CSE 8B Today

Don’t forget: Fill out your CAPES, TA evaluations, and your survey

PSA8 due WEDNESDAY week 10 (+1 slip day if you have any left)

PSA8 + Threads + Exceptions

Simulations? For what?

Cool-Amazing Scientific Computer Simulations
CSE 8B Today

Trying to make an animation...
(steps in this class example are very similar to PSA8 steps... SOME are already given in the starter files)

Threads

vs. SwingTimers
// The start button
JButton jbtStart = new JButton( "Start" );
ActionListener ssListen = new StartStopListener();
jbtStart.addActionListener( ssListen );

// The stop button
JButton jbtStop = new JButton( "Stop" );
jbtStop.addActionListener( ssListen );

THIS IS NOT A CLASS!!!
But we are allowed to use the "interface" name as the type.
It means "any" class that Implements that interface can be assigned to that variable.

1. ADD START/STOP animation button listener
class StartStopListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        JButton src = (JButton)e.getSource();
        if (src.getText().equals( "Start" )) {
            int panelNum = 0;
            for ( int i = 0; i < 20; i++ ) {
                if ( board[panelNum] == ' ' ) {
                    board[panelNum] = turn;
                } else {
                    board[panelNum] = ' ';
                }
                panelNum = (panelNum + 1) % 3;
            }
            repaint();
            try {
                Thread.sleep( 500 );
            } catch ( InterruptedException e ) {} 
        } else if (src.getText().equals( "Stop" )) {
            System.out.println( "STOP!!" );
        }
    }
}

2. Actually implement the START/STOP button listener class

What will happen when the user clicks the Start button?
C. The GUI will show only the last frame of 20 frames of animation, but it will take a lot longer!!

This Puts the MAIN GUI thread to sleep 500 ms ...

WE NEED "ANOTHER" THREAD ... ➔ DEMO!
public class ConnectFourAnimDemo extends JFrame implements Runnable {
    private boolean running; // Should the animation run?
    public ConnectFourAnimationDemo() {
        new Thread( this ).start();
    }
    public void run() {
        try {
            int panelNum = 0;
            while( true ) {
                if ( running ) {
                    if ( board[panelNum] == ' ' )
                        board[panelNum] = turn;
                    else
                        board[panelNum] = ' ';
                    panelNum = (panelNum + 1) % 3;
                    repaint();
                }
                Thread.sleep( 500 );
            }
        } catch (InterruptedException ex) { }
    }
    catch (InterruptedException ex) { }
}

A class variable sets the “state” of the animation
Launch the run method in a separate thread.
Can “run” separate threads
repaint asks the main GUI thread to repaint.
But sleep happens in a separate thread so is doesn’t affect the main thread as it happened before ...
public class CritterController extends JApplet implements Runnable

    // Instance variables that maintain various parts of the simulation
    private ArrayList<Critter> critterList;  /**< The Critters in the world */
    private Interactor actor; /**< ... object that controls how Critters interact */
    private boolean running;  /**< Whether or not the simulation is running */
    private CritterPanel world; /**< The custom JPanel that displays the Critters */
    private JLabel stateLabel; /**< The label at the top of the panel */
    private JLabel critLabel; /**< The label next to the Critter buttons */

    public init() { // ALL DONE, including how to start the thread...
        // rest of the init code . . .
        new Thread( this ).start();
    }

    public void run(){ // ALL DONE! But method
        try {
            while( true ) {
                if ( running ) {
                    Rectangle r = new Rectangle( . . .);
                    for ( int i = 0; i<critterList.size(); i++ ) {
                        Critter crit = (Critter) critterList.get(i);  /**< PICK A CRITTER!
                        actor.interact( crit, critterList, r );  /**< MAKE it INTERACT with others
                    }
                    repaint(); // UPDATE how things look like after ALL THE INTERACTIONS!
                }
                Thread.sleep( DELAY ); // let’s wait a bit before “interacting” again ...
            }
        } catch (InterruptedException ex) { //want to do something if this crashes?   }
    }
DON’T PANIC! Many of these steps are given in PSA8 starter files...

The code you need to use about Threads and Exceptions is given in the starter files! → **NO NEED TO WAIT TO START PSA8**! (in a minute... you’ll know already all you need to finish it!)

TODAY we’re learning to understand what the thread-related Code means

TUESDAY a few more PSA8 final tips and understand better what the Exception-handling related code means
2. Actually implement the START/STOP button listener class . . .

So . . . Run() is “doing” the board variable updates IF simulation is running; and the main is handling the repaint() request. DO I need to put something in the button listener then?

```java
class StartStopListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        JButton src = (JButton)e.getSource();
        if (src.getText().equals( "Start" )) {
            ___________ ?? __________________
        }
        else if (src.getText().equals( "Stop" )) {
            ___________ ?? __________________
        }
    }
}
```

Fill in the missing to correctly control the animation.
Trying to WRAP up... our ConnectFourAnimationDemo object IS our task object.

```java
public class ConnectFourAnimationDemo extends JFrame implements Runnable {
    // ...

    public void run() {
        try {
            int panelNum = 0;
            while (true) {
                if (running) {
                    if (board[panelNum] == ' ') {
                        board[panelNum] = turn;
                    } else {
                        board[panelNum] = ' ';  
                        panelNum = (panelNum + 1) % 3;
                    }
                    repaint();
                }
                Thread.sleep(500);
            }
        } catch (InterruptedException ex) {
        }
    }
}
```

In the ConnectFourAnimationDemo constructor, what line of code will execute the ConnectFourAnimationDemo’s run method in a new Thread?

A. `Thread.start(this);`
B. `new Thread(this).start();`
C. `this.run();`
D. `ConnectFourAnimationDemo.run();`
public class ConnectFourAnimationDemo extends JFrame implements Runnable {
    private boolean running;  // Should the animation run?
    public ConnectFourAnimationDemo() {
        new Thread( this ).start();  // Can be run in a separate thread

        public void run() {
            try {
                int panelNum = 0;
                while ( true ) {
                    if ( running ) {
                        if ( board[panelNum] == ' ' )
                            board[panelNum] = turn;
                        else
                            board[panelNum] = ' ';
                        panelNum = (panelNum + 1) % 3;
                        repaint();
                    }
                    Thread.sleep( 500 );
                }
            } catch (InterruptedException ex) { }
        }
        catch (InterruptedException ex) { } }
    }
}
public class AnimationDemoSwingTimer extends JFrame implements Runnable {
    private boolean running;  // Should the animation run?
    private Timer animationTimer;  // Timer to control the animation

    public ConnectFourAnimationDemo() {
        public AnimationDemoSwingTimer() {
            ...
            new Thread( this ).start();
            this.animationTimer = new Timer( 500, new AnimationListener() );
            this.animationTimer.stop();
        }
    }

    public void run() {
        class AnimationListener implements ActionListener {
            public void actionPerformed( ActionEvent e ) {
                if (board[panelNum] == ' ') {
                    board[panelNum] = turn;
                } else {
                    board[panelNum] = ' ';
                }
                panelNum = (panelNum + 1) % 3;
                repaint();
            }
        }
    }
}

Where did the loop go?

TIMER FIRES ACTION_EVENTS!
Every 500 ms
NEXT: What is an IOException??

... And what can we do with them