

# Intro to Swing

Using Java Swing and MVC Design Pattern

# Today's Overview

Git Exercise

Swing basics

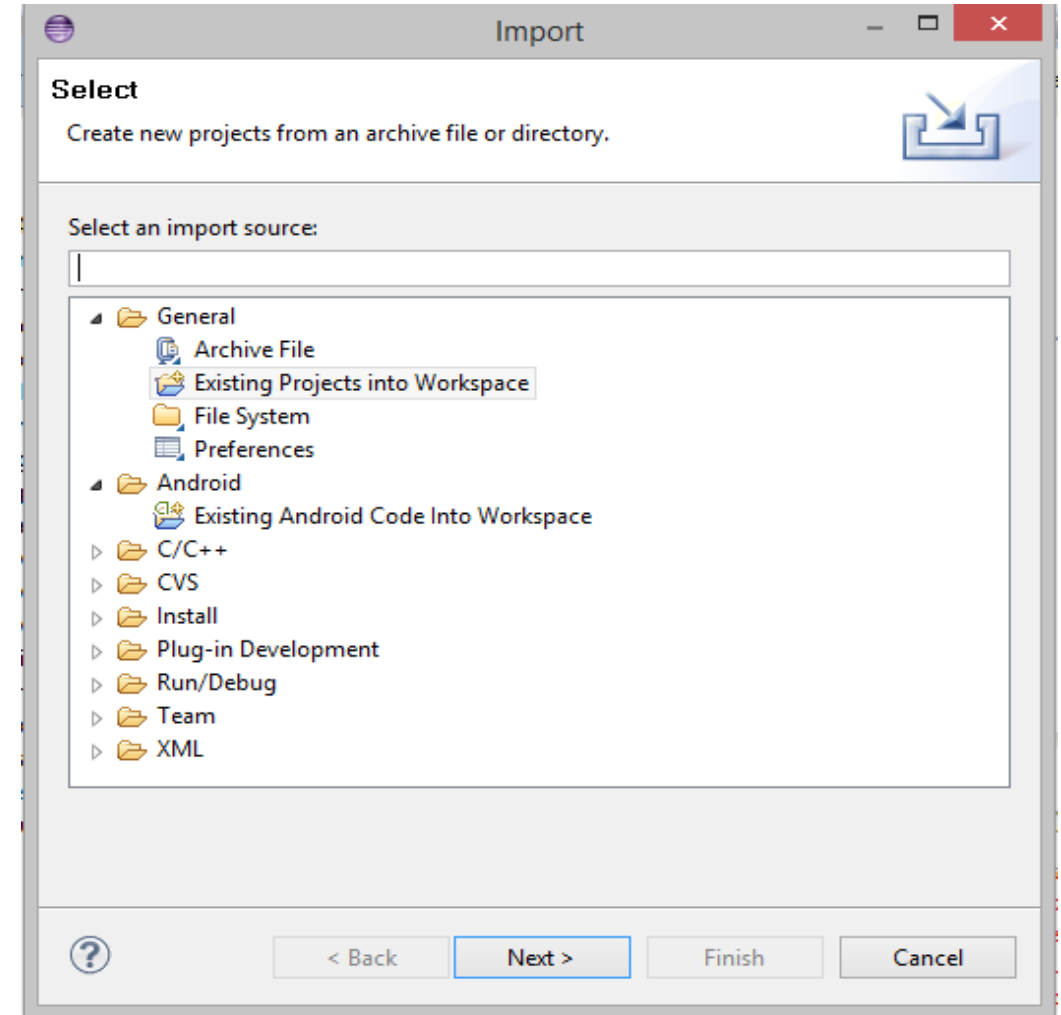
Swing Exercise

More Git exercise

# Importing a project

File -> Import -> Existing  
project into Workspace ->  
Browse for project

Go ahead and import today's example code



# Model View Controller Design Pattern

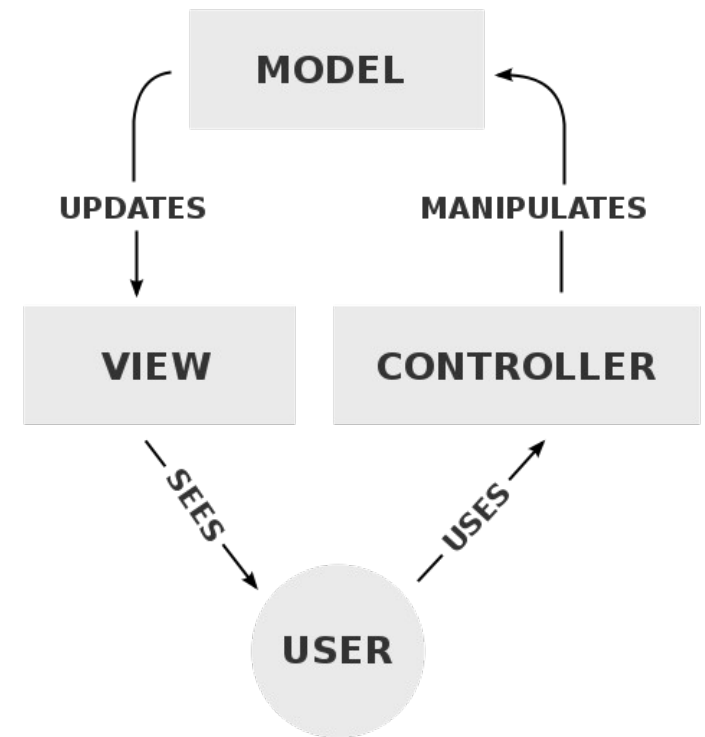
Why bother with MVC?

- High **Coupling**
- Low **Cohesion**
- **Modular**

**Model**: Representation of Data

**View**: Graphical Interface

**Controller**: Binds Model to View



# Model View Controller Design Pattern



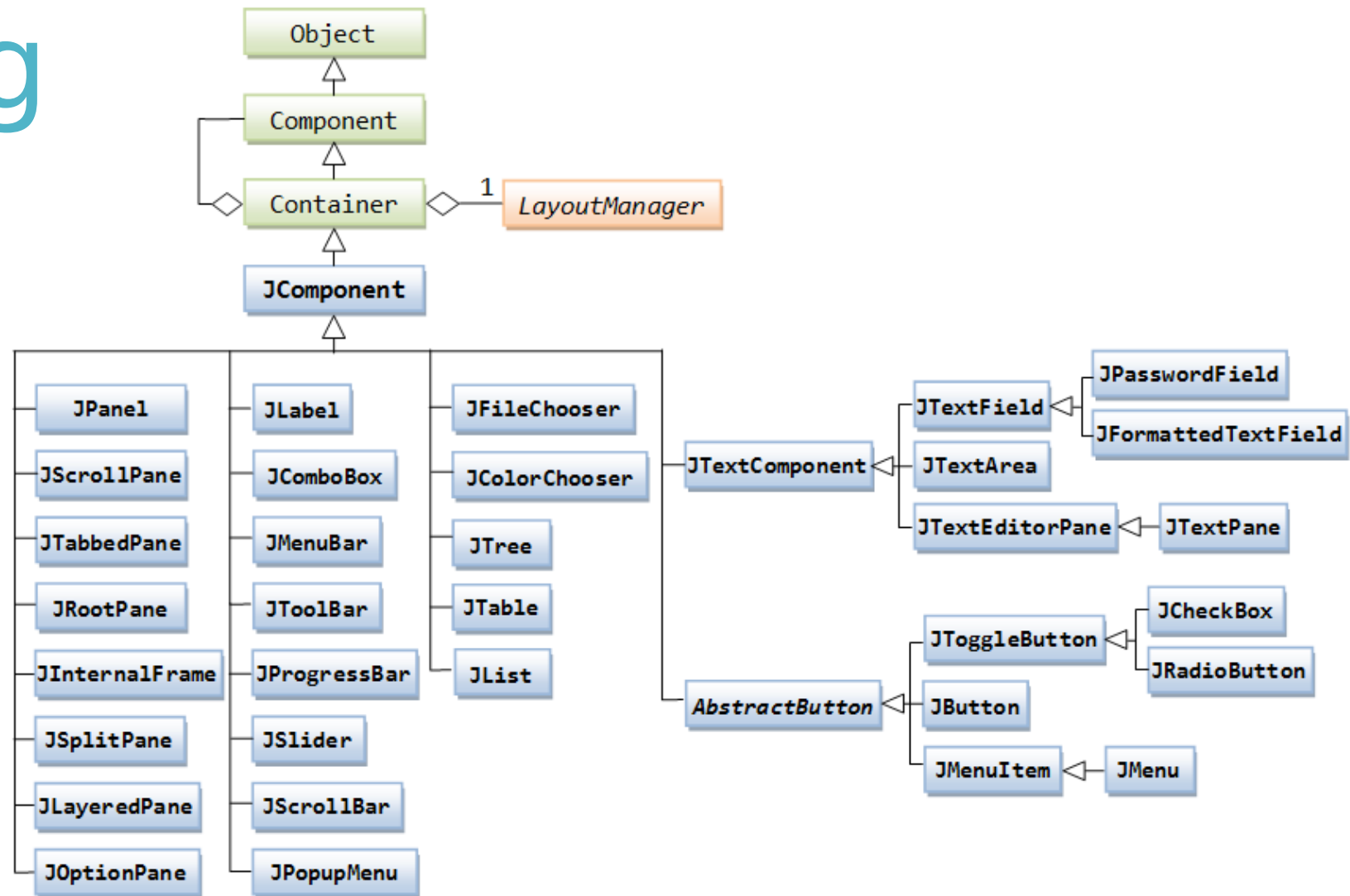


# Java Swing

The very basics

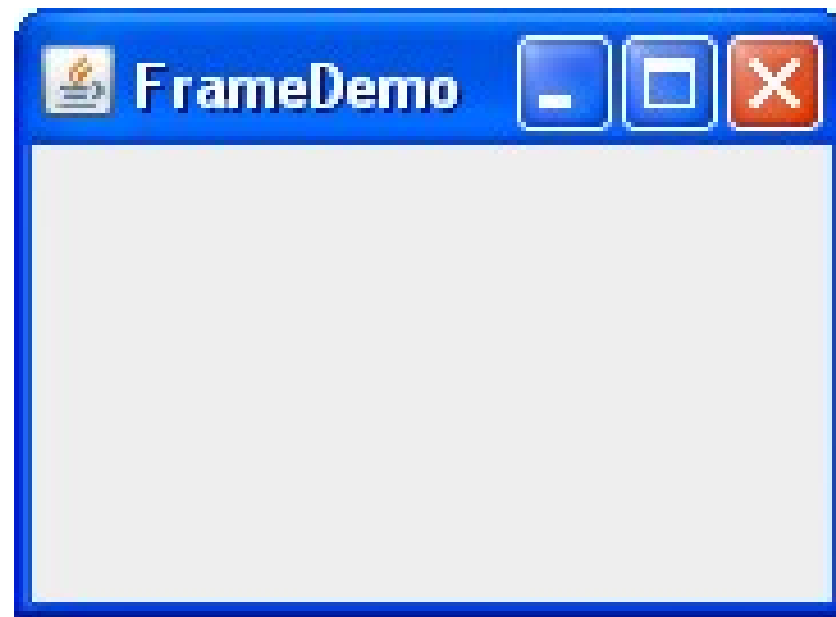
# Java Swing

Java Swing is the library used to make Graphical user interfaces in Java



# Swing Basics - JFrame

A JFrame object is your window



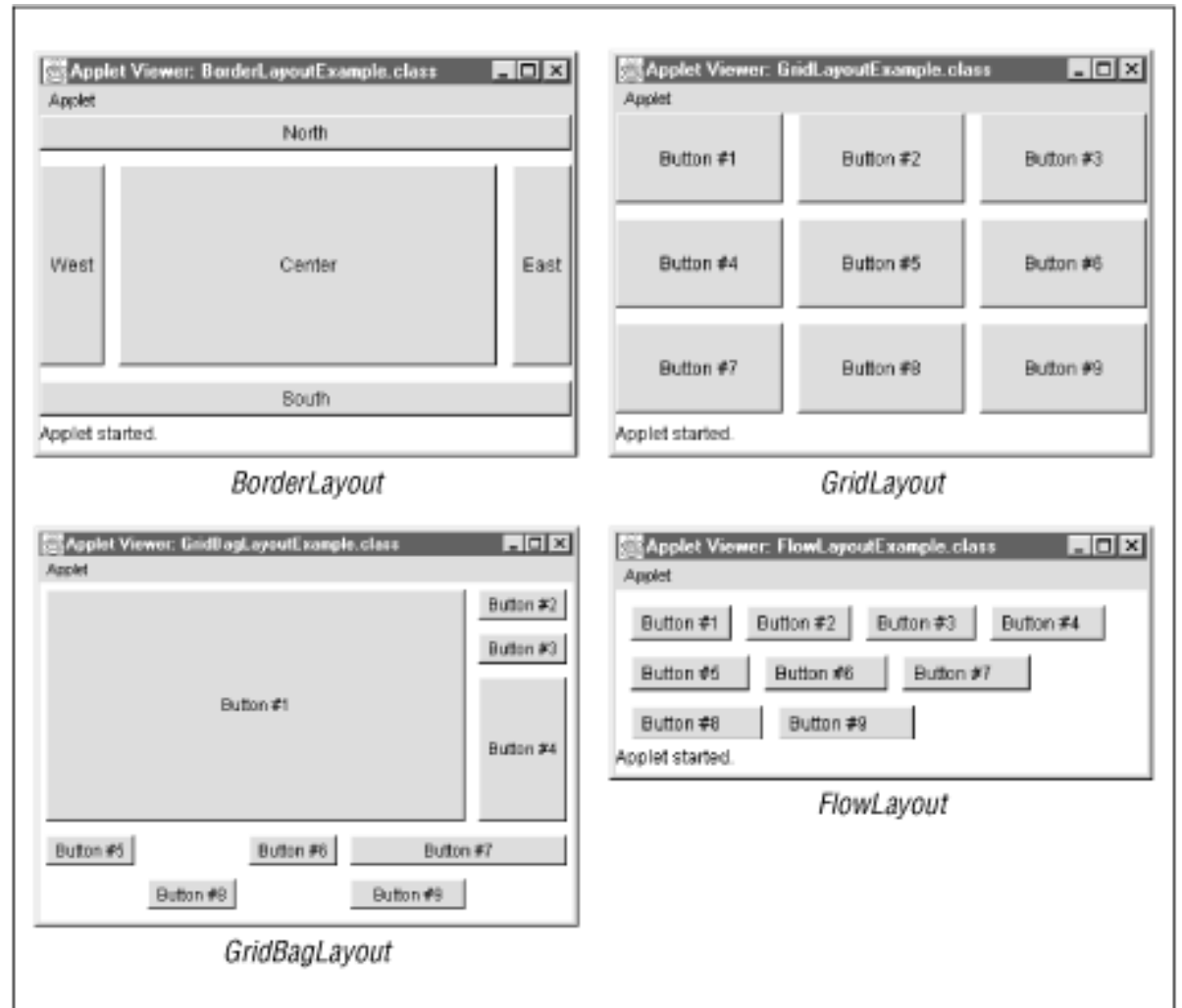


# Swing Basics - JFrame

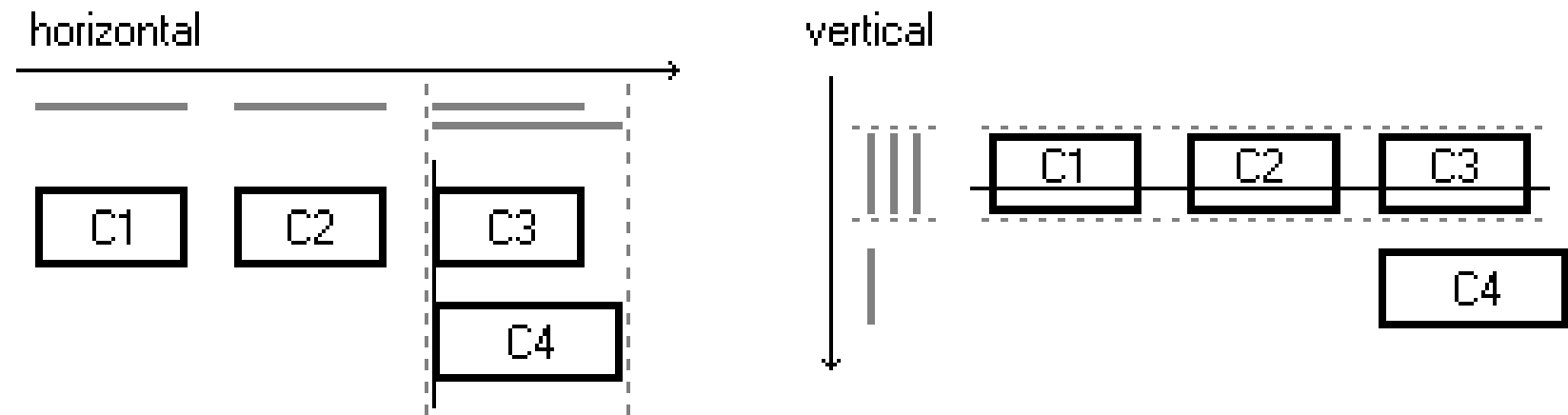
```
public class Application extends JFrame {  
  
    // constants  
    public static final int SCREEN_WIDTH = 400;  
    public static final int SCREEN_HEIGHT = 400;  
    public static final String APP_NAME = "Snake Game 2014";  
  
    public static final int TIMEOUT = 50;
```

# Swing Basics - Layouts

A layout is how **components** are **placed**



# Swing Basics - Layouts



horizontal layout = sequential group { c1, c2, **parallel group (LEFT)** { c3, c4 } }

vertical layout = **sequential group** { **parallel group (BASELINE)** { c1, c2, c3 }, c4 }

# Swing Basics – Action Listener

```
this.addParticipantButton.addActionListener(new ActionListener(){  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        addParticipantButtonActionPerformed(e);  
    }  
});
```

Action Listener **calls** a **method** when a particular **action** takes place

# Pair Programming exercise



# References

Java Game Tutorial: <http://zetcode.com/tutorials/javagamestutorial/>  
Java Group  
layout:https://docs.oracle.com/javase/tutorial/uiswing/layout/group.html