

CS 2334: Lab 12

Graphics and Binary File I/O

Menus

- Attached to a Frame or an entire program
- Drop down menus: easy access to high-level commands
 - Not uncommon to have at least a “File” menu

Menus

- Menu bar: bar at top of frame
 - Container of menus
- Menu (also a container): drop-down list of menu items:
 - Individual items
 - Sub-menus (also containers!)
- Selecting an individual item raises an event

Menu Example: Set up the Frame

```
// Frame creation
JFrame frame = new JFrame("Menu Test");
frame.setSize(400,600);
frame.setLocationRelativeTo(null);

// Will add menu code here

// Frame visibility and behavior
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setVisible(true);
```

Menu Example: Create Menu Bar, Menu and Items

```
// Menu Bar
JMenuBar menuBar = new JMenuBar();
frame.setJMenuBar(menuBar);

// Menu
JMenu menu = new JMenu("File");
menuBar.add(menu);

// Menu items
JMenuItem menuItem1 = new JMenuItem("Save first half");
JMenuItem menuItem2 = new JMenuItem("Save second half");

menu.add(menuItem1);
menu.add(menuItem2);
```

Menu Example: Menu Item Actions

```
// Attach an action to a menu item
menuItem1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        System.out.println("Item 1");
    }
});
```

Dialog Boxes

- Pop-up windows that exist for a very short period of time
- Ask a single question of the user & return
- Dialog boxes are typically “modal” windows
 - Only interaction may be with these windows (all other interaction is blocked)

Dialog Boxes

Many types, including:

- Confirmation
- File selection
- Color picking

Confirmation Dialog Box

```
int ret = JOptionPane.showConfirmDialog(null, "Are you sure?",  
                                       "Title", JOptionPane.YES_NO_OPTION);  
System.out.println(ret);
```

Notes:

- Returns 0 if “Yes” and 1 if “No”
- First parameter should generally be the parent object (this gives the dialog box a hint as to where it should appear)

File Selection Dialog Box

```
// File determines default directory (folder)
// Note: generally only need to do this once
JFileChooser fileChooser = new JFileChooser(new File("./data"));

// Prompt the user for a file to save to
// parent = reference to parent component
int returnVal = fileChooser.showSaveDialog(parent);
if (returnVal == JFileChooser.APPROVE_OPTION) {
    // Use the selected file
}
```

Color Picker Dialog Box

```
// parent = parent object
// color = default color
// colorChosen = selected color (could be null)
Color colorChosen = JColorChooser.showDialog(parent,
        "Choose a Color", color);
```

Binary I/O

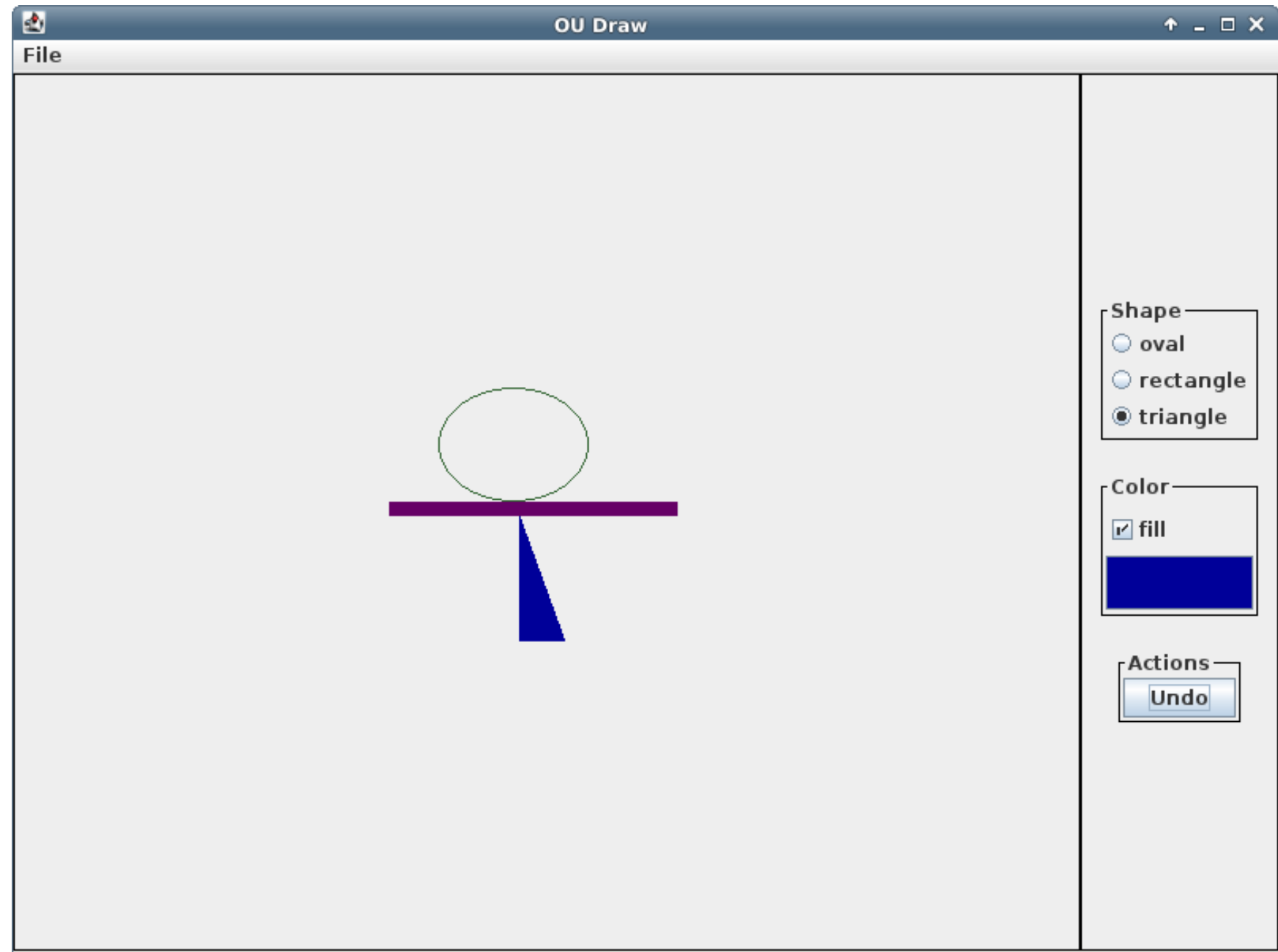
Already covered in class... Things to remember:

- Wrapping:
 - ObjectOutputStream around a FileOutputStream
 - ObjectInputStream around a FileInputStream
- Can write/read entire objects, which includes all sub objects
 - Must all be Serializable (a *marker* interface)
 - On read, must cast read object to the correct type
 - Order of object write/read matters

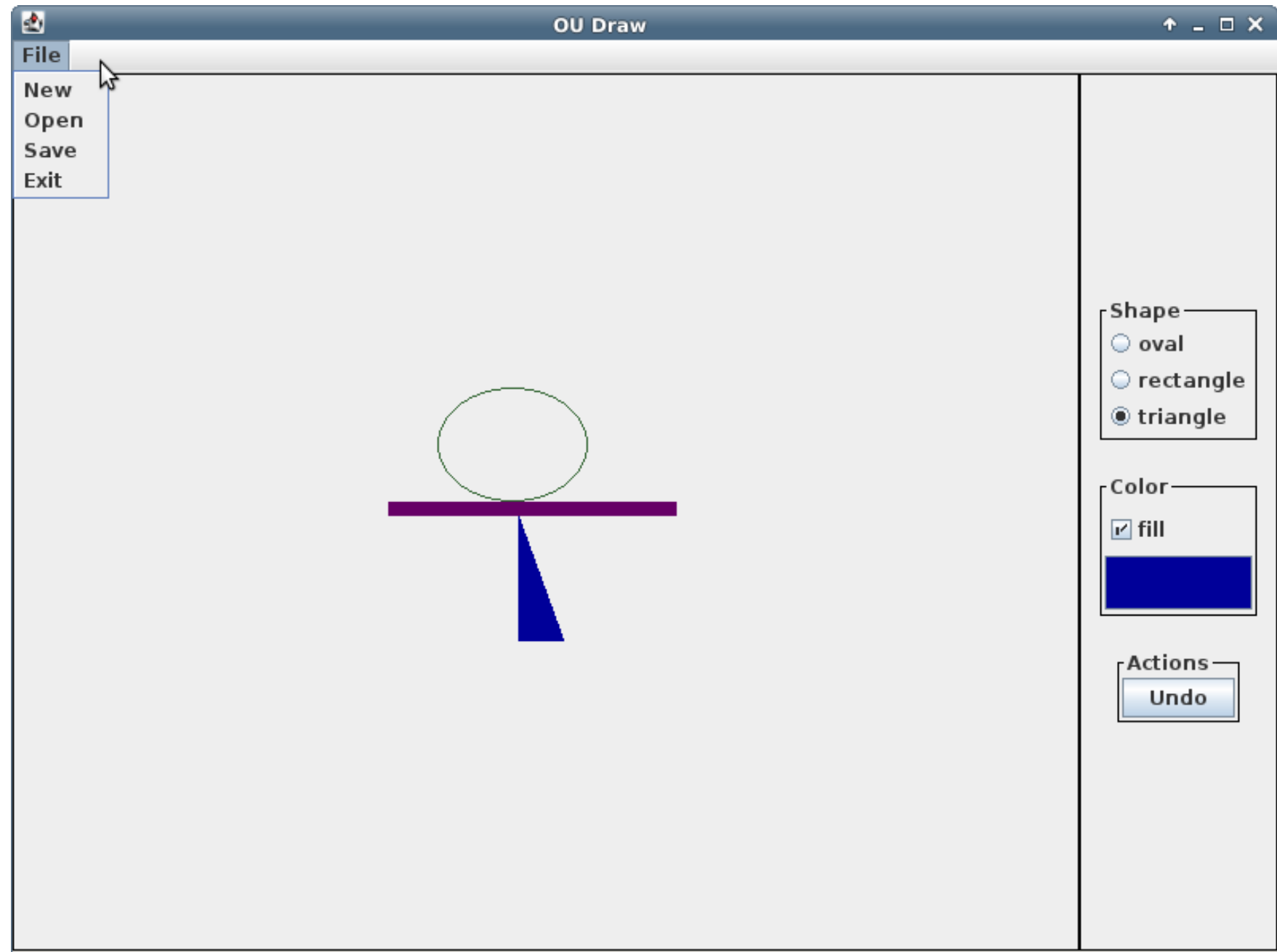
Lab 12

- Lab 10: we created and rendered a list of Shape objects
 - We will use this same implementation of Shapes
- Lab 12: user interface for drawing shapes
 - User selects: shape type, color and whether the shape is filled
 - User specifies shape location and size with mouse clicks in the Panel
 - User can save existing drawings and load old ones

Lab 12: GUI for Drawing

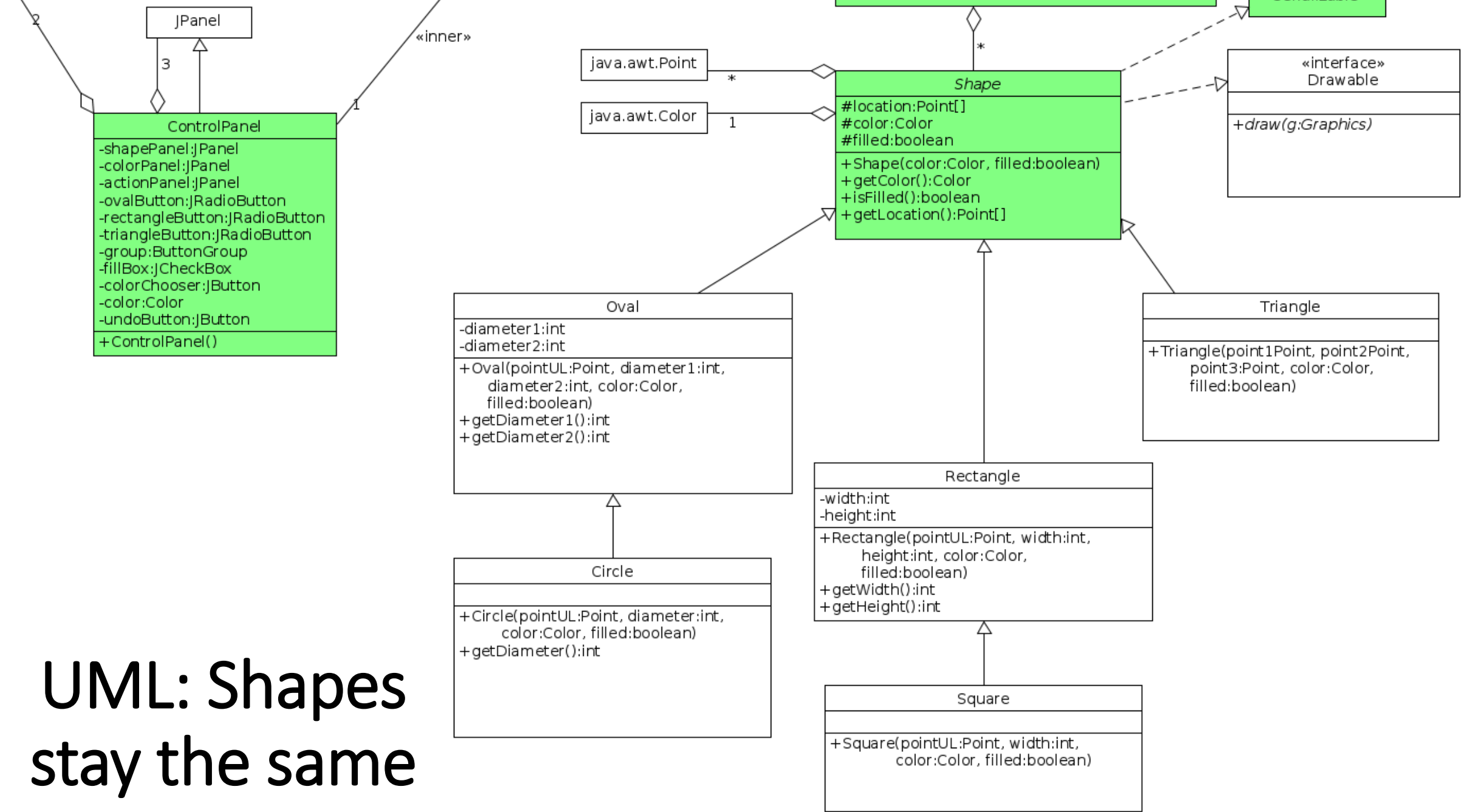


File Menu

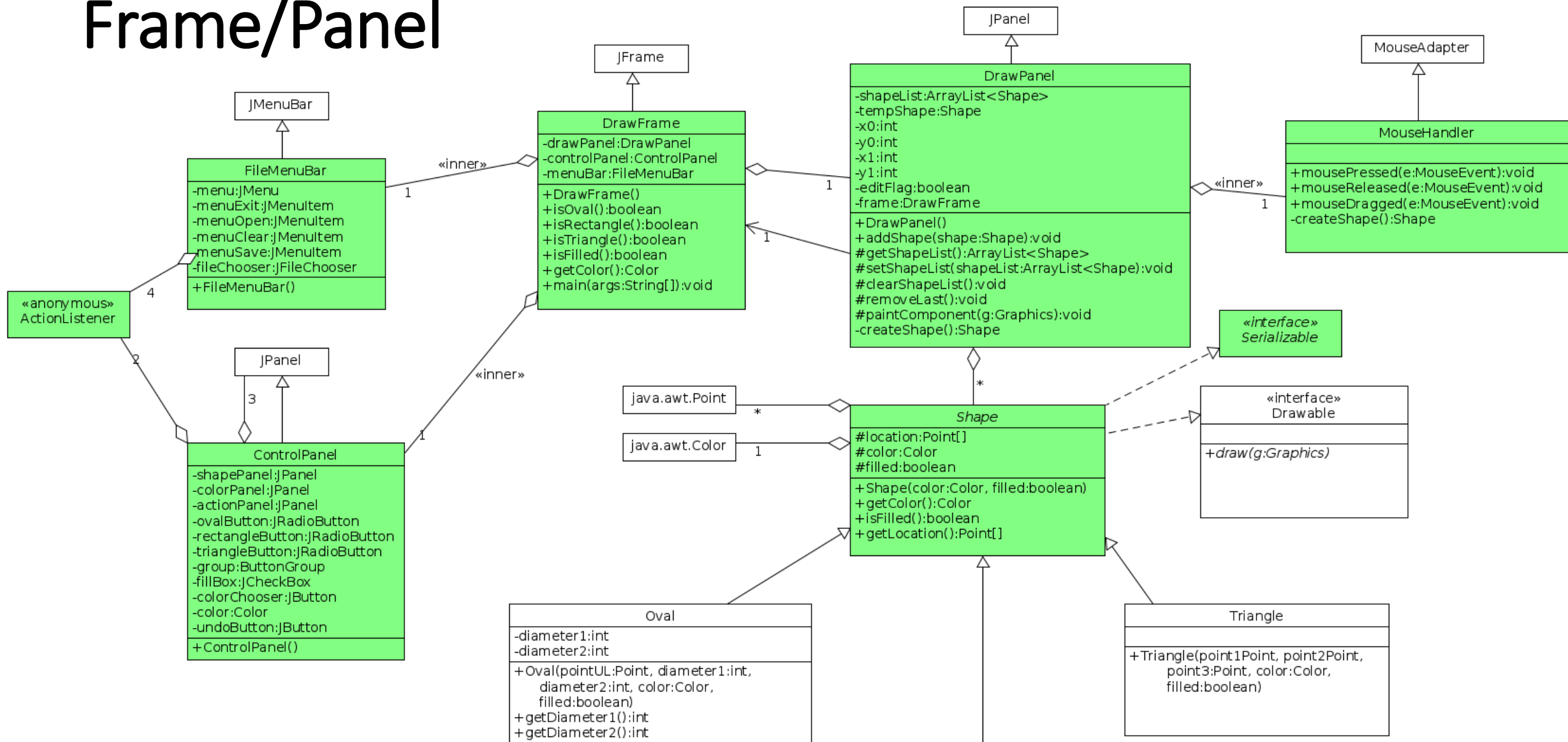


Demonstration ...

UML: Shapes stay the same



UML: Frame/Panel



Lab 12 Tasks

Classes to work on (and their inner classes)

- DrawPanel
- DrawFrame

- Read the documented API
- Look for “TODOs”

Lab 12: DrawPanel

- Complete implementation of `paintComponent()`
- Complete implementation of `createShape()`. Create a shape to render based on:
 - Points specified by the mouse button-up and button-down events
 - Choices made by the user: shape type, color and fill
 - Hint: `DrawFrame` provides methods that will help with this

Lab 12: DrawFrame

- Create the File menu:
 - Clear: remove all shapes from the picture
 - Open: read a new set of shapes from a file
 - Save: write the existing set of shapes to a file
 - Exit
- Add functionality for picking a color
 - Current color is default; new choice becomes the current color
- Add functionality for undoing the add of the last shape

Submission

- Submit only one file: lab12.zip (casing matters)
- Due date: Friday, November 13th @11:59pm
- Submit to lab12 dropbox on D2L