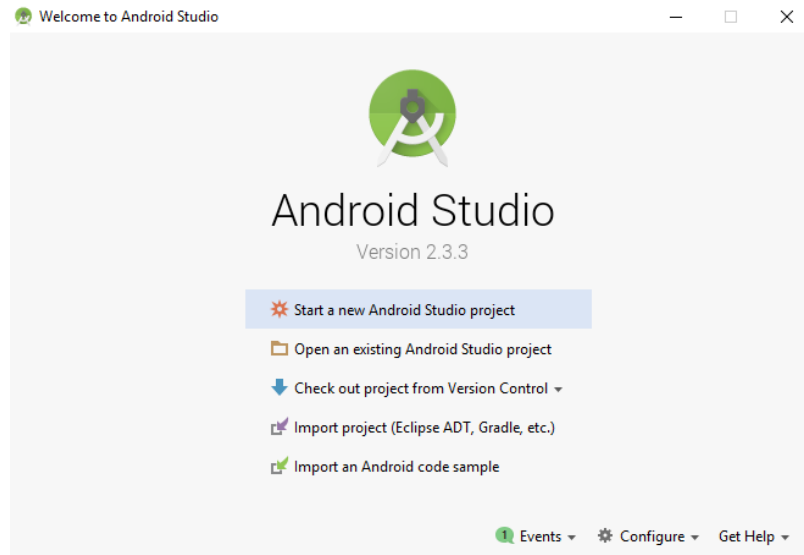


Laboration 17/18 – Android

Laboration 17a – Nytt projekt och starta emulator

Starta Android Studio och skapa projektet IntroProject.



Klicka på Start a new Android Studio project (Välj File – New - New Project... om du är redan jobbar med Android Studio)

Company Domain bör vara username.mah.se och location bör vara en katalog du skapat för att spara Android-projekt. Om du använder dator på MAH så spara på enhet M.

Klicka på *Next* två gånger, välj *Empty Activity* och klicka på *Next*

Låt defaultvärdena på *Activity Name* och *Layout Name* vara kvar och klicka på *Finish*.

Nu skapas projektet. Det tar en liten stund.

Det krävs ett litet tillägg i en fil för att projektet ska vara körbart på MAHs datorer (gäller ej om du har egen dator).

Klicka på pilen vid Gradle Scripts i projektfönstret till vänster och öppna build.gradle (Project: ...) och ändra enligt bilden nedan. När du gjort tillägget så klicka på "Try Again".



Skapa en emulator om du inte redan har en. Klicka på  (alternativt välj Tools – Android – AVD Manager). Klicka på + Create Virtual Device..., markera en emulator

Category	Name	Play Store	Size	Resolution	Density
TV	Pixel XL		5,5"	1440x2560	560dpi
Wear	Pixel		5,0"	1080x1920	xhdpi
Phone	Nexus S		4,0"	480x800	hdpi
Tablet	Nexus One		3,7"	480x800	hdpi
	Nexus 6P		5,7"	1440x2560	560dpi

Klicka på *Next* och väl sedan Nougat, API Level 25. Klicka på *Next* och sedan *Finish*.

Select a system image			
Recommended x86 Images Other Images			
Release Name	API Level	ABI	Target
Android 7.1.1 (Google APIs)	25	x86	Android 7.1.1 (Google APIs)
Android 7.0 (Google APIs)	24	x86	Android 7.0 (Google APIs)
Android 6.0 (Google APIs)	23	x86	Android 6.0 (Google APIs)

Starta emulaton genom att klicka på den gröna "start-pilen" i AVD Manager.

När Emulatorn startat så kan du stänga AVD Manager.

Laboration 17b

I föreläsningsunderlaget är det tre olika Activity-klasser, *MainActivity*, *GameActivity* och *GoalActivity*.

MainActivity skapades med projektet i Laboration 17a. Nu ska du skapa de andra två klasserna.

Markera app i projektfönstret till vänster och välj *File – New – Activity – Empty Activity*.

Activity Name:

☒ Generate Layout File

Layout Name:

☐ Launcher Activity

☒ Backwards Compatibility (AppCompat)

Package name:

Ditt användarnamn

Och klicka på *Finish*.

Upprepa för *GoalActivity*.

Nu ska filerna *GameActivity* och *GoalActivity* finnas i java-package-mappen och filerna *activity_game.xml* och *activity_goal.xml* i res-layout-mappen.

Laboration 17c

Applikationen kräver en del bilder och strängar. Placera bildfilerna i katalogen *res-drawable*. Lagg gärna till egna bilder. Tänk på att formatet ska vara .png / .jpg eller .gif och att filnamnet endast ska bestå av små bokstäver, siffror och _.

Applikationen använder en hel del strängar. Skriv in nedanstående i *strings.html*.

```
<resources>
  <string name="app_name">Location game</string>
  <string name="main_choose_game">Välj spel</string>
  <string name="main_btnStart">Starta spel</string>
  <string name="main_badPlace">Felaktig plats</string>
  <string name="game_goal">Gå till:</string>
  <string name="game_latitude">Latitude</string>
  <string name="game_longitude">Longitude</string>
  <string name="game_your_location">Här är du:</string>
  <string name="game_distance">Avstånd till målet:</string>
  <string name="goal_qoute_title">Ett citat:</string>
</resources>
```

Laboration 17d

Nu ska du lägga till element i *activity_main.xml*, *activity_game.xml* och *activity_goal.xml* så att de ungefär ser ut som i föreläsningsunderlaget. Det får vara hur snygga som helst men ska innehålla följande:

activity_main.xml

Ska innehålla minst en *TextView*, en *EditText*, en *Button* och en *ImageView*. Ge *EditText* och *Button* vettiga id (gärna även övriga komponenter).

activity_game.xml

Ska innehålla minst sju *TextView* och en *ViewGroup* (*LinearLayout*, bakgrundsfärgen ska

ändras). Ge TextView-komponenterna som ska visa latitude/longitud vettiga id. Det samma gäller ViewGroup-komponenten.

activity_goal.xml

Ska innehålla minst en TextView och en ImageView. Ge båda komponenterna vettiga id.

Laboration 18a

Applikationen innehåller ett par hjälp-klasser, *Database* och *GameLocation*. Kopiera dessa till din källkodskatalog (där *MainActivity* är).

Dags att skriva kod för *MainActivity*.

Klassen ska ha två instansvariabler och i *onCreate* ska dessa kopplas mot komponenter i UIet. Dessutom ska en lyssnare registreras för knappen.

```
public class MainActivity extends AppCompatActivity {
    private EditText etChooseGame;
    private Button btnStartGame;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        initializeComponents();
        registerListeners();
    }

    private void initializeComponents() {
        etChooseGame = (EditText)findViewById(R.id.etChooseGame);
        btnStartGame = (Button)findViewById(R.id.btStartGame);
    }

    private void registerListeners() {
        btnStartGame.setOnClickListener(new ButtonListener());
    }
}
```

Lyssnarklassen *ButtonListener* skrivs som en inre klass i *MainActivity*. När användaren klickar på knappen ska spelet startas (eller felmeddelande visas). Detta sköts av metoden *startGame*.

```
private class ButtonListener implements View.OnClickListener {
    public void onClick(View v) {
        String game = etChooseGame.getText().toString().toUpperCase();
        if(game.equals("MAH") || game.equals("NFU") || game.equals("OSC"))
            startGame(game);
        else
            Toast.makeText(MainActivity.this, R.string.main_badPlace, Toast.LENGTH_SHORT).show();
    }

    private void startGame(String game) {
        Intent intent = new Intent(this, GameActivity.class);
        intent.putExtra("game", game);
        this.startActivity(intent);
    }
}
```

Testa att köra appen. Om du matar in mah eller nfu i EditText-komponenten kommer *GameActivity* att startas.

Laboration 18b

Dags att skriva kod för *GameActivity* och den blir ganska omfattande. Varje block av kod kommenteras kortfattat. Får du inte rätt på koden kan du kopiera motsvarande kod från föreläsningsmaterialet.

Filen AndroidManifest (i app-katalogen) måste innehålla ett par permissions för att appen ska få använda mobilens GPS (avläsa mobilens position). Lägg till nedanstående rader.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="se.mah.tsroax.introproject">

    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Location game"
        android:roundIcon="@mipmap/ic_launcher_round"
        >
```

Det behövs många instansvariabler i klassen. Bl.a. 5 st av typen TextView och en av typen ViewGroup (LinearLayout). Övriga instansvariabler är kortfattat kommenterade.

```
public class GameActivity extends AppCompatActivity {
    private final int REQUEST_ACCESS_FINE_LOCATION = 1; // Konstant vid förfrågan om permisson
    private final int GOAL_REQUEST = 1; // Konstant vid retur från Activity
    private ViewGroup viewGroup;
    private TextView tvGoalLatitude;
    private TextView tvGoalLongitude;
    private TextView tvPlayerLatitude;
    private TextView tvPlayerLongitude;
    private TextView tvGoalDistance;
    private LocationManager locationManager; // requestLocationUpdates
    private LocationListener listener; // location updates
    private GameLocation[] locations; // locations to find
    private GameLocation gameLocation; // Current location to find
    private Database db = new Database(); // Lagrar ett par arrayer med GameLocation-objekt
    private int[] imageResources = {R.drawable.pokemon1, R.drawable.pokemon2, R.drawable.pokemon3};
    private int currentIndex = 0; // index för aktuell position + bild som ska visas
}
```

Då *onCreate()* exekveras så anropas först superklassens *onCreate()* och därefter skapas alla objekten som tillhör UIet vid anrop till metoden *setContentView()*. UIet beskrivs av *activity_game.xml*.

initializeComponents() ger instansvariablerna av typen TextView och ViewGroup referenser till motsvarande objekt i UIet.

getGameLocations() avläser ur Intent-instansen, vilken startade Activityn, värdet på "game". Värdet används som argument vid anrop till *getGameLocations()* (i Database). *locations* tilldelas en array med GameLocation-objekt och *gameLocation* det första GameLocation-objektet..

Det sista som händer i *onCreate()* är att

- *locationManager* tilldelas referens till ett objekt av typen LocationManager. Detta objekt kan sedan användas för att erhålla GPS-koordinater från mobilen.
- *listener* tilldelas referens till LocationListener-implementering. Metoden *onLocationChanged()* kommer att anropas av systemet då GPSen registrerat en ny position.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_game);
    initializeComponents();
    getGameLocations();

    locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
    listener = new Listener();
}

private void getGameLocations() {
    Intent gameIntent = getIntent();
    String game = gameIntent.getStringExtra("game");
    locations = db.getGameLocations(game);
    if (locations.length > 0) {
        gameLocation = locations[0];
    }
}

private void initializeComponents() {
    viewGroup = (ViewGroup) findViewById(R.id.activity_game);
    tvGoalLatitude = (TextView) findViewById(R.id.tvGoalLatitude);
    tvGoalLongitude = (TextView) findViewById(R.id.tvGoalLongitude);
    tvPlayerLatitude = (TextView) findViewById(R.id.tvPlayerLatitude);
    tvPlayerLongitude = (TextView) findViewById(R.id.tvPlayerLongitude);
    tvGoalDistance = (TextView) findViewById(R.id.tvGoalDistance);
}
```

Metoden `onResume()` anropas precis innan Ulet blir synligt. I metoden anropas metoderna:

`super.onResume()` – superklassens `onResume()`-metod

`updateDisplay()` – Positioner som man ska gå till uppdateras på skärmen

Sedan kontrolleras att appen har tillstånd att använda GPS-en.

Om så inte är fallet görs en förfrågan om tillstånd till användaren genom anrop till

`ActivityCompat.requestPermissions(...)`;

Resultatet av förfrågan kommer via anrop till metoden `onRequestPermissionsResult()` (kommenteras kort nedan).

Om så är fallet så meddelas `LocationManager` att appen vill ha meddelande om position av GPS-en. Uppdatering av position kommer ske genom anrop till metoden `onLocationChanged()` i `LocationListener`-implementeringen vilken `listener` refererar till.

Metoden `onPause()` anropas då activityn ej är synlig längre. I metoden meddelas `LocationManager` att appen inte vill ha fler meddelande om position av GPSen.

```
public void updateDisplay() {
    if (gameLocation != null) {
        tvGoalLatitude.setText(String.valueOf(gameLocation.getLatitude()));
        tvGoalLongitude.setText(String.valueOf(gameLocation.getLongitude()));
    }
}

@Override
protected void onResume() {
    super.onResume();
    updateDisplay();
    if (ContextCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) == PackageManager.PERMISSION_DENIED) {
        ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.ACCESS_FINE_LOCATION}, REQUEST_ACCESS_FINE_LOCATION);
    } else {
        locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 1000, 0, listener);
        locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 1000, 0, listener);
    }
}

@Override
protected void onPause() {
    super.onPause();
    if (ContextCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) == PackageManager.PERMISSION_GRANTED) {
        locationManager.removeUpdates(listener);
    }
}
```


Metoden *onRequestPermissionsResult()* anropas av systemet för att meddela om användaren har samtyckt till att GPSen används eller ej.

```
// callback requestPermissions()
public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {
    switch (requestCode) {
        case REQUEST_ACCESS_FINE_LOCATION:
            if (ContextCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) == PackageManager.PERMISSION_GRANTED) {
                locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 1000, 0, listener);
                locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 1000, 0, listener);
            }
            break;
    }
}
```

Om en annan Activity startas från denna Activity genom anrop till metoden *startActivityForResult()* så kommer metoden *onActivityResult()* anropas då den startade Activity är färdig (avslutas).

```
// callback startActivityForResult()
public void onActivityResult(int requestCode, int resultCode, Intent data) {
    currentIndex++;
    if (currentIndex < locations.length) {
        gameLocation = locations[currentIndex];
        updateDisplay();
    } else {
        finish();
    }
}
```

Kvar är klassen *Listener* vilken är modell för att erhålla GPS-uppdateringar. Klassen implementerar *LocationListener* vilket innebär att metoderna *onLocationChanged(Location)*, *onStatusChanged(...)*, *onProviderEnabled(...)* och *onProviderDisabled(...)* måste implementeras i klassen.

Så fort GPSen har uppdatering om position så anropas *onLocationChanged(Location location)* och det är i denna metod det mesta av spellogiken är.

1. Uppdatera *TextViews* som visar var användaren är just nu
2. Beräkna avståndet till mål-positionen och visa avståndet i en *Textview*.
3. Ändra bakgrundsfärgen på skärmen
4. Om avståndet till mål-positionen är mindre än 10 m så starta *GoalActivity*. Detta sker genom anrop till *startActivityForResult()*.

```
// LocationListener - callback interface for LocationManager
private class Listener implements LocationListener {
    public void onLocationChanged(Location location) {
        int redBlue, color;
        tvPlayerLatitude.setText(String.valueOf(location.getLatitude()));
        tvPlayerLongitude.setText(String.valueOf(location.getLongitude()));
        if (gameLocation != null) {
            double distance = location.distanceTo(gameLocation.getLocation());
            tvGoalDistance.setText(String.format("%1d m", (int) distance));
            redBlue = Math.min((int)distance, 255);
            color = Color.rgb(redBlue, 255-redBlue, 0);
            viewGroup.setBackgroundColor(color);

            if (distance < 10) {
                Intent intent = new Intent(GameActivity.this, GoalActivity.class);
                intent.putExtra("name", gameLocation.getContent());
                intent.putExtra("image", imageResources[currentIndex]);
                startActivityForResult(intent, GOAL_REQUEST);
            }
        }
    }

    public void onStatusChanged(String provider, int status, Bundle extras) {}

    public void onProviderEnabled(String provider) {}

    public void onProviderDisabled(String provider) {}
}
```

Du kan testköra appen i emulatorn. Men för att ange en position krävs att du klickar på tre prickar till höger om emulatorn och skriver in

Laboration 18c

Kvar är att skriva kod i GoalActivity.

Det är endast kod i GoalActivity, och förutom anrop till *super.onCreate()* och till *setContentView()*, så:

- *tvMessage* respektive *ivGoal* tilldelas referens till motsvarande objekt i UIet.
- Data som skickat, från Activityn som startade denna Activity, avläses ("name" och "image"). Dessa värde används för att visa "name" i TextView-komponenten och bilden i ImageView-komponenten.

```
public class GoalActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_goal);
        TextView tvMessage = (TextView)findViewById(R.id.tvMessage);
        ImageView ivGoal = (ImageView)findViewById(R.id.ivGoal);
        Intent intent = getIntent();
        tvMessage.setText(intent.getStringExtra("name"));
        ivGoal.setImageResource(intent.getIntExtra("image", 0));
    }

    @Override
    protected void onPause() { super.onPause(); }

    @Override
    protected void onResume() { super.onResume(); }
}
```


Laboration 18d

Nu är det dags att testköra appen.

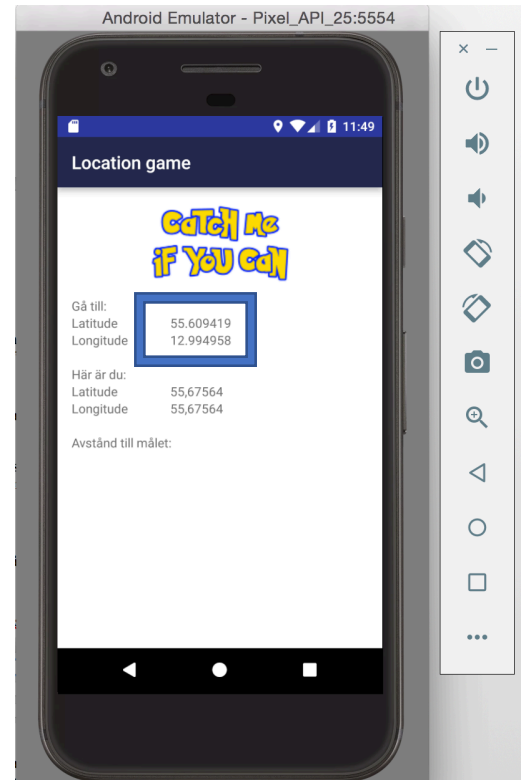
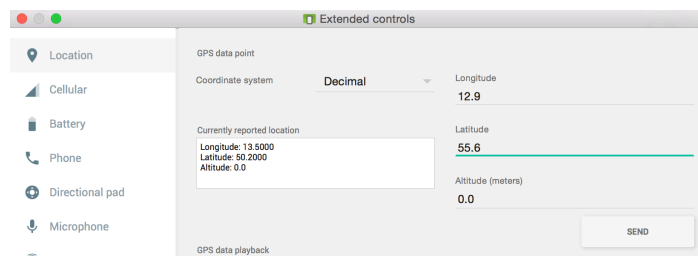
Om du har egen android-enhet som du kör appen på, och befinner dig på Malmö högskola, kan du skriva *mah* i EditText-komponenten och klicka på knappen ”Starta spel”. I närheten av Niagara ska tre positioner finnas.

Om du kör appen i emulatorn så startar du med att skriva in mah i EditText-komponenten och klickar på ”Starta spel”. Nu måste du simulera positionen som mobilen befinner sig på. Klicka på ... till höger om emulatorn och välj Location. Här har du möjlighet att ange din position genom att mata in värde efter Longitude och Latitude och därefter klicka på ”SEND”.

I emulatorn ser du pokemons position (Gå till).

Gör så här:

Mata in pokemons position (Longitude och Latitude) med en decimal och klicka på SEND.



Nu ändras avståndet från mobilens position till pokemons position. Upprepa ovanstående men lägg till en decimal före varje klick på SEND. När du anger en position tillräckligt nära pokemonen kommer denne att visa sig i appen.

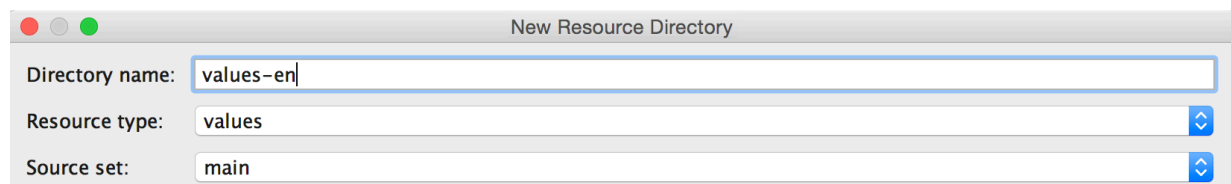
Laboration 18f

Internationaliser appen genom att även medge engelska som språkval. Gör följande:

Skapa katalogen *values-en* i *res*-katalogen.

Högerklicka res och väl *New – New Resource Directory*

I dialogen som visas ska *Directory name* vara *values-en* och *Resource type* vara *values*




Klicka sedan på OK. Katalogen dyker inte upp i res.

I katalog *values-en* ska du placera en kopia av *strings.xml*. Sedan ska du ändra de svenska strängarna till motsvarande engelska. Om du i din mobil / i emulatorn väljer engelska så ska den nya uppsättningen strängar användas.

Kopiera *strings.xml*, högerklicka *values* och klistra in *strings.xml*. Du får en förfrågan om vilken katalog *strings.xml* ska vara i och anger *values-en*.

Copy file /Users/tsroax/Andr...troProjectNFU/app/src/main/res/values/strings.xml

New name:

To directory: 

Kör din app men ändra språkinställningen till engelska. Ändra tillbaka till svenska. Ändra till norska.

Laboration 18g

Lägg till egna positioner i **Database** under något namn. Se till att du har minst lika många bilder i arrayn *imageResources* (GameActivity) som du har GameLocation-objekt i arrayen *gameLocations* (GameActivity, arrayen kommer från Database)

Utforska

Gå till www.android.com. Du är nu på Androids hemsida med massor av resurser. Som utvecklare av programvara är det intressant med material för utvecklare.

Klicka på *Developers* (längst ner på sidan). Nya möjligheter varav *Develop* just nu är intressantast.

Klicka på *Develop*. Nu har du kommit till en sida med tutorials (Training, API Guides), API-dokumentation (Reference) och diverse utvecklingsverktyg (Tools).

Utforska innehållet!