

Tietovisa  
Mobiiliohjelmointi

Jani Sillanpää

KURSSITYÖ

Elokuu 2021

Tieto- ja viestintätekniikka

Ohjelmistotekniikka

SISÄLLYS

[1 SOVELLUKSEN KUVAUS 3](#_Toc69710015)

[1.1 Sovelluksen arkkitehtuuri 3](#_Toc69710016)

[1.2 MainActivity 4](#_Toc69710017)

[1.3 SettingsActivity 4](#_Toc69710018)

[1.4 GameActivity 5](#_Toc69710019)

[1.5 Weather 6](#_Toc69710020)

[2 VAATIMUKSET 8](#_Toc69710021)

[3 APIN KUVAUS 9](#_Toc69710022)

[4 PROSESSI (POHDINTA) 10](#_Toc69710023)

[5 POHDINTA 11](#_Toc69710024)

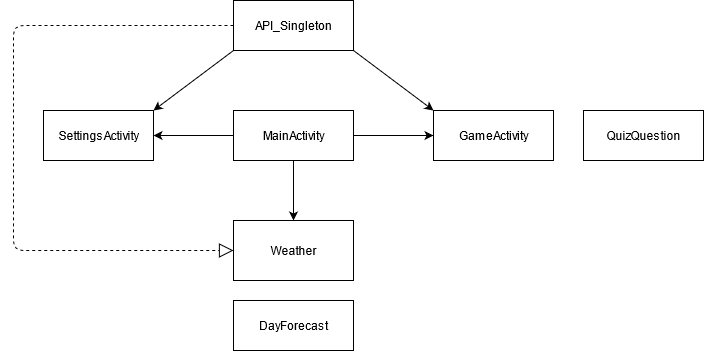
# SOVELLUKSEN KUVAUS

Projektin aiheeksi valikoitui lukuisien ”project ideas” ja ”free api list” hakusanojen jälkeen tietovisa, kun löydettiin saman aiheinen fiksulta ja yksinkertaiselta vaikuttava API. Ohjelma on rakennettu pitkälti tämän ympärille. Ohjelmassa voi valita haluamansa peliasetukset, jonka jälkeen visualisoidaan APIlta tulevat kysymykset tietovisaksi.

Ohjelmaan on myös tehty seitsemän päivän sääennuste. Allekirjoittanut on halunnut tietynlaiset nopean tavan katsoa säätä, joten tämä tuli aitoon tarpeeseen. Myös käytetty listadapteria, vaikkakin käytännössä samalla tavalla kuin tunnin esimerkissä.

## Sovelluksen arkkitehtuuri

Sovelluksen rakenne on karkeasti seuraavanlainen kuin kuviossa 1.



KUVIO 1. Sovelluksen rakenne

Sovelluksen datan hakemisen ja tarjoamisen aktiviteeteille hoitaa staattinen luokka API\_Singleton. Ennen tunneilla näytettyjä esimerkkejä kyseinen luokka ei hoitanut API-kutsuja, vaan ne hoidettiin aktiviteeteissa. Eli API\_Singleton tarjosi ainoastaan staattisen RequestQueuen, jota tarjottiin aktiviteeteille.

Nyt kuitenkin koodi on tosiaan refaktoroitu siten, että pelidatan kutsuminen hoidetaan kokonaisuudessaan API\_Singletonissa ja getterillä tarjotaan data muille luokille. Weather-aktiviteetissä on vielä ”vanha” toteutus, koska koodinluettavuuden kannalta ajateltiin olevan fiksumpaa jättää API-kutsun tekeminen ja parsiminen sinne sen sijaan, että paisutettaisiin API\_Singletonia.

## MainActivity

MainActivity on sovelluksen päämenu, josta ohjelman käyttäminen alkaa. Käyttäjällä on mahdollisuus aloittaa suoraan pelaan, koska aktiviteetin alkaessa lähdetään tekemään API-kutsua perusasetuksilla, kymmenen kappaletta mitä tahansa kysymyksiä.

Alussa haetaan myös kaikki kategoriat (ja niitä vastaavat ID:et) SettingsActivitya varten. Nämä säilötään HashMappiin String, Integer pareina.

MainActivitysta myös pääsee Weather-aktiviteettiin. Virhehallinta on ohjelmoitu API\_Singletonille booleanit, jotka pitävät kirjaa, kuinka data on ladattu. Jos dataa ei ole ladattu, niin aktiviteettiin mentäessä tulostuu virheilmoitus Toastina.

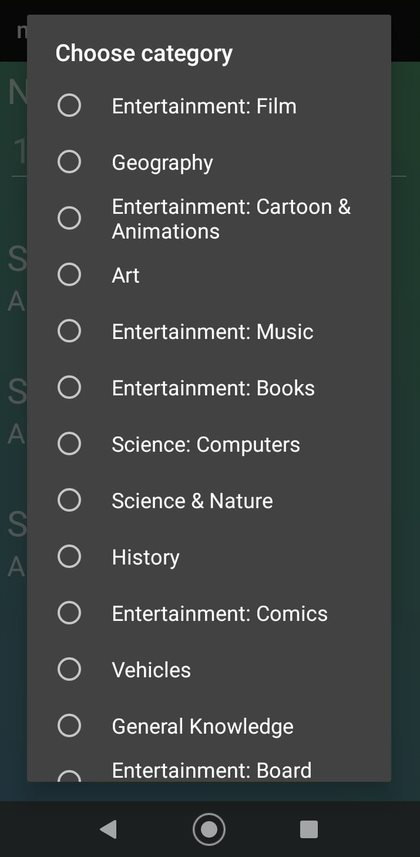
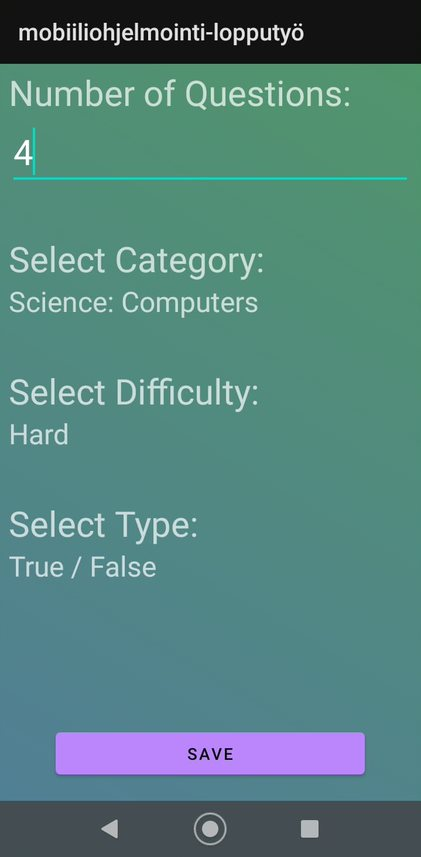
## SettingsActivity

SettingsActivityssa valitaan pelin asetukset, jotka ovat kysymyksien lukumäärä, kategoria, vaikeusaste ja onko totuus/valhe tehtäviä vai monivalinta (vai molempia).

Kysymysten lukumäärä otetaan perinteiseen tapaan numeerisella texteditillä. Muihin tietoihin haluttiin rakentaa erillinen dialogi, jossa on valmiiksi asetettuna valittavat vaihtoehdot. Tämä on rakennettu AlertDialogin, jolla homma onnistui hyvin. Myöhemmin selvisi, että Androidissa spinneri olisi hoitanut asian myös mainiosti ja varmaankin vähemmällä koodiriveillä.

Virheenhallinta ei ole tarvinnut muuten tehdä, paitsi numerovalinta on 1 – 50 väliltä. Aktiviteetti toimii myös mainiosti landscape-ruudulla, tilat tallentuu asianmukaisesti.

Kun käyttäjä on valinnut haluamansa asetukset, generoituu kutsuttava API-linkki aktiviteetin resultina, josta MainActivity tekee kutsun.

KUVA 1, 2. SettingsActivity kuvakaappaukset

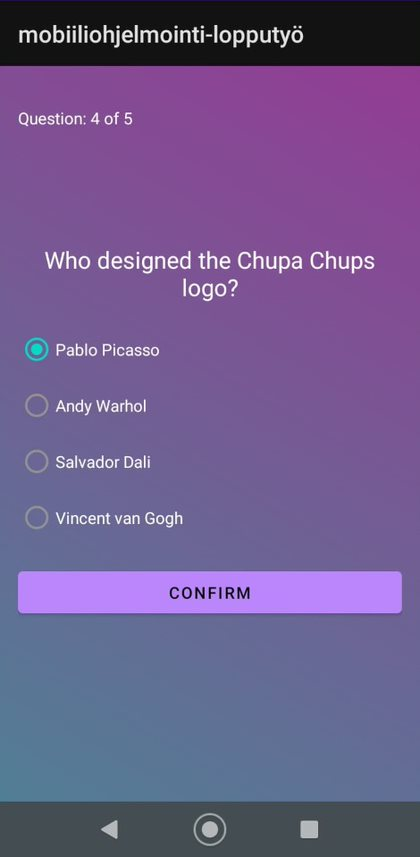
## GameActivity

GameActivityn ”pääkierto” on näyttää kysymys, klikata nappia ja tarkastaa vastaus, näytetään oikea (ja väärä) vastuas, minkä jälkeen näytetään uusi kysymys jne., kunnes kysymykset loppuvat.

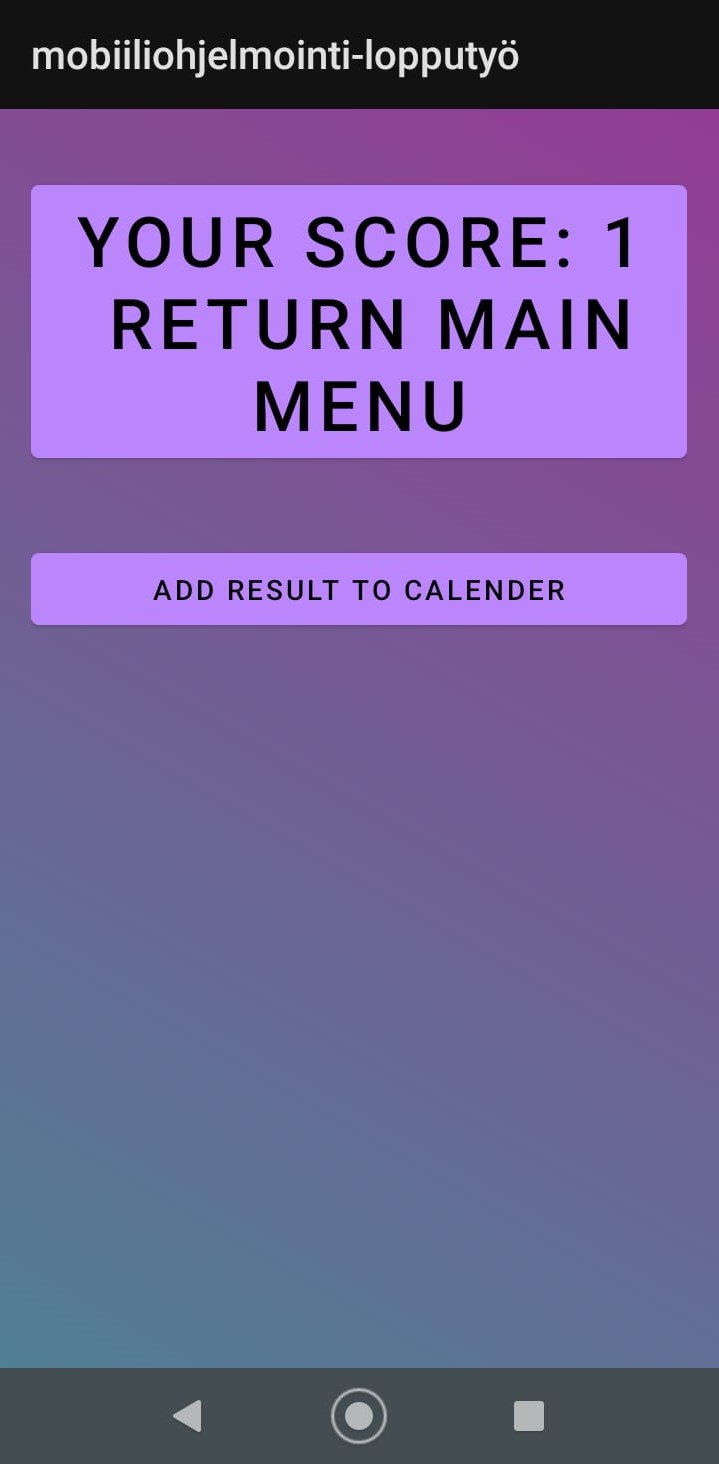
Visailukysymyksistä on tehty oma luokka, ja Singleton\_API tarjoaa listan näistä aktiviteetille.

Väärä ja oikea vastaus highlightataan timerilla 2 sekunnin ajaksi vaihtamalla radiobuttonin taustaväriä. Tämän ajaksi vastausnappula disabloidaan, jotta käyttäjä ei pysty hakkaamaan timerin aikana ”vastauksia jonoon”.

Kun kysymykset loppuvat, eli kysättävän kysymyksen indeksi on samankokoinen kuin listan koko, tulostuu vain kaksi buttonia. Toisella palataan MainActivitetyyn ja toisella voi lisätä tuloksensa Intentillä kalenteriin halutessaan.

KUVA 3, 4. GameActivity ja oikea vastaus



KUVA 5. Lopputulema GameActivityssa

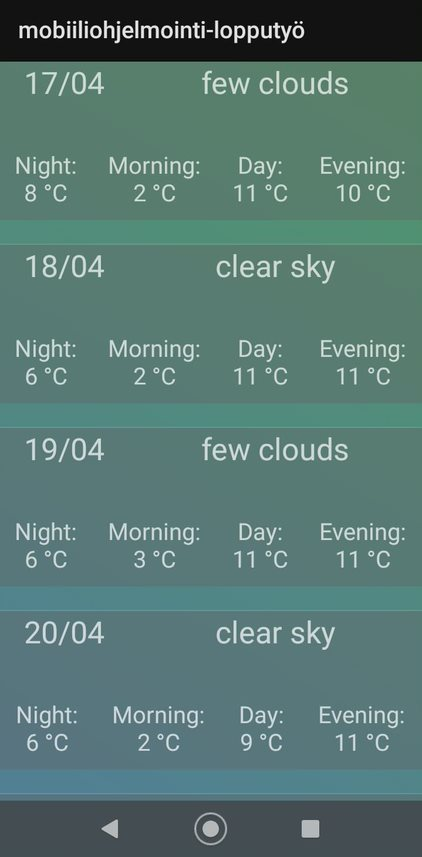
## Weather

Aktiviteetissa haetaan viikon sääennuste koordinaattien perusteella eri vuorokauden ajoille ja listataan ne SimpleAdapterin avulla.

Aktiviteetin aluksi haetaan viimeisin sijainti, jos sellainen on olemassa. Jos ei ole, käyttäjälle tulostuu Toastina virheilmoitus asiasta. Tämän jälkeen haetaan data, parsitaan se DayForecast olioksi listaan. Matkalla tehdään triviaaleja hienosäätöjä (esim. päiväyksen muuttaminen).

Tämän jälkeen luodaan SImpleAdapterin tarvitsema HashMap ja näiden avulla visualisoidaan data.

Aktiviteetista poistuessa data tuhotaan ja haetaan uudelleen kun saavutaan aktiviteettiin taas.



KUVA 5. Weather-aktiviteetin näkymä

# VAATIMUKSET

Sovellus täyttää seuraavat vaatimukset:

* REST API
* Laiteapi (GPS, Kalenteri)
* Toimiva, bugiton UI
* Aktiviteeteissa tilat säilyvät
* ListView, SimpleAdapter
* Ei kaatuile
* Resurssitiedostojen hyödyntäminen

Tavoitteena työstä on 6 pistettä. Sovellus toimii moiteetta, ei kaatuile ja on ulkonäöltään moitteettoman näköinen. Sovellus tukee landscape-kuvaa muulloin, paitsi GameActivityssa se on disabloitu, koska en saanut kaikkea mahtumaan näytölle fiksusti enkä pitänyt ScrollViewiä vaihtoehtona.

GameActivityssa tila häviää, mikä on tietoinen valinta. SettingsActivityssa tila säilyy näyttöä kääntäessä ja muulloinkin onSaveInstanceState metodi-laukaistaan. (Eli käytännössä aktiviteetti keskeytyy muusta syystä kuin käyttäjä itse poistuu).

Stringit, kuvat, teemat on ulkoistettu resurssitiedostoihin, arkkitehtuurissa on pyritty noudattamaan Singletonia APIn suhteen. Itse koodissa varmasti olisi hiottavia kohtia, mutta on suhteellisen toimivaa.

# APIN KUVAUS

APIna toimi Open Trivia Database: <https://opentdb.com/>

OpenTdb oma dokumentaatio: <https://opentdb.com/api_config.php>

Lyhykäisyydessään kyseessä REST API. Mahdollista valita kysymysten lukumäärä (1 – 50), kategoria, vaikeusaste, onko kysymykset true/false vai monivalintoja ja encoding.

APIsta mahdollisuus käyttää myös session tokenia, joka mahdollistaa että ei tule samoja kysymysiä.

APIn vastauskoodit ovat:

Code 0: Success Returned results successfully.

Code 1: No Results Could not return results. The API doesn't have enough questions for your query. (Ex. Asking for 50 Questions in a Category that only has 20.)

Code 2: Invalid Parameter Contains an invalid parameter. Arguements passed in aren't valid. (Ex. Amount = Five)

Code 3: Token Not Found Session Token does not exist.

Code 4: Token Empty Session Token has returned all possible questions for the specified query. Resetting the Token is necessary.