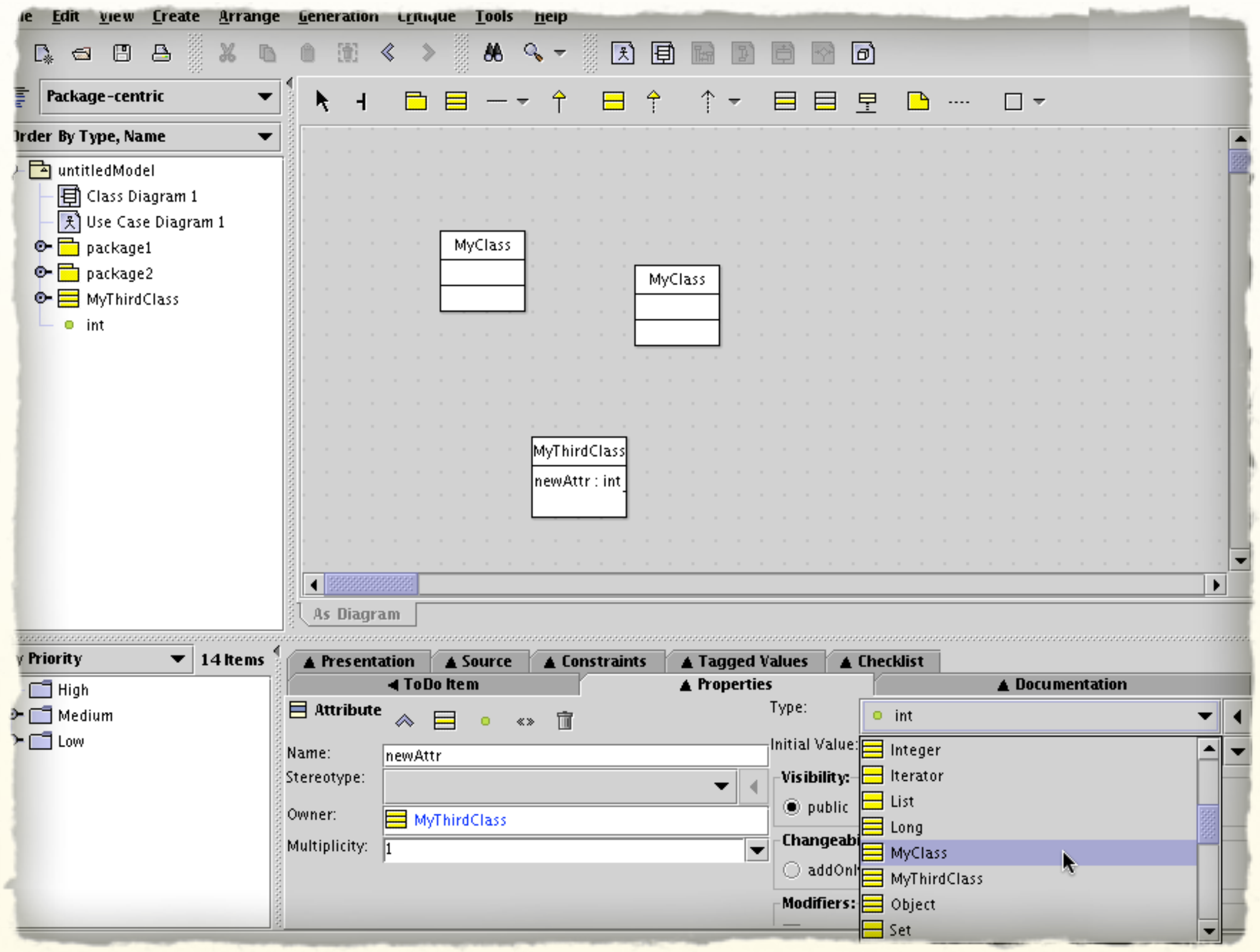
pausing live program

editing live program

arbitrary print statements

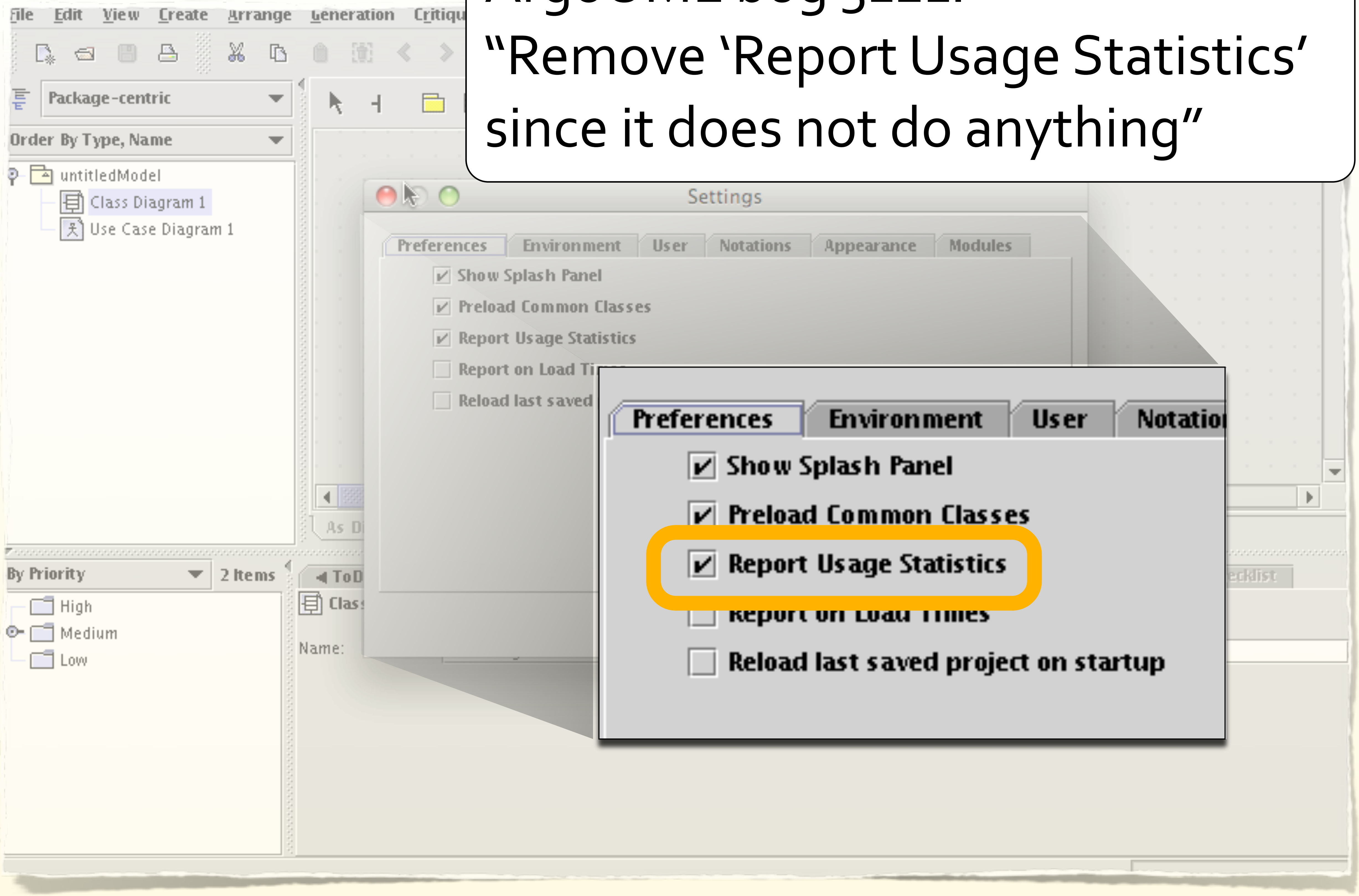
subject program

- **ArgoUML**, an open source software design tool
- ~150,000 lines of code
- 22 external libraries
- chose **two bug reports** from version 18.1
  - one easy
  - one difficult



Name: newAttr
Stereotype:
Owner: MyThirdClass
Multiplicity: 1
Type: int
Initial Value:

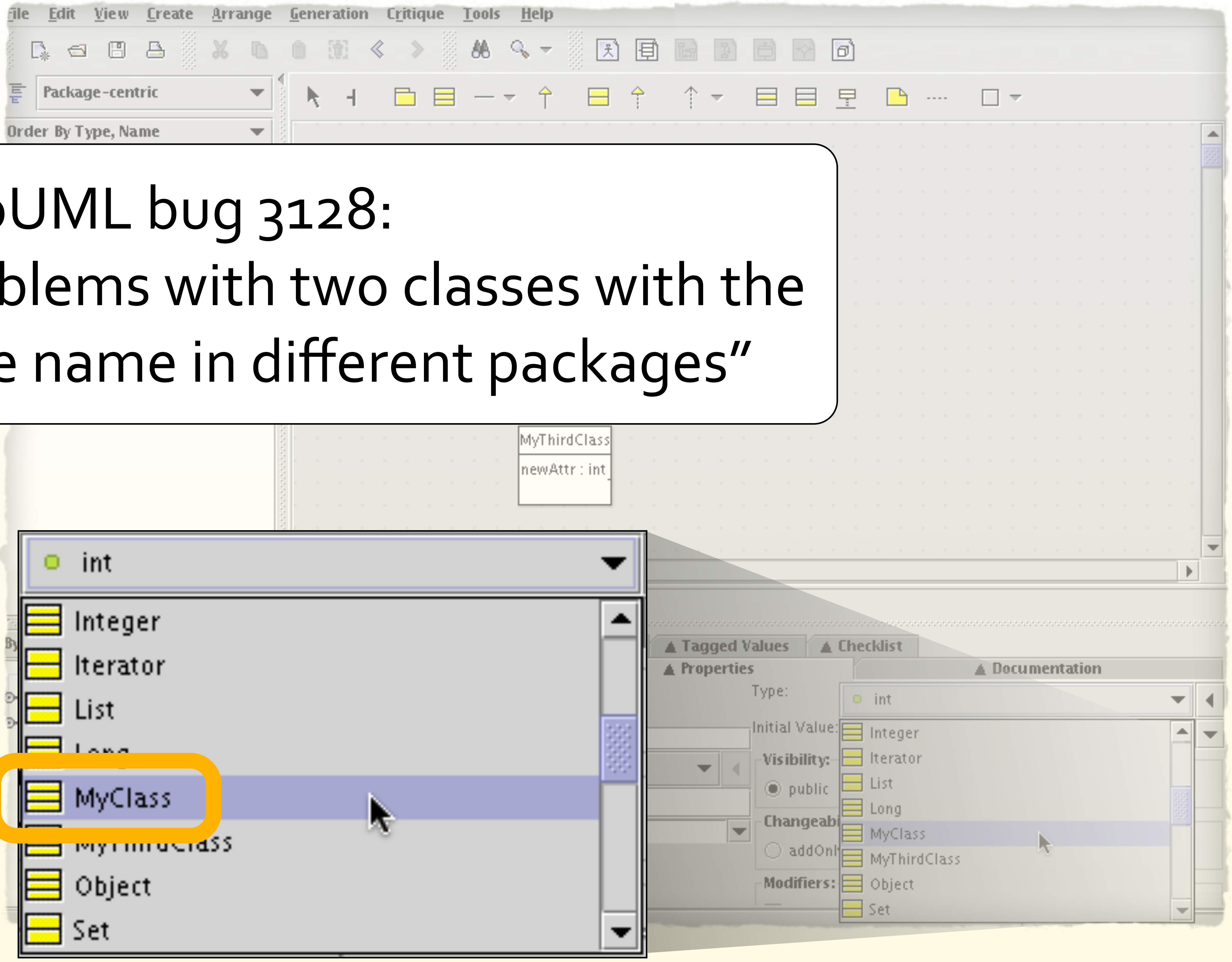
task one



- [✓] Show Splash Panel
- [✓] Preload Common Classes
- [✓] Report Usage Statistics
- [ ] Report on Load Times
- [ ] Reload last saved project on startup

ArgoUML bug 3121:
"Remove 'Report Usage Statistics' since it does not do anything"

task two



ArgoUML bug 3128:
“Problems with two classes with the same name in different packages”

participants were told...

- for each task

  identify cause of the problem

  write **change recommendation** to a fictional boss

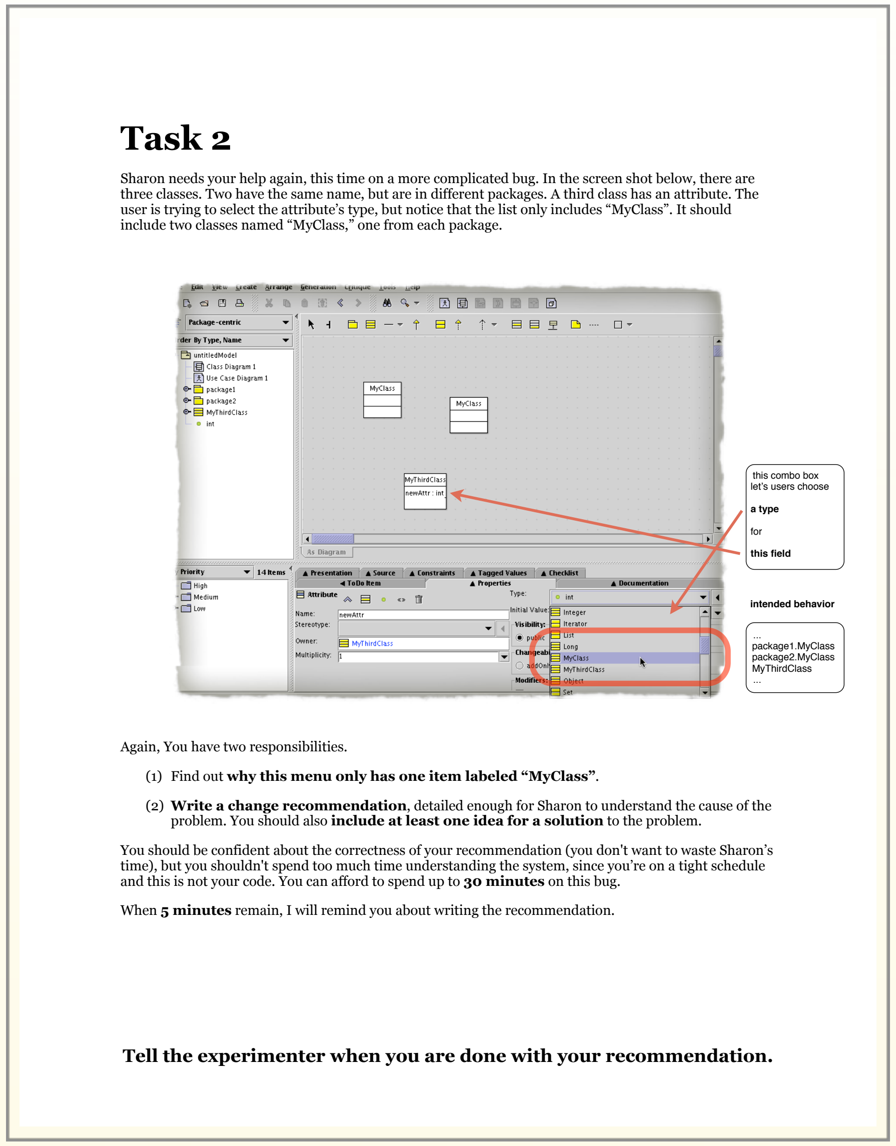
- 30 minutes to work

  emphasize **speed** over **confidence**

- measured

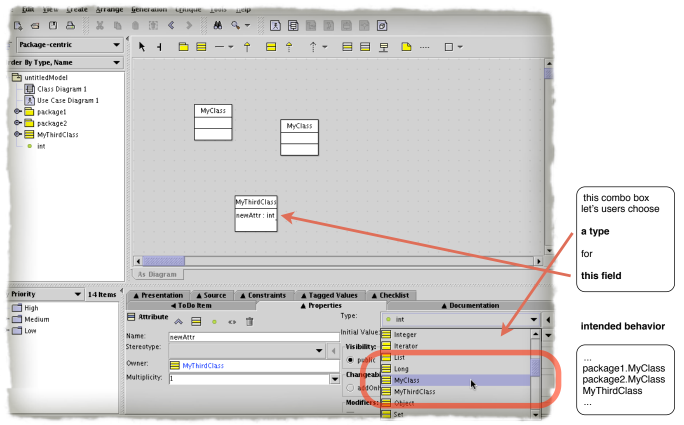
  time on task

  success



Task 2

Sharon needs your help again, this time on a more complicated bug. In the screen shot below, there are three classes. Two have the same name, but are in different packages. A third class has an attribute. The user is trying to select the attribute's type, but notice that the list only includes "MyClass". It should include two classes named "MyClass," one from each package.



Again, You have two responsibilities.

(1) Find out **why this menu only has one item labeled "MyClass"**.

(2) **Write a change recommendation**, detailed enough for Sharon to understand the cause of the problem. You should also **include at least one idea for a solution** to the problem.

You should be confident about the correctness of your recommendation (you don't want to waste Sharon's time), but you shouldn't spend too much time understanding the system, since you're on a tight schedule and this is not your code. You can afford to spend up to **30 minutes** on this bug.

When **5 minutes** remain, I will remind you about writing the recommendation.

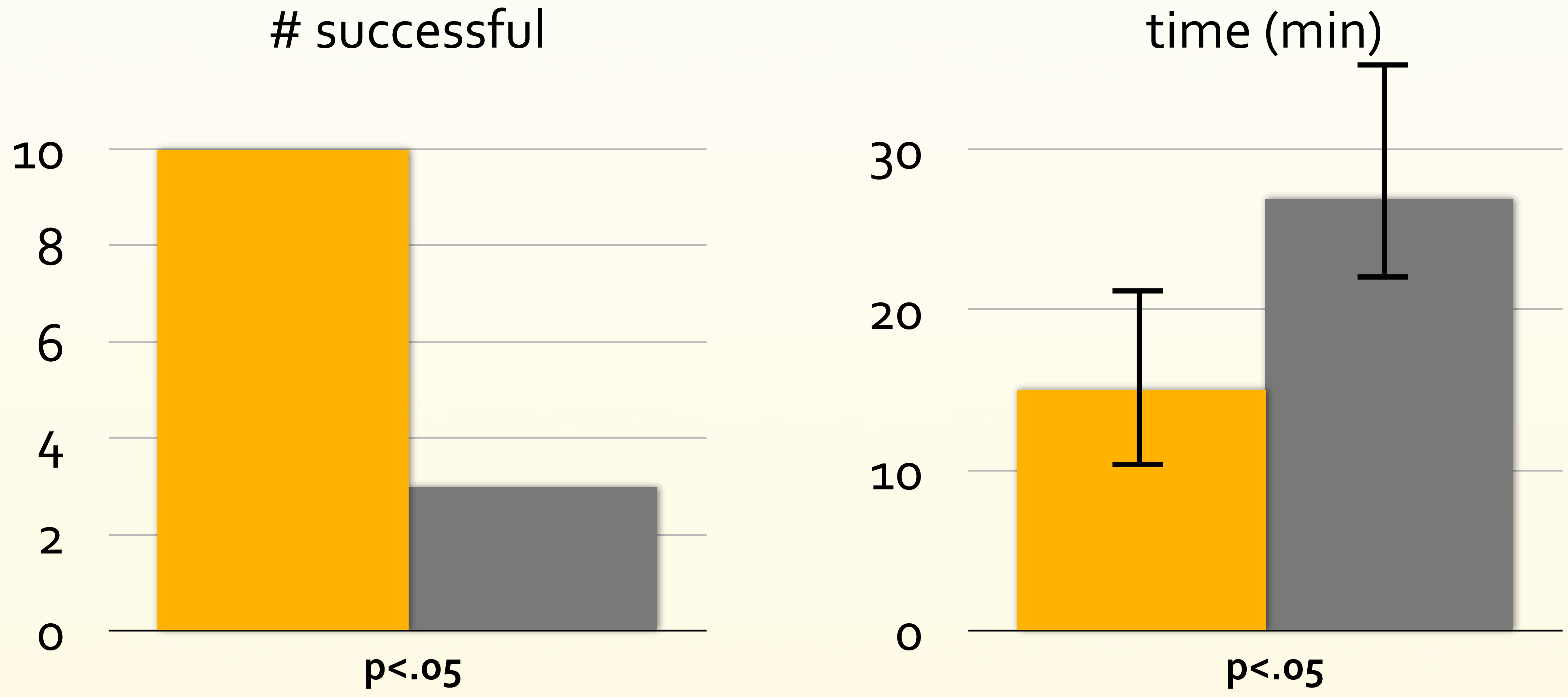
Tell the experimenter when you are done with your recommendation.

sample

- 20 **masters** students in **software engineering**
  - all **non-native** English speakers
  - **0-10** years in the software industry, median **1.5 years**
  - **average** self-rated Java expertise (“beginner” to “expert” scale)
- groups **did not** significantly **differ** on any measures

task 1 results

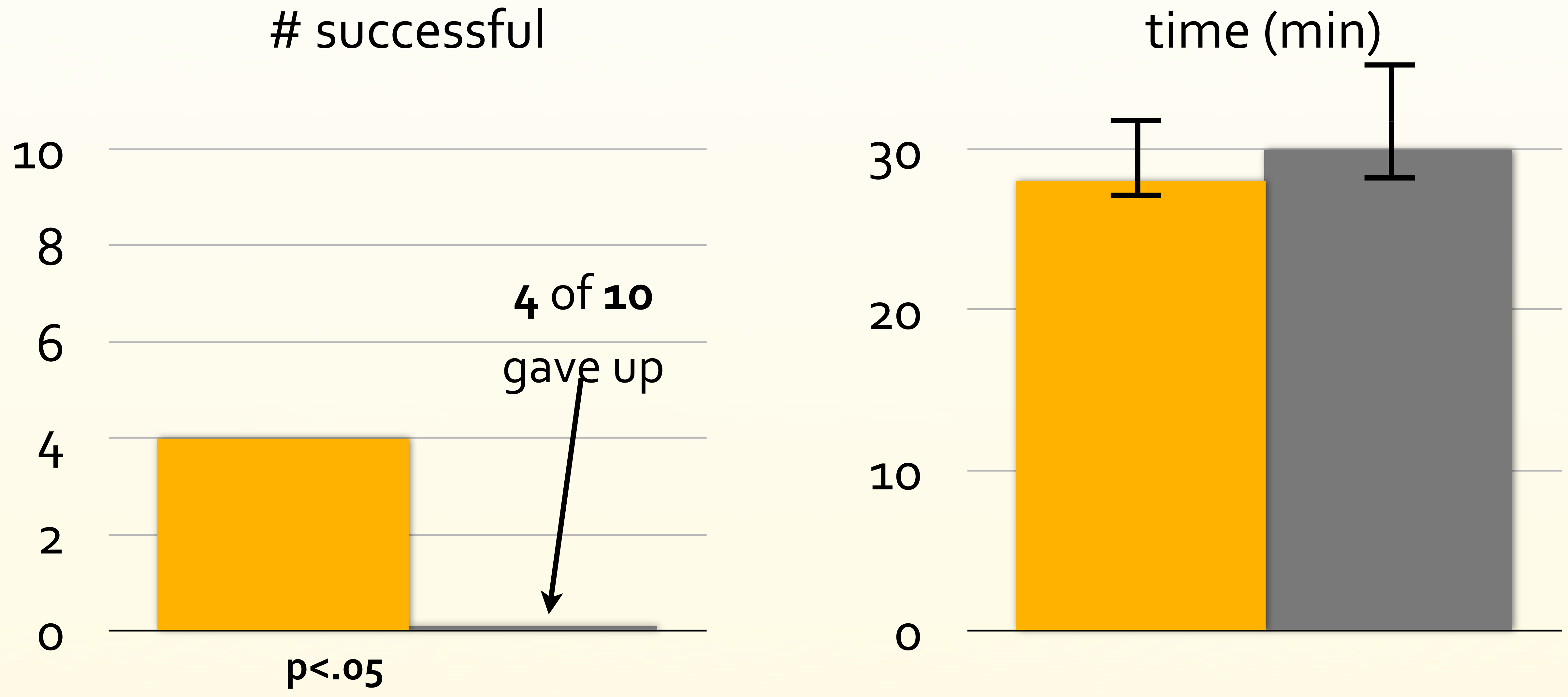
🟧 whyline ⬛ control



more successful in **half** the time

task 2 results

🟧 whyline ⬛ control



more successful in the **same** time

observations

- still need to choose question **carefully**
  - makes choice **explicit**, unlike current tools
- right questions take **closer** to bug, get you there **faster**
  - **less relevant** questions get you there, but with **more work**
- whyline gives **confidence** about **causality**?
  - control condition got near the bugs but didn’t know it

quotes

“It's so nice and straight and simple…”

“My god, this is so cool…”

“When can I get this for C?”

some limitations

memory and **performance** can be bottlenecks

infeasible for **long** executions, **real time** software

quality of **question phrasing** $\propto$ quality of **identifiers**

question and answer **precision** $\propto$ **type** information

some limitations

no **change suggestions**, just **causal** explanations

good for **functional correctness**, less for other **qualities**

good for ‘**where** is the buggy code’, not
‘why is the **code** buggy’

summary

current tools require **guessing**, costing time, money and accuracy of knowledge

the **whyline** limits guesswork by supporting queries on **program output**

the **whyline** saves time, improves **success** rates

future work

whyline for **education**

whyline for **teams**

- discovering **collaboration** requirements
- designing **annotations** and communication tools

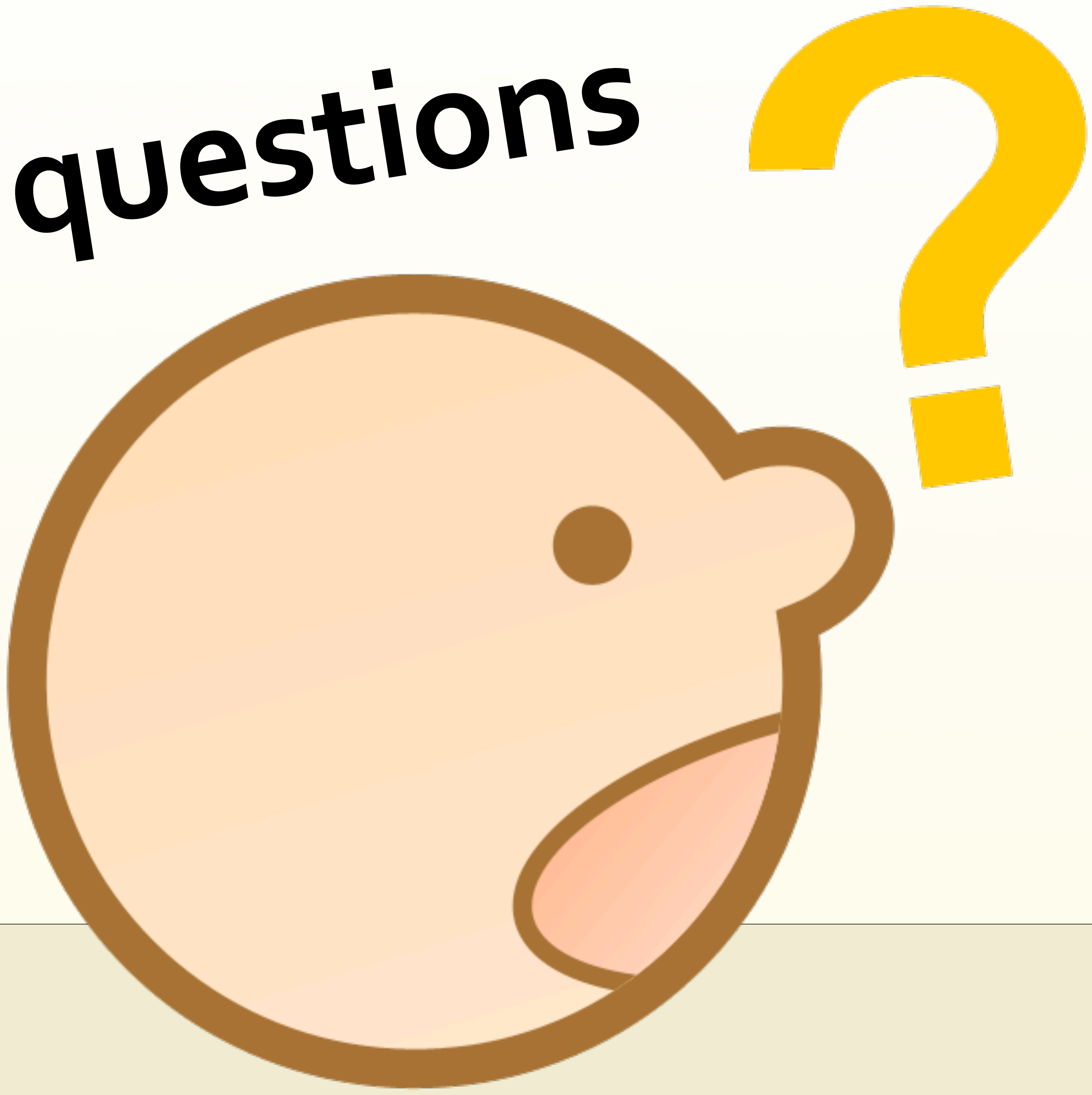
the other half of fixing a bug

- understanding **design rationale** behind code
- **why** is the code written this way?
- is this bug **important** to fix?

future work

- information work
  - **interaction designers'** collaboration with developers
  - **scientists**' use of technology
  - **students'** use of **statistics**
  - **engineers'** use of **specifications**
- democratizing **access** to computing
  - new domain-specific languages and tools

questions



thanks to
Polo Chau
for this icon!

thank you to

my thesis committee

Brad Myers
Bonnie John
Jonathan Aldrich
Gail Murphy

past and present
HCII students

my family

and many others…

This work was supported by the National Science Foundation under NSF grant IIS-0329090 and the EUSES consortium under NSF grant ITR CCR-0324770. The author is also supported by an NDSEG fellowship and by a NSF Graduate Research Fellowship.