Android sweatshop II

7.7.16

Side note

- During the session we **didn't got to the TODO app**
- We did cover **Intents** and how to **start a new activity**

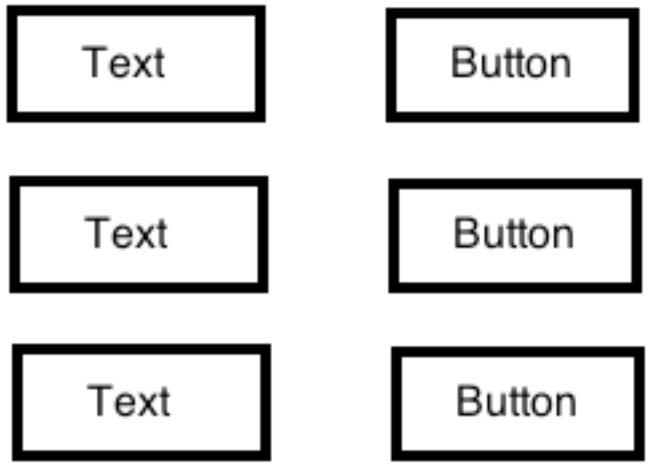
Last episode

- Activity lifecycle
- Logcat
- Layouts
  - MATCH_PARENT vs WRAP_CONTENT
- Views
  - OnClickListener

Home work

- A client asked for a grid like app, each TextView should hold a separate counter

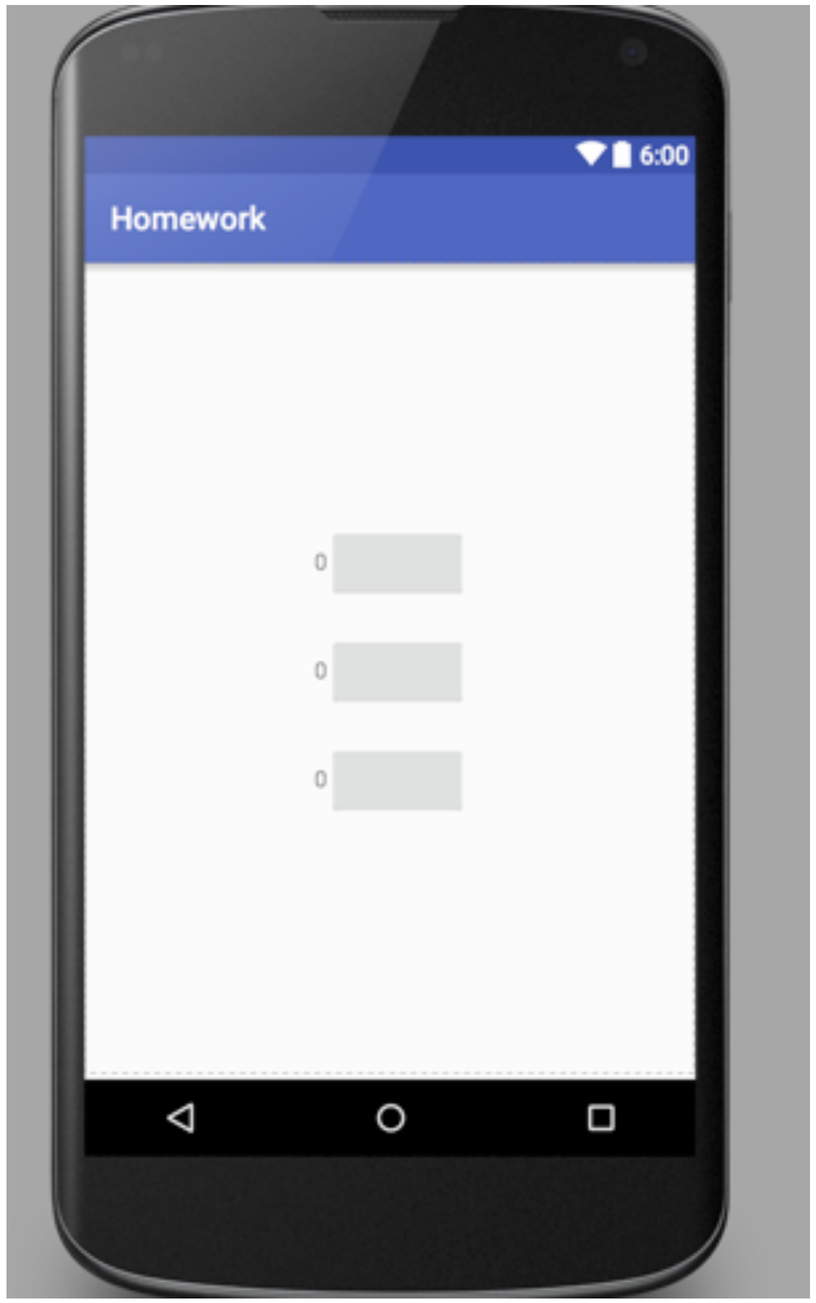
- Bonus - use three horizontal LinearLayout for each row, and a vertical LinearLayout for their parent



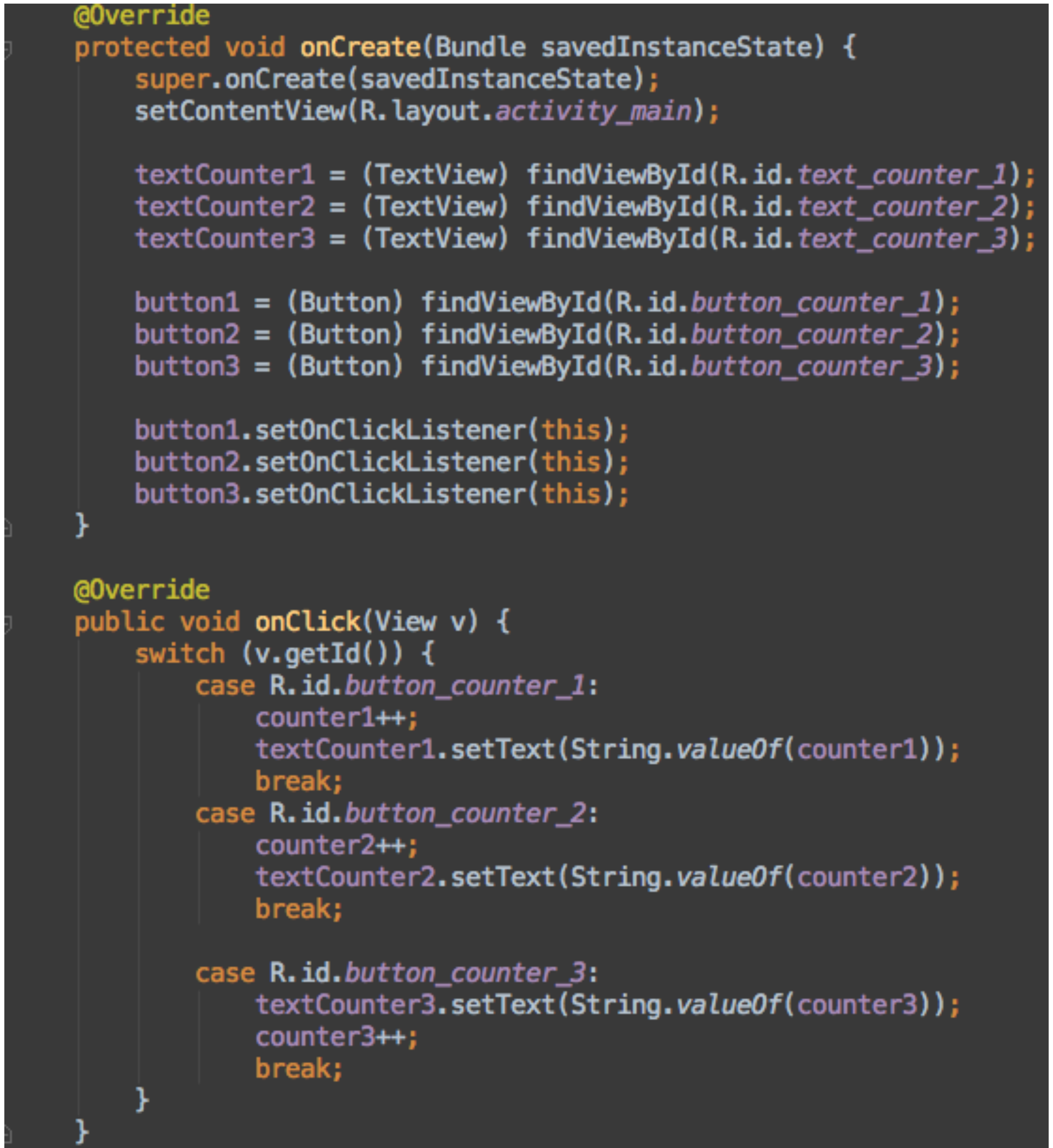
Home work

- Challenge - how to align them ?
- Hint: LinearLayout + LinearLayouts
- Any one wanna try ?

Home work UI



Homework solution



```java
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    textCounter1 = (TextView) findViewById(R.id.text_counter_1);
    textCounter2 = (TextView) findViewById(R.id.text_counter_2);
    textCounter3 = (TextView) findViewById(R.id.text_counter_3);

    button1 = (Button) findViewById(R.id.button_counter_1);
    button2 = (Button) findViewById(R.id.button_counter_2);
    button3 = (Button) findViewById(R.id.button_counter_3);

    button1.setOnClickListener(this);
    button2.setOnClickListener(this);
    button3.setOnClickListener(this);
}

@Override
public void onClick(View v) {
    switch (v.getId()) {
        case R.id.button_counter_1:
            counter1++;
            textCounter1.setText(String.valueOf(counter1));
            break;
        case R.id.button_counter_2:
            counter2++;
            textCounter2.setText(String.valueOf(counter2));
            break;

        case R.id.button_counter_3:
            textCounter3.setText(String.valueOf(counter3));
            counter3++;
            break;
    }
}
```

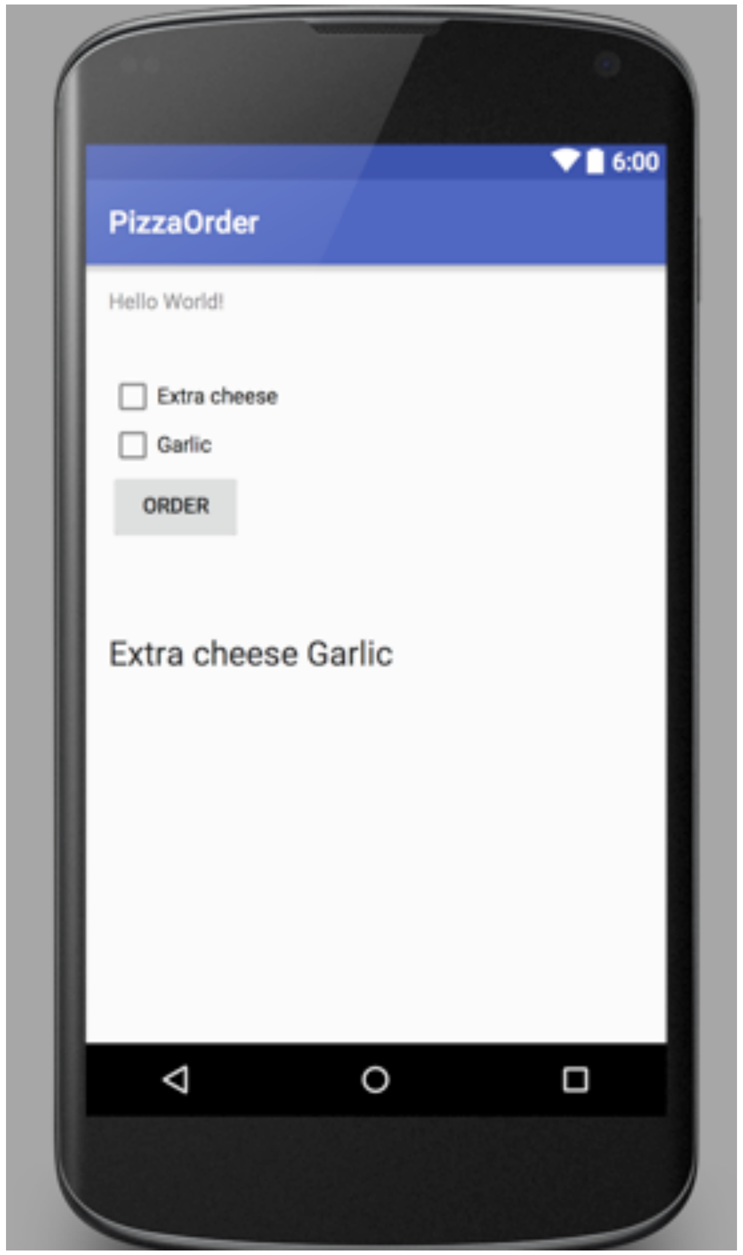
Home work activity_main.xml

- RelativeLayout (why?)
  - LinearLayout
    - LinearLayout
      - TextView
      - Button
    - LinearLayout
      - TextView
      - Button
    - LinearLayout
      - TextView
      - Button

Content table

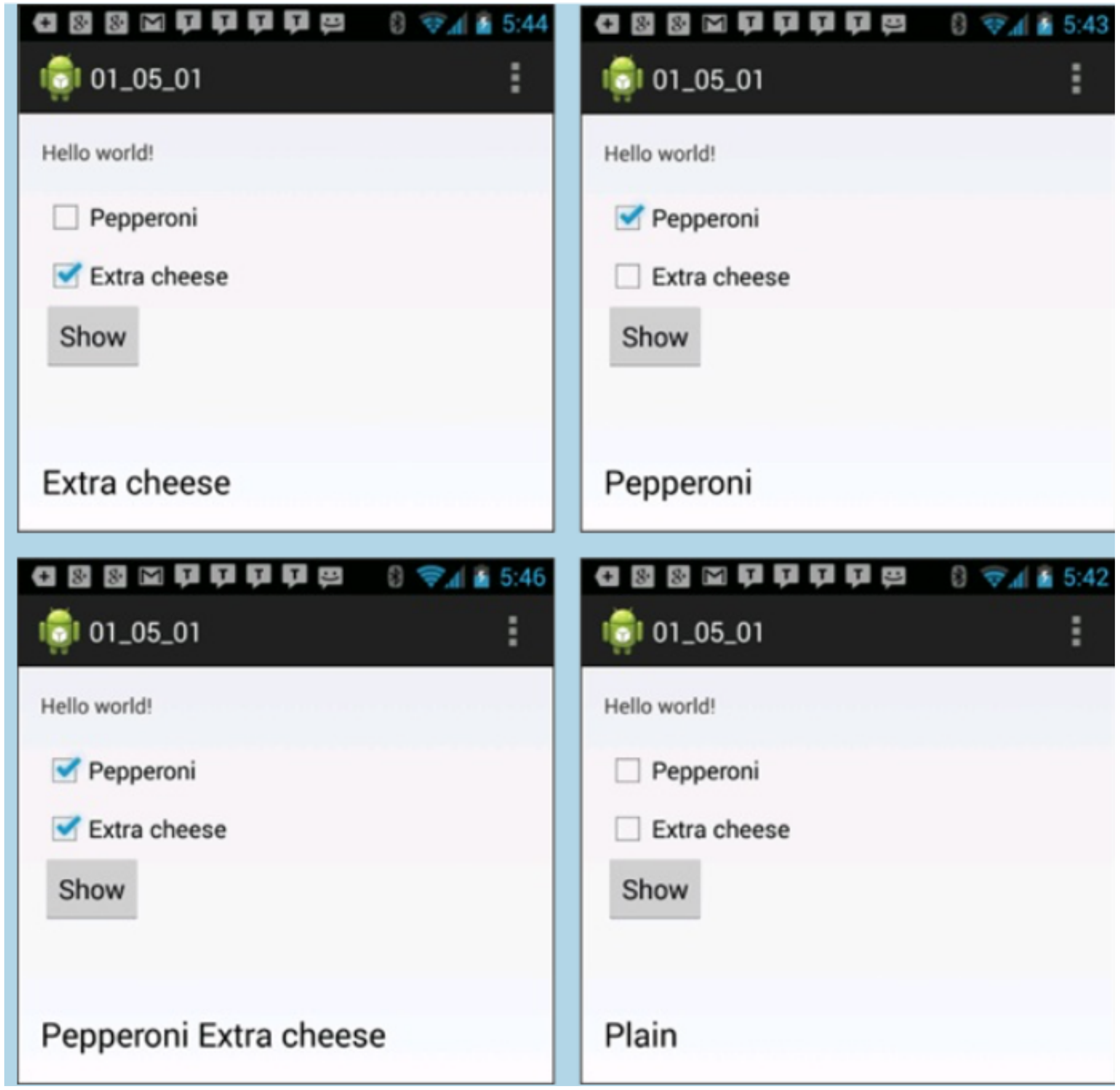
- Pizza app (from Android For Dummies)
  - More views and layouts
  - Bitamp
  - Passing data to another activity
    - Intent
- TODO app
  - ListView
    - Array adapter

PIZZA !



Extra cheese Garlic

More examples


5:44
01_05_01
Hello world!
☐ Pepperoni
✓ Extra cheese
Show
Extra cheese

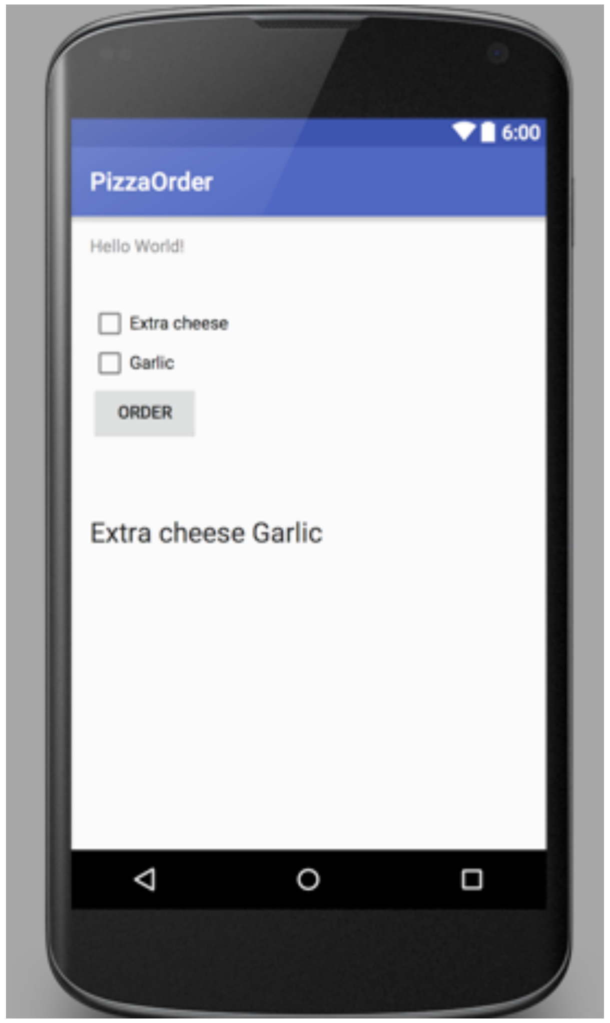
5:43
01_05_01
Hello world!
✓ Pepperoni
☐ Extra cheese
Show
Pepperoni

5:46
01_05_01
Hello world!
✓ Pepperoni
✓ Extra cheese
Show
Pepperoni Extra cheese

5:42
01_05_01
Hello world!
☐ Pepperoni
☐ Extra cheese
Show
Plain

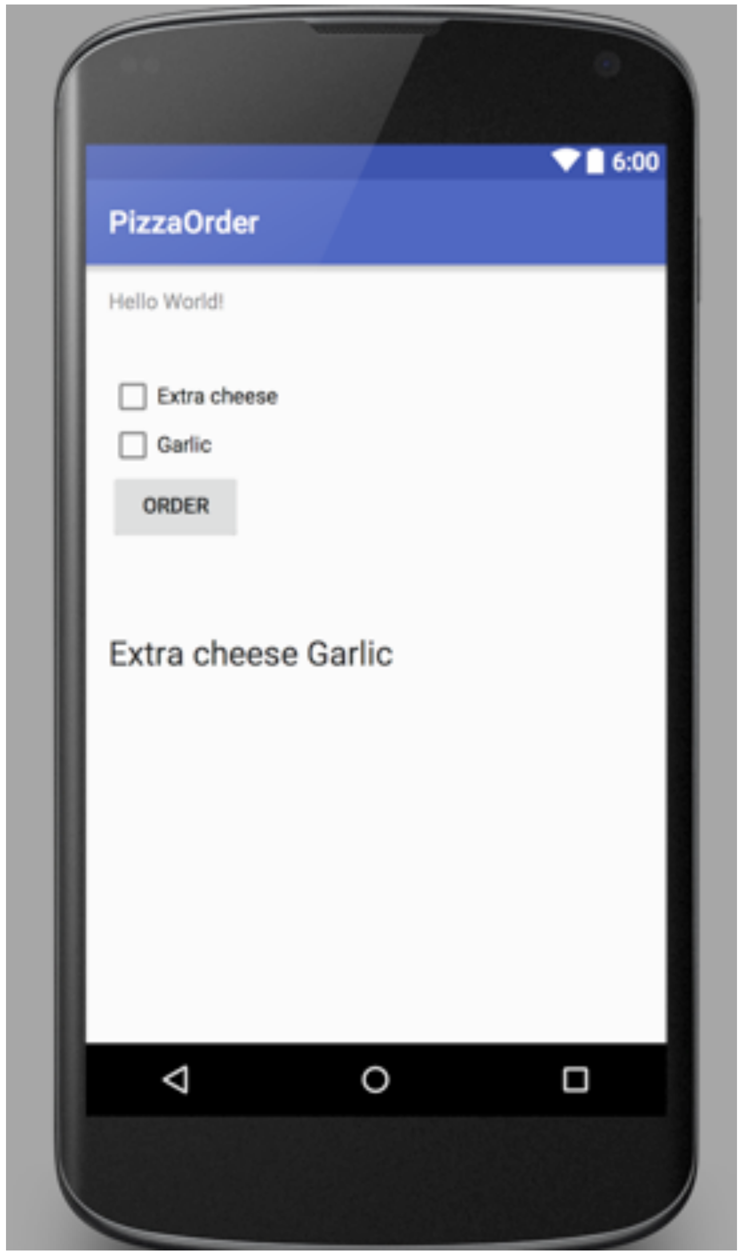
PIZZA - what does it do ?

- Press on Order
    - Check which checkboxes are press
    - Set the order string inside the textview



Pizza instructions - layout designer

- Drag checkboxes
- Drag buttons
- Drag a large TextView
- Remove the HelloWorld
- Change the id's to match



PIZZA - MainActivity

- add Class members - all of the views
    - private CheckBox …
- set them using findViewById(R.id.view_id)
    - garlicCheckBox = findByViewId…

MainActivity - onClickListener

- Set our Order button OnClickListener
  - Check checkboxes ;)
  - Update string

Checkbox != Chessbox

Adding a bitmap to an ImageView

- Search for a pizza photo on google
- Add it to the drawable folder
- Add an ImageView to the layout
- Load

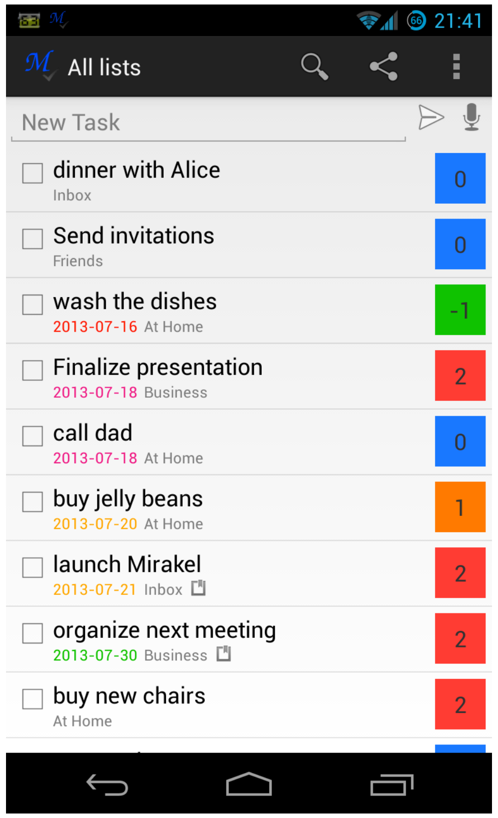
TODO app

Waba laba dab dab

TODO lists are useful

- We're gonna learn more about Android UI
- Activities and intents
- Later: fragments, action bars, preferences, sql

simple

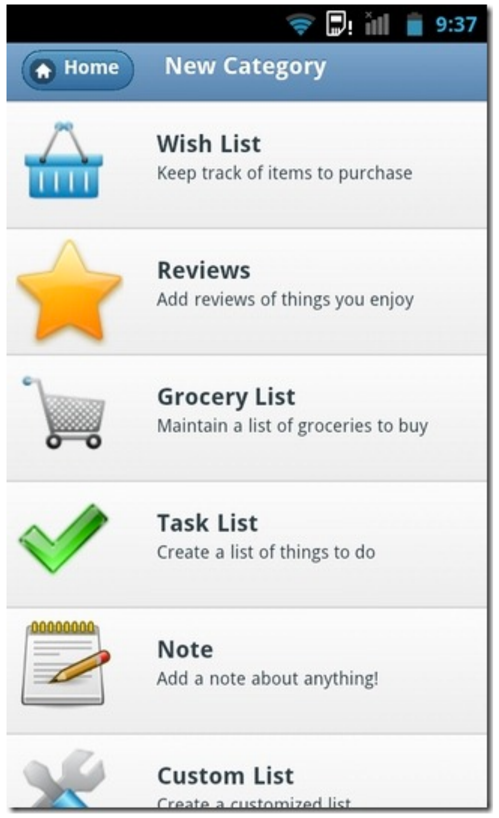


21:41

All lists

New Task

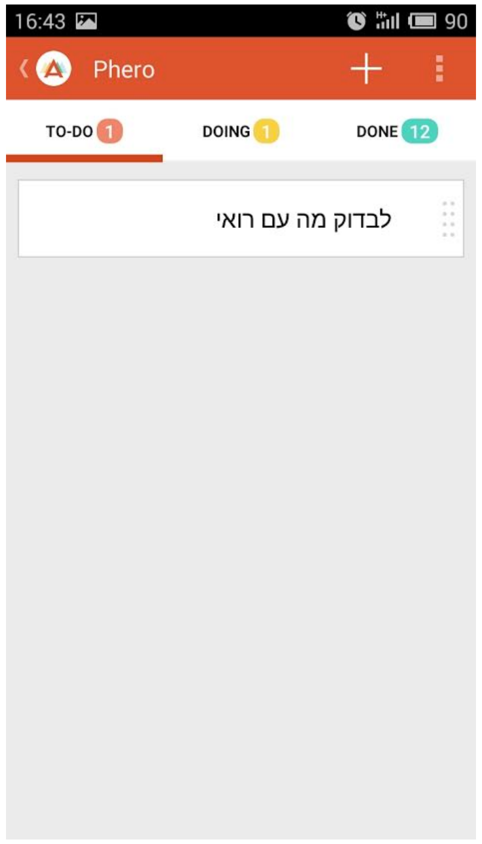
dinner with Alice	Inbox	0
Send invitations	Friends	0
wash the dishes	2013-07-16 At Home	-1
Finalize presentation	2013-07-18 Business	2
call dad	2013-07-18 At Home	0
buy jelly beans	2013-07-20 At Home	1
launch Mirakel	2013-07-21 Inbox	2
organize next meeting	2013-07-30 Business	2
buy new chairs	At Home	2



9:37

Home New Category

Wish List	Keep track of items to purchase
Reviews	Add reviews of things you enjoy
Grocery List	Maintain a list of groceries to buy
Task List	Create a list of things to do
Note	Add a note about anything!
Custom List	Create a customized list



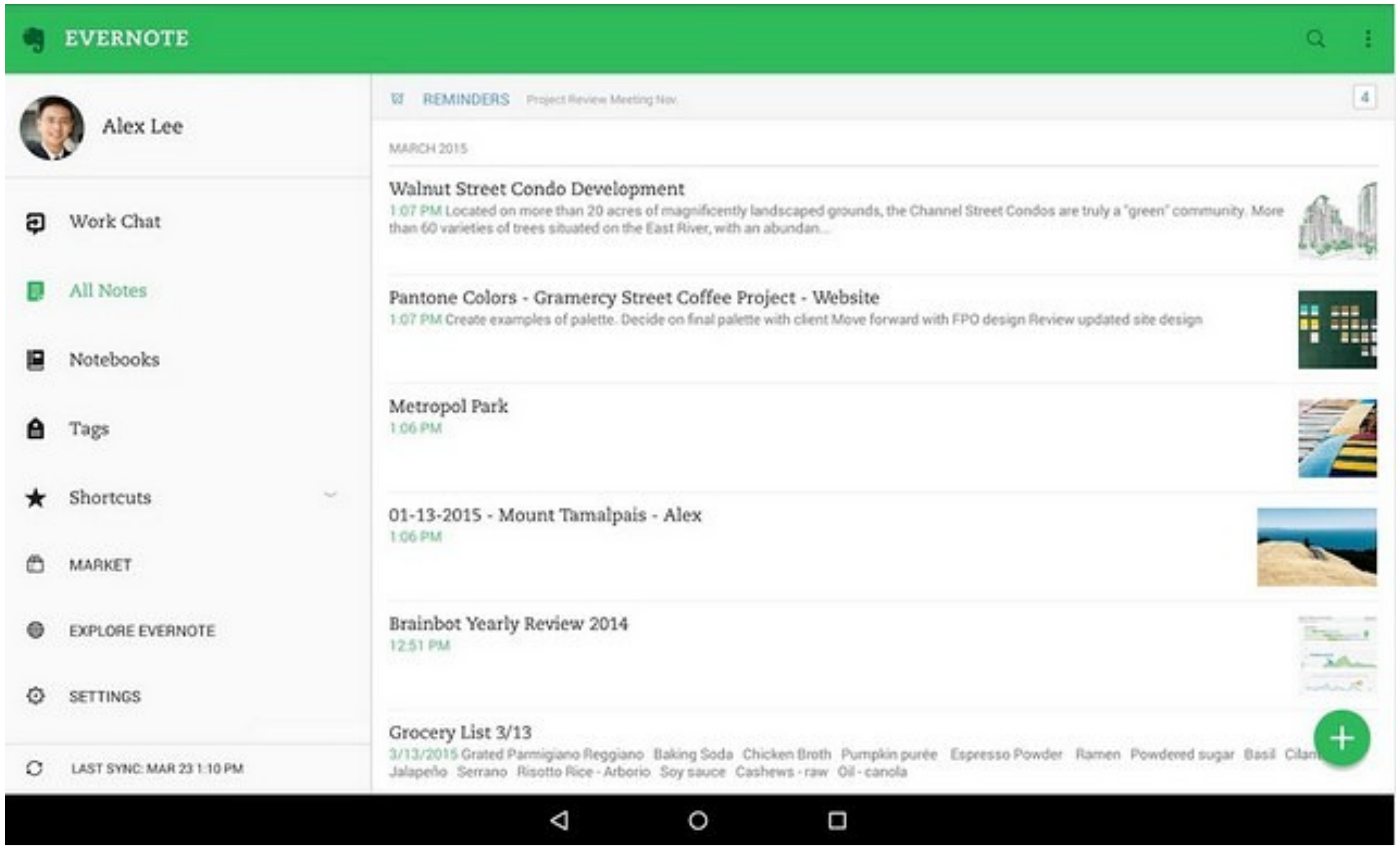
16:43 90

Phero

TO-DO 1 DOING 1 DONE 12

לבדוק מה עם רואי

Complex



EVERNOTE

Alex Lee

Work Chat

All Notes

Notebooks

Tags

Shortcuts

MARKET

EXPLORE EVERNOTE

SETTINGS

LAST SYNC: MAR 23 1:10 PM

REMINDERS Project Review Meeting Nov 4

MARCH 2015

Walnut Street Condo Development
1:07 PM Located on more than 20 acres of magnificently landscaped grounds, the Channel Street Condos are truly a "green" community. More than 60 varieties of trees situated on the East River, with an abundan...

Pantone Colors - Gramercy Street Coffee Project - Website
1:07 PM Create examples of palette. Decide on final palette with client Move forward with FPO design Review updated site design

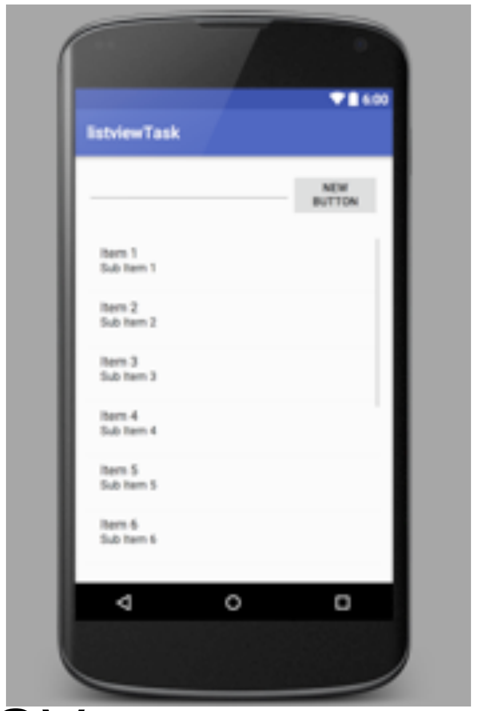
Metropol Park
1:06 PM

01-13-2015 - Mount Tamalpais - Alex
1:06 PM

Brainbot Yearly Review 2014
12:51 PM

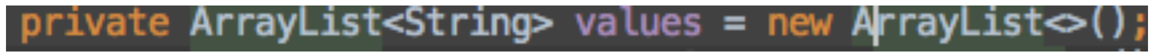
Grocery List 3/13
3/13/2015 Grated Parmigiano Reggiano Baking Soda Chicken Broth Pumpkin purée Espresso Powder Ramen Powdered sugar Basil Cilan
Jalapeño Serrano Risotto Rice - Arborio Soy sauce Cashews - raw Oil - canola

NoteListActivity layout design

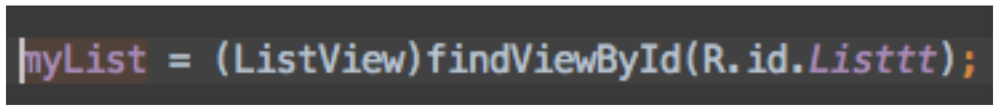


- ListView
- EditText field
- Add TODO item button
    - add the text from the text field to the array
    - notify the adapter list changed
- Later: onClick() of item starts EditActivity - now show a toast

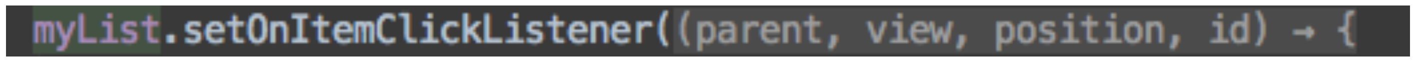
ListView - code parts



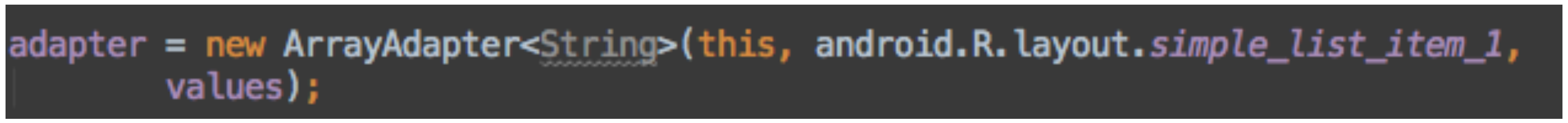
```java
private ArrayList<String> values = new ArrayList<>();
```



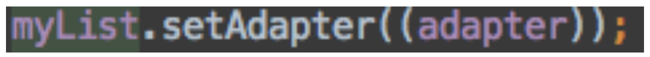
```java
myList = (ListView)findViewById(R.id.Listtt);
```



```java
myList.setOnItemClickListener((parent, view, position, id) → {
```



```java
adapter = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1,
        values);
```



```java
myList.setAdapter((adapter));
```

Adapter class as an Android design pattern

- The Adapter pattern allows two incompatible classes to work together by converting the interface of one class into an interface expected by the clients.

Coding time !

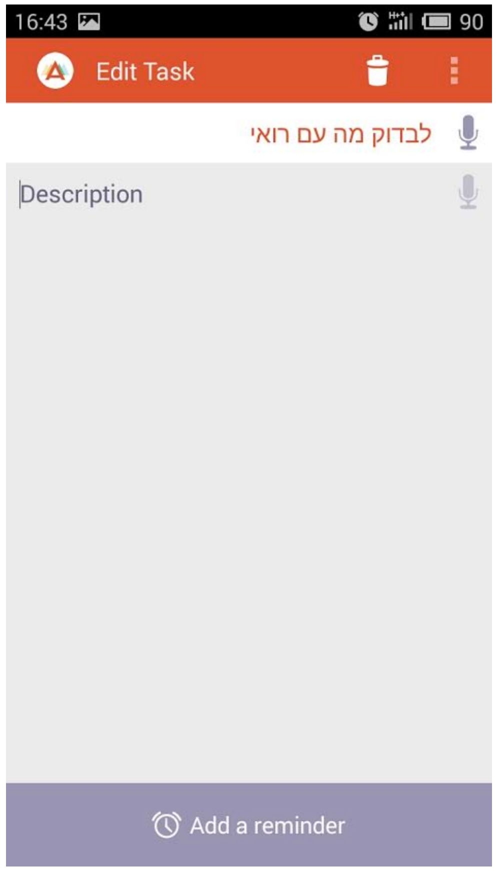
- Start with NoteListActivity

How do we add the content of the note ?

- We need another activity !

EditNoteActivity

- **Title** of note
- **Body** content
- Edit button



Starting another activity

- How do we move from screen to screen ?

Intent - special Android component

- The most common use of intents is to **bind** your application **components**. Intents are used to **start**, **stop** and transition the **activities** within an application.

Explicitly Starting New Activities

```
Intent intent = new Intent(MyActivity.this, MyOtherActivity.class);
startActivity(intent);
```

- Of course we can add variables to this intent

Implicit Intents and Late Runtime Binding

```
if (somethingWeird && itDontLookGood) {
  Intent intent = new Intent(Intent.ACTION_DIAL,
                             Uri.parse("tel:555-2368"));
  startActivity(intent);
}
```

Receiving side

- Intent intent = getIntent()

Returning results from activity

```
Intent result = new Intent(null, data);
result.putExtra(IS_INPUT_CORRECT, inputCorrect);
result.putExtra(SELECTED_PISTOL, selectedPistol);
```

```
setResult(RESULT_OK, result);
```

```
finish();
```

<u>receive on the activity end</u>

```
@Override
public void onActivityResult(int requestCode,
                             int resultCode,
                             Intent data) {

  super.onActivityResult(requestCode, resultCode, data);

  switch(requestCode) {
    case (SHOW_SUB_ACTIVITY_ONE) : {
      if (resultCode == Activity.RESULT_OK) {
        Uri horse = data.getData();
```

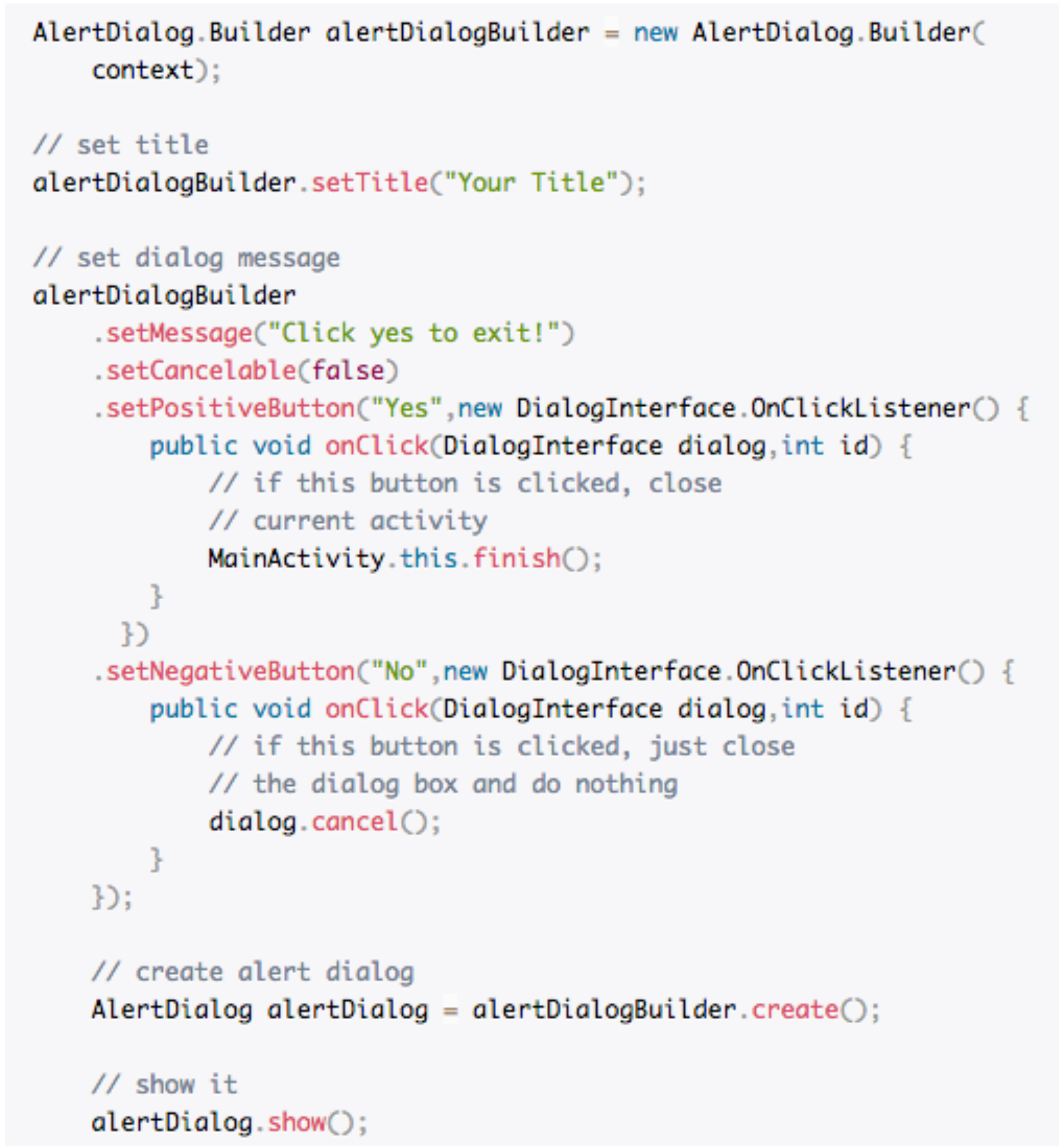
Practice

- Start EditNoteActivity with the intent as in the example
  - Explicit intent
  - startActivityForResult(intent)
  - add to it a test string “test” (putExtra)
- Add an edit box and a go back button and return the result back the NoteActivity
  - set OnClickListener for the go back button
  - init a new Intent
  - setResult(intent)
  - finish

What's left for the note?

- Long click on a note to delete
  - Dialog Interface
- Saving the data

Alert dialog



```
AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(
    context);

// set title
alertDialogBuilder.setTitle("Your Title");

// set dialog message
alertDialogBuilder
    .setMessage("Click yes to exit!")
    .setCancelable(false)
    .setPositiveButton("Yes",new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog,int id) {
            // if this button is clicked, close
            // current activity
            MainActivity.this.finish();
        }
      })
    .setNegativeButton("No",new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog,int id) {
            // if this button is clicked, just close
            // the dialog box and do nothing
            dialog.cancel();
        }
    });

    // create alert dialog
    AlertDialog alertDialog = alertDialogBuilder.create();

    // show it
    alertDialog.show();
```

Practice

- Start a dialog on a long click of the item list view, it should ask either to delete it or not. On delete, delete.

Persistance

- Why ? What happens when turn off the mobile ?
- Exit the app ?
- Too much resources are on hold ?

- We lost the data -> lets save it (persist it)

Android Techniques for Saving Data

The data persistence techniques in Android provide options for balancing speed, efficiency, and robustness:

- **Shared Preferences** When storing the UI state, user preferences, or application settings, you want a lightweight mechanism to store a known set of values. Shared Preferences let you save groups of key/value pairs of primitive data as named preferences.
- **Files** It's not pretty, but sometimes writing to, and reading from, files directly is the only way to go. Android lets you create and load files on the device's internal or external media.
- **SQLite Databases** When managed, structured data is the best approach, Android offers the SQLite relational database library. Every application can create its own databases over which it has total control.
- **Content Providers** Rather than a storage mechanism in their own right, Content Providers let you expose a well-defined interface for using and sharing private data. You can control access to Content Providers using the standard permission system.

Creating and Saving Preferences

```
public static final String MYPREFS = "mySharedPreferences";

protected void savePreferences(){
  // Create or retrieve the shared preference object.
  int mode = Activity.MODE_PRIVATE;
  SharedPreferences mySharedPreferences = getSharedPreferences(MYPREFS,
                                                               mode);
  // Retrieve an editor to modify the shared preferences.
  SharedPreferences.Editor editor = mySharedPreferences.edit();

  // Store new primitive types in the shared preferences object.
  editor.putBoolean("isTrue", true);
  editor.putFloat("lastFloat", 1f);
  editor.putInt("wholeNumber", 2);
  editor.putLong("aNumber", 3l);
  editor.putString("textEntryValue", "Not Empty");

  // Commit the changes.
  editor.commit();
}
```

Retrieving Shared Preferences

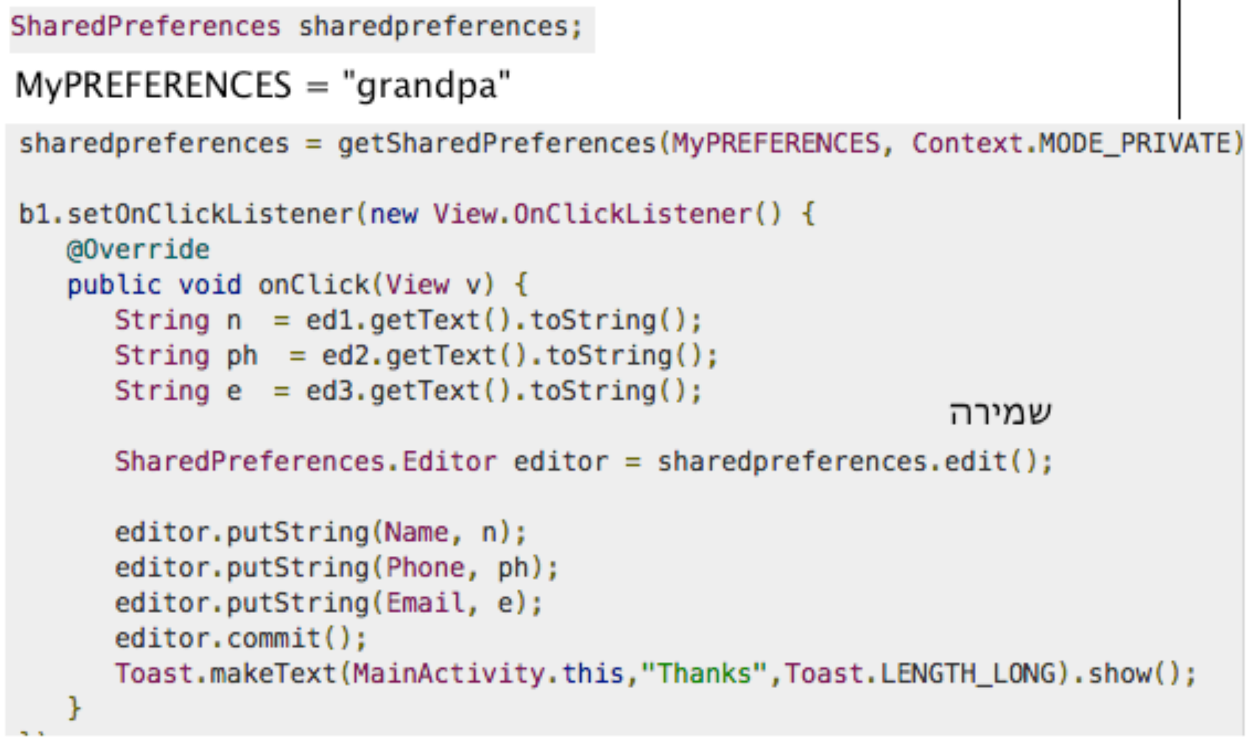
```
int mode = Activity.MODE_PRIVATE;
SharedPreferences mySharedPreferences = getSharedPreferences(MYPREFS,
                                                             mode);

// Retrieve the saved values.
boolean isTrue = mySharedPreferences.getBoolean("isTrue", false);
float lastFloat = mySharedPreferences.getFloat("lastFloat", 0f);
int wholeNumber = mySharedPreferences.getInt("wholeNumber", 1);
long aNumber = mySharedPreferences.getLong("aNumber", 0);
String stringPreference;
stringPreference = mySharedPreferences.getString("textEntryValue",
                                                 "");
```

דוגמה של שמירת נתונים

יצירה של אובייקט שאחראי על שמירה וטעינה של משתנים מהזכרון

```
SharedPreferences sharedpreferences;
```

MyPREFERENCES = "grandpa"



```
sharedpreferences = getSharedPreferences(MyPREFERENCES, Context.MODE_PRIVATE)

b1.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {
      String n  = ed1.getText().toString();
      String ph  = ed2.getText().toString();
      String e  = ed3.getText().toString();

      SharedPreferences.Editor editor = sharedpreferences.edit();

      editor.putString(Name, n);
      editor.putString(Phone, ph);
      editor.putString(Email, e);
      editor.commit();
      Toast.makeText(MainActivity.this,"Thanks",Toast.LENGTH_LONG).show();
   }
```

שמירה

דוגמה של טעינת נתונים

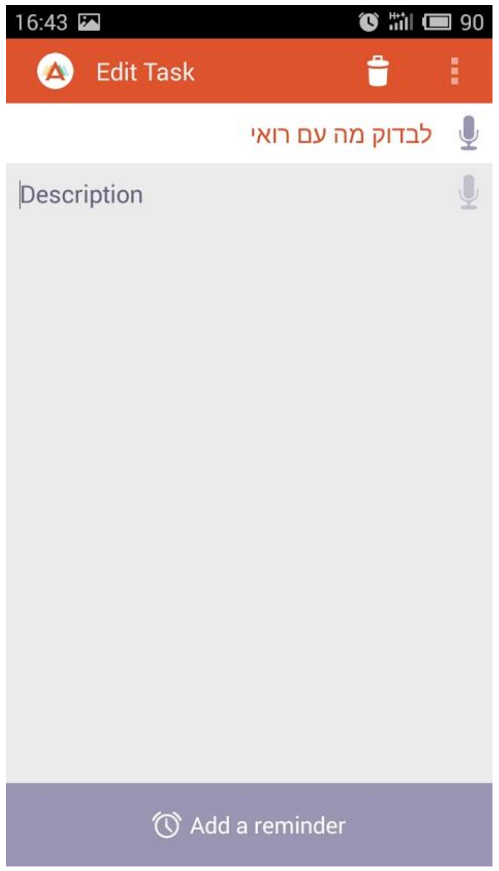
```
int value = shared preferences.getString(Name, "");
```

מפתח

ערך דיפולטי במידה ולא שמרנו שום דבר במפתח מסוים

Practice

- Create a shared preferences
- Save a string
- Save all title notes
- Save all note details



Summary

- Pizza app
  - new view - checkboxes
- TODO app
  - Lists and adapters
  - Intents
  - AlertDialog
  - SharedPreferences

Homework

- Pizza
    - On each checkbox switch to a different imageView
    - What if two checkboxes are picked ?
- TODO
    - Different users may have different tasks
        - Add a login screen - username and password
        - Save it using SharedPreferences
        - Load the specific notes of the user