

OPPIMISPÄIVÄKIRJA

Merja Partanen

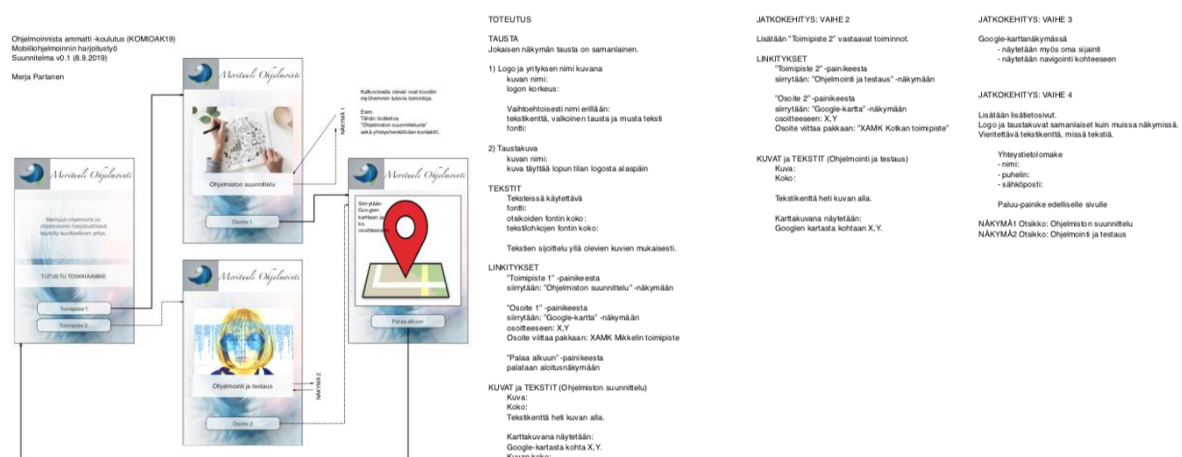
ALKUTILANNE

Koulutus starttasi kahdella lähipäivällä, minkä jälkeen saatiin tehtäväksi suunnitella ja toteuttaa oma mobiiliohjelmointiprojekti. Lähipäivien aikana tutustuin ensimmäisen kerran Android studio -ympäristöön, sekä ohjelmoinnissa käytettävään Java-ohjelmointikieleen. iOS-laitteiden käyttäjänä, myös Android laitteen toiminta emuloinnin kautta tuli uutena asiana.

Lähipäivillä käytössäni on ollut kannettavatietokone, missä käyttöjärjestelmänä Windows 10. Varsinaisen projektin tein Mac Mini –koneella millä työskentelyolosuhteet olivat huomattavasti paremmat. Projektin suunnitelman toteutin **Visual Designer** –sovelluksella ja muistikirjanani toimi **Quiver** –sovellus, mihin pystyy helposti tallentamaan koodia sekä muuta ohjeistusta. Tämä oppimispäiväkirja muodostuu näistä projektin toteutuksen aikana muistiin kirjoitetuista ohjeista.

PROJEKTIN AIHE JA PAPERIPROTO

Projektin aiheeksi valitsin kuvitteellisen yrityksen, mille tulisi yritystä esittelevä sovellus sisältäen myös Google-kartan käytön. Karttatoiminnot koin sen verran haastavaksi toteuttaa, että jaoin toteutuksen useampaan osaan. Alkaen kiinteiden koordinaattien mukaan näytettävistä kohteista edeten oman sijainnin näyttämiseen, sekä lopulta navigoimiseen kohteeseen. Valitsin ja latsin jo suunnitteluvaiheessa, sovellukseen käyttöön tulevat kuvat palvelusta www.pixabay.com.



Paperiproto suunnittelu tapahtui kohtuullisen helposti loogisesti, että teknisesti.

OHJELMOINNISSA ALKUUN PÄÄSEMINEN

Itse mobiiliohjelmoinnin aloittaminen tapahtui lataamalla ja asentamalla sovellus Android Studio (for Mac), koneelle osoitteesta: <https://developer.android.com/studio>.

Parin ensimmäisen viikon aikana aloitin useamman projektin, yritin hahmottaa sovellusympäristön toimintaa sekä itse ohjelmointia että koodia. Googlasin hakusanoilla "Android studio ...". Keräsin löytämiäni linkkejä talteen. Yritin saada otetta mobiiliohjelmoinnista siinä kuitenkin kovinkaan hyvin onnistumatta.

Täältä aloitin

<https://developer.android.com/training/basics/firstapp/creating-project>

ja tämän kohdalle olin kirjoittanut, että täältä löytyy lähes kaikki=)

<https://www.tutorialspoint.com/android>

Linkkejä tuli kerättyä paljon lähinnä kirjoitettuja ohjeita.

Latasin tunnilla tehdyt esimerkit. Suoritin niitä, tein pieniä muutoksia, sekä yritin ymmärtää. Edelleenkin en oikein tiennyt mistä päin oikein pitäisi aloittaa oman projektin toteutus. Se etten ollut ennen tätä projektia, tutustunut Java-koodiin tai ohjelmointiin, toi mukanaan oman haasteensa.

Hiljalleen projektien kansio- ja tiedostorakenne alkoi hahmottumaan:

- **activity_main.xml** -tiedosto sisältää koodin sovelluksen ulkoasusta
- **MainActivity.java** -sisältää toimintaan vaikuttavan koodin
- **AndroidManifest.xml** -tiedostossa esitellään kaikki käytettävissä olevat aktiviteetit

Lopulta löysin video-ohjeita, joiden avulla pääsin alkuun oman projektin toteutuksen suhteen. YouTube ohjeita oli kohtuullisen helppo etsiä sen jälkeen, kun tiesi mitä haluaa koodin suorittavan.

Moodlen esimerkeissä on hyvin tuotu esille eri toimintojen koodi. Näin koodaamisen alkuvaiheessa en niitä kuitenkaan osannut vielä hyödyntää.

Projektin toteutuksessa tuon esille miten sain projektin koodin aikaan; asioiden syvällisempään oppimiseen ja sisäistämiseen menee vielä jonkin verran aikaa.

Työssä käytettyjen kuvien lähdetiedot:

| <i>Kuvien alkuperäiset nimet</i> | <i>kuvaaja/piirtäjä</i> |
|---|------------------------------|
| Taustakuva: feather-3010848_1920.jpg | Sponchia: Christine Sponchia |
| Logo: logo-1933884_640.png | ptr: Petra |
| Suunnittelu: hand-3190204_640.jpg | rawpixel |
| Ohjelmointi ja testaus: woman-3600557_640.jpg | geralt: Gerd Altmann |
| Kartta: location-162102_640.png (tämä vain paperiprotossa) | OpenClipart-Vectors |

PROJEKTIN TOTEUTUS

Tästä se sitten alkoi, oman sovelluksen rakentaminen.

Android Studio –sovellus käyntiin ja uusi pohja esille:

“Empty Activity”, *Language: “Java”* ja *Minimun API level: “API 29 Android 10.0 Q”*

Nämä asetukset olivat tuttuja harjoitustöistä. Mikäli projekti tähtäisi todelliseen tuotantoon, niin olisi kannattanut valita jokin aikaisempi API-versio, mitä tukevia Androidi-laitteita olisi enemmän käytössä.

Lähdin liikkeelle kuvatiedostoista. Ensimmäisenä pyrin selvittämään missä muodossa ja minkä kokoisina kuvat kannattaa sovellukseen tuoda. Kokeilin kuvien muuttamiseen: **NativeScript Image Builder** - sivustoa osoitteesta <http://nsimage.brosteins.com/> .

Kopioin koneeltani png-muotoiset kuvat Android Studion projektin kansioon: **app/res/drawable/**
Kuvien nimet: *tausta.png*, *logo.png*, *suunnittelu.png* ja *ohjelmointi.png*

Taustakuvan lisääminen

Taustakuvan määrittelin tiedostossa: **app/res/layout/activity_main.xml**
komennolla **android:background="@drawable/tausta"**

Apuna oli video-ohje taustakuvan lisäämiseen:

<https://www.youtube.com/watch?v=7OvsWwbvYsM>

Logon lisääminen

Logon olin muodostanut Visual Designer-sovelluksessa, lisäämällä “Merituuli Ohjelmointi” -tekstin logon oikealle puolelle. Näin sain elementin mikä olisi helposti vaihdettavissa.

Lisäsin kuvan tiedostoon: **app/res/layout/activity_main.xml**

Valitsin **Palette -> Common -> ImageView** ja edelleen **Drawable -> Project -> logo**

Kiinnitin reunapisteistä kuvan näkymän yläreunaan. Koska kuvalle varattu alue oli oletuksena suurempi kuin kuva, niin alueen kulmapisteitä sain alueen koon muutettua kuvan kokoiseksi.

activity_main.xml koodia

```
<ImageView
    android:id="@+id/igLogo"
    android:layout_width="409dp"
    android:layout_height="89dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/logo"/>
```



Lisää näkymiä

Taustakuvan ja logon lisäämisen jälkeen lisäsin suunnitelman mukaiset lisänäkyvät: "Suunnittelu", "Ohjelmointi" ja "Kartta".

Tämän toteutusta pohdiskelin, googlasin ja luin esimerkkejä, miten toteuttaisin tai miten ylipäätään toteutetaan. Onnistuisiko pääohjelmassa piilottamalla ja tuomalla näkyviin aktiviteetteja. Lopulta löysin seuraavan video-ohjeen, minkä mukaan lähdin etenemään uuden aktiviteetin muodostamiseen ja aktiviteetista toiseen siirtymiseen: <https://www.youtube.com/watch?v=bglUdb-7Rqo>

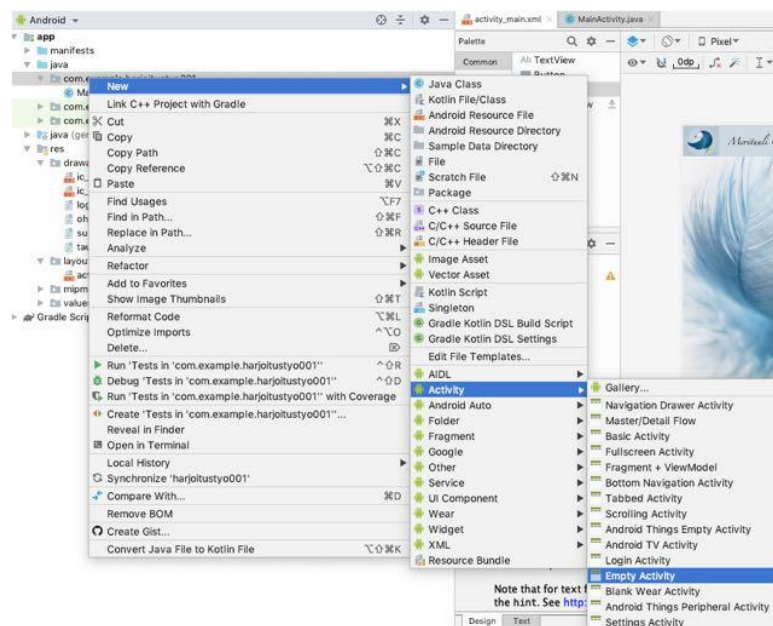
Uuden näkymän eli aktiviteetin lisääminen

Etsin näkyville: **app/java/com.example.harjoitustyo001/** tästä sain hiiren oikealla valikon esille mistä valitsin **New -> Activity -> Empty Activity**

ActivityName-kenttään tuli "toisen näkymän" nimi eli "**Suunnittelu**".

Samalla tavalla toteutin myös kolmannen aktiviteetti nimeltään "**Ohjelmointi**".

Neljännän aktiviteetin eli **Google-karttapohjan** muodostin seuraavan polun mukaisesti: New -> Google-> Google Maps Activity ja nimesin sen nimellä "**Sijainti**".



Tämän jälkeen lisäsin taustakuvan ja logon "Suunnittelu" ja "Ohjelmointi" -näkyymiin, samalla koodilla mitä aloitusnäkylässä. "Sijainti" -näkymään jätin alkuvaiheessa oletusasetuksille. Myöhemmin sitten täydensin myös sinne taustan ja logon.

Tekstien, painikkeiden ja kuvien lisääminen

Kuvien, tekstien ja painikkeiden lisääminen olikin jo tuttua asiaa lähitunneilta. Käsittelin kaikki "näkymät" erikseen ja tein niihin samalla kaavalla paikat kuville, teksteille ja painikkeille.

Useamman rivin tekstin lisääminen halutulla tavalla, tuottikin hieman päänvaivaa. Jossain vaiheessa tietoisuuteen tuli tiedosto `app/res/values/strings.xml`. Tuolla määritellään tekstimuuttujien arvoja, ja lopulta tutulla \n -merkinnällä rivinvaihdon sain onnistumaan.

Näyttää tältä: `<string name="merituuli">\nMerituuli Ohjelmointi on\nkuvitteellinen yritys\nmitä käytän ohjelmoinnin\nharjoituksissa.\n</string>`

Kuvan lisääminen tapahtuu samalla tavalla kuin logon lisääminen.

`app/res/layout/activity_main.xml` -tiedostossa

Palette - Common – ImageView, raahataan näkymään ja valitaan **Drawable - Project – suunnittelu.png**.

Tekstikentän lisääminen

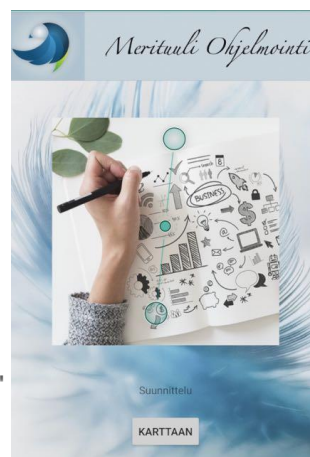
Palette - Common – TextView, raahataan ja sovitellaan näkymään.

Painikkeiden lisääminen

Palette - Common – Button, raahataan ja sovitellaan näkymään.

Tekstin ja painikkeen määrittely `Activity_suunnittelu.xml` tiedostossa:

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Suunnittelu"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="true"/>
<Button
    android:id="@+id/button3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="140dp"
    android:layout_marginTop="512dp"
    android:text="Karttaan"
    app:layout_constraintBottom_toTopOf="@+id/button3"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```



Jokaiselle näkymään lisättyyn kohteeseen määritellään yksilöllinen id, mihin sitten koodissa tullaan myöhemmässä vaiheessa viittaamaan. Tässä esimerkissä tämä id-koodi oli **button3**.

Painikkeiden toiminnan määrittely

Kun kuvat, tekstit ja painikkeet oli määriteltä "näkyymiin" alkoi seuraava haaste; miten toteutetaan siirtyminen aktiviteetista toiseen.

Muistissa oli mielikuva painikkeesta ja onClick -kentästä Android Studio oikeassa reunassa ja että toiminta määriteltiin MainActivity-tiedostossa. Tämä oli jo hyvä lähtökohta. Joten etsimään sopivaa ohjetta. Kahlasin ensin lukuisia tekstimuotoisia ohjeita lävitse, sen jälkeen löysin seuraavanlaisen video-ohjeen, mistä jo aikaisemminkin mainitsin: <https://www.youtube.com/watch?v=bglUdb-7Rqo> ei välttämättä paras ohje, mutta eteenpäin tällä pääsin.

Esimerkkinä ensimmäisen näkymän painikkeen toiminnan lisäämisestä: **app/java/com.example.harjoitustyo001/MainActivity** -tiedostossa.

```
private Button button;

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

    button = findViewById(R.id.button);
    button.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            opensuunnittelu();
        }
    });
}

public void opensuunnittelu(){
    Intent intent = new Intent(this, Suunnittelu.class);
    startActivity(intent);
}
```

Toiminta menee jotenkuten näin: "Määritellään paikallinen painike nimeltään **button**. Kuunnellaan buttonia, ja kun sitä on klikattu, niin siirrytään suorittamaan **opensuunnittelu** kohtaa ja sieltä siirrytään näkymään **Suunnittelu**."

Tämän jälkeen määrittelin painikkeiden toiminnat myös "Suunnittelu" ja "Ohjelmointi" -näkyymiin. "Sijainti" -näkymään toiminnot lisäsin myöhemmin.

Google-kartan käyttöönotto

Ilman tunnilla käytyä esimerkkiä olisin tässä vaiheessa ollut täysin pihalla, tai ainakin työn eteneminen olisi hidastunut. Luultavasti en olisi huomannut toteuttaa API-avaimen hakemista, vaikka se tuolla automaattisesti muodostetussa koodissa mainitaan.

Alussa jo mainitsin, että Google-karttapohjan muodostin seuraavan polun mukaisesti: New -> Google-> Google Maps Activity ja nimesin sen nimellä **"Sijainti"**. Tämä toiminto loi automaattisesti myös tiedoston `app/res/values/google_maps_api.xml`

Tiedostossa oli teksti: *"TODO: Before you run your application, you need a Google Maps API key."* Joten ei muuta kuin API-koodia muodostamaan.

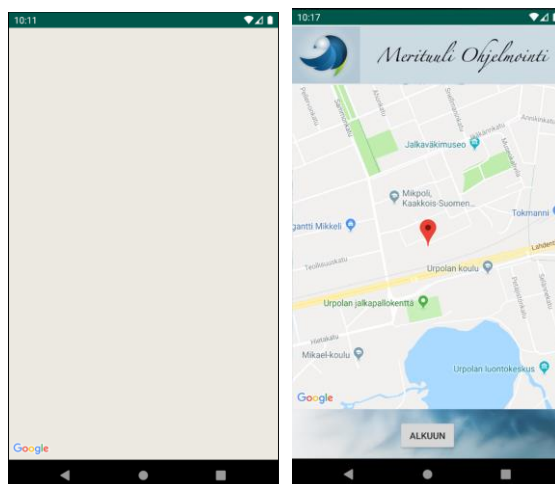
Kirjauduin Gmail-tunnuksella koodissa olevaan linkkiin ja pääsin luomaan uuteen projektiin oman "API key" -avaimen. Kopioin saadun avaimen ja liitin sen samaisen tiedoston loppuun.

```
<string name="google_maps_key" templateMergeStrategy="preserve"
translatable="false">AlzaSyC4Ytässäesimerkkiavaindznuku5Y</string>
```

Samainen koodi näkyy myös `AndroidManifest.xml` -tiedostossa kohdassa:

```
<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="@string/google_maps_key" />
```

Ilman API-avainta google-karttapohja avautuu emulaattoriin ilman karttaa; alla olevista kuvista vasemmalla.



Oikealla oleva kuva on jo lähes valmiista työstä, missä näkyy kartta ja, seuraavana esittelyyn tuleva, koordinaatti punaisena pisteinä.

Kartta ja koordinaattien välitys edeltävistä näkymistä

Seuraava haaste oli miten saada google-karttaan toimipisteiden sijainnit näkyville. Karttaan tullaa kahdesta eri näkymästä, ja näytettävillä pistellä on eri koordinaatit. Muuttujien avulla, kyllä, mutta miten se toteutetaan koodissa. Tästä videosta poimin esille koordinaattien käytön: <https://www.youtube.com/watch?v=qS1E-Vrk60E> ja sen jälkeen vain soveltamaan omaan koodiin.

Tiedostossa app/java/com.example.harjoitustyo001/**Sijainti** lisäsin aluksi koordinaatit numeroilla oletuskoodiin. Tämän jälkeen muutin koordinaatit muuttujiksi x1 ja y1.

```
@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;
    double x1 = 61.6790134;
    double y1 = 27.2572596;

    // Add a marker in "merituuliohjelmointi" and move the camera
    LatLng placeName = new LatLng(x1, y1);
    mMap.addMarker(new MarkerOptions().position(placeName).title("Merituuli Ohjelmointi"));
    mMap.moveCamera(CameraUpdateFactory.newLatLng(placeName));
}
```

Seuraavana selvitin, miten saan koordinaatit siirrettyä edeltävästä aktiviteetista. Ohjetta löysin täältä: <https://www.youtube.com/watch?v=p7BVdP0ljWo>.

Määrittelin "Suunnittelutoimiston" koordinaatin tiedostossa app/java/com.example.harjoitustyo001/**Suunnittelu**.

```
public void openkartta(){
    Intent intent = new Intent(this, Sijainti.class);
    startActivity(intent);
    double x1 = 61.6795217;
    double y1 = 27.2591599;
    Intent passdata_intent = new Intent(this, Sijainti.class);
    startActivity(passdata_intent);
    passdata_intent.putExtra("x1",x1);
    passdata_intent.putExtra("y1",y1);
    startActivity(passdata_intent);
}
```

Tiedostossa app/java/com.example.harjoitustyo001/**Sijainti** otetaan muuttujien sisältö vastaan ja käytetään sijainnin näyttämiseen.

```
@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;

    Bundle bundle = getIntent().getExtras();
    double x1 = bundle.getDouble("x1");
    double y1 = bundle.getDouble("y1");
    // Add a marker in "merituuliohjelmointi" and move the camera
    LatLng placeName = new LatLng(x1, y1);
    mMap.addMarker(new MarkerOptions().position(placeName).title("Merituuli Ohjelmointi"));
    mMap.moveCamera(CameraUpdateFactory.newLatLng(placeName));
    mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(placeName,15));
}
```


Koordinaattipisteiden jälkeen selvittelin koodiin "kameran" asetukset, eli miten saadaan karttakuva niin lähelle, että kadut näkyvät. Tämä toiminto saatiin aikaan tällä rivillä: `mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(placeName,15));`

Numero 15 määrittelee sen, kuinka korkealta kohdetta katsotaan. Sopivan epälooginen; mitä pienempi luku sen kauempaa katsotaan, ja mitä isompi niin sitä lähempänä maanpintaa ollaan.

LOPUKSI

Aikataulullisesti tämän projektin työstäminen tulee jatkumaan opiskelujakson ulkopuolella tästä eteenpäin. Seuraavat askeleet ovat oman sijainnin selvittäminen ja miten navigoidaan haluttuun koordinaattiin.

Miten projekti mielestäni meni. Alku oli haastava, kun kokonaiskuva ei ollut tarkka. Pari viikkoa meni kuin olisi pyörinyt kuumen nuotion ympärillä. Tietoa löytyi netistä, mutta mikä siitä tietomäärästä olisi sitä millä pääsee alkuun. Kaikkea ei voi käydä lävitse, mikä on oleellista, miten koodi muodostuu, mihin tiedostoon mitäkin pitäisi kirjoittaa.

Haasteet on tehty voitettaviksi; alkuun kun lopulta pääsi oli hieno tunne, kun oivalsi selviävänsä eteenpäin. Vaikka vasta pienen pieniä vauvanaskeleita mobiiliohjelmoinnissa on otettu, niin silti luottamus oppimiseen on olemassa.

Tiivistettynä: 1) tiedä mitä haluat, 2) etsi ohje ja 3) toteuta se!

Näin kirjoitin tänään 10.10.2019.

-Merja-