

# IS 543 Fall 2023

Mobile Platform Development

Project 2 Q&A

Modal Views

Midterm Answers

Dr. Stephen W. Liddle

782 TNRB, office hours Tu 9:30–10:30am, Th 2–3pm

[is543@byu.edu](mailto:is543@byu.edu)

801-422-8792 (rings on my cell)

Zoom: <https://bit.ly/liddlezoom>





# Today

- How was the forum with Dr. Wilcox?
- Project 2 Q&A
- Modal Views
  - Alerts
  - Action sheets (a.k.a. confirmation dialogs)
  - Popovers
  - Sheets
- Review midterm questions



# Project 2 Q&A

- Any questions you'd like to discuss together as a class?  
(If we end early, I'll stick around and answer individual questions too)



# Modal Views

- From the HIG:

“Modality is a design technique that presents content in a separate, dedicated mode that prevents interaction with the parent view and requires an explicit action to dismiss.”

- Reasons to present content modally:

Ensure that the user receives critical information and acts on it if needed

E.g. battery at 20%, go to low power mode?

Provide options that let users confirm or modify their most recent action

E.g. user shakes the phone and iOS asks if they want to undo

Help users perform a distinct, narrowly scoped task without losing track of their previous context

E.g. user taps share button while looking at a photo

Give users an immersive experience or help them concentrate on a complex task

E.g. user wants to edit a recipe



# Modal Views

## • Alerts

The most fundamental modal view

Create using the `.alert()` view modifier

Alert is displayed when the bound Boolean variable is true

Here's a simple example (see the system documentation for more options)

```
struct ContentView: View {  
    @State private var timeIsUp = false  
  
    var body: some View {  
        Button("Indicate Time Up") {  
            timeIsUp = true  
        }  
        .alert("Time is up", isPresented: $timeIsUp) {  
            Button("OK", role: .cancel) { }  
        }  
    }  
}
```

Role can be `.cancel`  
or `.destructive`



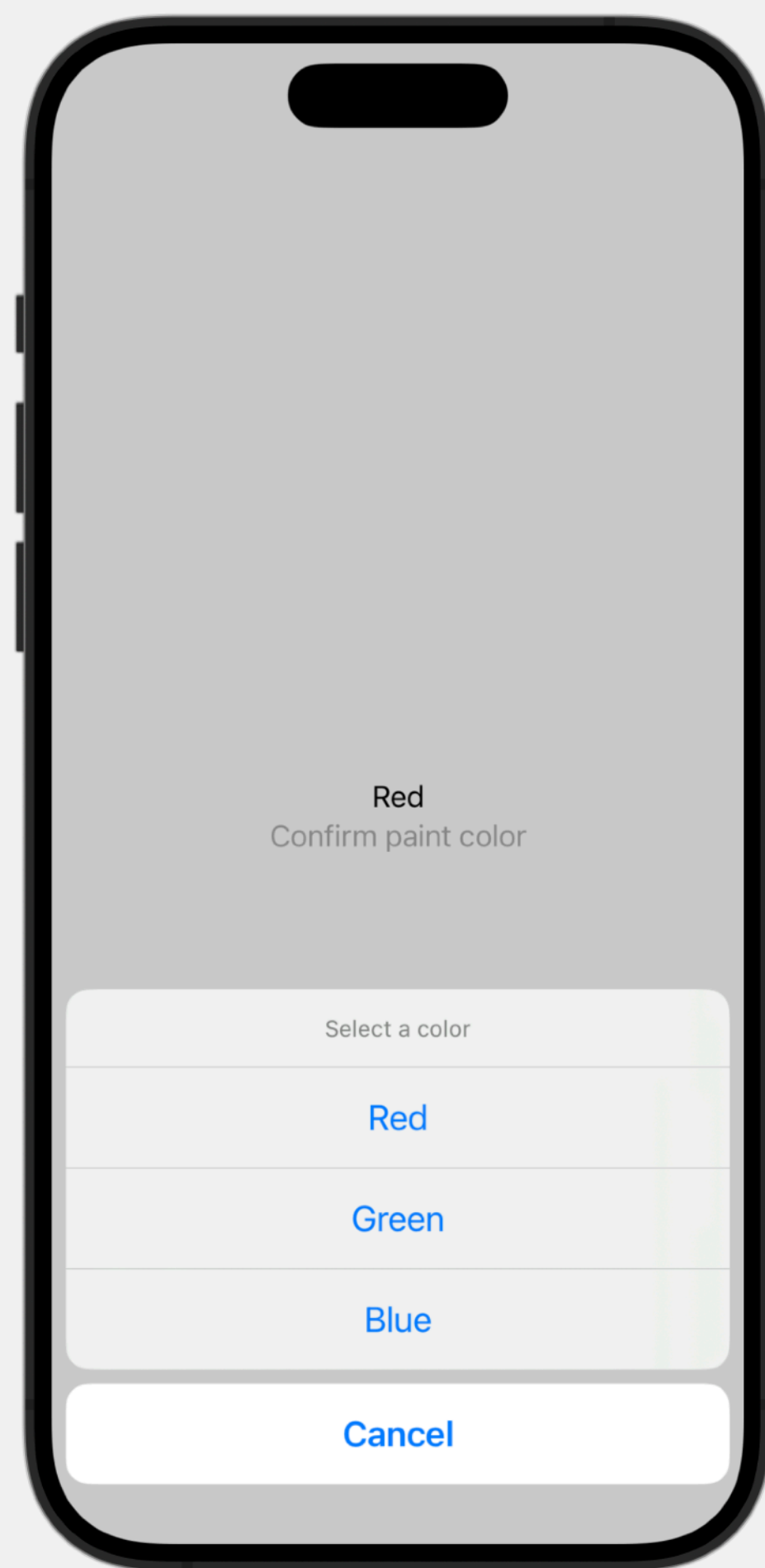
# Modal Views

## • Action sheets (“confirmation dialogs” in iOS 15)

```
struct ConfirmationDialogView: View {  
    @State private var showDialog = false  
    @State private var selection = "None"  
  
    ...  
    .confirmationDialog("Select a color",  
                        isPresented: $showDialog,  
                        titleVisibility: .visible) {  
        Button("Red") {  
            selection = "Red"  
        }  
        Button("Green") {  
            selection = "Green"  
        }  
        Button("Blue") {  
            selection = "Blue"  
        }  
    }  
    ...  
}
```

Visibility is `.automatic`,  
`.visible`, or `.hidden`

As before, buttons  
could specify roles





# Modal Views

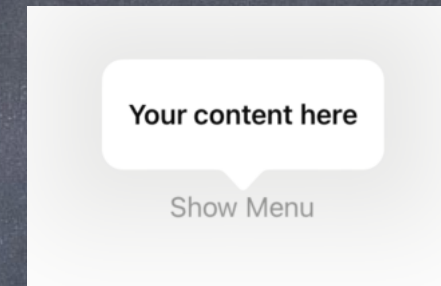
## 👁 Popovers

Renders differently on iPad vs. iOS

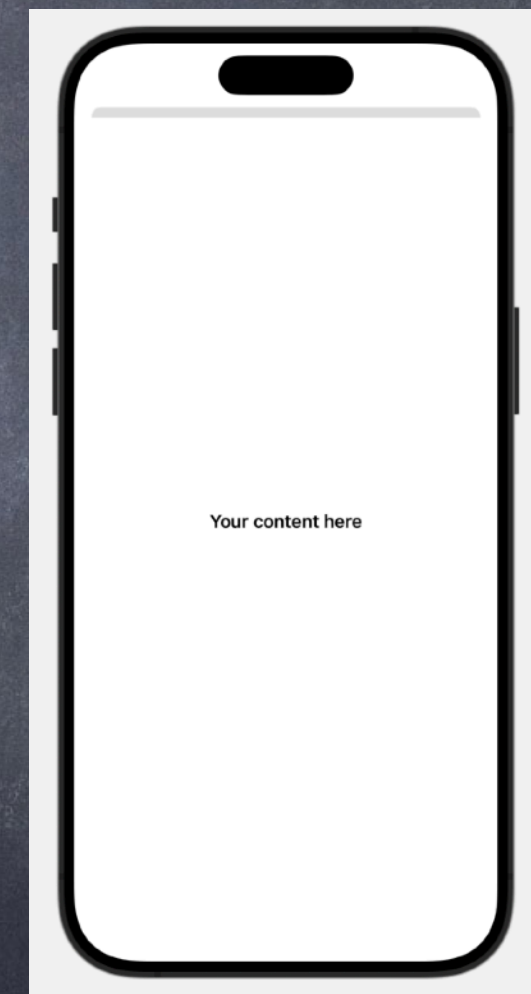
On iPad, it points to the source and it sized to the content

On iPhone, it takes the full width and displays a modal sheet  
(So I would generally use these only for iPad interfaces)

```
struct PopoverView: View {  
    @State private var showingPopover = false  
  
    var body: some View {  
        Button("Show Menu") {  
            showingPopover = true  
        }  
        .popover(isPresented: $showingPopover) {  
            Text("Your content here")  
                .font(.headline)  
                .padding()  
        }  
    }  
}
```



iPad



iPhone



# Modal Views

## 👁 Sheets

Sometimes you want your modal view to use most of the screen

On iPhone a Sheet uses the full width of the display

On iPad, it's only a portion of the width, but centered

Dismiss using a closure that comes from the environment

```
struct SheetDemoView: View {
    @State private var showingSheet = false

    var body: some View {
        Button("Show Sheet") {
            showingSheet.toggle()
        }
        .sheet(isPresented: $showingSheet) {
            SheetView()
        }
    }
}
```

```
struct SheetView: View {
    @Environment(\.dismiss) var dismiss

    var body: some View {
        Button("Press to dismiss") {
            dismiss()
        }
        .padding()
    }
}
```



# Modal Views

- Read more:  
<https://developer.apple.com/documentation/swiftui/modal-presentations>
- How might we use modal views in Project 2?  
To add or edit a recipe  
(Display the edit view in a modal sheet)



# Midterm Questions

- Let's review the midterm questions

We'll leaf through the key and you can ask about questions you want to discuss



# HW 9

- HW 9 is due this coming Tuesday

Get a rough prototype of your app's UI put together in SwiftUI. You won't have everything there, but show that you have navigation to most of the main UI features and that your app runs on iPad and iPhone devices. Upload two files: (1) a zip file of your project and (2) a PDF that shows screenshots of the main elements of your UI in both iPhone and iPad configurations.



# Coming Up...

- I'll be back in person on Tuesday
- More Project 2 Q&A, plus miscellaneous topics as we have time  
This is much we haven't talked about yet  
Let me know if there are topics you're especially interested in