

## **№6-amaliyot ishi**

### **Ilovalarda smartfonning imkoniyatlarini ishlatilishi**

#### **1. Ishdan maqsad**

- Android-ilovalarni ishlab chiqish instrumentlari bilan tanishish;
- Eng oddiy dasturlar misolida oddiy Android-ilovaning tuzilmasini o‘rganish.
- Emulyatorda ilovani ishga tushirishni o‘rganish.
- Dalvik Debug Monitor Server (DDMS) yordamida ilovani testlashni o‘rganish.

#### **2. Topshiriq**

- x86 emulyatorni yaratish;
- quyidagi ilovalarni o‘rganish, ularni emulyatorda ishga tushirish va har bir misol uchun qo‘shimcha topshiriqni bajarish:

1. Hello, World!
2. Tugmalar bilan ishlash.
3. Animatsiya bilan ishlash.
4. GPS bilan ishlash

#### **3. Zarur dasturiy ta’minot**

- Java Development Kit;
- Android Development Tools plugin o‘rnatilgan Eclipse;

- Android SDK.
- Android NDK.
- Intel Hardware Accelerated Execution Manager.

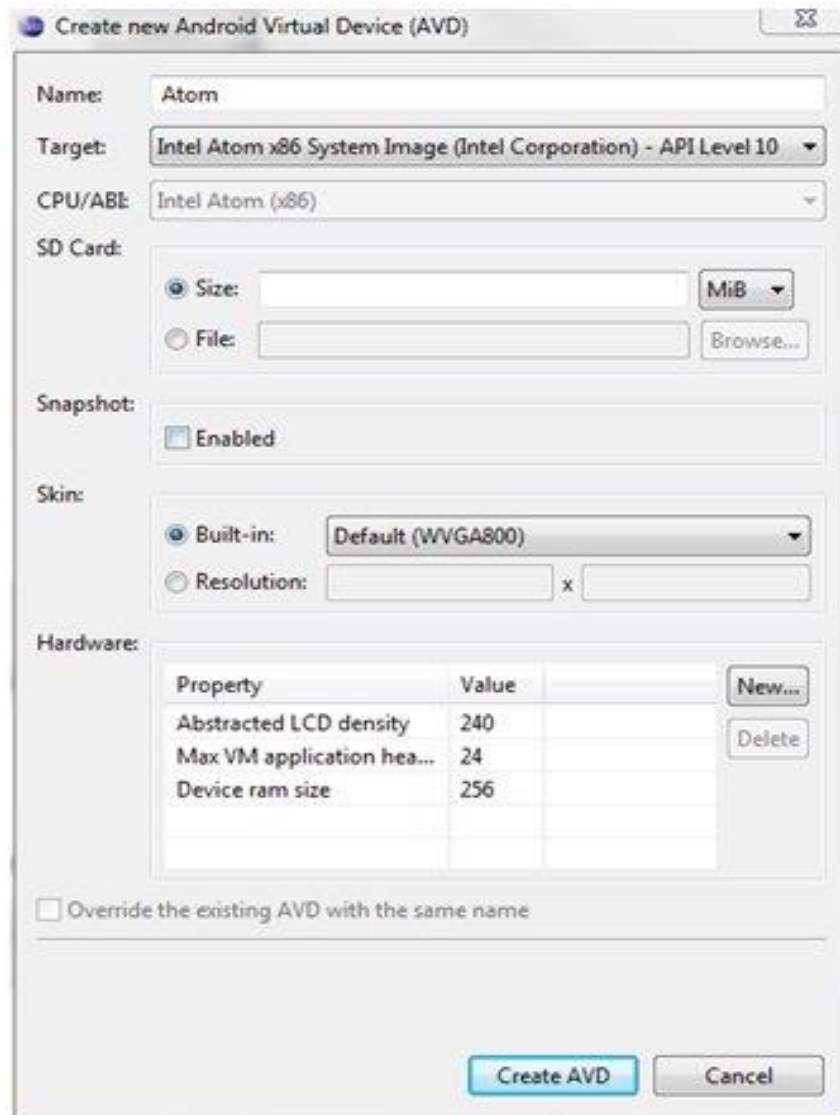
#### **4. Amaliyot ishini bajarish bo'yicha ko'rsatmalar**

2-4 ilovalar ustida ishlash tayyorlangan loyihalarni ishchi muhitga (Workspace) import qilishdan boshlanadi. Import quyidagi tarzda amalga oshiriladi:

- **File -> New -> Project -> Android Project from Existing Code**
- Keyin loyiha joylashgan direktoriya ko'ratiladi va misoldan ishchi muhitga nusxa ko'chiriladi (Copy project into workspace).

#### **Emulyatorni yaratish va sozlash**

- Android Virtual Devices Managerni ishga tushirish: Window → AVD Manager;
- Yangi virtual qurilmani (New) yarating;
- Emulyatorni sozlang: yangi qurilmaga nom bering, qabul qilish Intel Atom x86 platformasini tanlang (kerak bo'lsa SD-karta hajmini berish mumkin) va Create AVDni bosing (6.1-rasm).

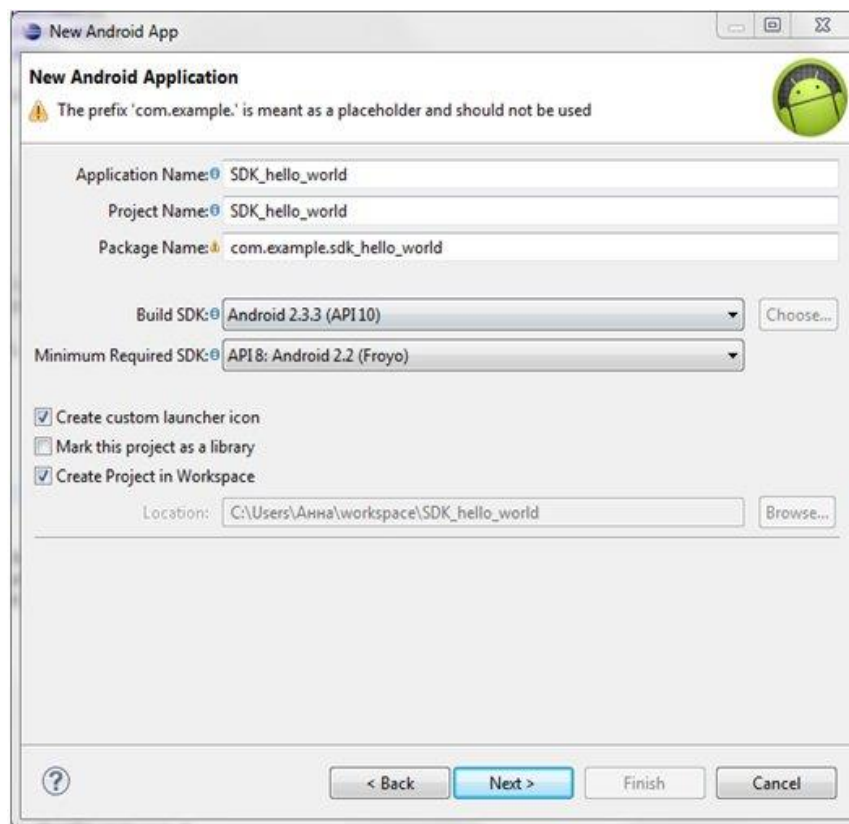


6.1-rasm. Emulyatorni yaratish

Yaratilgan emulyator virtual qurilmalar ro'yxatida paydo bo'ladi.

**Hello, World!**

- Yangi loyihani yarating: **File** → **New** → **Project** → **Android Application Project**
- Loyihani zarur sozlashlarni amalga oshirish: nomni bering, SDKni tanlang, kerak bo'lsa o'z ilovangizni belgisini chizish mumkin (5.2-rasm).



6.2-rasm. Yangi Android-loyihani yaratish

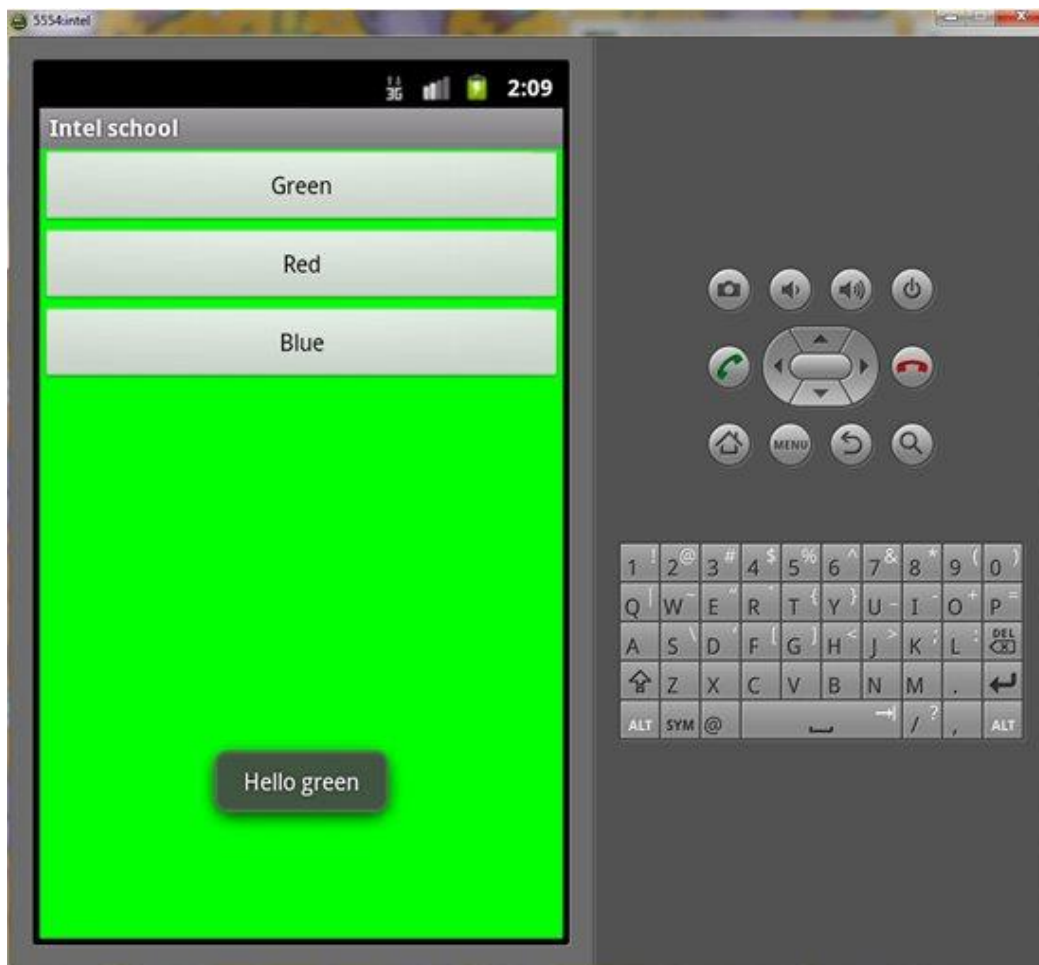
Keyin Activityni yaratish taklif etiladigan oyna paydo bo‘ladi. BlankActivityni tanlang va unga qandaydir nom bering.

- Sizing birinchi ilovangiz tayyor. Uni emulyatorda ishga tushiring.
- **res/layout/activity\_main.xml**, **src/MainActivity.java** va **AndroidManifest.xml** fayllarini oching. Ularda nima borligiga qarang. Tayyor ilovaning qanday ko‘rinishini bilish bilan bu fayllardan har biri nimaga javob berishini tushunib olishga harakat qiling.

### **Tugmalar bilan ishlash**

Ekranida uchta tugmalar mavjud (Red, Green, Blue), ular bosilganida fon rangi o‘zgaradi va mos bildirish paydo bo‘ladi (5.3-rasm).

- 1) Button Example loyihani ishchi sohaga import qiling;
- 2) Uni emulyatorda ishga tushiring va testlang;
- 3) Tugmachani yaratishga (**res/layout/activity\_main.xml**) va uni bosilishiga ishlov berilishiga (**src/MainActivity.java**) misol bilan tanishing;
- 4) Bosilganida fon sifatida qandaydir tasvir o'rnatilishi uchun ilovani yana bitta tugmacha bilan to'ldiring (tasvirni **res/drawable** direktoriyaga ko'chiring).



6.3-rasm. Emulyatorda ishga tushirilgan Button Example ilovasi

### **Animatsiya bilan ishlash**

Ekranida uchta tugmachalar (Frame animation, Transform animation, Cancel animation) joylashgan. Birinchi tugmacha bosilganida kadrlab animatsiya, ikkinchi

tugmacha bosilganida o‘zgartirishlar animatsiyasi qayta tiklanadi, uchinchi tugmacha bosilganida animatsiya to‘xtaydi.

- 1) Animaton Example loyihani ishchi sohaga import qiling;
- 2) Uni emulyatorda ishga tushiring va testlang;
- 3) Kadrlab animatsiya va o‘zgartirishlar animatsiyasini ([res/anim/frame\\_anim.xml](#) va [res/anim/transform\\_anim.xml](#)) yaratishga misollar va ularni ob’ektga ([src/MainActivity.java](#)) qo‘llanilishiga misollar bilan tanishing;
- 4) Ilovani o‘zingizninganimatsiyalashga misollaringizni ishlatilishi bilan to‘ldiring. Bo‘lishi mumkin o‘zgartirishlar animatsiyasi elementlariga misollar 5.1-jadvalda keltirilgan.

6.1-jadval

O‘zgartirishlar animatsiyasi elementlariga misollar

Element	Atributlar
<alpha> shaffoflikni o‘zgartirilishi animatsiyasi	fromAlpha – shaffoflikning boshlang‘ich qiymati toAlpha – shaffoflikning oxirgi qiymati
<scale> o‘lchamni o‘zgartirilishi animatsiyasi	fromxScale – X bo‘yicha boshlang‘ich masshtab toxScale – X bo‘yicha oxirgi masshtab fromYScale – Y bo‘yicha boshlang‘ich masshtab toYScale –Y bo‘yicha oxirgi masshtab pivotX – biriktirilgan markaz X-koordinatasi pivotY – biriktirilgan markaz Y-koordinatasi

<translate> harakatlanish animatsiyasi (vertikal/gorizontal)	fromXDelta – X bo'yicha boshlang'ich holat toXDelta – X bo'yicha oxirgi holat fromYDelta – Y bo'yicha boshlang'ich holat toYDelta – Y bo'yicha oxirgi holat
<rotate> aylanish animatsiyasi	fromDegrees – boshlang'ich aylanish burchagi toDegrees – oxirgi aylanish burchagi pivotX – aylanish burchagi X koordinatasi pivotY – aylanish burchagi Y koordinatasi

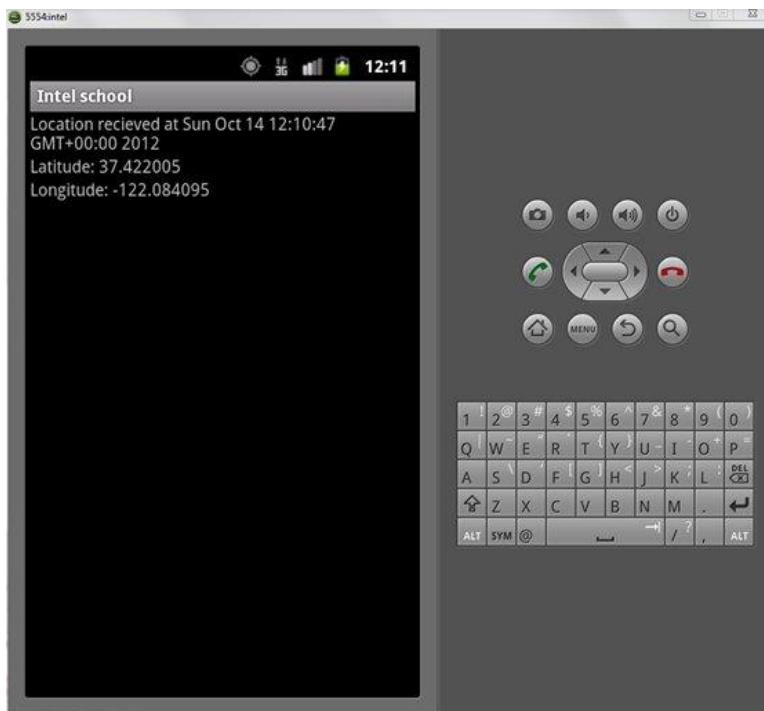
### GPS bilan ishlash

Ekranda joylashish o'rni haqidagi quyidagi ma'lumotlar berilgan (5.4-rasm):

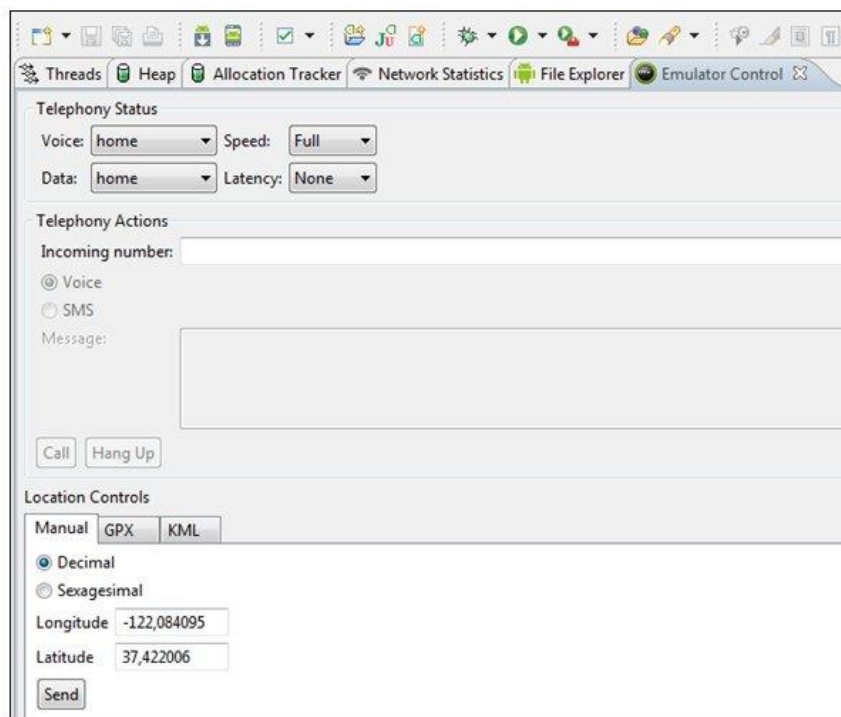
- Status (ma'lumotlar qachon olinganligi haqidagi bildirish);
- Kenglik;
- Uzunlik;
- Location Example loyihani ishchi sohaga import qiling;
- Uni emulyatorda ishga tushiring. Ekranda "Waiting for location"

bildirishi paydo bo'ladi;

- Endi virtual qurilmaga joylashish o'rni haqidagi ma'lumotlarni uzatish zarur. Buning uchun emulyator oynasini yopmasdan DDMSni ishga tushiring (**Window → Open Perspective → Other → DDMS**) va Emulator Control qo'yilmasi orqali qurilmaga zarur ma'lumotlarni jo'nating (6.5-rasm).



6.3-rasm. Emulyatorda ishga tushirilgan Location Example ilovasi



6.5-rasm.Uzunlik va kenglik qiymatlari emulyatorga DDMS yordamida jo‘natiladi



- Dastur to'g'ri ishlayotganiga va emulyator joylashish o'rni haqidagi ma'lumotlarni chiqarganligiga ishonch hosil qiling;
- AndroidManifest.xml faylidagi **<uses-permission>** tegga e'tibor bering va nima uchun u bu ilovaga yozilishini o'ylab ko'ring;
- **src/MainActivity.java** faylini oching va bu ilovaning Activity hayot sikli qanday bo'lib o'tishini va har bir bosqichda qanday amallar bajarilishini tushunib olishga harakat qiling.

### **Amaliyot ishiga ilova**

Bu ilova misollarda ko'ribchiqilgan dasturlarning matnlaridan iborat.

#### **1. Button Example**

- o res/layout/activity\_main.xml
- o <manifest xmlns:android="http://schemas.android.com/apk/res/android"
- o package="com.example.application"
- o android:versionCode="1"
- o android:versionName="1.0" >
- o <uses-sdk
- o android:minSdkVersion="8"
- o android:targetSdkVersion="15" />
- o <application
- o android:icon="@drawable/ic\_launcher"
- o android:label="@string/app\_name">
- o <activity
- o android:name=".MainActivity"
- o android:label="@string/title\_activity\_main" >
- o <intent-filter>

- <action android:name="android.intent.action.MAIN" />
- <category android:name="android.intent.category.LAUNCHER" />
- </intent-filter>
- </activity>
- </application>
- </manifest>
- src/MainActivity.java
- package com.example.application;
- import android.app.Activity;
- import android.graphics.Color;
- import android.os.Bundle;
- import android.view.View;
- import android.view.View.OnClickListener;
- import android.widget.Button;
- import android.widget.LinearLayout;
- import android.widget.Toast;
- public class MainActivity extends Activity implements OnClickListener {
- private Button switchToGreen;
- private Button switchToRed;
- private Button switchToBlue;
- private LinearLayout screenLayout;
- private Toast informationToast;
- @Override
- public void onCreate(Bundle savedInstanceState) {
- super.onCreate(savedInstanceState);
- setContentView(R.layout.activity\_main);
- // init buttons

- switchToBlue = (Button) findViewById(R.id.switchBlue);
- switchToGreen = (Button) findViewById(R.id.switchGreen);
- switchToRed = (Button) findViewById(R.id.switchRed);
- screenLayout = (LinearLayout) findViewById(R.id.screenLayout);
- // setup listeners
- switchToBlue.setOnClickListener(this);
- switchToRed.setOnClickListener(this);
- switchToGreen.setOnClickListener(this);
- informationToast = Toast.makeText(this, "", Toast.LENGTH\_SHORT);
- }
- public void onClick(View view) {
- if (switchToBlue.equals(view)) {
- screenLayout.setBackgroundColor(Color.BLUE);
- showToast("Hello blue");
- } else if (switchToRed.equals(view)) {
- screenLayout.setBackgroundColor(Color.RED);
- showToast("Hello red");
- } else if (switchToGreen.equals(view)) {
- screenLayout.setBackgroundColor(Color.GREEN);
- showToast("Hello green");
- }
- }
- private void showToast(String text) {
- informationToast.cancel();
- informationToast.setText(text);
- informationToast.show();
- }

- }

## 2. Animation Example

- res/anim/frame\_anim.xml

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

- <animation-list

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

- android:oneshot="false" >

- <item

- android:drawable="@drawable/ic\_launcher"

- android:duration="200"/>

- <item

- android:drawable="@drawable/ic\_launcher1"

- android:duration="200"/>

- <item

- android:drawable="@drawable/ic\_launcher2"

- android:duration="200"/>

- <item

- android:drawable="@drawable/ic\_launcher3"

- android:duration="200"/>

- </animation-list>

- res/anim/transform\_anim.xml

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

- <set xmlns:android="http://schemas.android.com/apk/res/android"

- android:shareInterpolator="false" >

- <scale

- android:duration="700"

- android:fillAfter="false"

- android:fromXScale="1.0"
- android:fromYScale="1.0"
- android:interpolator="@android:anim/accelerate\_decelerate\_interpolator"
- android:pivotX="50% "
- android:pivotY="50% "
- android:toXScale="1.4"
- android:toYScale="0.6" />
- <set android:interpolator="@android:anim/decelerate\_interpolator" >
- <scale
- android:duration="400"
- android:fillBefore="false"
- android:fromXScale="1.4"
- android:fromYScale="0.6"
- android:pivotX="50% "
- android:pivotY="50% "
- android:startOffset="700"
- android:toXScale="0.0"
- android:toYScale="0.0" />
- <rotate
- android:duration="400"
- android:fromDegrees="0"
- android:pivotX="50% "
- android:pivotY="50% "
- android:startOffset="700"
- android:toDegrees="-45"
- android:toYScale="0.0" />
- </set>

- </set>
- src/MainActivity.java
- package com.example.application;
- import android.app.Activity;
- import android.graphics.Color;
- import android.graphics.drawable.AnimationDrawable;
- import android.os.Bundle;
- import android.view.View;
- import android.view.View.OnClickListener;
- import android.view.animation.Animation;
- import android.view.animation.AnimationUtils;
- import android.widget.Button;
- import android.widget.ImageView;
- public class MainActivity extends Activity implements OnClickListener {
- private Button startFrameAnim;
- private Button startTransformAnim;
- private Button cancelAnim;
- private ImageView animationView;
- @Override
- public void onCreate(Bundle savedInstanceState) {
- super.onCreate(savedInstanceState);
- setContentView(R.layout.activity\_main);
- startFrameAnim = (Button) findViewById(R.id.frameAnimationStart);
- startTransformAnim= (Button)
- findViewById(R.id.transformAnimationStart);
- cancelAnim = (Button) findViewById(R.id.cancelAnimation);
- animationView = (ImageView) findViewById(R.id.animationView);

- startFrameAnim.setOnClickListener(this);
- startTransformAnim.setOnClickListener(this);
- cancelAnim.setOnClickListener(this);
- }
- public void onClick(View v) {
- if (startFrameAnim.equals(v)) {
- animationView.setBackgroundResource(R.anim.frame\_anim);
- AnimationDrawable animation =
- (AnimationDrawable) animationView.getBackground();
- animation.start();
- } else if (startTransformAnim.equals(v)) {
- animationView.setBackgroundResource(R.drawable.ic\_launcher);
- Animation transformAnimation =
- AnimationUtils.loadAnimation(this, R.anim.transform\_anim);
- animationView.startAnimation(transformAnimation);
- } else if (cancelAnim.equals(v)) {
- animationView.setBackgroundColor(Color.BLACK);
- }
- }
- }

### 3. Location Example

- src/MainActivity.java
- package com.example.application;
- import java.util.Date;
- import android.app.Activity;
- import android.location.Criteria;
- import android.location.Location;

- import android.location.LocationListener;
- import android.location.LocationManager;
- import android.os.Bundle;
- import android.widget.TextView;
- public class MainActivity extends Activity implements LocationListener {
- private TextView latitudeLabel;
- private TextView longitudeLabel;
- private TextView statusLabel;
- private LocationManager locationManager;
- @Override
- public void onCreate(Bundle savedInstanceState) {
- super.onCreate(savedInstanceState);
- setContentView(R.layout.activity\_main);
- latitudeLabel = (TextView) findViewById(R.id.latitudeLabel);
- longitudeLabel = (TextView) findViewById(R.id.longitudeLabel);
- statusLabel = (TextView) findViewById(R.id.statusLabel);
- locationManager = (LocationManager)
- getSystemService(Activity.LOCATION\_SERVICE);
- }
- @Override
- protected void onResume() {
- super.onResume();
- // construct a criteria with best accuracy
- Criteria criteria = new Criteria();
- criteria.setAccuracy(Criteria.ACCURACY\_FINE);
- // get best ENABLED provider that meets the criteria
- String provider = locationManager.getBestProvider(criteria, true);



- // request the updates
- locationManager.requestLocationUpdates(provider, 0, 0, this);
- }
- @Override
- protected void onPause() {
- super.onPause();
- locationManager.removeUpdates(this);
- }
- public void onLocationChanged(Location location) {
- statusLabel.setText("Location recieved at " + new Date());
- latitudeLabel.setText("Latitude: " + location.getLatitude());
- longitudeLabel.setText("Longitude: " + location.getLongitude());
- }
- public void onProviderDisabled(String provider) {
- }
- public void onProviderEnabled(String provider) {
- }
- public void onStatusChanged(String provider, int status, Bundle extras) {
- }
- }
- AndroidManifest.xml
- <manifest xmlns:android="http://schemas.android.com/apk/res/android"
- package="com.example.application"
- android:versionCode="1"
- android:versionName="1.0" >
- <uses-sdk
- android:minSdkVersion="8"

- `android:targetSdkVersion="15" />`
- `<uses-permission`  
`android:name="android.permission.ACCESS_COARSE_LOCATION"/`  
`>`
- `<uses-permission`  
`android:name="android.permission.ACCESS_FINE_LOCATION"/>`
- `<application`
- `android:icon="@drawable/ic_launcher"`
- `android:label="@string/app_name">`
- `<activity`
- `android:name=".MainActivity"`
- `android:label="@string/title_activity_main" >`
- `<intent-filter>`
- `<action android:name="android.intent.action.MAIN" />`
- `<category android:name="android.intent.category.LAUNCHER" />`
- `</intent-filter>`
- `</activity>`
- `</application>`
- `</manifest>`

#### **4. Hisobning tarkibi**

1. Ishlab chiqilgan loyihaning Xml fayli printskrinini keltirish.
2. Ishlab chiqilgan loyiha kodining listingini keltirish.

#### **Nazorat savollari**

1. Emulyatorni sozlash tatibini tushuntiring
2. Emulyatordagi tugmalarni yaratish va tugmachalar bilan ishlashni tushuntiring?
3. Kadrlab animatsiya qanday amalga oshiriladi?
4. O'zgartirishlar animatsiyasi qanday bajariladi?
5. GPS bilan ishlashni tushuntiring?
6. Joylashi o'rni haqidagi ma'lumotlarni virtual qurilmaga qanday uzatiladi?
7. Bu ilovaning Activity hayot sikli qanday bo'lib o'tadi va har bir bosqichda qanday amallar bajariladi?