



Practis

ABC-218

Hands-on  
Project

הכנה פרקטית לה-טק

# Course syllabus

## Android development – course syllabus

- **Session 1 – First android app**
- Session 2 – Multiple screens & Menus
- Session 3 – Files & Network
- Session 4 – Touch, Permission and feedback
- Session 5 – Location And basic animations
- Session 6 – Ads, Uploading to Google Play



# Session 1 overview

- What is Android
- First Android App
- Testing applications
- Android project structure
- XML
- Enhancing First Android App

# Session 1 – Hands on

## Exercise – Basic Android App –

- Create new Android App – Coins collector
  - UI:
    - Player (Image)
    - Coin (Image)
    - Coins collected (counter)
    - 4 buttons to move player in required direction
  - Once coin picked up:
    - Move coin to a new random location
    - Score increase by 1



# Session 1 – Hands on

## Exercise – Basic Android App –

- Created in steps:
  - Add player image
  - Add coins image
  - Add 4 buttons
  - Each button click moves the player in direction
  - After each click check if coin picked up
  - Add Coin pick up logic
    - Generate random location
    - Place coin in that random location
    - Update Score



# Session 1 – Hands on

## Exercise – Basic Android App –

- Tip for importing images: file->Settings



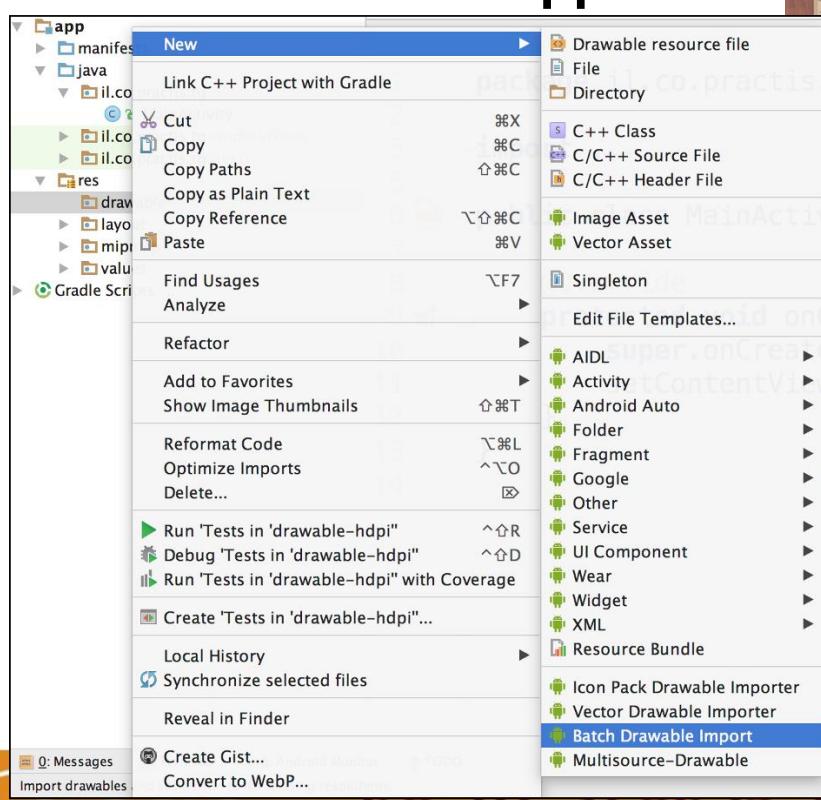
The screenshot shows the 'Preferences' window in Android Studio, specifically the 'Plugins' section. The left sidebar lists various settings categories like Appearance & Behavior, System Settings, Keymap, Editor, Plugins, Version Control, Build, Execution, Deployment, Languages & Frameworks, and Tools. The 'Plugins' category is selected. In the main pane, a search bar at the top has 'batch' typed into it. Below the search bar, a list of available plugins is shown, with 'Android Drawable Importer' highlighted. To the right of the list, detailed information about the plugin is displayed, including its name ('Android Drawable Importer'), version ('0.5~81'), and three numbered features: 1. AndroidIcons and Material Icons Drawable Import, 2. Batch Drawable Import, and 3. Multisource-Drawable. At the bottom of the pane, there are buttons for 'Install JetBrains plugin...', 'Browse repositories...', and 'Install plugin from disk...'. A footer bar at the bottom contains 'Cancel', 'Apply', and 'OK' buttons, along with a question mark icon.

6

# Session 1 – Hands on

## Exercise – Basic Android App –

- Then:



# Session 1 – Hands on

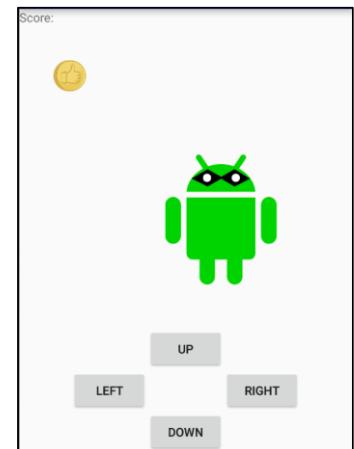
## Exercise – Basic Android App –

- Created in steps: Layout Format:



▼ **RelativeLayout**

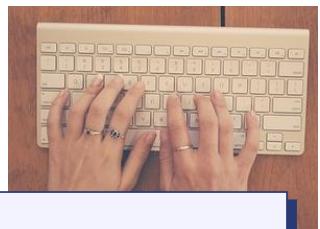
- Ab **txtScore** (TextView) – "@string/txt\_score"
- ▼ **layoutGame** (RelativeLayout)
  - player** (ImageView)
  - coin** (ImageView)
- ▼ **layoutControls** (GridLayout)
  - OK **btnUp** (Button) – "@string/btn\_up\_text"
  - OK **btnDown** (Button) – "@string/btn\_down\_text"
  - OK **btnLeft** (Button) – "@string/btn\_left\_text"
  - OK **btnRight** (Button) – "@string/btn\_right\_text"



# Session 1 – Hands on

## Exercise – Basic Android App –

- Check ImageView collisions:



```
int[] location = new int[2];  
  
player.getLocationInWindow(location);  
Rect rectPlayer = new Rect(location[0],  
location[1], location[0] + player.getWidth(), location[1] +  
player.getHeight());  
  
coin.getLocationInWindow(location);  
Rect rectCoin = new Rect(location[0],  
location[1], location[0] + coin.getWidth(), location[1] +  
coin.getHeight());  
  
// collision is detected  
if (Rect.intersects(rectPlayer, rectCoin)) {  
    // do collision action  
}
```

9

# Session 1 – Hands on

## Exercise – Basic Android App –

- Move coin to a random location:



```
Random randomGenerator = new Random();  
  
View gameField =  
        (View) findViewById(R.id.layoutGame);  
int maxHeight = gameField.getHeight();  
int maxWidth = gameField.getWidth();  
  
int x = randomGenerator.nextInt(maxWidth);  
int y = randomGenerator.nextInt(maxHeight);  
  
coin.setX(x);  
coin.setY(y);
```

10

## Session 2 overview

- **Activity**
- **Menus**
- **Intents**
- **Context**
- **Android debugging**

## Session 2 – Hands on

### Exercise – Basic Android App –

- Add Menu with options:
  - Restart game
    - Places player and coin in default locations
    - Resets score count
  - Show about screen:
    - Add a new Activity screen with info about us
    - Show high score (In Toast: not implemented yet)
    - Show our web page



## Session 2 – Hands on

### Exercise – Adding Activities –

- Menu XML file:



```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android" >

    <item
        android:id="@+id/menuNewGame"
        android:title="@string/menu_new_game_text"/>
    <item
        android:id="@+id/menuShowAbout"
        android:title="@string/menu_show_about_text"/>
    <item
        android:id="@+id/menuShowHighScore"
        android:title="@string/menu_show_highscore_text"/>
    <item
        android:id="@+id/menuShowWebPage"
        android:title="@string/menu_show_web_page"/>

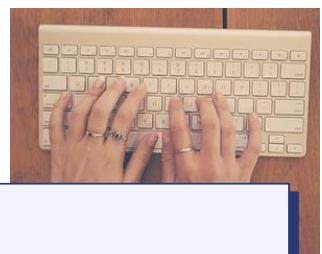
</menu>
```

13

## Session 2 – Hands on

### Exercise – Adding Activities –

- Add menu to Activity:

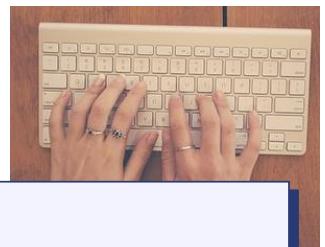


```
@Override  
public boolean onCreateOptionsMenu(Menu menu) {  
  
    MenuInflater inflater = getMenuInflater();  
    inflater.inflate(R.menu.mymenu, menu);  
    return true;  
}
```

## Session 2 – Hands on

### Exercise – Adding Activities –

- Handle menu events:



```
@Override  
public boolean onOptionsItemSelected(MenuItem item) {  
    switch (item.getItemId()) {  
  
        case R.id.menuNewGame:  
            startNewGame();  
            return true;  
  
        case R.id.menuShowAbout:  
            showAboutScreen();  
            return true;  
  
        default:  
            return super.onOptionsItemSelected(item);  
    }  
}
```

15

## Session 2 – Hands on

### Exercise – Adding Activities –

- Move to another Activity:

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



- Open webpage using Intent:

```
// Prepare Intent to open URL  
String url = "http://www.practis.co.il";  
Intent intent = new Intent(Intent.ACTION_VIEW);  
intent.setData(Uri.parse(url));  
  
// Start Intent  
startActivity(intent);
```

## Session 3 overview

- Working with files
- Working with the network
- Android Project

## Session 3 – Hands on

### Exercise – Adding files –

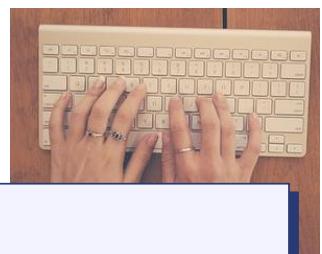
- Let's add game ending logic
  - Each game is 15 seconds long
  - Once game over: freeze timer and buttons
  - New game starts: unfreeze timer and buttons
- Save and Load high score from Shared preferences
- Add menu option to save screenshot to new external file



## Session 3 – Hands on

### Exercise – Adding files –

- Adding timer logic: (new variables)



```
// Duration of a game in ms
public final int GAME_TIMER_START_VALUE = 15000;

// UI update interval in ms
public final int GAME_TIMER_TICK_VALUE = 100;

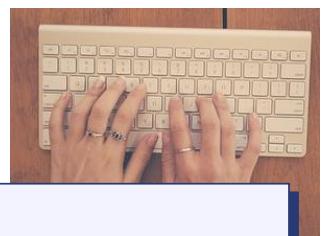
// Initial delay when new game starts
public final int NEW_GAME_DELAY = 300;
```

19

## Session 3 – Hands on

### Exercise – Adding files –

- Adding timer logic: (inside startNewGame())



```
// restart game timer
if (timer != null) {
    timer.cancel();
}

timeRemaining = GAME_TIMER_START_VALUE;
timer = new Timer();

timer.scheduleAtFixedRate(new TimerTask() {
    public void run() {
        // signal UI thread to do timer actions
        mHandler.obtainMessage(1).sendToTarget();
    }
}, NEW_GAME_DELAY, GAME_TIMER_TICK_VALUE);
```

20

# Session 3 – Hands on

## Exercise – Adding files –

- Handling timer tick logic:



```
public Handler mHandler = new Handler(new Handler.Callback() {
    @Override
    public boolean handleMessage(Message msg) {
        timeRemaining-= GAME_TIMER_TICK_VALUE;
        TextView timeView = (TextView)findViewById(R.id.txtTime);
        String message = "time:" + timeRemaining;
        timeView.setText(message);

        // if time to finish the game
        if (timeRemaining <= 0) {
            gameEnded();
        }

        return true;
    } });
}
```

21

## Session 3 – Hands on

### Exercise – Adding files –

- Handling game ended logic:



```
private void gameEnded() {  
    timer.cancel();  
  
    // disable buttons logic  
    Button upBtn = (Button) findViewById(R.id.btnUp);  
    upBtn.setEnabled(false);  
  
    ...  
}  
  
// don't forget to enable them starting new game
```

## Session 3 – Hands on

### Exercise – Adding files –

- Adding timer logic: (inside onCreate())



```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
  
    ...  
    // run "startNewGame" logic only when UI is ready  
    RelativeLayout mainLayout =  
        (RelativeLayout) findViewById(R.id.layoutMain);  
  
    mainLayout.post(new Runnable() {  
        @Override  
        public void run() {  
            startNewGame();  
        }  
    });  
}
```

23

## Session 3 – Hands on

### Exercise – Adding files –

- Take screenshot:



```
// get screenshot
View view = getWindow().getDecorView().getRootView();

Bitmap bitmap = Bitmap.createBitmap(view.getWidth(),
                                    view.getHeight(),
                                    Bitmap.Config.ARGB_8888);

Canvas canvas = new Canvas(bitmap);

view.draw(canvas);

// screenshot is now in memory inside the bitmap variable
```

24

## Session 3 – Hands on

### Exercise – Adding files –

- Save screenshot to file:



```
// generate filename
String filename = System.currentTimeMillis() + ".png";
File externalStoragePath =
        Environment.getExternalStorageDirectory();
File newFilePath = new File(externalStoragePath, filename);

// save screenshot to file
FileOutputStream out = new FileOutputStream(newFilePath);
bitmap.compress(Bitmap.CompressFormat.PNG, 100, out);
out.close();

// add image to phone gallery
MediaStore.Images.Media.insertImage(getApplicationContext(),
        bitmap, "", "");
```

25

## Session 3 – Hands on

### Exercise – Adding files –

- Ask external storage permissions:
  - **Don't forget to add in the manifest file**



```
int checkResult = ContextCompat.checkSelfPermission(this,  
        Manifest.permission.WRITE_EXTERNAL_STORAGE);  
  
if (checkResult != PackageManager.PERMISSION_GRANTED) {  
    ActivityCompat.requestPermissions(this, new  
        String[] {Manifest.permission.WRITE_EXTERNAL_STORAGE}, 0);  
  
} else {  
    ...  
}
```

26

## Session 4 overview

- Touch gestures
- Permissions
- Common feedback tools
  - Vibration
  - Flash
  - Sounds

## Session 4 – Hands on

### Exercise – Adding feedback –

- Add swipe control instead of buttons
  - Add timer that moves player in current direction automatically
  - Remove buttons
- Add sound for coin collection
- Add background music
  - Add menu option to disable music
- Add vibration upon game over



## Session 4 – Hands on

### Exercise – Adding Swipe –

- Step #1 – auto movement by timer
  - Don't remove buttons yet



```
private int moveX = DELTA;
private int moveY = 0;

public void doRight(View view) {
    setMoveDirection(DELTA, 0);
}

private void setMoveDirection(int x, int y) {
    moveX = x;
    moveY = y;
}

// in the timer function add a call to move player:
movePlayer();
```

29

## Session 4 – Hands on

### Exercise – Adding Swipe –

- Step #2 – add swipe
  - Extend GestureDetector.SimpleOnGestureListener



```
public class MySimpleGestureListener extends
    GestureDetector.SimpleOnGestureListener {

    @Override
    public boolean onDown(MotionEvent event) {
        return true;
    }

    // implement the swipe function here:
    @Override
    public boolean onFling(MotionEvent e1, MotionEvent e2,
        float velocityX, float velocityY) {
        ...
    }
}
```

30

## Session 4 – Hands on

### Exercise – Adding Swipe –

- Step #2 – add swipe: implement onFling



```
public boolean onFling(MotionEvent e1, MotionEvent e2,
                      float velocityX, float velocityY) {
    boolean result = false;
    // find amount of change in both X and Y axis
    float diffX = e2.getX() - e1.getX();
    float diffY = e2.getY() - e1.getY();

    if (Math.abs(diffX) > Math.abs(diffY)) {
        if (Math.abs(diffX) > SWIPE_THRESHOLD &&
            Math.abs(velocityX) > SWIPE_VELOCITY_THRESHOLD) {
            if (diffX > 0) {
                parent.onSwipeRight();
            } else {
                parent.onSwipeLeft();
            }
        }
        result = true;
    }
}
```

31

## Session 4 – Hands on

### Exercise – Adding Swipe –

- Step #2 – add swipe: implement onFling



```
else if (Math.abs(diffY) > SWIPE_THRESHOLD &&
         Math.abs(velocityY) > SWIPE_VELOCITY_THRESHOLD) {
    if (diffY > 0) {
        parent.onSwipeBottom();
    } else {
        parent.onSwipeTop();
    }
    result = true;
}

return result;
}

private MainActivity parent = null;
private static final int SWIPE_THRESHOLD = 100;
private static final int SWIPE_VELOCITY_THRESHOLD = 100;
```

32

## Session 4 – Hands on

### Exercise – Adding Swipe –

- Step #3 – add swipe code to MainActivity:



```
private GestureDetectorCompat mDetector;  
  
protected void onCreate(Bundle savedInstanceState) {  
    ...  
    mDetector = new GestureDetectorCompat(this,  
                                         new MySimpleGestureListener());  
}  
  
public boolean onTouchEvent(MotionEvent event) {  
    this.mDetector.onTouchEvent(event);  
    return super.onTouchEvent(event);  
}  
}
```

33

## Session 4 – Hands on

### Exercise – Adding Swipe –

- Step #3 – add swipe handlers to MainActivity:
  - Same logic as move buttons



```
public void doUp(View view) {  
    setMoveDirection(0, -1 * DELTA);  
}  
public void onSwipeTop() {  
    setMoveDirection(0, -1 * DELTA);  
}
```

- Once working we can remove buttons!

34

## Session 4 – Hands on

### Exercise – Adding Coin sound –

- import mp3 to res/raw and then:



```
private void initializeSoundPool() {  
  
    AudioManager audioManager = (AudioManager) getSystemService(AUDIO_SERVICE);  
    maxVolume =  
        (float) audioManager.getStreamMaxVolume(AudioManager.STREAM_MUSIC);  
  
    // Load the sound  
    soundPool = new SoundPool(10, AudioManager.STREAM_MUSIC, 0);  
    soundPool.setOnLoadCompleteListener(new  
        SoundPool.OnLoadCompleteListener() {  
            @Override  
            public void onLoadComplete(SoundPool soundPool, int sampleId,  
                int status) {  
                loaded = true;  
            }  
        } );  
  
    soundID = soundPool.load(this, R.raw.coin, 1);  
}
```

35

## Session 4 – Hands on

### Exercise – Adding Coin sound –

- import mp3 to res/raw and then:



```
// sound related variables
private SoundPool soundPool;
private int soundID;
float maxVolume;
boolean loaded = false;

@Override
protected void onCreate(Bundle savedInstanceState) {
    ...
    initializeSoundPool();
}

private void performSound() {
    // Is the sound loaded
    if (loaded) {
        soundPool.play(soundID, maxVolume, maxVolume, 1, 0, 1.0f);
        soundID = soundPool.load(this, R.raw.coin, 1); //reload file
    }
}
```

36

## Session 4 – Hands on

### Exercise – Adding background music –

- import mp3 to res/raw and then:



```
private MediaPlayer mp;
private void performStartMusic() {
    mp = MediaPlayer.create(this, R.raw.music);

    mp.setOnCompletionListener(new
        MediaPlayer.OnCompletionListener() {

        @Override
        public void onCompletion(MediaPlayer mp) {
            mp.start(); // play in a loop
        }
    });
    mp.start();
}

private void performStopMusic() {
    mp.stop();
}
```

37

## Session 4 – Hands on

### Exercise – Adding background music –

- Add menu to allow stop music:
  - In the menu file:



```
<item  
    android:id="@+id/menuToggleSound"  
    android:checkable="true"  
    android:checked="true"  
    android:title="@string/menu_toggle_sound"/>
```

- In the code:

```
case R.id.menuToggleSound:  
    toggleSound();  
    item.setChecked(!item.isChecked());  
    return true;
```

38

## Session 4 – Hands on

### Exercise – Adding vibration (game over) –

- Call function from gameEnded()
  - **Don't forget to ask Vibrate permission!**



```
private void performVibrate() {  
    // Get Vibrator from the current Context  
    Vibrator vibrator =  
        (Vibrator) getSystemService(Context.VIBRATOR_SERVICE);  
  
    // perform actual Vibrate if possible  
    if (vibrator.hasVibrator()) {  
        vibrator.vibrate(VIBRATE_LENGTH);  
    }  
}
```

## Session 5 overview

- Location
- Raw Camera
- Animations
- 2D graphics

## Session 5 – Hands on

### Exercise – Adding advanced content –



- Animation:
  - Add player animation when game over
  - Add coin animation when picked up
- Add Location logic:
  - HUD contains current location details
- Add raw camera logic:
  - Snap a picture of front camera when game over

41

# Session 5 – Hands on

## Exercise – Add Animations –

- Game over player rotate animation:



```
private int mLongAnimationDuration = 0;

@Override
protected void onCreate(Bundle savedInstanceState) {
    ...
    mLongAnimationDuration =
        getResources().getInteger(android.R.integer.config_longAnimTime);
}

private void gameEnded() {
    ...
    player.animate()
        .rotationBy(360f)
        .setDuration(mLongAnimationDuration)
        .setListener(null);
}
```

42

# Session 5 – Hands on

## Exercise – Add Animations –

- Coin pick up animation #1:



```
private void addAnimatedCoin() {  
    RelativeLayout layoutGame =  
        (RelativeLayout) findViewById(R.id.layoutGame);  
  
    ImageView coin = (ImageView) findViewById(R.id.imgCoin);  
  
    ImageView animatedCoin = new ImageView(this);  
    animatedCoin.setImageResource(R.drawable.coin);  
    animatedCoin.setX(coin.getX());  
    animatedCoin.setY(coin.getY());  
    layoutGame.addView(animatedCoin);  
  
    animatedCoin.animate()  
        .alpha(0f)  
        .setDuration(mLongAnimationDuration)  
        .setListener(null);  
}
```

43

# Session 5 – Hands on

## Exercise – Add Animations –

- Coin pick up animation #2:



```
animatedCoin.animate()
    .alpha(0f)
    .setDuration(mLongAnimationDuration)
    .setListener(new Animator.AnimatorListener() {

        @Override
        public void onAnimationStart(Animator animation) { }

        @Override
        public void onAnimationEnd(Animator animation) {
            layoutGame.removeView(animatedCoin);
        }

        @Override
        public void onAnimationCancel(Animator animation) { }

        @Override
        public void onAnimationRepeat(Animator animation) { }
    });
ABC-218
```

## Session 5 – Hands on

### Exercise – Add Location logic –

- Prepare GPS location settings
  - **Don't forget to ask manifest permission!**



```
private void prepareGPSLocation() {  
    locationManager = (LocationManager)  
        this.getSystemService(Context.LOCATION_SERVICE);  
  
    // Listener to handle incoming location updates  
    locationListener = new LocationListener() {  
        public void onLocationChanged(Location location) {  
            // Executed when a location update is received  
            newLocationReceived(location);  
        }  
        public void onStatusChanged(String provider, int status, Bundle extras) { }  
        public void onProviderEnabled(String provider) { }  
        public void onProviderDisabled(String provider) { }  
    };  
}
```

45

# Session 5 – Hands on

## Exercise – Add Location logic –

- Register for GPS location updates



```
// location variables
private LocationManager locationManager = null;
private LocationListener locationListener = null;
private String currentLocation = "No GPS data...";

private void requestGPSLocation() {
    if (handlePermissions(Manifest.permission.ACCESS_FINE_LOCATION)) {
        try {
            locationManager.requestLocationUpdates(
                LocationManager.GPS_PROVIDER,
                0, 0, locationListener);
        } catch (SecurityException e) {
            }
    }
}
```

46

# Session 5 – Hands on

## Exercise – Add Location logic –

- Handling GPS permission in Android > 6.0:



```
private boolean handlePermissions(String permissionsToCheck) {  
    // check if we have needed permission  
    int checkResult = ContextCompat.checkSelfPermission(this,  
        permissionsToCheck);  
  
    if (checkResult == PackageManager.PERMISSION_DENIED) {  
        // Do we need an explanation?  
        if (ActivityCompat.shouldShowRequestPermissionRationale(this,  
            permissionsToCheck)) {  
            // Show window with rationale  
            Toast.makeText(this, "I need permission", Toast.LENGTH_SHORT).show();  
        }  
  
        // Now ask for permission  
        ActivityCompat.requestPermissions(this, new  
            String[]{permissionsToCheck}, 123);  
        return false; // don't have permission yet. Try again later  
    }  
  
    ABC-218return true; // we have required permission, can continue  
}
```

## Session 5 – Hands on

### Exercise – Add Location logic –

- Handling GPS permission in Android > 6.0:



```
@Override  
public void onRequestPermissionsResult(int requestCode,  
                                         String permissions[],  
                                         int[] grantResults) {  
  
    if ((grantResults.length > 0) && (grantResults[0] == PackageManager.PERMISSION_GRANTED)) {  
  
        try {  
  
            locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,  
                                                    0, 0, locationListener);  
        } catch (SecurityException e) {  
  
        }  
    }  
}
```

48

# Session 5 – Hands on

## Exercise – Add Location logic –

- Parse current city from location:



```
private void newLocationReceived(Location location) {  
    double lat = location.getLatitude();  
    double lng = location.getLongitude();  
  
    Geocoder geoCoder = new Geocoder(this, Locale.getDefault());  
    StringBuilder builder = new StringBuilder();  
  
    List<Address> address = geoCoder.getFromLocation(lat, lng, 1);  
    int maxLines = address.get(0).getMaxAddressLineIndex();  
    for (int i=0; i < maxLines; i++) {  
        String addressStr = address.get(0).getAddressLine(i);  
        builder.append(addressStr);  
        builder.append(" ");  
    }  
  
    currentLocation = builder.toString();  
    updateLocationDisplay();
```

49

## Session 5 – Hands on

### Exercise – Add Location logic –

- Register for location updates:
  - Also unregister upon losing focus



```
@Override  
protected void onPause() {  
    super.onPause();  
    locationManager.removeUpdates(locationListener);  
}
```

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    ...  
    prepareGPSLocation();  
    requestGPSLocation();  
}
```

50

## Session 5 – Hands on

### Exercise – Add Location logic –

- Add location to UI:



```
<LinearLayout>
    ...
<TextView
    android:id="@+id/txtLocation"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="@string/txt_empty_location" />
</LinearLayout>
```

- Add to Strings file:

```
<string name="txt_location">Location:</string>
<string name="txt_empty_location">Location: no GPS data</string>
```

51

## Session 5 – Hands on

### Exercise – Add Location logic –

- Update UI with new location data:



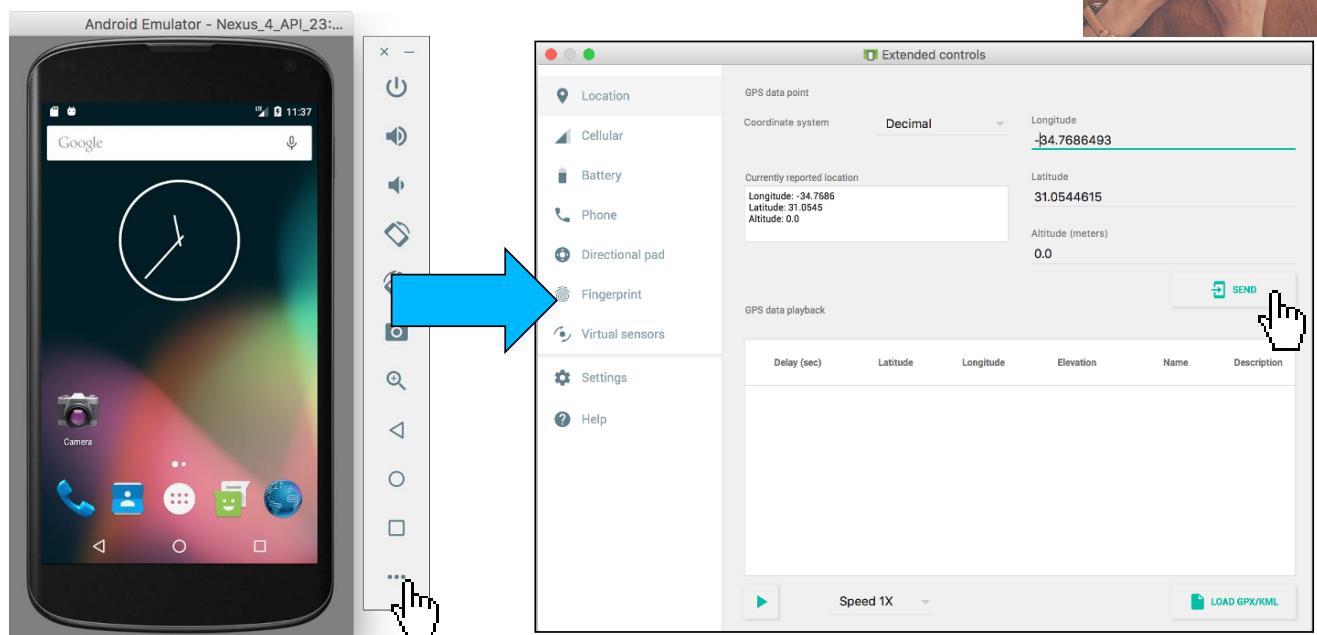
```
private void updateLocationDisplay() {  
  
    TextView txtLocation = (TextView) findViewById(R.id.txtLocation);  
  
    String message = getString(R.string.txt_location) +  
                    currentLocation;  
  
    txtLocation.setText(message);  
  
}
```

52

# Session 5 – Hands on

## Exercise – Add Location logic –

- Send fake location to Emulator #1:



53

ABC-218

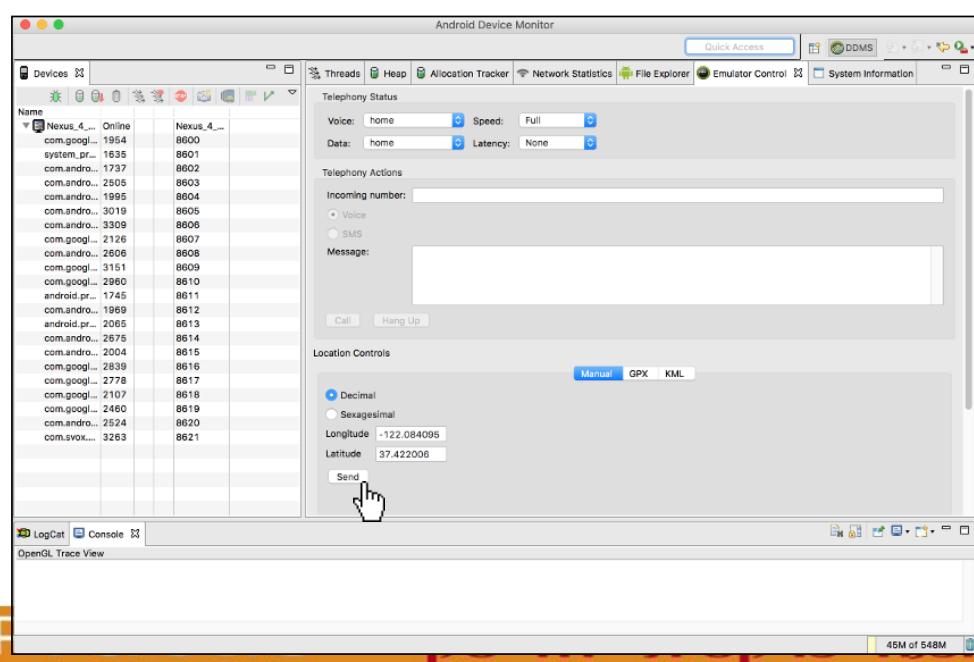
Practis

הכנה פרקטית להי-טק

# Session 5 – Hands on

## Exercise – Add Location logic –

- Send fake location to Emulator #2:
  - Android Studio: Tools -> Android -> Android Device Monitor

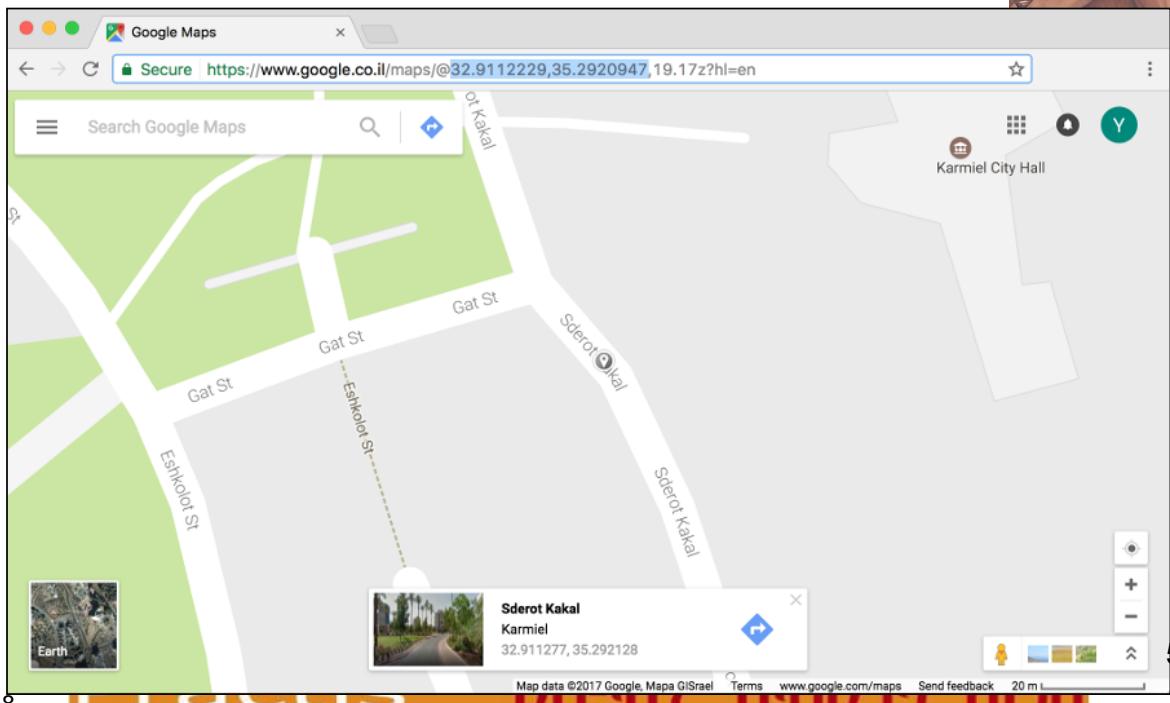


54

## Session 5 – Hands on

### Exercise – Add Location logic –

- Get test locations from Google Maps:



## Session 5 – Hands on

### Exercise – Add Raw Camera Logic –

- Take forward facing camera snapshot
  - **Don't forget to ask manifest permission!**
  - **Also add Camera permission check like in Location!**
  - **Don't forget to add camera in AVD manager**



```
Camera mCamera = null;  
ImageView snap = null;  
  
private void gameEnded() {  
    ...  
  
    // take a snapshot  
    takeCameraPicture();  
}
```

56

## Session 5 – Hands on

### Exercise – Add Raw Camera Logic –

- Take forward facing camera snapshot



```
private void takeCameraPicture() {  
  
    Camera.CameraInfo cameraInfo = new Camera.CameraInfo();  
    for (int i = 0; i < Camera.getNumberOfCameras(); i++) {  
  
        Camera.getCameraInfo(i, cameraInfo);  
        if (cameraInfo.facing == Camera.CameraInfo.CAMERA_FACING_FRONT) {  
            RelativeLayout layoutGame = (RelativeLayout) findViewById(R.id.layoutGame);  
            snap = new ImageView(this);  
            layoutGame.addView(snap);  
  
            mCamera = Camera.open(i); // get a Camera instance  
            SurfaceTexture st = new SurfaceTexture(MODE_PRIVATE);  
            mCamera.setPreviewTexture(st);  
            mCamera.startPreview();  
            mCamera.takePicture(null, null, mPicture);  
            return;  
        }  
    }  
}
```

## Session 5 – Hands on

### Exercise – Add Raw Camera Logic –

- Take forward facing camera snapshot



```
Camera.PictureCallback mPicture = new Camera.PictureCallback() {  
  
    public void onPictureTaken(byte[] data, Camera camera) {  
  
        Bitmap bitmap = BitmapFactory.decodeByteArray(data, 0, data.length);  
  
        // Set ImageView to contain picture taken  
        snap.setImageBitmap(bitmap);  
  
        // release objects to free memory  
        data = null;  
        mCamera.release();  
        mCamera = null;  
    } };
```

## Session 5 – Hands on

### Exercise – Add Raw Camera Logic –

- Take forward facing camera snapshot



```
private void startNewGame() {  
  
    ...  
    // clear snapshot  
    if (snap != null) {  
        RelativeLayout layoutGame = (RelativeLayout) findViewById(R.id.layoutGame);  
  
        snap.setImageBitmap(null);  
        layoutGame.removeView(snap);  
        snap = null;  
    }  
}
```

59

## Session 6 overview

- Using Google maps
- Monetizing Apps
- Uploading to Google Play

## Session 6 – Hands on

### Exercise – Adding Ads –



- Let's monetize our app
- Add banner ads to the about screen
- Add full screen ad in a new menu option (Interstitial)

## Session 6 – Hands on

### Exercise – Add banner ad –

- Will be shown at the bottom of About screen



62

## Session 6 – Hands on

### Usage – Using Google Play services SDK In **AndroidStudio**

Add in your build.Gradle file, under dependencies:

compile '**com.google.android.gms:play-services-ads:8.4.0**'

```
dependencies {  
    compile fileTree(dir: 'libs', include: ['*.jar'])  
    testCompile 'junit:junit:4.12'  
    compile 'com.android.support:appcompat-v7:23.2.0'  
    compile 'com.google.android.gms:play-services-ads:8.4.0'  
}
```

Click Sync now to activate changes:

Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work properly. [Sync Now](#)

## Session 6 – Hands on

### Usage – Activity Layout –

```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:ads="http://schemas.android.com/apk/res-auto"
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    tools:ignore="MergeRootFrame" >

<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/hello_world" />

    --- Ad XML code goes here ---

</RelativeLayout>
```

## Session 6 – Hands on

### Usage – Activity Layout –

```
<RelativeLayout  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_alignParentBottom="true"  
    android:layout_centerInParent="true" >  
  
    <com.google.android.gms.ads.AdView  
        android:id="@+id/adView"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:layout_centerHorizontal="true"  
        ads:adSize="BANNER"  
        ads:adUnitId="@string/banner_ad_unit_id" /> This is a test  
        Ad ID  
  
</RelativeLayout>
```

Set position  
at the bottom  
center of  
Activity

## Session 6 – Hands on

### Usage – strings.xml –

```
<?xml version="1.0" encoding="utf-8"?>  
  
<resources>  
    <string name="banner_ad_unit_id">ca-app-pub-3940256099942544/6300978111</string>  
</resources>
```

This is a just  
a test Ad ID,  
used for  
debugging

## Session 6 – Hands on

### Usage – Activity Code –

```
private AdView mAdView = null;

@Override
protected void onCreate(Bundle savedInstanceState) {
    ...
    initAd();
}

private void initAd() {
    mAdView = (AdView) findViewById(R.id.adView);
    AdRequest adRequest = new AdRequest.Builder()
        .addTestDevice(AdRequest.DEVICE_ID_EMULATOR)
        .build();

    // Start loading the ad in the background.
    mAdView.loadAd(adRequest);
}
```

67

## Session 6 – Hands on

### Usage – Activity Code –

```
public void onPause() {  
    if (mAdView != null) {  
        mAdView.pause();  
    }  
    super.onPause();  
}  
  
public void onResume() {  
    super.onResume();  
    if (mAdView != null) {  
        mAdView.resume();  
    }  
}  
  
public void onDestroy() {  
    if (mAdView != null) {  
        mAdView.destroy();  
    }  
    super.onDestroy();  
}
```

68

## Session 6 – Hands on

### Usage – AndroidManifest.xml –

- Permissions required:

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

- Under <Application> tag:

```
<meta-data android:name="com.google.android.gms.version"  
        android:value="@integer/google_play_services_version"/>
```

## Session 6 – Hands on

### Usage – AndroidManifest.xml –

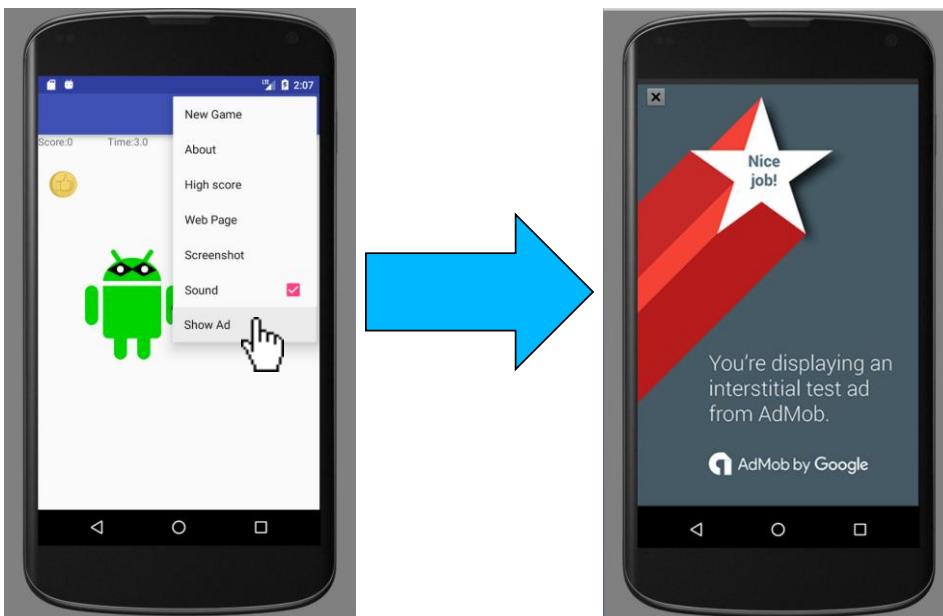
- Also Under <Application> tag:

```
<activity android:name="com.google.android.gms.ads.AdActivity"  
        android:configChanges="keyboard|keyboardHidden|orientation|screenLayo  
        ut|uiMode|screenSize|smallestScreenSize"  
        android:theme="@android:style/Theme.Translucent" />
```

## Session 6 – Hands on

### Exercise – Menu Interstitial ad –

- Will be shown after menu option clicked



71

## Session 6 – Hands on

### Exercise – Menu Interstitial ad –

- Menu file:

```
<item  
    android:id="@+id/menuShowAd"  
    android:title="@string/menu_show_ad"/>
```



- Strings file:

```
<string name="menu_show_ad">Show Ad</string>
```

# Session 6 – Hands on

## Exercise – Menu Interstitial ad –

- Actual code:



```
InterstitialAd mInterstitialAd = null;  
  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    ...  
    initializeInterstitialAd();  
}  
  
private void requestNewInterstitial() {  
    AdRequest adRequest = new AdRequest.Builder()  
        .addTestDevice("SEE_YOUR_LOGCAT_TO_GET_YOUR_DEVICE_ID")  
        .build();  
  
    mInterstitialAd.loadAd(adRequest);  
}
```

73

## Session 6 – Hands on

### Exercise – Menu Interstitial ad –

- Actual code:



```
private void initializeInterstitialAd() {  
    mInterstitialAd = new InterstitialAd(this);  
    mInterstitialAd.setAdUnitId("ca-app-pub-3940256099942544/1033173712");  
  
    mInterstitialAd.setAdListener(new AdListener() {  
        @Override  
        public void onAdClosed() {  
            requestNewInterstitial();  
  
            // your code here such as start new game;  
        }  
    });  
  
    requestNewInterstitial();  
}
```

74

## Session 6 – Hands on

### Exercise – Menu Interstitial ad –

- Actual code:



```
@Override  
public boolean onOptionsItemSelected(MenuItem item) {  
  
    switch (item.getItemId()) {  
        case R.id.menuNewGame:  
            startNewGame();  
            return true;  
  
        case R.id.menuShowAd:  
            if (mInterstitialAd.isLoaded()) {  
                mInterstitialAd.show();  
            } else {  
                Toast.makeText(this, "Not ready yet..",  
                    Toast.LENGTH_SHORT).show();  
            }  
            return true;  
    }  
}
```

75



ABC-218

Practis

For Questions:  
[yoav@practis.co.il](mailto:yoav@practis.co.il)

הנה רקיית לה-טק 76