Mobile Application Programming: Android
Model View Controller
Activities

- Apps are composed of activities
- Activities are self-contained tasks made up of one screen-full of information
- Activities start one another and are destroyed commonly
- Apps can use activities belonging to another app
Application
Model View Controller (MVC) Application
The View

- Built how we’ve been building them since the beginning of class
- Concerned mainly with presentation of data and receiving input from the user
- Data is generally passed to views from the controller through accessor methods to simple data types
  - Booleans, Integers, Floating Point Numbers, Strings
- User input events are sent from views to the controller through listeners provided to the view by the controller
Why Separate Views from Data?
The Model

- Source of data for the application
- Contains all or nearly all of the program’s state
- Should be thought of as persistent, that is, it stays the same over separate runs of the program
- Could be specifics of a running simulation, network and caching code for interfacing with a server, interface into a database running on the phone itself, or a simple object that saves the state of the app to file
Why Separate Models?
The Controller

- Acts as an **adapter** between model and view
- **Keeps the model in sync** with changes in the view (user interaction), and **the view in sync** with changes to the model (network access, etc.)
- Is the **least portable** part of the program
- Typically implemented using the **Activity class**
Why Link with Controllers?
Why Link with Controllers?
Why Link with Controllers?

“Model-View-Adapter”
Why Link with Controllers?
Notification

How do these happen?
Notification

How do these happen? Listeners!
Model View Adapter (MVA)
Model View Adapter (MVA)
Model View Adapter (MVA)

User action

Send Message

onSendMessageListener
onSendMessage()
Model View Adapter (MVA)

User action

Update

Notify

Send Message

onSendMessageListener
onSendMessage()
Model View Adapter (MVA)

User action

Send Message

SendMessageListener

sendMessage()

Update

Notify

MessagesChangedListener

MessagesChanged()
Model View Adapter (MVA)

User action

Send Message

onSendMessageListener
onSendMessage()

Update

Send Message

addMessage()

Notify

onMessagesChangedListener
onMessagesChanged()

invalidateViews()
Model View Adapter (MVA)

User action
- ListAdapter getView()
- getMessage()

Update
- addMessage()
- invalidateViews()
- onMessagesChanged()
- onSendMessage()
- onSendMessageListner

Send Message
- Send Message

Notify
- onMessagesChangedListner
Launching Activities

- Create `Intent` instance
- Set intent `class` property
- Fill in intent `extras` using `putExtra(key, value)`
  - Get keys from activity-to-be-launched class
- Call `Activity.startActivity(Intent)`
- Can open activities from other apps similarly as well
Launching Activities
Launching Activities

Intent

startActivity()

Extras
Launching Activities

`startActivityForResult()`

`onActivityResult()`

`setResult()`
Sharing Data

startActivity()
Sharing Data

startActivity()