

CS1007: Object Oriented Design and Programming in Java

Lecture #23

April 13

Shlomo Hershkop

shlomo@cs.columbia.edu

Outline

- Combining threads and graphics
- Animation
- Some code examples
- Help with graphic part of homework

Announcement

- Passover holiday
- Thanks for TA's for running the class
- Regular class next Tuesday

Graphics

- So we've done some basic swing graphics
- Most of your programming has been
 - Setup window
 - Add bunch of elements
 - Register Events for specific component
 - Add logic to help events

 - Wait for something.....

Game programming

- When you code a game, its almost the same
- With the exception if you just sit there...the game player dies
 - If the player dies a horrible death, game is more popular (or so it seems)

Games ?

- So how do you think this is different than what we did ?66

Threaded events

- The main difference is that a thread of execution is manipulating parts of the game environment
- Lets start to play with some of this

Warning

- One of the big advantages is that Java comes with many libraries for you to use
- Lots of graphic ability packaged with java
 - Awt
 - Swing
- Many many libraries available for you to use
- So before reinventing the wheel

- Design your graphical environment
- Think of the components
- Try to find someone else's library to reuse

Big advantage

1. Don't waste time reinventing wheel
2. They've probably worked out a bunch of bugs you haven't even imagined yet 😊
3. Sometimes buggy, so good idea to keep an open mind during debugging
4. If many people use this library, more chance a bug has been discovered and fixed already

Where to find libraries ?

- First decide what you want
- Start searching sun java's website
- Else use your favorite engine to find one

Question

- When would it be a bad idea to use someone else's package ??

Answers

- If you are selling a product
 - Licensing issues
- If you want control, and source code isn't available
- No one has built it yet

Simple window with no action

```
public static void main(String[] args) {  
  
    JFrame easyWindow = new JFrame();  
    easyWindow.setLayout(new FlowLayout());  
    easyWindow.setSize(300, 300);  
    easyWindow.setTitle("This is your first window");  
    easyWindow.setDefaultCloseOperation(JFrame.EXIT_ON_C  
        LOSE);  
    easyWindow.setVisible(true);  
  
}
```

Progress

- Many times when you have a long running task you want to show progress
- So user wont run away
- Anyone used this in java ?

JProgressBar

- Allows you to draw a progress bar
- Can set it to show percentage progress

Tool tip

- You can register strings or html information to show up when you move the mouse over a component
- Its called `.setToolTipText()`

Enabled

- If you have a component, but don't want it to be used, can set it to be “grayed” out
- Call `.setEnabled(boolean)` to turn it on/off

Lets add to our simple window a progress bar

```
public static void main(String[] args) {  
  
    JFrame easyWindow = new JFrame();  
    easyWindow.setLayout(new FlowLayout());  
    easyWindow.setSize(300, 300);  
    easyWindow.setTitle("This is your first window");  
    easyWindow.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
  
    JProgressBar progress = new JProgressBar(1,100);  
    progress.setToolTipText("this will show up when you move  
        the mouse over");  
  
    easyWindow.add(progress);  
    easyWindow.setVisible(true);  
}
```

- Pretty straightforward, but two things:
- Its just sitting there
- User has no idea what the progress bar is for
 - Any ideas for solving these ?

Solutions

- Can create a thread to update it every X seconds
- Add a JLabel to tell us what the progress bar is

```
public static void main(String[] args) {

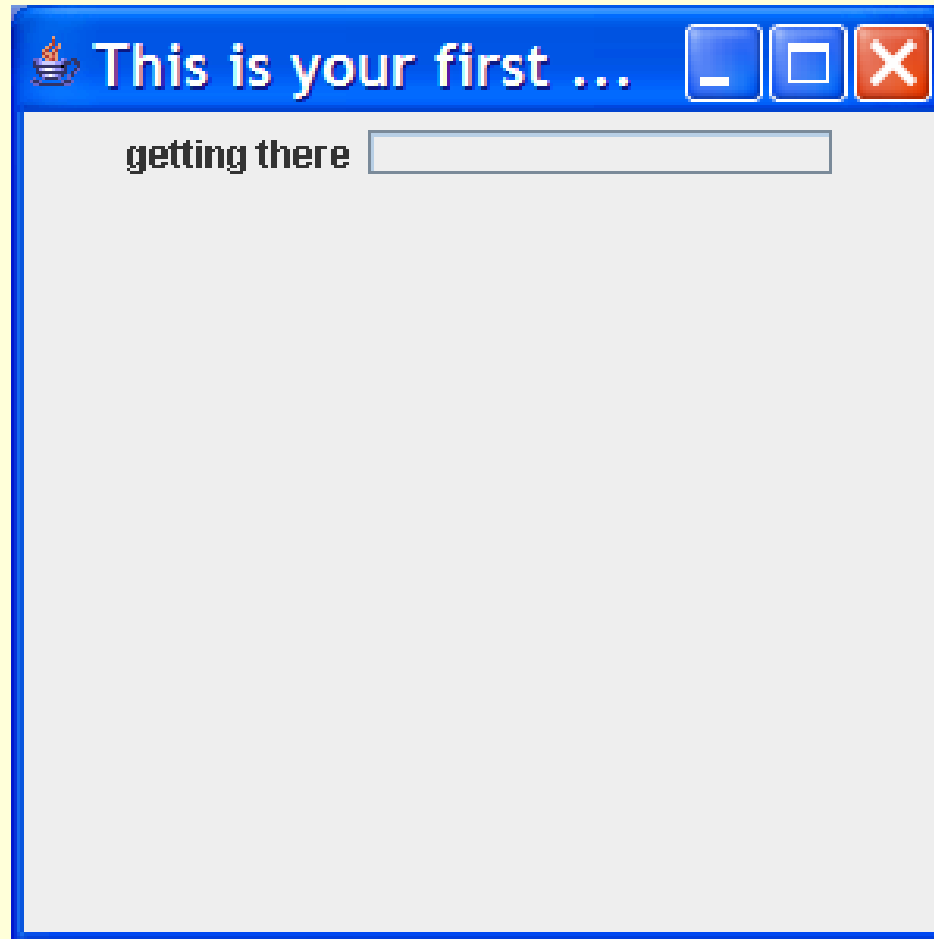
    JFrame easyWindow = new JFrame();
    easyWindow.setLayout(new FlowLayout());
    easyWindow.setSize(300, 300);
    easyWindow.setTitle("This is your first window");

    easyWindow.setDefaultCloseOperation(JFrame.EXIT_ON_C
        LOSE);

    JProgressBar progress = new JProgressBar(1,100);
    progress.setToolTipText("this will show up when you
        move the mouse over");
    easyWindow.add(new JLabel("getting there"));

    easyWindow.add(progress);
    easyWindow.setVisible(true);
}
```

Screen shot



- Now lets create a thread to do something to our progress bar

- Oh yes: `setStringPainted(true);`
 - Will show percentage done

Code we are adding

```
new Thread() {  
  
    public void run() {  
        for(int i=0;i<100;i++){  
            progress.setValue(i);  
            try{  
                sleep(100);  
            }catch(InterruptedException e){}  
        }  
    } } .start();
```

```
JFrame easyWindow = new JFrame();
    easyWindow.setLayout(new FlowLayout());
    easyWindow.setSize(300, 300);
    easyWindow.setTitle("This is your first window");
    easyWindow.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    final JProgressBar progress = new JProgressBar(1,100);
    progress.setToolTipText("this will show up when you move the mouse over");
    easyWindow.add(new JLabel("getting there"));
    easyWindow.add(progress);
    easyWindow.setVisible(true);

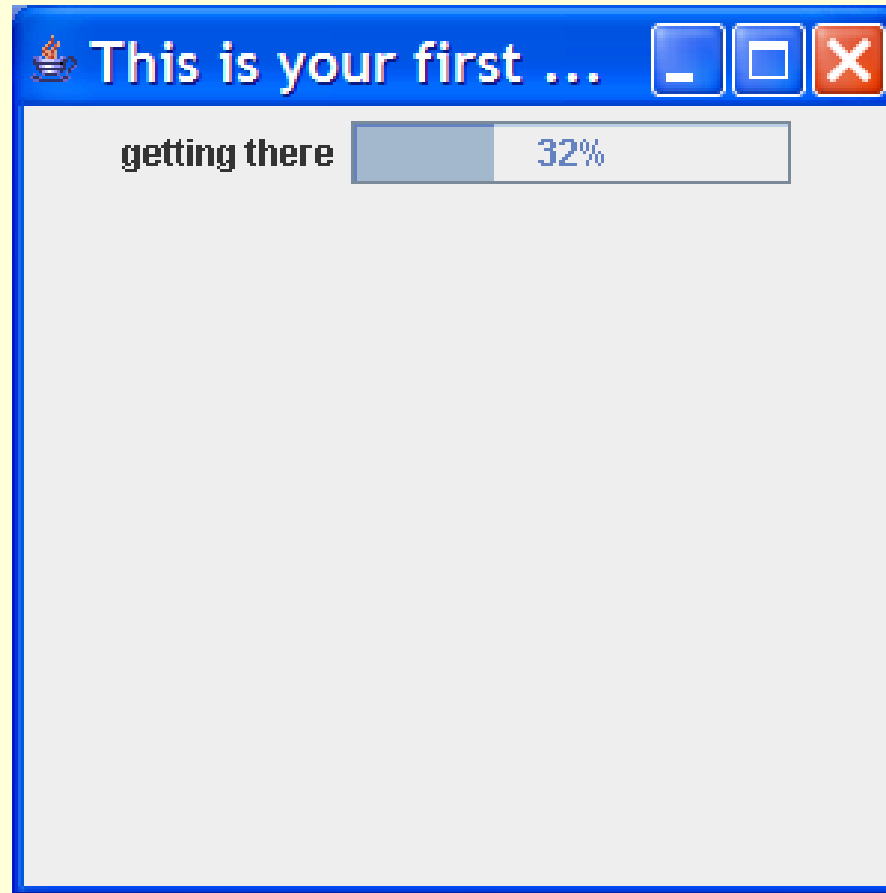
    progress.setStringPainted(true);

    new Thread(){
        public void run() {
            for(int i=0;i<100;i++){
                progress.setValue(i);

                try{
                    sleep(100);
                }catch(InterruptedException e){}

            }
        }
    }.start();
```

Screen shot



Challenge

- Can you write this class with 3 progress bars each running on their own random time.....when any one is done, it is declared the winner and others are stopped where they are??
- Hint: think of thread signaling ...
- We will cover this in next class

Reading

- 9-9.3
- Now we will talk about the homework