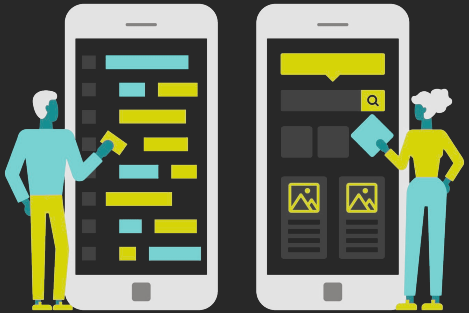
UI Basics and JavaFX

Dr. Isaac Griffith Idaho State University

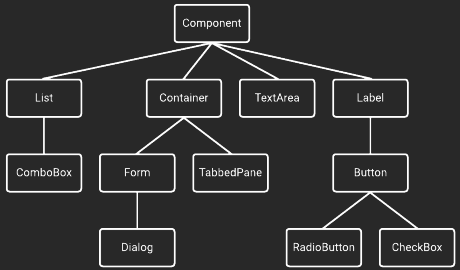
Two Questions

Idaho State University | Computer Science

What makes a **good** UI?



What UI toolkits have **YOU** used?



Outcomes

Idaho State University | Computer Science

After today's lecture you will be able to:

- Understand the basics of UI
- Understand how to use JavaFX to implement a UI

UI/UX Basics

CS 2263

Key Terms

Idaho State University | Computer Science

- **User Experience (UX)** - all aspects of a user's interaction with a software application
  - Actions
  - Responses
  - Perceptions
  - Feelings
- **User Interface (UI)** - Set of inputs and outputs that the user interacts with to invoke an application's functions
- **User-Centered Design** - Design technique that embodies the view that the UI appears to be the entire system
- **Usability** - Degree to which a system is easy to learn and use

User-Centered Design

Idaho State University | Computer Science

- **Three Principles**
  - Focus early on users and their work
  - Evaluate designs to ensure usability
  - Use iterative development

- **Goal:** high usability
  - System should be easy to learn and use
  - These two may be in conflict

UI Design Considerations

Idaho State University | Computer Science

- **Human Interface Objects** - Objects that users can manipulate or navigate with
  - These should reflect the function they perform
  - *Tooltips* should be provided to assist the user
  - Should provide visual/audio feedback when activated

- **Consistency**
  - Be consistent when using icons
  - Icons uses are familiar with should not be used for new things
  - Maintain consistency with the underlying platform
  - Maintain consistency within your application/suite of applications

UI Design Considerations

Idaho State University | Computer Science

- **Discoverability** - it should be easy for users to find and figure our operations and features in a UI
  - Make the UI “intuitive”
  - **Active Discovery** - designer includes clues for the user to avoid trial and error
    - *Tooltips*
    - Mouse pointer changes
    - Hover changes action colors
    - Popups to discover functions

- **Closure** - Let a user know they completed an operation
  - Visual or audible feedback
  - Undo should be provided so users can “back up” if needed

UI Design Considerations

Idaho State University | Computer Science

- **Readability**
  - Text must be readable by type, size, and color
  - The best systems allow users to change these properties
    - Try to prevent combinations which are difficult to read
- **Navigation**
  - Should be obvious and easily traversed
  - Navigation reversal should be provided
    - *Breadcrumbs* are a common technique which shows hierarchy of screens traversed
- **Usability and Efficiency**
  - Design for easy to use systems
  - Allow power users to have speed via shortcut keys for most functions
  - Provided clear error messages that both explain the error and how to fix it
  - Follow the KISS principle

UI and JavaFX

CS 2263

JavaFX Overview

Idaho State University | Computer Science

- **JavaFX**
  - Newer UI toolkit for Java to replace the aging Swing toolkit
  - Provides many features above and beyond Swing, and it looks better
  - Can be used on various OS's and devices including
    - Windows
    - Linux
    - Mac
    - iOS
    - Android
    - Raspberry Pi
  - Has a component library consistent with modern UI toolkits
    - This lecture is mainly about setting up JavaFX, rather than going into detail, I leave that to you

Using JavaFX with Gradle

- To use with gradle, you need to use the JavaFX plugin

```
plugins {
  id 'application'
  id 'org.openjfx.javafxplugin' version '0.0.10'
}
```

- Next, add the required modules

```
javafx {
    version = "17"
    modules = [ 'javafx.controls' ]
}
```

- The version should correspond to your JDK version

Setting the Stage

- In order to use JavaFX you need a class which extends

  `javafx.application.Application`

- Additionally, you will need to implement the method `start(Stage)`
  - Where the Stage is our window and is provided by JavaFX

- Finally, you should have a `main(String[])` method which calls

  `Application.launch()`

Example

```
import javafx.application.Application;
import javafx.stage.Stage;

public class MyApp extends Application {

  @Override
  public void start(Stage stage) throws Exception {
    stage.setTitle("First JavaFX Application");

    stage.show();
  }

  public static void main(String[] args) {
    Application.launch(args);
  }
}
```

A Simple Scene

- JavaFX uses an idea called a scene graph to create a hierarchy of the components that will be displayed
- These can be UI components, 2D/3D graphics, etc.
- We then construct a scene and add this to the stage

```
public void start(Stage stage) throws Exception {
  stage.setTitle("First JavaFX Application");

  Label label = new Label("Hello World, JavaFX!");
  Scene scene = new Scene(label, 400, 200);
  stage.setScene(scene);

  stage.show();
}
```

Event Handling

Idaho State University | Computer Science

- To handle simple events (such as button clicks or menu selections)
  - we add an appropriate event handler
- There two primary ways to handle events
  - Using an anonymous inner class

```
button.setOnAction(new EventHandler() {
  public void handle(ActionEvent actionEvent) {
    // do something
  }
})
```

  - Using a Lambda Expression

```
button.setOnAction(actionEvent -> {
  // do something
})
```

Event Handling Example

Idaho State University | Computer Science

```
import javafx.application.Applicaton;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scenelayout.VBox;
import javafx.stage.Stage;

import java.io.FileInputStream;
import java.io.FileNotFoundException;

public class ButtonActions extends Application {

  public void start(Stage stage) throws Exception {
    stage.setTitle("Button Experiment");

    Label label = new Label("Not Clicked!");
    Button button = new Button("Click");

    button.setOnAction(value -> {
      label.setText("Clicked!");
    });

    VBox vbox = new VBox(button, label);

    Scene scene = new Scene(vbox, 200, 100);
    stage.setScene(scene);
    stage.show();
  }

  public static void main(String[] args) {
    Application.lauch(args);
  }
}
```

Other Controls

Idaho State University | Computer Science

- Controls are the components that provide some kind of control functionality within the application.
- To be visible they must be attached to the scene graph of some `Scene` object
- Controls tend to be nested within layout components which manage control layout relative to one another

Available Controls:

- Accordion
- Button
- CheckBox
- ChoiceBox
- ColorPicker
- ComboBox
- Label
- ListView
- Menu
- MenuBar
- PasswordField
- ProgressBar
- RadioButton
- Slider
- Spinner
- SplitMenuButton
- SplitPane
- TableView
- TabPane
- TextArea
- TextField
- TitledPane
- ToggleButton
- ToolBar
- TreeTableView
- TreeView

Layouts

Idaho State University | Computer Science

- These are components which contain other components.
- They manager the layout of the contained components
- Must be attached to a scene graph of a `Scene` object to be visible

Available Layouts:

- Group
- Region
- Pane
- HBox
- VBox
- FlowPane
- BorderPane
- StackPane
- TilePane
- GridPane
- AnchorPane
- TextFlow

- To achieve a specific layout it is often necessary to nest layouts
  - For example to get horizontal rows which are not laid out in a grid but different for each row, you can nest multiple HBox layouts inside a VBox.

Advanced Capabilities

Idaho State University | Computer Science

- As you learn more about JavaFX, you will encounter several concepts, including
  - Properties and property bindings
  - FXML - an xml data-binding capability
    - This uses the MVC pattern (we will discuss much later)
    - Binds XML and CSS to classes thus allowing the design to be separate from the code
  - There are tools for creating FXML, including plugins for all IDEs

For Next Time

- Review JavaFX Tutorial
- Review this Lecture
- Come to class
- Read the Course Project Overview

Are there any questions?