



Saint Vincent College

Department of Computing
and Information Science



A Programming Paradigm: Event-Driven Programming Game Example

Fr. Boniface, OSB
Lecture 21

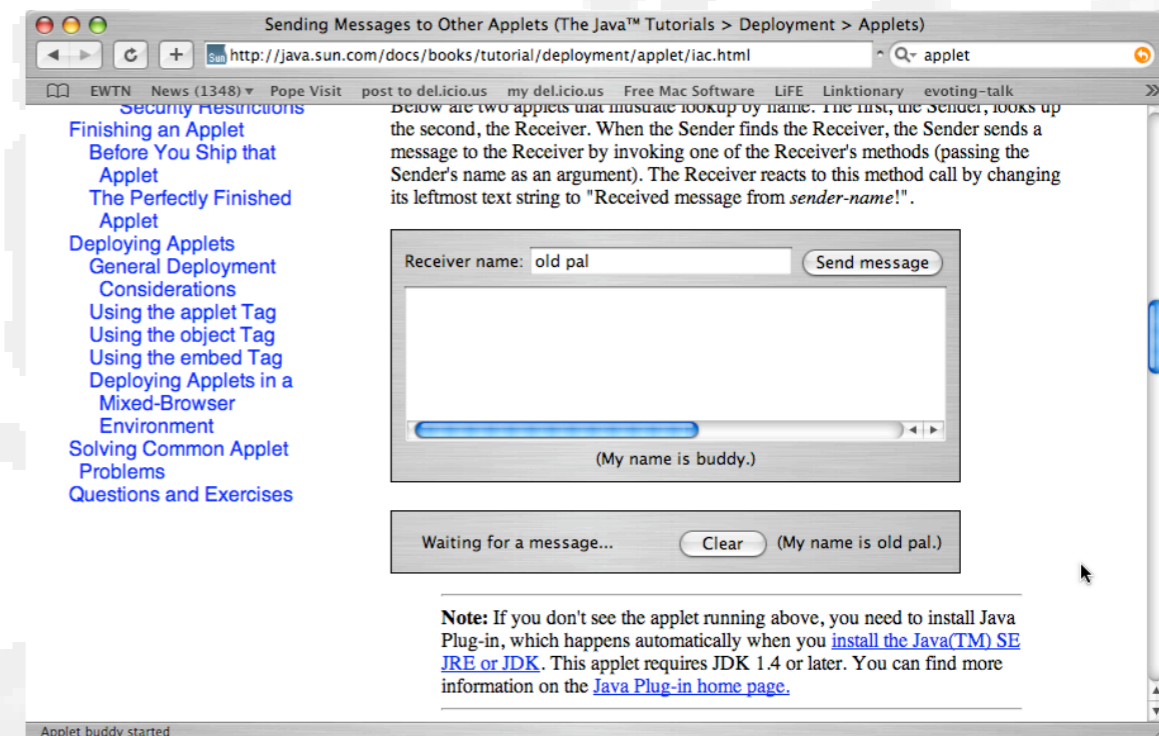
(some slides modified from Tucker/Noonan, copyright McGraw-Hill, 2006)

“And the Lord was not in the earthquake” (1 Kings 19:11)

Applets

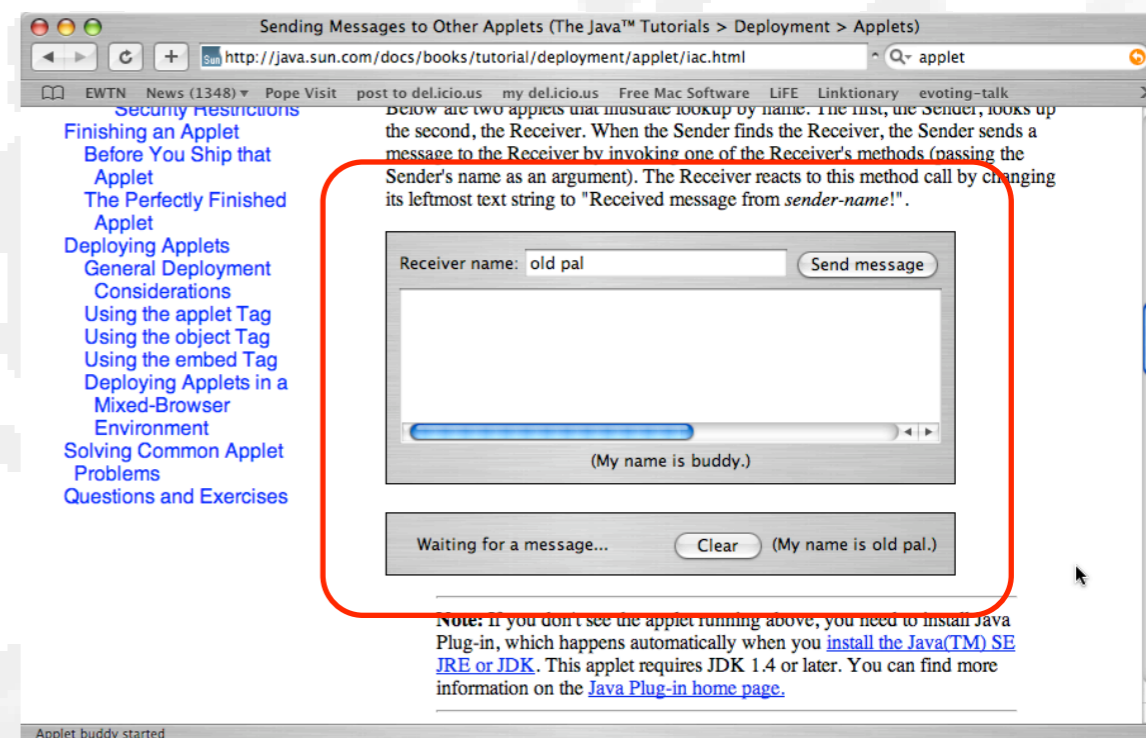


- An applet runs inside a web browser
- Can convert previous application
- Differences
 - ▶ Extend `JApplet`
 - ▶ Lacks a `main` method
 - ▶ Method `init` replaces constructor



Applets

- An applet runs inside a web browser
- Can convert previous application
- Differences
 - ▶ Extend `JApplet`
 - ▶ Lacks a `main` method
 - ▶ Method `init` replaces constructor



Converting Skeleton

```
public class SkeletonApplet extends JApplet {  
    private int lastX = 0;           // first click's x-y coordinates  
    private int lastY = 0;
```

...

```
public void init( ) {  
    JPanel panel = new JPanel( );  
  
    // Set the background color and mouse listener  
    setBackground(Color.white);  
    addMouseListener(new MouseHandler());
```

- **extends JApplet, not JPanel**
- **replace Skeleton() constructor with init()**

Converting Skeleton cont...



```
public void init( ) {  
    JPanel panel = new JPanel( );  
  
    setBackground(Color.white);  
    addMouseListener(new MouseHandler());  
    ...  
    echoArea = new JTextArea(2, 40);  
    echoArea.setEditable(false);  
    panel.add(echoArea);  
  
    setSize(500,500);  
    getContentPane().add(panel);  
}
```

```
public static void main(String args[] ) {  
    JFrame frame = new JFrame();  
    frame.setDefaultCloseOperation(  
        JFrame.EXIT_ON_CLOSE);  
  
    Skeleton panel = new Skeleton( );  
  
    frame.getContentPane().add(panel);  
    frame.setSize(500, 500);  
    frame.setVisible(true);  
}
```

- no main method (merged into init())
- this instead of Frame
- JPanel panel instead of Skeleton panel
- panel.add(echoArea) instead of add(echoArea)

Converting Skeleton cont...



```
public void init( ) {
    JPanel panel = new JPanel( );

    setBackground(Color.white);
    addMouseListener(new MouseHandler());
    ...
    echoArea = new JTextArea(2, 40);
    echoArea.setEditable(false);
    panel.add(echoArea);

    setSize(500,500);
    getContentPane().add(panel);
}
```

```
public static void main(String args[] ) {
    JFrame frame = new JFrame();
    frame.setDefaultCloseOperation(
        JFrame.EXIT_ON_CLOSE);

    Skeleton panel = new Skeleton( );

    frame.getContentPane().add(panel);
    frame.setSize(500, 500);
    frame.setVisible(true);
}
```

- no main method (merged into `init()`)
- this instead of `Frame`
- `JPanel panel` instead of `Skeleton panel`
- `panel.add(echoArea)` instead of `add(echoArea)`

Converting Skeleton cont...



```
public void init( ) {  
    JPanel panel = new JPanel( );  
  
    setBackground(Color.white);  
    addMouseListener(new MouseHandler());  
    ...  
    echoArea = new JTextArea(2, 40);  
    echoArea.setEditable(false);  
    panel.add(echoArea);  
  
    setSize(500,500);  
    getContentPane().add(panel);  
}  
  
public static void main(String args[] ) {  
    JFrame frame = new JFrame();  
    frame.setDefaultCloseOperation(  
        JFrame.EXIT_ON_CLOSE);  
  
    Skeleton panel = new Skeleton( );  
  
    frame.getContentPane().add(panel);  
    frame.setSize(500, 500);  
    frame.setVisible(true);  
}
```

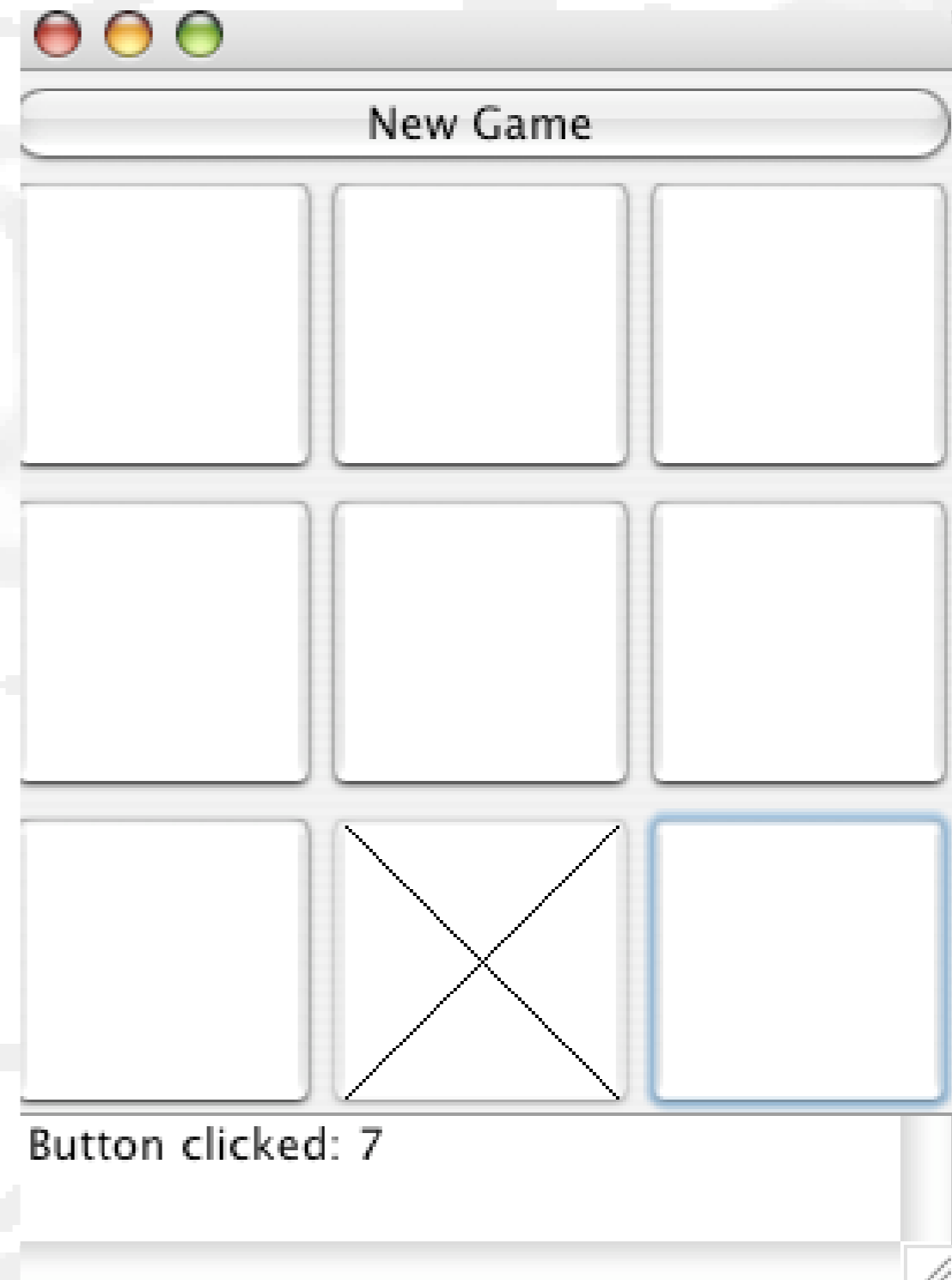
- no main method (merged into init())
- this instead of Frame
- JPanel panel instead of Skeleton panel
- panel.add(echoArea) instead of add(echoArea)

Tic Tac Toe



- **Model**
- **View**
- **Controller**

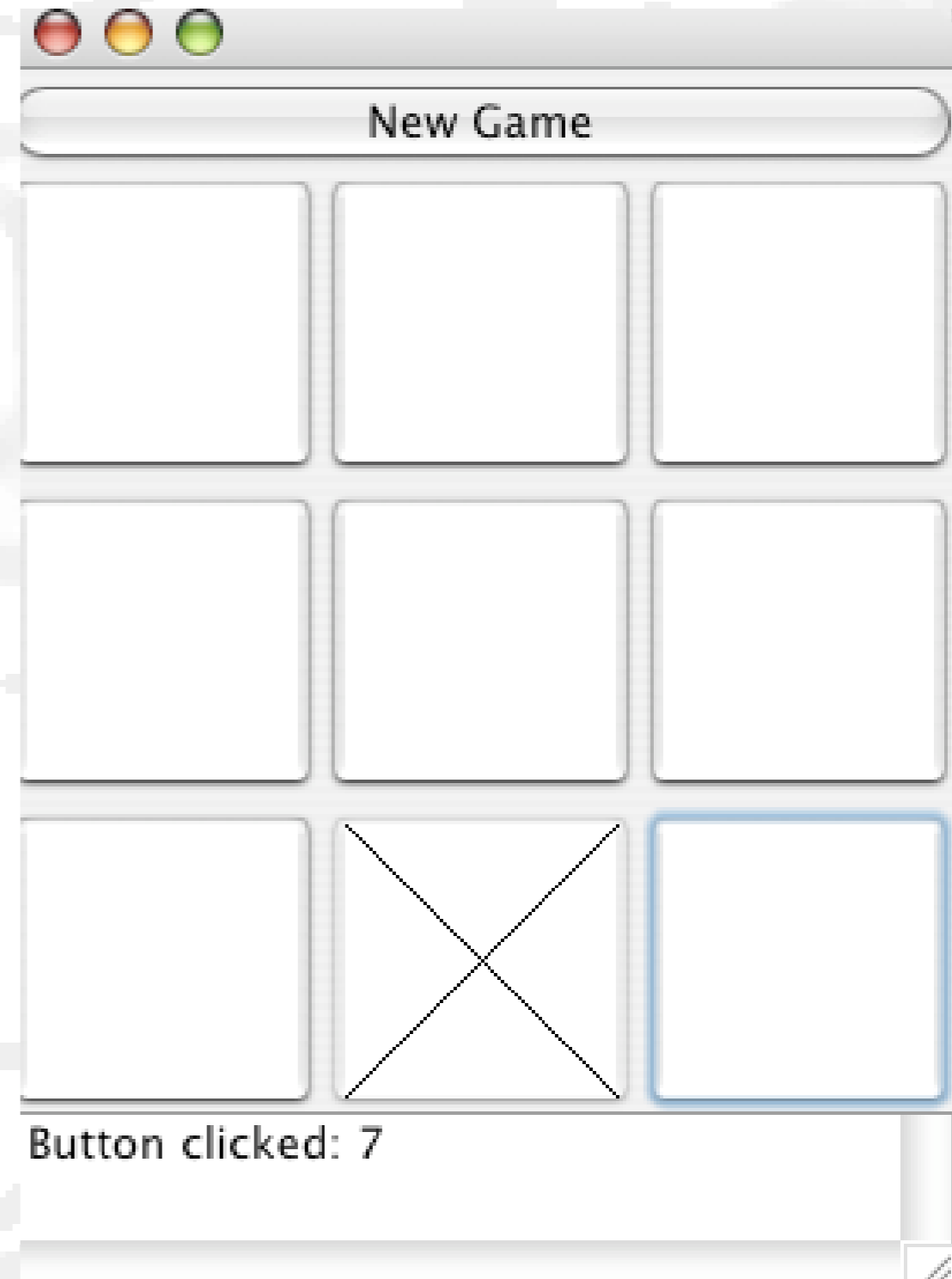
- Each square
 - ▶ X, O or empty
- Output log
 - ▶ last move
 - ▶ whose turn
 - ▶ winner
 - ▶ new game



Controller



- Each square (button)
 - ▶ ActionHandler per button
 - ▶ Each button knows own #
 - ▶ Updates model
 - ▶ Updates view
 - ▶ Can check for a win
- New Game button
 - ▶ resets model
 - ▶ resets view



Button ActionListener



```
private class ActionListener implements ActionListener {
    private int button;

    public ActionListener(int button) {
        this.button = button;
    }

    public void actionPerformed(ActionEvent e) {
        Icon icon = game.move(button);
        square[button].setIcon(icon);
        square[button].setEnabled(false);
        if (game.won()) {
            setButtons(false);
            echoArea.setText(game.player() + " wins!");
        } else
            echoArea.setText("Button clicked: " + button);
    }
}
```

Model



```
public class Game {  
  
    private final int ICON_SIZE = 80;  
    private final CellIcon off = new CellEmptyIcon(ICON_SIZE);  
    private final CellIcon ex = new CellXIcon(ICON_SIZE);  
    private final CellIcon oh = new CellOIcon(ICON_SIZE);  
    private CellIcon[] initial = { off, off, off, off, off, off, off, off, off,  
off };  
    private final CellIcon[] cell = { ex, oh };  
  
    private int player = -1; // Current player: EX, or OH  
    private int[] board = new int[9];  
}
```

- Just whose turn
- state of the board (stuff for drawing state of the board)
- check for winner

Model



```
public class Game {  
  
    private final int ICON_SIZE = 80;  
    private final CellIcon off = new CellEmptyIcon(ICON_SIZE);  
    private final CellIcon ex = new CellXIcon(ICON_SIZE);  
    private final CellIcon oh = new CellOIcon(ICON_SIZE);  
    private CellIcon[] initial = { off, off, off, off, off, off, off, off, off,  
off };  
    private final CellIcon[] cell = { ex, oh };  
  
    private int player = -1; // Current player: EX, or OH  
    private int[] board = new int[9];  
}
```

- Just whose turn
- state of the board (stuff for drawing state of the board)
- check for winner