# Using Options Menus and Action Bar Actions



Jim Wilson
MOBILE SOLUTIONS DEVELOPER & ARCHITECT
@hedgehogjim blog.jwhh.com

# What to Expect from This Module



Adding options menu items

Handling menu item selections

Options as app bar actions

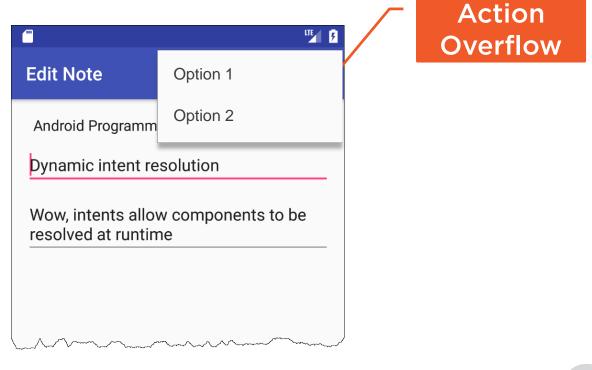
Runtime changes to menu items

Kotlin null safety

# Options Menus

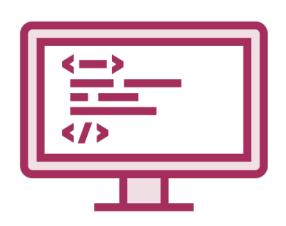
#### Provide actions for an Activity

- Actions available in app bar
- Appear under action overflow by default





### Creating Options Menus





Attach options to root menu

Normally created with Android Studio

designer

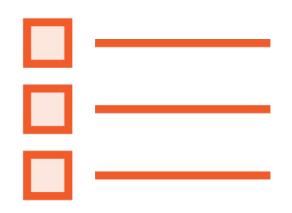


#### onCreateOptionsMenu

Attaches options to Activity
Receives a Menu reference
Inflate menu resource onto received
menu reference



# Handling Options Menu Item Selections





#### Each option defined as menu item

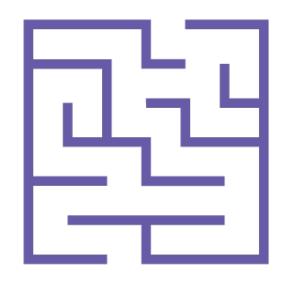
Has display text
Has a unique ID within the menu

#### onOptionsItemSelected

Receives a MenuItem reference
Use ID value to determine selection
Perform appropriate work for selection



# Menu Items as App Bar Actions



Action overflow challenges
Limited discoverability
Multiple steps



App bar actions
Visible on app bar
Normally have icon
Easily discoverable
Easy access



showAsAction property
Enables menu item to
be app bar action

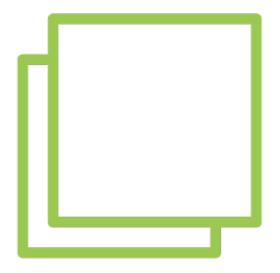


# showAsAction Property Values



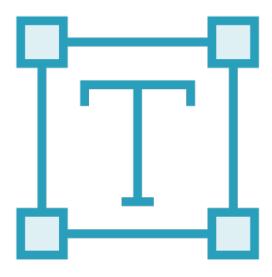
ifRoom

When space allows
Preference in
top-to-bottom order



always

Always means always
May cause crowding
Use sparingly



#### withText

Include text when space allows

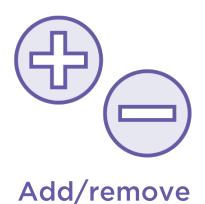
Can combine with ifRoom/Always



# Changing Menu Items at Runtime



### Application or Activity state can change menu items



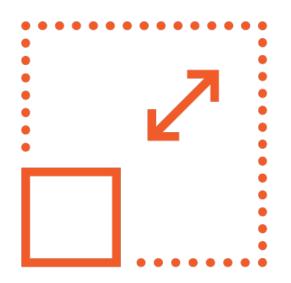
menu items







### Changing Menu Items at Runtime



#### onPrepareOptionsMenu

Override to modify menu
Receives reference to current menu
Initially called before menu displayed

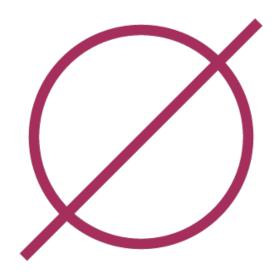


#### invalidateOptionsMenu

Call when menu needs to change Schedules call to onPrepareOptionsMenu



# Null Handling



Null references

Refence does not refer to a valid object



Source of much pain

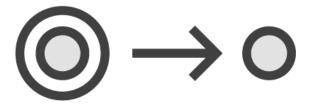
Common cause of app crashes

Must be diligent in checking for null



### Kotlin Null Safety





Nullability must be explicit

No implicit nullable/non-nullable types

Must indicate nullability

Accessing nullable types
Must include null handling



```
fun showName(p: Person) {
  textName.text = p.name
fun showName(p: Person?) {
  if(p != null)
    textName.text = p.name
```

- ◆ p cannot be null
- ◆ p is safe to use

- ◆ p can be null
- Must check that p is not null
- **◄** Compile-time error

# Null Safety Operators





#### Safe call operator

Safely access member of referenced type
Returns member if reference non-null
Otherwise return null

#### Elvis operator

Return a value based on null test
Returns value if not null
Returns alternate value if null



```
fun showName(p: Person?) {
 var n: String? = null
  if(p != null)
    n = p.name
  textName.text =
    if(n != null) n else "XX"
```

■ Variable to hold name

■ Return name if p not null

■ Return n if n not null

```
fun showName(p: Person?) {
 val n = p?.name
  textName.text =
    if(n != null) n else "XX"
```

■ Return name if p not null

**◄** Return n if n not null

```
fun showName(p: Person?) {
 val n = p?.name
  textName.text =
    n ?: "XX"
```

■ Return n if n not null

```
fun showName(p: Person?) {
  textName.text =
    p?.name ?: "XX"
```

■ Return name if p & name non-null



### Creating an options menu

- Normally defined as menu resource
- Composed of menu items

#### Associating menu with activity

- Override on Create Options Menu
- Inflate menu resource





#### Handling menu item selection

- Override on Options Item Selected
- Receives MenuItem reference
- Take action based on itemId property

#### App bar actions

- Menu items visible on app bar
- Use showAsAction Property
- Normally have icon associated





#### Setting menu state

- Override on Prepare Options Menu
- Receives Menu reference
- Can access individual menu items

#### Triggering change to menu state

- Call invalidateOptionsMenu





#### **Null safety**

- Must explicitly indicate nullability
- Kotlin enforces safe null handling
- Provides operators to simplify task

