IT – Entrepreneurship
Activity Lifecycle

Activity Lifecycles

- Application comprises of one or more activities.
- An activity is governed by a set of states called the lifecycle.
- Unlike programming paradigms in which apps are launched with a main() method, the Android system initiates code in an Activity instance by invoking specific callback methods that correspond to specific stages of its lifecycle.

The concept of activities

- The Activity class serves as the entry point for an app’s interaction with the user, providing the window in which the app draws its UI.
- You implement an activity as a subclass of the Activity class.
- Most apps contain multiple screens, which means they comprise multiple activities.

The concept of activities

- Typically, one activity in an app is specified as the main activity, which is the first screen to appear when the user launches the app.
- Each activity can then start another activity in order to perform different actions.

Configuring the manifest

- To declare your activity, open your manifest file and add an <activity> element as a child of the <application> element.
- The manifest file provides essential information about your app to the Android system, which the system must have before it can run any of the app's code.

```
<manifest ... >
  <application ... >
      <activity android:name=".ExampleActivity" />
      ...
  </application ... >
  ...
</manifest >
```

Intents and Intent Filters

- Intent filters are declared in the manifest file.
- Intent filter describes a capability of the component(like activities)
- An Intent is a messaging object you can use to request an action from another app component.

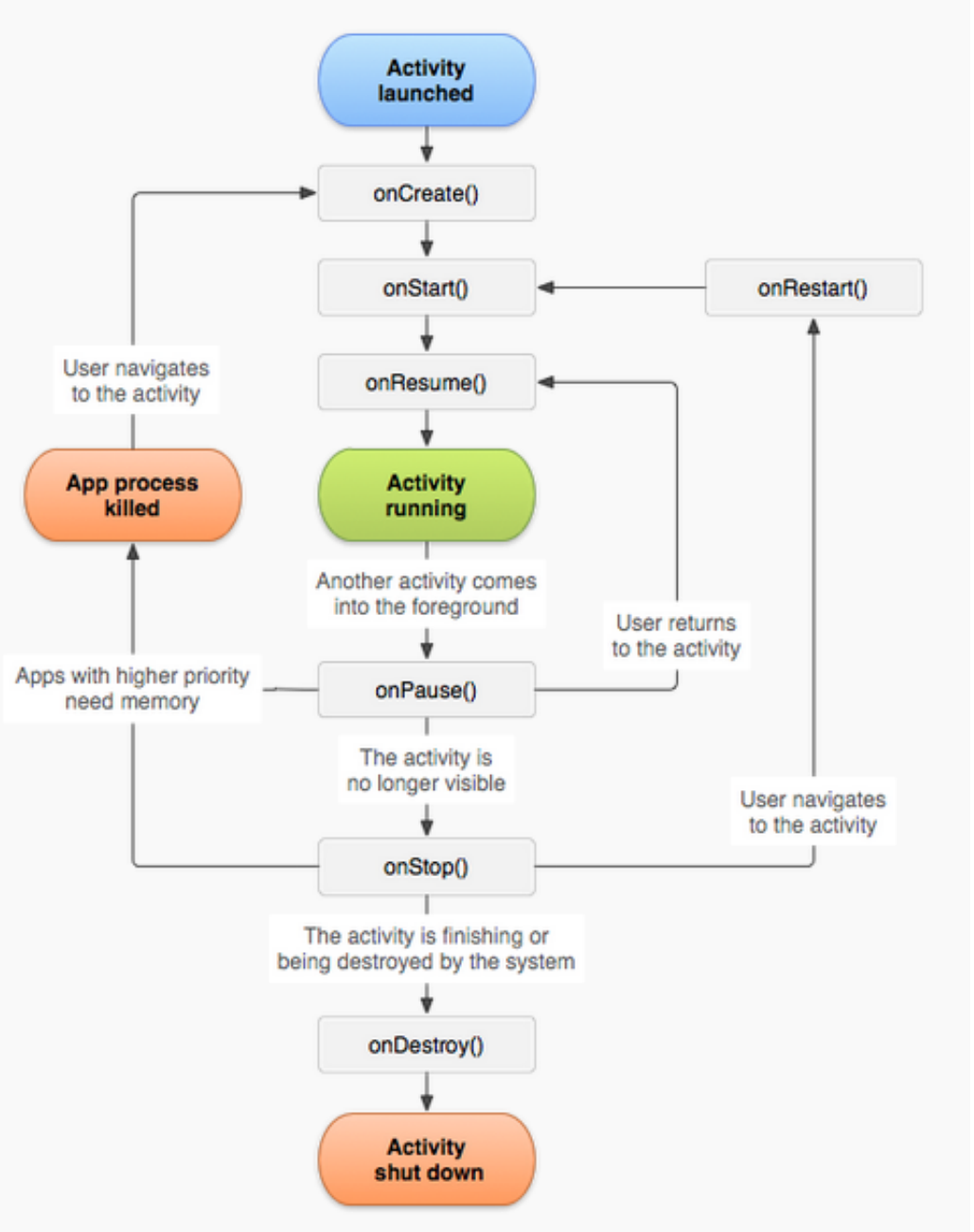
Intents and Intent Filters

- There are two types of intents; Explicit and Implicit.
- An explicit request might tell the system to “Start the Send Email activity in the Gmail app"
- An implicit request tells the system to “Start a Send Email screen in any activity that can do the job."
- When the system UI asks a user which app to use in performing a task, that’s an intent filter at work.

Managing the activity lifecycle

- onCreate(), fires when the system creates your activity.
- When onCreate() finishes, the next callback is always onStart().
- onStart(), the activity enters the Started state, and the activity becomes visible to the user.
- onResume(), system invokes this callback just before the activity starts interacting with the user.

Activity Lifecycle

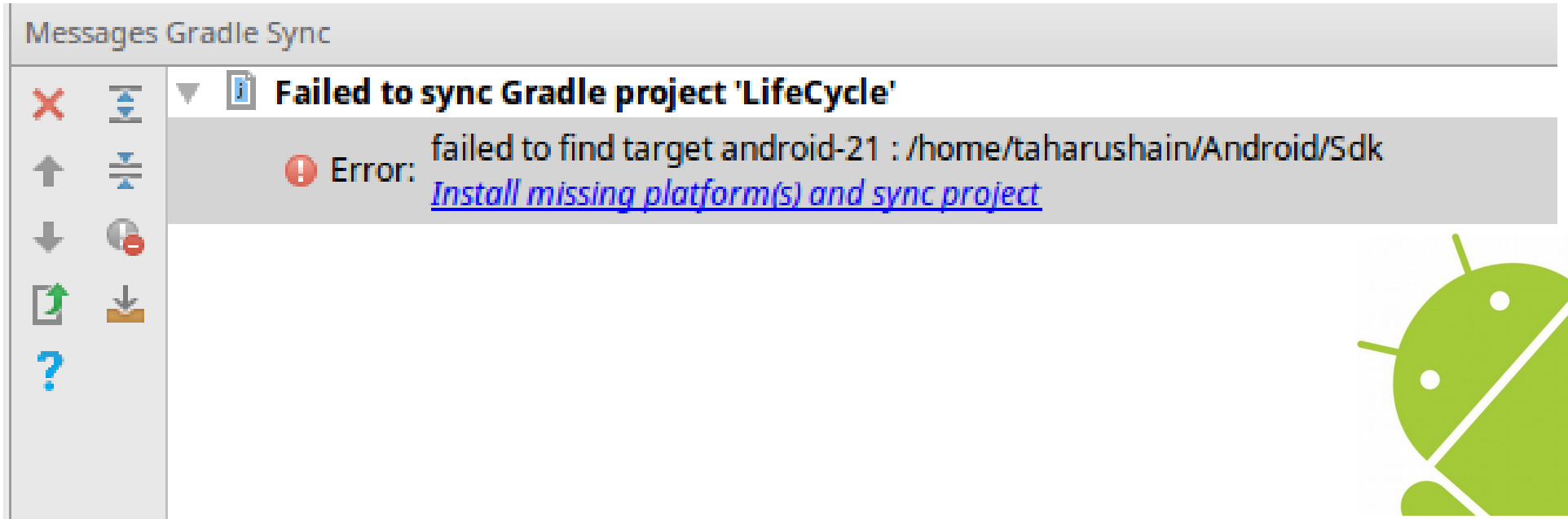


Managing the activity lifecycle

- The system calls onPause() when the activity loses focus and enters a Paused state.
- The system calls onStop() when the activity is no longer visible to the user.
- The system invokes this callback when an activity in the Stopped state is about to restart.
- The system invokes this callback before an activity is destroyed.

Importing our project

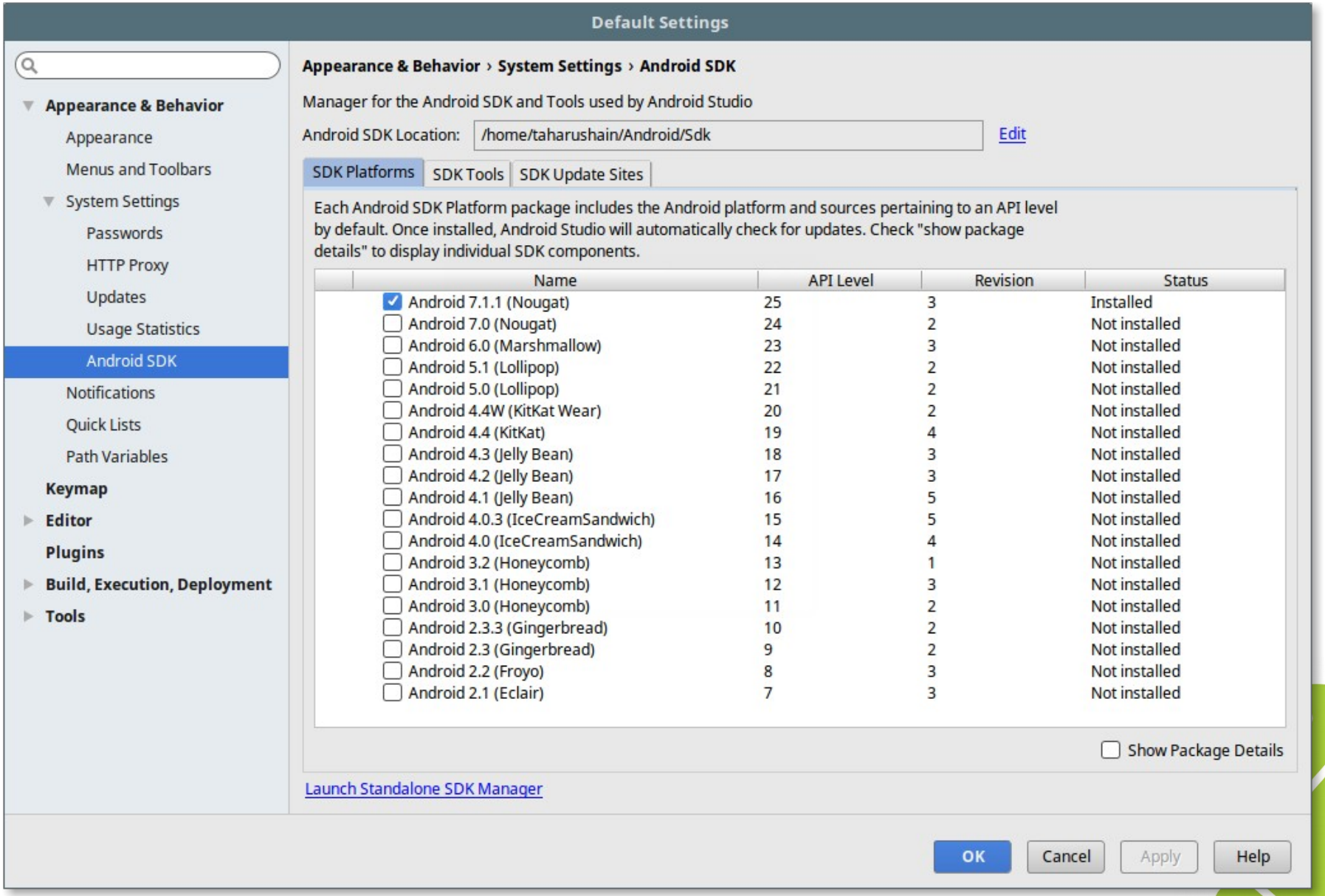
- The problem is that this project is created with some other ‘SDK’ version than the one we have installed.
- To solve this, we can either install the missing SDK by clicking the ‘Install missing ...’



Importing our project

- Or we can solve it by changing it to the one we have installed.
- For that, we first need to know what we have installed.
- Open the ‘SDK Manager’
- It is the second last icon, next to the Question mark.

Importing our project



Importing our project

- After identifying our SDK version and build-tools version.
- We open our app gradle file
- App → build.gradle
- At the starting, we have the sdk and buildtools used by our project. It should look something like this.

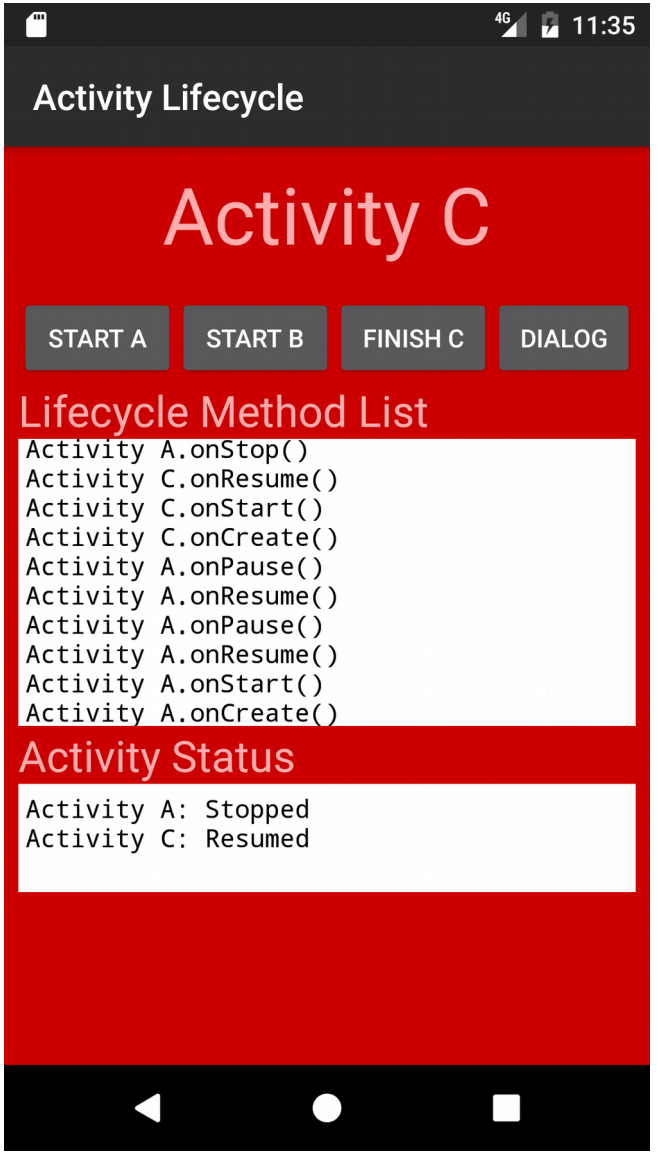
```
android {
    compileSdkVersion 21
    buildToolsVersion "21.1.2"
```

Importing our project

- After changing the sdk and buildTools version to appropriate values, press ‘Try Again’ or ‘Sync’.
- Now, the gradle should look something like this.

```
android {
    compileSdkVersion 25
    buildToolsVersion "25.0.2"
```

Identify the activity transitions



References

- Android Developers
- cs.dartmouth.edu