Mobile Applications
CSCI 448
Lecture 14

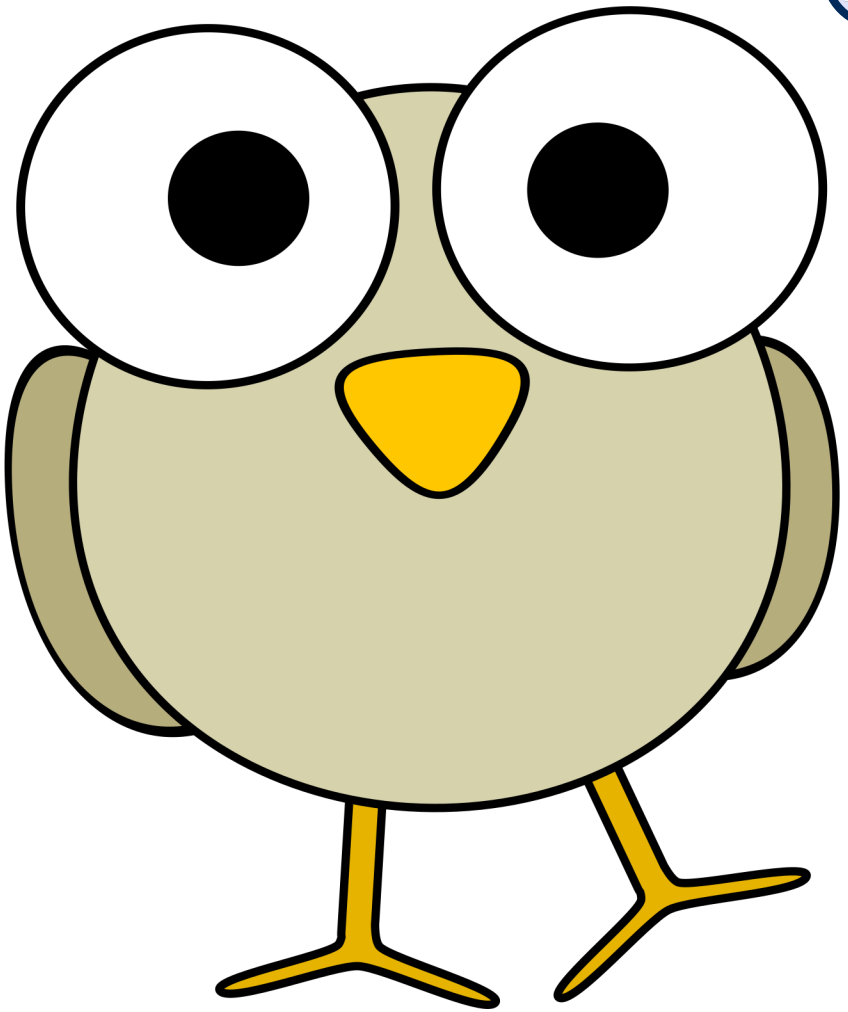
Jetpack Compose

Navigation

Previously in CSCI 448

- Screen Size
  - Components are sized once when drawing
  - Size/pad using **`dp`** (fonts use **`sp`**)
  - Apply weights for relative sizing/placements
    - Weights distribute unused spaced amongst components
  - Use Box composable to aid with alignment of inner components
- Landscape Layouts
  - UI Logic based on current configuration

Questions?



Learning Outcomes For Today

- Discuss how to implement Jetpack Navigation in the Compose framework
- Describe how NavGraphs implement the Builder Design Pattern

On Tap For Today

- Navigation Components

On Tap For Today

- Navigation Components

Jetpack Navigation

- Provides consistent UX by following established principles
- Simplifies moving through app via
  - Button Clicks
  - Menu Bars
  - Tabs
  - Navigation Drawers

Jetpack Nav: Step 0

- Include the dependency

```
// app/build.gradle

dependencies {
  implementation "androidx.navigation:navigation-runtime-ktx:2.5.3"
  implementation "androidx.navigation:navigation-compose:2.6.0-alpha05"

}
```

Navigation Components

- NavController
- NavHost
- NavGraph

(1) NavController

- Object that manages navigation through NavHost
- In Compose Framework, accessed via remember call
- Events trigger NavController actions to navigate through app

```
// inside composable
val navController = rememberNavController()
// inside an event handler
navController.navigate( ... )
```

(2) NavHost

- Composable container for navigation via NavController
- Handles controller state
- Manages back stack

(2) NavHost Cont.

- In Compose Framework, accessed via a Composable

```
NavHost(
  navController = navController,
  startDestination = "startingRoute"
) {
  // specify NavGraph
}
```

(3) NavGraph

- Collection of destinations
  - Specify start destination
- In Compose Framework, destinations are a Composable

```
NavHost(navController = navController, startDestination = "myNavGraph") {
    navigation (route = "myNavGraph", startDestination = "myRoute") {
        composable(route = "myRoute") {
            Text("This is my destination!")
        }
    }
}
```

(3) NavGraph Cont.

- In Compose Framework, destinations are a Composable

```
NavHost(navController = navController, startDestination = "myNavGraph") {
  navigation (route = "myNavGraph", startDestination = "myRoute") {
    composable(route = "myRoute") {
        Text("This is my destination!")
    }
    composable(route = "listScreen") {
        ListScreen(...)
    }
    composable(route = "detailScreen") {
        DetailScreen(...)
    }
  }
}
```

Design Pattern #6: Builder

- Separate the construction of a complex object from its representation so that the same construction process can create different representations
- Participants:
  - `Builder`: specifies an abstract interface for creating parts of a `Product` object
  - `Director`: constructs an object using the `Builder` interface
  - `ConcreteBuilder`: constructs & assembles part of the `Product` keeping track of the representation and providing an interface for retrieving the `Product`
  - `Product`: represents the complex object under construction, includes classes that define the constituent parts and interfaces for assembling the parts into the final result

Navigation Builder

- `Builder` →
- `Director` →
- `ConcreteBuilder` →
- `Product` →

Android Design Patterns

- Behavioral Patterns
  1. Command – UI Event Handling
  2. Observer – State
- Creational Patterns
  3. Builder – Compose NavGraph
  4. Factory – ViewModelFactory
  5. Singleton – ViewModelProvider, Repository
- Structural Patterns
  6. Decorator – View Model

Factory vs. Builder

- `Factory`
  - Builds the whole `Product` at once
  - Always builds the `Product` the same way and always makes similar instances of the `Product`
- `Builder`
  - Builds the `Product` piece by piece
  - Can build different versions of the same `Product`

Navigating

- Navigate to a destination

- In Compose Framework, performed via

```
navController.navigate("routeString")
```

(3) NavGraph Cont.

- In Compose Framework, destinations are a Composable

```
NavHost(navController = navController, startDestination = "myNavGraph") {
  navigation (route = "myNavGraph", startDestination = "myRoute") {
    composable(route = "myRoute") {
        Text("This is my destination!")
        Button( onClick = { navController.navigate("listScreen") } ) {
            Text("Go To List")
        }
    }
    composable(route = "listScreen") {
        ListScreen(onGoToDetail = { navController.navigate("detailScreen") })
    }
    composable(route = "detailScreen") {
        DetailScreen(...)
    }
  }
}
```

On Tap For Today

- Navigation Components

To Do For Next Time

- Continue on Lab03 - due Thu Feb 16
- Lab04 due Thu Feb 23
- Lab05 & A2 will be posted ASAP
- Alpha Release due Mon Mar 13
  - Have screens and navigation in place