

## פרק 12 - זיכרון חיצוני (external files)

בפרק הקודם למדנו על זיכרון פנימי ואיך להשתמש בו. (בקובץ בטלפון, מחיקת האפליקציה מוחקת גם את הקבצים האלו)

בפרק הזה נלמד על זיכרון חיצוני בטלפון. (ב sd card, מחיקת האפליקציה לא תמחק את הקבצים)

אם תיגשו ל DDMS תוכלו לראות שבתיקה data יושבות כל האפליקציות שבטלפון שלכם.

Name	Size	Date	Time	Permission
▶ folder acct		2017-12-20	02:44	drwxr-xr-x
▶ folder cache		2017-12-20	01:40	drwxrwx--
▶ folder config		2017-12-20	02:44	dr-x-----
▶ folder d		2017-12-20	02:44	lrwxrwxrwx
▶ folder data		2017-12-14	03:08	drwxrwx--
▶ file default.prop	116	1969-12-31	19:00	-rw-r--r--

תוכלו לגשת ל SD CARD ולשמור שם קבצים שאליהם יוכלו לגשת כל האפליקציות שבטלפון.

dev	2017-12-20	02:45	drwxr-xr-x
etc	2017-12-20	02:44	lrwxrwxrwx
file_contexts	8753 1969-12-31	19:00	-rw-r--r--
fstab.goldfish	495 1969-12-31	19:00	-rw-r--r--
init	154500 1969-12-31	19:00	-rwxr-xr--
init.goldfish.rc	2660 1969-12-31	19:00	-rwxr-xr--
init.rc	19930 1969-12-31	19:00	-rwxr-xr--
init.trace.rc	1795 1969-12-31	19:00	-rwxr-xr--
init.usb.rc	3915 1969-12-31	19:00	-rwxr-xr--
mnt	2017-12-20	02:44	drwxrwxr-x
asec	2017-12-20	02:44	drwxr-xr-x
obb	2017-12-20	02:44	drwxr-xr-x
sdcard	2017-12-20	02:44	lrwxrwxrwx
secure	2017-12-20	02:44	drwx-----
shell	2017-12-20	02:44	drwx-----
proc	1969-12-31	19:00	dr-xr-xr-x
property_contexts	2109 1969-12-31	19:00	-rw-r--r--

בשביל לכתוב לקובץ חיצוני בטלפון יש צורך בהרשאה בקובץ המניפסט:

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

בשביל לקרוא מקובץ חיצוני בטלפון יש צורך בהרשאה בקובץ המניפסט:

```
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
```

בשביל לגשת לקבצים בזיכרון חיצוני נבנה קודם כל פונקציה לבדיקת הרשאה.

יכול להיות שלא תהיה לנו הרשאה לגשת לקובץ הספציפי ולכן בשביל לא לקרוס נשתמש בפונקציה:

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {
```

```
    boolean mExternalStorageAvailable=false;
```

נצהיר תחילה על משתנה בוליאני שיוגדר true אם הקובץ קיים

```
    boolean mExternalStorageWriteable =false;
```

נצהיר על משתנה בוליאני נוסף, true אם אפשר לכתוב לקובץ אחרת false

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_main);
```

```
    setPermissitios();
```

נקרא לפונקציה שתבדוק האם יש לנו הרשאה לגשת לתיקיה

```
}
```

```
public void setPermissitios()
```

```
{
```

```
    String state =Environment.getExternalStorageState();
```

החלק הראשון של הפונקציה, בדיקה אם אפשר לקרוא ולכתוב

```
    if (Environment.MEDIA_MOUNTED.equals(state))
```

שליפת המצב של הזיכרון החיצוני

```
{
```

בדיקה האם המצב של הזיכרון החיצוני הוא mounted, כלומר קריאה וכתיבה

```
        //We can read and write the media
```

```
        mExternalStorageAvailable =mExternalStorageWriteable =true;
```

שינוי הבוליאנים ל true

```
        Toast.makeText(this,"We can read and write",Toast.LENGTH_LONG).show();
```

```
}
```

```

else if (Environment.MEDIA_MOUNTED_READ_ONLY.equals(state)) {
    //We can only read the media

    mExternalStorageAvailable = true;
    mExternalStorageWriteable = false;

    Toast.makeText(this, "We can only read", Toast.LENGTH_LONG).show();
}

else
{
    //Something else is wrong. we can neither read nor write

    mExternalStorageAvailable = mExternalStorageWriteable = false;

    Toast.makeText(this, "we can neither read nor
write", Toast.LENGTH_LONG).show();
}
}

```

החלק השני של הפונקציה - בדיקה אם אפשר רק לקרוא

שינוי המשתנה שמגדיר שהקובץ קיים ל true

המשתנה שמגדיר שאפשר לכתוב לקובץ ישאר false

החלק השלישי של הפונקציה - אנחנו לא יכולים לקרוא או לכתוב לזיכרון החיצוני

בשביל לפתוח הרשאה לקובץ זיכרון חיצוני נשתמש בקוד הבא:

## קבלת הנתיב לקובץ בזיכרון החיצוני:

```
string root = Android.OS.Environment.getExternalStoragePublicDirectory  
(Android.OS.Environment.DirectoryPictures).ToString();
```

```
Java.IO.File myDir = new Java.IO.File(root + "/saved_images");
```

```
String fname = "Image-1.jpg";
```

```
Java.IO.File file = new Java.IO.File(myDir, fname);
```

**mnt/sdcard/image1.jpg - נתיב הקובץ יהיה**

## ניתן לקבל נתיב לתיקיות מיוחדות כגון :

DIRECTORY\_PICTURE  
DIRECTORY\_RINGTONES

```
string root = Android.OS.Environment  
.getExternalStoragePublicDirectory(Android.OS  
.Environment.DirectoryPictures).ToString();
```

למעשה הפונקציה שאנו קוראים לה מקבלת String:

**Environment.getExternalStorageFilesDir(string name)**

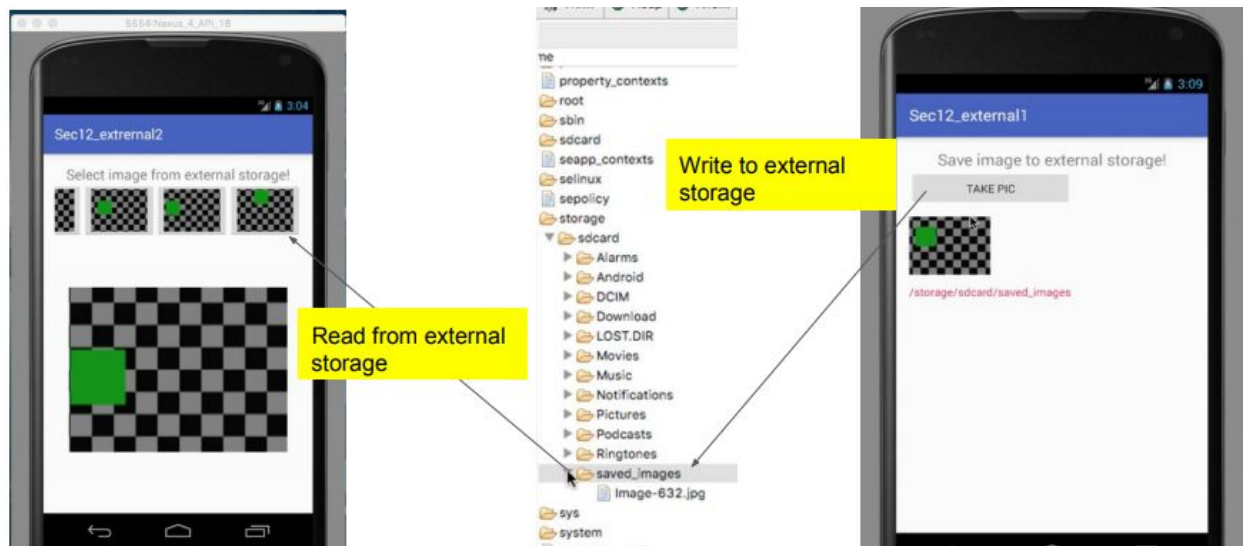
```
File file =new File(Environment.getExternalStorageDirectory(), "myfile.txt");
```

הקובץ יהיה ב- storage/sdcard/myfile.txt

```
File file =new File(getExternalFilesDir(Environment.DIRECTORY_MUSIC),  
"myfile.txt");
```

בדוגמה הראשונה שלנו נלמד לכתוב לתיקיה בזיכרון החיצוני תמונה.

נקרא מהתיקיה את התמונה ונשים בגלריה.



שלב 1 - עיצוב MainActivity

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="iil.co.appschool.sec12_external1.MainActivity">

    <TextView
        android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
```

```
android:text="Select image from external storage!"
```

```
android:textSize="20sp"
```

```
android:layout_gravity="center"
```

```
/>
```

```
<HorizontalScrollView
```

```
android:layout_width="match_parent"
```

```
android:layout_height="wrap_content">
```

```
<LinearLayout
```

```
android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"
```

```
android:orientation="horizontal"
```

```
android:id="@+id/gallery"
```

```
>
```

```
</LinearLayout>
```

```
</HorizontalScrollView>
```

```
<ImageView
```

```
android:layout_width="match_parent"
```

```
android:layout_height="match_parent"
```

```
    android:id="@+id/iv"  
    android:layout_margin="20dp"  
    />  
    <TextView  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:id="@+id/tvPathFile"  
        android:textColor="#ed0a54"  
    />  
</LinearLayout>
```

שלב 2 - מתן הרשאות בקובץ המניפסט:

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />  
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
```

שלב 3 - הצהרות ב MainActivity:

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {  
  
    ImageView iv;  
    Button btnTakePic;  
    TextView tv;
```



שלב 4 - הפניות והאזנה לכפתורים ב MainActivity:

**@Override**

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    btnTakePic=(Button)findViewById(R.id.btnPic);
    btnTakePic.setOnClickListener(this);
    iv=(ImageView)findViewById(R.id.iv);
    tv=(TextView)findViewById(R.id.tvPathFile);
    getAllFile();
}
```

נממש את הפונקציה getAllFile בהמשך

שלב 5 - מימוש onClick:

**@Override**

```
public void onClick(View v) {
```

```
    if(btnTakePic==v)
    {
        Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        startActivityForResult(intent, 0);
    }
}
```

בלחיצה על btnTakePic המשתמש יעבור למסך מצלמה עם requestCode 0

שלב 6 - מימוש onActivityResult לחזרה ממסך המצלמה:

**@Override**

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
```

```
    if(requestCode==0&&resultCode==RESULT_OK)
    {
        Bitmap bitmap=(Bitmap)data.getExtras().get("data");
        iv.setImageBitmap(bitmap);
        saveImageToExternalStorage_version1(bitmap);
    }
}
```

נשנה את התמונה ב iv להיות התמונה שהמשתמש צילם

נקרא לפונקציה שתשמור את התמונה בזיכרון חיצוני

שלב 7 - מימוש הפונקציה שתשמור את התמונה בזיכרון חיצוני

```
private void saveImageToExternalStorage_version1(Bitmap finalBitmap) {
    String root =
    Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES).toString
    ();
```

```
    File myDir = new File(root + "/saved_images");
    myDir.mkdirs();
```

```
    Random generator = new Random();
```

```
    int n = 10000;
```

```
    n = generator.nextInt(n);
```

```
    String fname = "Image-" + n + ".jpg";
```

```
    File file = new File (myDir, fname);
```

```
    if (file.exists ())
```

```
        file.delete ();
```

נגריל מספר שיהיה השם לתמונה

ניצר קובץ לתמונה עם השם fname  
שהגדרנו ועם הנתיב (path)

במידה וכבר קיימת תמונה  
עם השם הזה, נמחק אותה

```
try {
```

```
    FileOutputStream out = new FileOutputStream(file);
```

```
    finalBitmap.compress(Bitmap.CompressFormat.JPEG, 100, out);
```

```
    tv.setText(myDir.getAbsolutePath());
```

```
    out.flush();
```

```
    out.close();
```

```
} catch (Exception e) {
```

```
    tv.setText("file didnt save");
```

```
}
```

```
}
```

נחבר את הנתיב המלא של  
הקובץ עם השם של הקובץ

נדחוס את התמונה  
לקובץ שיצרנו

נשנה את הטקסט ב TextView  
להיות ה path של הקובץ שיצרנו

נסגור את הנתיב ל  
FileStream שיצרנו

במקרה שנכשלנו, נראה מה  
הייתה הבעיה ונציג ב TextView

עד כה נוכל לראות ב DDMS שנשמרות לנו התמונות:

seapp_contexts	611	1969-12-31	19:00	-rw-r--r--
sepolicy	63747	1969-12-31	19:00	-rw-r--r--
storage		2017-12-20	03:51	d---r-x---
sdcards		1969-12-31	19:00	d---rwxr-x
Alarms		2017-12-14	08:09	d---rwxr-x
Android		2017-12-14	03:15	d---rwxr-x
DCIM		2017-12-20	03:02	d---rwxr-x
Download		2017-12-14	08:09	d---rwxr-x
LOST.DIR		2017-12-14	08:07	d---rwxr-x
Movies		2017-12-14	08:09	d---rwxr-x
Music		2017-12-14	08:08	d---rwxr-x
Notifications		2017-12-14	08:09	d---rwxr-x
Pictures		2017-12-20	03:57	d---rwxr-x
saved_images		2017-12-20	03:57	d---rwxr-x
Image-5382.jpg	980	2017-12-20	03:57	----rwxr-x
Podcasts		2017-12-14	08:09	d---rwxr-x
Ringtones		2017-12-14	08:09	d---rwxr-x
sys		2017-12-20	03:51	dr-xr-xr-x
system		2015-02-18	23:41	drwxr-xr-x
ueventd.goldfish.rc	272	1969-12-31	19:00	-rw-r--r--

שלב 8 - מימוש פונקציה שתציג את כל התמונות ב HorizontalScrollView:

```

public void getAllFile() {
    String root =
    Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES).toString();

    File myDir = new File(root + "/saved_images");
    if(myDir.isDirectory()){
        String[] dirList = myDir.list();
        for(int i=0;i<dirList.length;i++){
            File file = new File(myDir, dirList[i]);
            Bitmap tmp = BitmapFactory.decodeFile(file.getAbsolutePath());
            ImageButton btnimg=new ImageButton(this);
            btnimg.Tag = tmp;
            btnimg.setOnClickListener(this);
            btnimg.setImageBitmap(tmp);
            linearLayout.addView(btnimg);
        }
    }
}

```

ניגש לתיקיה בה שמרנו (DirectoryPictures)

ניגש לתיקיה שבפנים בה שמרנו את הקובץ

במידה והגענו לתיקיה

ניצור מערך String של שמות הקבצים

ניקח את הקובץ במקום ה i בתיקיה myDir

נשלוף את התמונה ששמורה בקובץ

נשמור את ה Bitmap

ניצור תמונה שאפשר ללחוץ עליה

ניתן אפשרות האזנה

נשמור את התמונה

נוסיף את התמונה ל linearLayout שבתוך ה HorizontalScrollView

שלב 9 - הוספה לפונקציה onClick לחיצה על ה ImageButton:

```
public void OnClick(View v)
```

```
{
```

```
    if (btnTakePic == v)
```

```
    {
```

```
        Intent intent = new Intent(Android.Provider.MediaStore.ActionImageCapture);
```

```
        StartActivityResult(intent, 0);
```

```
    }
```

```
else{
```

```
    Bitmap bitmap = (Bitmap)v.Tag;
```

```
    iv.SetImageBitmap(bitmap);
```

```
}
```

לחיצה על ה ImageButton תשנה  
את התמונה המוצגת ב  
ImageView לתמונה שלחצנו עליה

בשביל להשתמש בפונקציות של Bitmap  
חובה להמיר את ה View כלפי מטה

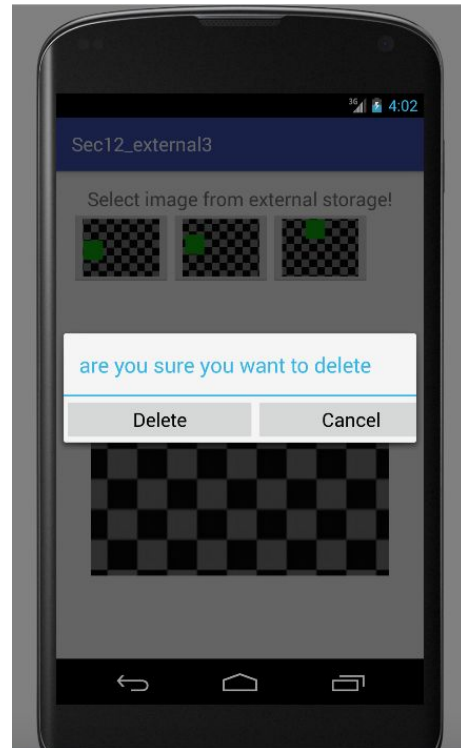
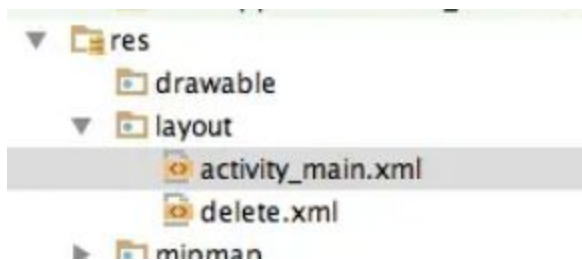
נגדיר את התמונה ששמרנו

סיימנו לבנות את הגלריה!

כעת תוכלו להריץ את האפליקציה, לצלם תמונות ולבנות גלריה משלכם.

נוסיף לאפליקציה שלנו אפשרות שלחיצה ארוכה על התמונה מוחקת אותה:

נוסיף לאפליקציה שבנינו custom layout לדיאלוג שיוצג לנו כשהמשתמש ירצה למחוק  
תמונה:



נערוך את העיצוב של ה custom dialog :

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<LinearLayout
```

```
xmlns:android="http://schemas.android.com/apk/res/android"
```

```
android:orientation="horizontal"
```

```
android:layout_width="match_parent"
```

```
android:layout_height="match_parent"
```

```
>
```

```
<Button
```

```
android:layout_width="200dp"
```

```
android:layout_height="wrap_content"
```

```
android:id="@+id/btnDelete"
```

```

    android:textSize="20sp"
    android:text="Delete"
  />

```

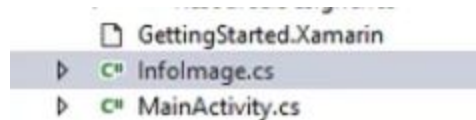
<Button

```

    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:id="@+id/btnCancel"
    android:textSize="20sp"
    android:text="Cancel"
  />

```

בנוסף, נבנה class עזר:



```
public class InfoImage {
```

```
    private File file;
```

```
    private Bitmap bitmap;
```

נצהיר על File ו Bitmap  
בהם נשתמש בהמשך



ניתן לקלאס Constructor, Getter, Setter:

```
public Infolmage(File file, Bitmap bitmap) {  
  
    this.file = file;  
  
    this.bitmap = bitmap;  
  
}  
  
public File getFile() {  
  
    return file;  
  
}  
  
public void setFile(File file) {  
  
    this.file = file;  
  
}  
  
public Bitmap getBitmap() {  
  
    return bitmap;  
  
}  
  
public void setBitmap(Bitmap bitmap) {  
  
    this.bitmap = bitmap;  
  
}  
}
```

בשביל למחוק קובץ נצטרך את הntיב שלו. לכן הפעם נשמור גם את התמונה וגם את הntיב.

נבנה פונקציה אחרת ב MainActivity שמקבלת את כל הקבצים בתיקה, והפעם נשמור גם את התמונה וגם את הntיב:



```

public void getDirFile() {

    linearLayout.removeAllViews();

    String root = Environment.getExternalStoragePublicDirectory(
        Environment.DIRECTORY_PICTURES).toString();

    File myDir = new File(root + "/saved_images");

    if(myDir.isDirectory()) {

        String[] dirList = myDir.list();

        for(int i=0;i<dirList.length;i++) {

            File file = new File(myDir, dirList[i]);

            Bitmap tmp = BitmapFactory.decodeFile(file.getAbsolutePath());

            ImageButton btnimg=new ImageButton(this);
            btnimg.setOnClickListener(this);

            Infomage infomage = new Infomage(file,tmp);
            btnimg.setTag(infomage);

            btnimg.setOnLongClickListener(this);
            btnimg.setImageBitmap(tmp);

            linearLayout.addView(btnimg);

        }
    }
}

```

נצטרך נתיב בשביל למחוק את התמונה. בגלל זה הפעם נשמור גם את התמונה וגם את הנתיב

הפעם נשמור את התמונה והנתיב שלה בקלאס שיצרנו

ב infomage יש נתיב ותמונה

ניתן ל ImageButton האזנה ללחיצה ארוכה ונממש את onLongClickListener

נצהיר על הדיאלוג ועל הכפתורים שבדיאלוג:

```

public class MainActivity extends AppCompatActivity implements View.OnClickListener, View.OnLongClickListener {

    ImageView iv;

    TextView tv;

    LinearLayout linearLayout;

    Dialog d;

    Button btnDelete,btnCancel;
}

```

הפניה לתמונה האחרונה שבחרנו ולנתיב שלה

ניצור פונקציה שיוצרת לנו את הדיאלוג:

```
public void createDialog(Infolmage infolmage){
```

```
    d=new Dialog(this);
```

```
    d.setContentView(R.layout.delete);
```

```
    d.setTitle("are you sure you want to delete this file?");
```

```
    btnCancel=(Button)d.findViewById(R.id.btnCancel);
```

```
    btnCancel.setOnClickListener(this);
```

```
    btnDelete=(Button)d.findViewById(R.id.btnDelete);
```

```
    btnDelete.setOnClickListener(this);
```

```
    btnDelete.setTag(infolmage);
```

```
    d.show();
```

```
}
```

הפונקציה תקבל תמונה ואת הנתיב שלה

נשמור את התמונה והנתיב שלה במקרה  
שהמשתמש לחץ על btnDelete

נממש את הפונקציה להאזנה ללחיצה ארוכה:

```
@Override
```

```
public boolean onLongClick(View v) {
```

```
    Infolmage infolmage = (Infolmage)v.getTag();
```

```
    createDialog(infolmage);
```

```
    return true;
```

```
}
```

נשלוף את התמונה והנתיב שלה ששמרנו  
כשיצרנו את התמונות ב getAllFile

נוסיף ל onClick האזנה ללחיצה על הכפתורים שבדיאלוג:

@Override

```
public void onClick(View v) {  
    if(v==btnCancel) {  
        Toast.makeText(this, "user cancel", Toast.LENGTH_LONG).show();  
    }  
    else if(v==btnDelete) {  
        Infolmage infolmage = (Infolmage)v.getTag();  
        deleteFromExternalStorage_v1(infolmage.getFile());  
    }  
    else {  
        Infolmage infolmage = (Infolmage)v.getTag();  
        iv.setImageBitmap(infolmage.getBitmap());  
    }  
}
```

נקרא לפונקציה שאותה  
נממש בהמשך

נממש את פונקציית המחיקה:

```
public void deleteFromExternalStorage_v1(File file) {
```

```
    Try {
```

```
        if(file.exists()) {
```

נבדוק אם הקובץ קיים

```
            Toast.makeText(this, "file " + file.getAbsolutePath() + "not deleted", Toast.LENGTH_SHORT).show();
```

```
        if(file.delete()) {
```

בדיקה אם צריך למחוק

```
            Toast.makeText(this, "file " + file.getAbsolutePath() + "deleted", Toast.LENGTH_SHORT).show();
```

```
        d.dismiss();
```

נסגור את הדיאלוג

```
        getAllFile();
```

נשים את כל האובייקטים שוב  
על ה LinearLayout, הפעם  
בלי האובייקט שמחקנו

```
    }
```

```
    else Toast.makeText(this, file.getAbsolutePath(), Toast.LENGTH_SHORT).show();
```

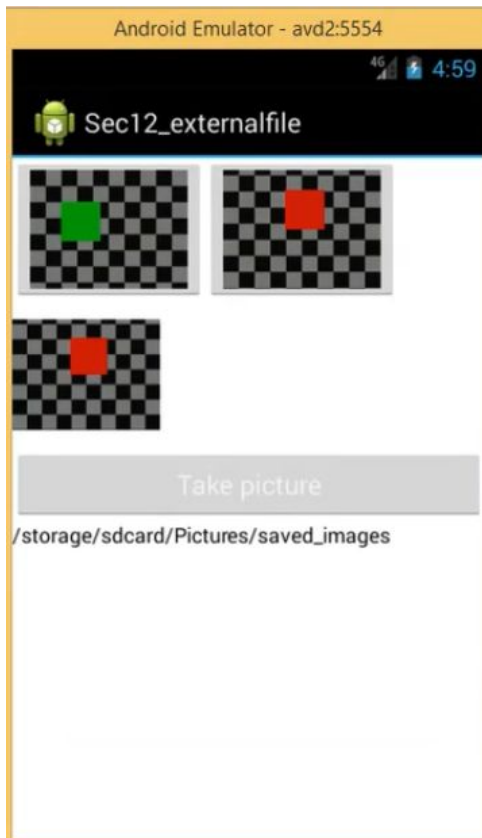
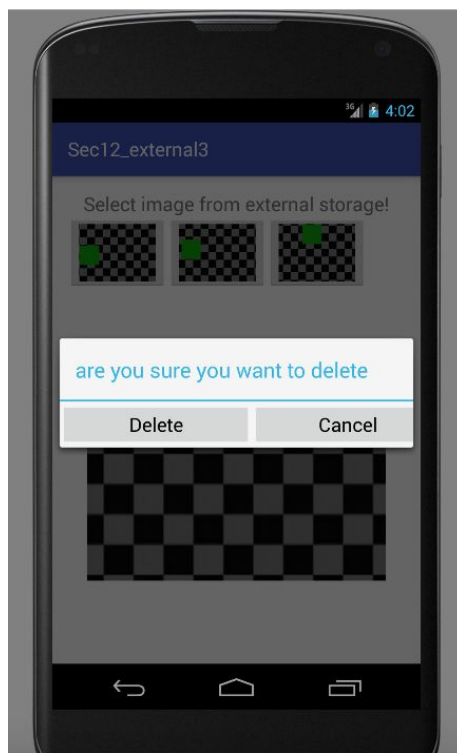
```
}
```

```
catch (Exception e) {
```

```
    Log.e("App", "Exception while deleting file " + e.getMessage());
```

```
}
```

```
}
```



תוכלו להריץ את האפליקציה, להוסיף ולמחוק תמונות מהגלריה ומהזיכרון החיצוני.  
הפונקציה האחרונה שנוסיף לאפליקציה שלנו היא פונקציה למחיקת תיקיה מהזיכרון החיצוני:

```
public void deleteDirectory(String name)
{
    String root = Environment.getExternalStorageDirectory().toString();
    File myDir = new File(root + "/saved_images");
    if(myDir.exists())
        myDir.delete();
}
```

מקבלת את שם התיקה שרוצים למחוק

במקום "saved\_images/" יכולנו לרשום name, כיוון שהפונקציה מקבלת שם התיקה