Saimaan ammattikorkeakoulu
Tekniikka Lappeenranta
Tietotekniikan koulutusohjelma
Organisaation IT-palvelut

Otto Luttinen

Moodlen käyttö Android-ohjelmoinnin opetuksessa

Opinnäytetyö 2016

Tiivistelmä

Otto Luttinen
Moodlen käyttö ohjelmoinnin opetuksessa
Saimaan ammattikorkeakoulu
Tekniikka Lappeenranta
Tietotekniikan koulutusohjelma
Organisaation IT-palvelut
Opinnäytetyö 2016
Ohjaajat: lehtori Perttu Laivamaa, Saimaan ammattikorkeakoulu

Opinnäytetyön tavoitteena oli tehdä Android-ohjelmointikurssi Saimaan ammattikorkeakoululla käytössä olevaan Moodle-opetusalustaan. Samalla oli tavoitteena tutkia, kuinka hyvin Moodle-opetusalusta soveltuu ohjelmoinnin opettamiseen. Kurssin tarkoitus oli olla itsenäisesti opiskeltava verkkokurssi, joka antaa opiskelijalle perusteet Android-ohjelmointiin. Omat tavoitteeni olivat oppia Androidin perusteet, jotta voisin jatkossa tehdä omia Android-sovelluksia.

Kurssi toteutettiin Saimaan ammattikorkeakoululla käytössä olevaa Moodle-alustaa käyttäen. Kurssi jaettiin aiheisiin. Jokaiselle aiheelle tehtiin teoriaosuus, pieni tentti testaamaan opiskelijan oppimista sekä kerättiin aiheeseen liittyviä linkkejä. Opinnäytetyössä käsitellään myös kahta Moodleen saatavaa lisäosaa, jotka lisäisivät Moodleen editorin, kääntäjän ja työkalut ohjelmien tarkastamiseen automaattisesti.

Avainsanat: Android, Moodle, verkkokurssi, ohjelmointi

Abstract

Otto Luttinen
The use of Moodle platform in education of programming
Saimaa University of Applied Sciences
Technology Lappeenranta
Degree Programme in Information Technology
Organizations IT-services
Bachelor´s Thesis 2016
Instructor(s): Mr. Perttu Laivamaa, Saimaa University of Applied Sciences (UAS)

The goal of this thesis was to create an Android programming course for the Moodle education platform which was in use in Saimaa University of Applied Sciences. The goal was also to study how suitable the Moodle platform is for teaching programming. The course was meant to be a self-study course which gives the learner the basics of Android programming. My own goal was to learn the basics of Android programming so I could make Android software.

The course was created using Moodle platform which was already in use at Saimaa University of Applied Sciences. The course was divided in chapters. Each chapter has theory, a little test and some useful links. The thesis also covers two plugins for the Moodle platform which add a text editor, compiler and tools to automatically check the programs.

Keywords: Android, Moodle, virtual course, programming

Sisältö

Lyhenteet ja termit 5
1 Johdanto 7
2 Android 7
3 Moodle 8
3.1 Kuvaus 8
3.2 Tilastoja 8
3.3 Rakenne 8
4 Android Studio 9
4.1 Kuvaus 9
4.2 Hyödyt 10
5 Kurssin tavoitteet 11
6 Opetusmenetelmät 11
6.1 Teoria 12
6.2 Tehtävät 12
6.3 Hyödylliset linkit 12
7 Kurssin esittely 12
7.1 Teorian sisältö 13
7.2 Tentit 16
8 Moodlen lisäosat 17
8.1 Viope 17
8.2 Virtual Programming Lab 18
9 Yhteenveto 19
9.1 Moodlen soveltuminen ohjelmoinnin opettamiseen 19
9.2 Projektin onnistuminen 20
9.3 Kuinka kurssia voisi kehittää tulevaisuudessa 20
9.4 Projektin oppimistavoitteet 20
Kuvaluettelo 22
Lähteet 23

Lyhenteet ja termit

Android	Googlen kehittämä mobiilikäyttöjärjestelmä.
Android Studio	Googlen kehitysympäristö Android-käyttöjärjestelmälle.
Moodle	Ilmainen virtuaalinen oppimisalusta.
Google Play	Androidin sovelluskauppa.
IOS	Applen mobiilikäyttöjärjestelmä.
Windows Phone	Microsoftin mobiilikäyttöjärjestelmä.
Avoin lähdekoodi	Lähdekoodi, joka julkaistaan ilmaiseksi kenen tahansa muokattavaksi ja käytettäväksi.
Wiki	Verkkosivusto, jonka sisältö perustuu käyttäjien yhteistyöhön.
Emulaattori	Ohjelma, joka mahdollistaa yhdelle tietokonejärjestelmälle suunniteltujen ohjelmien käytön toisessa tietokonejärjestelmässä.
Kääntäjä	Ohjelma, joka kääntää lähdekoodin konekielelle, jotta se voidaan ajaa tietokoneessa.
Java	Olio-ohjelmointikieli.
XML	Merkintäkieli, jolla voidaan muuttaa asiakirjat muotoon, joka on ihmisluettava ja koneluettava.
Word	Microsoftin tekstinkäsittelyohjelma.
Ohjelmointirajapinta	Määritelmä, jonka mukaan eri ohjelmat voivat vaihtaa tietoja.
Voice Over Ip	Tekniikka, joka siirtää ääntä Internetin yli.
Manifestitiedosto	Tiedosto, joka tarjoaa olennaista tietoa ohjelmasta Android-järjestelmälle.

Resurssitiedosto	Tiedosto, joka sisältää resursseja, joita ohjelma tarvitsee.
Aktiviteetti	Näkymä, jossa on käyttöliittymäelementtejä.
Palvelu	Komponentti, joka suoritetaan taustalla. Ei ole vuorovaikutuksessa käyttäjän kanssa.
Näkymä	Ruutu, jonka käyttäjä näkee.
Tapahtumakäsittelijä	Käyttäjän toimiin reagoiva metodi.
Fragmentti	Itsenäinen komponentti, joka liitetään osaksi aktiviteettiä.
Asetustiedosto	Tiedosto, johon voidaan tallentaa pysyvää tietoa.
HTTP-protokolla	Protokolla, jota selaimet ja WWW-palvelimet käyttävät tiedonsiirtoon.
Resoluutio	Näytön kuvantarkkuus.
Kryptografia	Salakirjoitustekniikka. Tiedon salaus.
Google Play	Googlen sovelluskauppa.
Kysymyspankki	Kysymystietokanta, joka sisältää useita kysymyksiä.
HTML	Kuvauskieli, jolla internetsivut on kirjoitettu.

1 Johdanto

Tämän opinnäytetyön tarkoituksena on toteuttaa Android-ohjelmoinnin perusteet -kurssi käyttäen Moodle-opetusalustaa ja testata Moodlen soveltuvuutta ohjelmoinnin itseopiskeluun. Kurssi sisältää opetusmateriaalit ja tehtävät. Opetusmateriaalit sisältävät Android-ohjelmoinnin teoriaa, jossa selitetään kuinka ohjelmat ja erilaiset ohjelmoinnin käskyt toimivat. Opetusmateriaalit sisältävät myös linkkejä Internet-sivuille, joista voi saada hyödyllistä lisätietoa sekä malliesimerkkejä, jotka auttavat ymmärtämään kuinka koodi toimii.

Kurssin kohderyhmänä ovat sellaiset henkilöt, joilla on hieman kokemusta ohjelmoinnista sekä ohjelmoinnin peruskäsitteet hallussa. Kurssi on myös tarkoitettu niille, jotka ovat kiinnostuneita opiskelemaan ohjelmointia mobiililaitteille.

Tavoite on käyttää ilmaisia ja vapaita ohjelmistoja, jotta kynnys aloittaa kurssi olisi mahdollisimman pieni.

2 Android

Android on Googlen ylläpitämä käyttöjärjestelmä mobiililaitteille. Sen pääasiallisia käyttökohteita ovat kosketusnäytöillä varustetut kännykät ja tabletit, mutta siitä on tehty versiot myös älytelevisioihin, autojen viihdelaitteisiin ja rannekelloihin.

Android on tällä hetkellä maailman käytetyin mobiilikäyttöjärjestelmä. Google arvioi vuonna 2014, että maailmassa on yli miljardi aktiivista Androidin käyttäjää. Googlen sovelluskaupassa Google Playssä on yli miljoona julkaistua sovellusta ja 50 miljardia latausta. Androidin pahimmat kilpailijat tällä hetkellä ovat Applen iOS ja Microsoftin Windows Phone -käyttöjärjestelmät. (Ars Technica; Cnet; Androidcentral.)

Android käyttää avoimen lähdekoodin lisenssiä. Tämä tarkoittaa sitä, että Androidin lähdekoodi julkaistaan vapaaseen jakeluun ja kuka tahansa voi käyttää ja muokata sitä. Google on tehnyt myös suljetun lähdekoodin ohjelmia Androidille ja laitteet tulevat niiden kanssa. Jotkut valmistajat ovat rakentaneet omia käyttöliittymiä ja sovelluksia, joiden kanssa heidän puhelimensa tulevat.

Yhtiöt ovat tehneet tämän siksi, että erottuisivat kovassa kilpailussa joukosta. (Ars Technica; Cnet; Androidcentral; Harju 2013, s. 11.)

3 Moodle

Tässä luvussa esitellään Moodle-opetusalusta. Luvussa käydään läpi myös sen tilastoja ja rakennetta.

3.1 Kuvaus

Moodle on ilmainen vapaan lähdekoodin oppimisalusta, josta käytetään myös nimitystä virtuaalinen oppimisympäristö. Moodlen nimi tulee englanninkielisestä akronyymistä ”Modular Object-Oriented Dynamic Learning Environment". Moodlen on kehittänyt Martin Dougiamas ja sen kehitys aloitettiin vuonna 1999. Moodlen uusin versio on tämän kirjoitushetkellä 3.0. (Moodle yhteisö.)

3.2 Tilastoja

Moodle on laajamittaisessa käytössä ympäri maailmaa. Moodlen tilastojen mukaan Moodle on käytössä 229 maassa. Moodlea käyttäviä sivustoja on noin 75 tuhatta kappaletta ja niillä on käyttäjiä yli 86 miljoonaa (Moodle tilastot.)

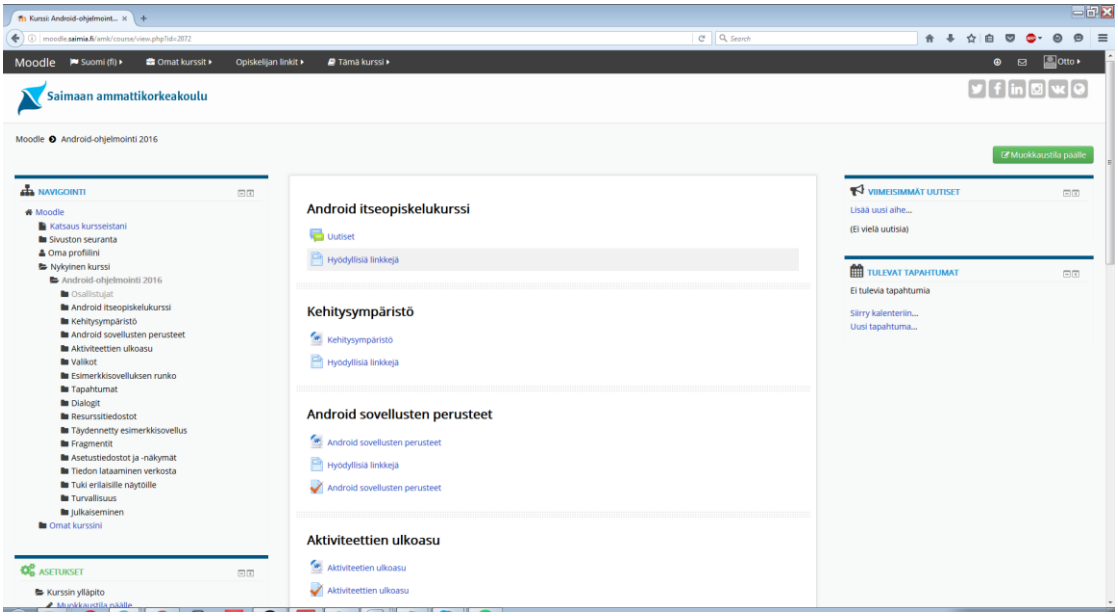
3.3 Rakenne

Moodlen rakenne koostuu kursseista, joihin opiskelijat voivat ilmoittautua vapaasti tai saatuaan kurssin aukaisevan avaimen. Kursseilla opiskelijat ja opettajat voivat keskustella keskenään viestien avulla. Opiskelijat voivat myös opiskella yhteistyössä tekemällä wikejä. Kursseilla voidaan julkaista materiaalia sekä tehdä erilaisia tehtäviä ja kokeita. (Moodle yhteisö.)

Kurssit on mahdollista jakaa aihepiirien tai ajan mukaan. Tentit ja tehtävät voivat olla aikarajoitettuja tai vapaita tehdä milloin käyttäjä kerkeää. Tentit ja tehtävät voivat vaikuttaa tai olla vaikuttamatta arvosanoihin. (Moodle yhteisö.)

Moodle on asennettu palvelimelle, josta sitä voidaan käyttää verkkoselaimilla. Moodleen on tehty ja on mahdollista tehdä erilaisia teemoja, joilla voi muuttaa Moodlen ulkonäköä sopivaksi omaan tarpeeseen. Moodleen on myös tehty

paljon liitännäisohjelmia, joilla saa Moodleen haluamiansa ominaisuuksia. Kuvassa 1 on nähtävissä Moodlen kurssisivu. (Moodle yhteisö.)



Kuva 1. Moodlen kurssisivu.

4 Android Studio

Tässä luvussa esitellään kurssilla toimiva kehitysympäristö Android Studio ja esitellään sen hyödyt.

4.1 Kuvaus

Kehitysympäristönä kurssilla käytetään Android Studiota. Se on Googlen kehittämä kehitysympäristö ja Androidin virallinen kehitysympäristö, jonka ensimmäinen vakaa versio julkaistiin joulukuussa 2014. Androidille on olemassa muitakin kehitysympäristöjä, kuten esimerkiksi Eclipse ja Netbeans. Periaatteessa ohjelmiston kehittämiseen ei tarvita kuin tekstieditori ja ohjelmointikielen kääntäjä, mutta kehitysympäristöt helpottavat työskentelyä huomattavasti. (Android Developers; ZDNet; Tekeye; NBAndroid.)

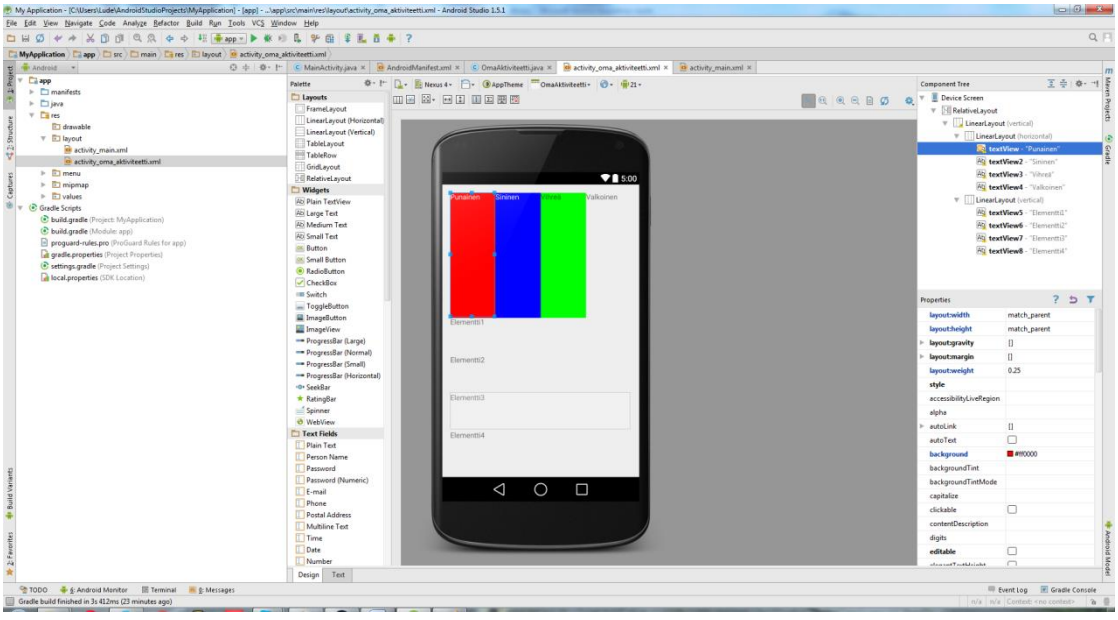
Ennen Android Studiota Androidin virallinen kehitysympäristö oli Eclipse, johon on lisäosa nimeltä Android Development Tools (ADT). Kesäkuussa 2015 Google ilmoitti lopettavansa virallisen tuen ja kehityksen Android Development Toolsille vuoden 2015 lopussa ja keskittyvänsä Android studion kehittämiseen. (Android Developers; ZDNet.)

4.2 Hyödyt

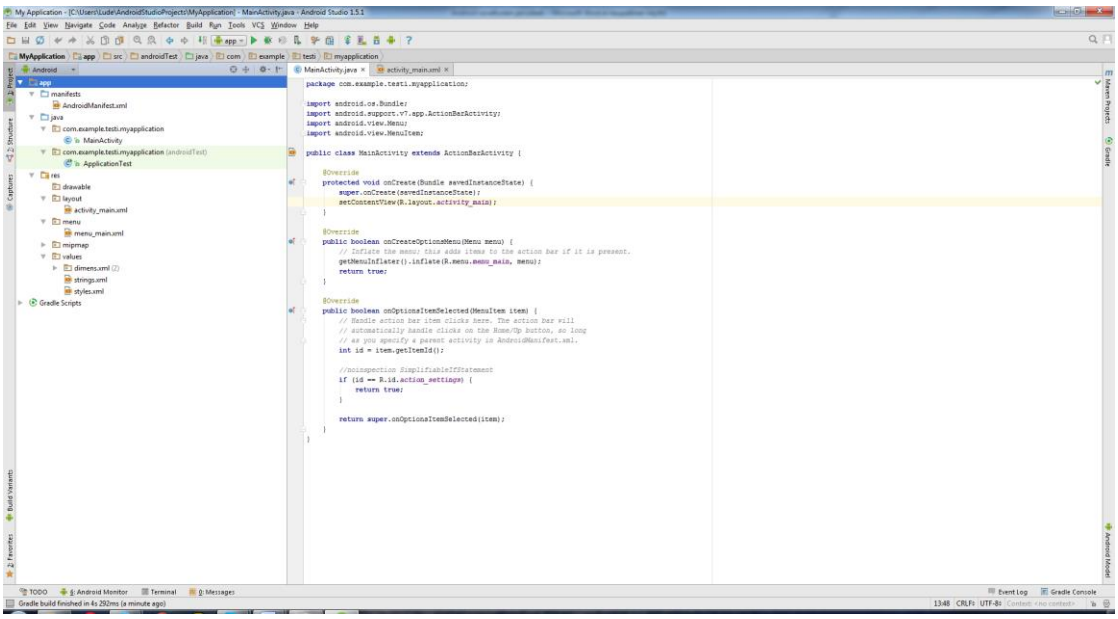
Android Studio -kehitysympäristön käytössä on monenlaisia hyötyjä. Android Studiossa on kehittynyt koodin täydennys, joka nopeuttaa työskentelyä. Tämä tarkoittaa sitä, että Android Studio yrittää päätellä, mitä käyttäjä on tekemässä ja antaa ehdotuksia lähdekoodiin. Ohjelmisto myös oikolukee lähdekoodia ja ilmoittaa, jos siinä on kirjoitusvirheitä. Kuvassa 3 on näkyvissä Android Studion ohjelmointityökalu.

Android Studio sisältää paljon valmiita malleja, jotta uusia ohjelmointiprojekteja ei tarvitse aloittaa ihan tyhjältä pöydältä. Android Studio sisältää myös graafisen käyttöliittymätyökalun, joka näkyy kuvassa 2. Sen avulla on paljon helpompi suunnitella käyttöliittymiä, koska muutokset näkyvät heti esikatselunäkymästä eikä muutoksia tarvitse kirjoittaa koodina.

Android Studiossa on sisäänrakennettuna kääntäjä, joka kääntää lähdekoodin toimivaksi ohjelmaksi. Jos lähdekoodissa on virheitä, niin kääntäjä ilmoittaa niistä ja missä kohtaa ne sijaitsevat. Android Studiossa on myös emulaattorityökalu, jolla voi testata ohjelman toimivuutta. Emulaattorityökalulla voidaan tehdä virtuaalisia laitteita, johon ohjelma ladataan käytettäväksi. Emulaattorityökalulla voidaan emuloida erilaisia älypuhelimia, tabletteja ja Androidia käyttäviä televisioita. Tämä mahdollistaa ohjelman testauksen erikokoisilla laitteilla, jotka käyttävät eri versioita Android-käyttöjärjestelmästä. (Android Studio.)



Kuva 2. Android Studion ulkoasutyökalu.



Kuva 3. Android Studion ohjelmointityökalu.

5 Kurssin tavoitteet

Kurssin tavoitteena on antaa opiskelijalle tiedot ja taidot Android-ohjelmoinnin perusteisiin. Kurssille osallistuvan toivotaan hallitsevan jonkun olio-ohjelmointikielen, kuten Javan ja XML:n perusteet. Kurssi on tarkoitus toteuttaa itseopiskelukurssina, jossa tarvitaan mahdollisimman vähän opettajan vuorovaikutusta. Optimaalisimmassa tapauksessa opettajaa ei tarvittaisi ollenkaan, edes harjoitusten tarkistamiseen. Jos kurssista halutaan antaa arvosana, tarvitaan opettajaa, koska kurssien arvosanat ovat toisessa järjestelmässä.

Kurssin tavoitteena on myös tutkia Moodlen soveltuvuutta ohjelmoinnin itseopiskeluun. Tarkoitus on löytää mahdolliset ongelmakohdat ja Moodlen rajoitteet ohjelmoinnin opetuksessa.

6 Opetusmenetelmät

Tässä luvussa esitellään Android-ohjelmointikurssilla käytetyt pääasialliset opetusmenetelmät.

6.1 Teoria

Kurssin teoriaosuus on jaettu viiteentoista lukuun, joista jokainen keskittyy tiettyyn asiaan. Luvuissa käsiteltyjä asioita ovat esimerkiksi kehitysympäristön asentaminen, ohjelman ulkoasu, valikot, resurssitiedostot, tietoturva ja ohjelman julkaisu. Luvuissa selitetään, miten asiat toimivat ja mihin niitä käytetään. Luvut sisältävät myös esimerkkikoodia, josta näkee miten asioita käytetään lähdekoodissa. Koodiesimerkit on myös selitetty auki kohta kohdalta, jotta opiskelija oppii, mitä mikäkin osa koodista tekee.

Jokaisen luvun teoria on tallennettu Microsoft Word -dokumenttiin. Tämä mahdollistaa sen, että opiskelijat voivat helposti ladata ne omille koneilleen ja lukea ja opiskella niitä ilman nettiyhteyttäkin.

6.2 Tehtävät

Tehtävien tarkoitus kurssilla on testata ja tukea opiskelijaa. Jokaiseen kurssin aiheeseen on tehty pieni tentti, joka koostuu muutamasta tehtävästä. Tentit voidaan suorittaa useita kertoja ja niillä ei ole aikarajaa. Kurssilla ei ole tarkoitus olla arvosanaa, siksi myöskään tentit eivät vaikuta siihen.

Moodlen tehtävistä kurssille sopivat parhaiten monivalinta- ja totta vai tarua -tehtävät. Nämä soveltuivat parhaiten, koska niissä ei tarvita opettajan vuorovaikutusta. Tentin suoritettuaan saa heti tietää, olivatko vastaukset oikein vai ei. Moodleen on mahdollista asentaa erilaisia tenttikysymystyyppejä lisäosien avulla, mutta niistä ei tuntunut löytyvän tähän kurssiin sopivia.

6.3 Hyödylliset linkit

Jokaiseen kurssin aiheeseen on myös kerätty hyödyllisiä linkkejä, joiden kautta saa aiheesta lisätietoa. Yksi paljon käytetty on Googlen oma Androidin ohjelmointirajapinnan kotisivu. Nämä linkit ovat useimmiten englannin kielellä.

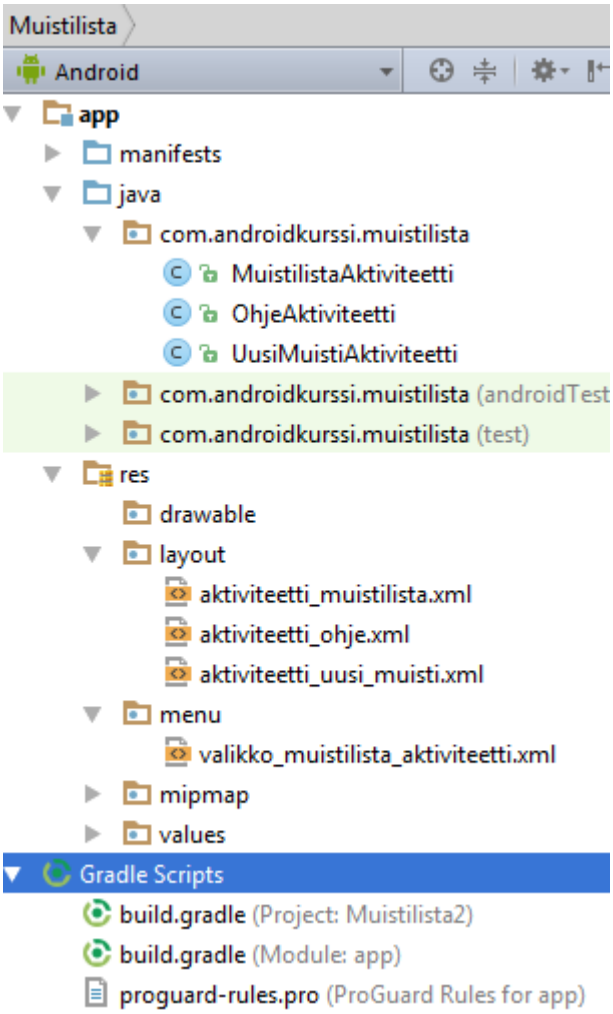
7 Kurssin esittely

Tässä luvussa esitellään, millaista sisältöä kurssille tuli. Luvussa esitellään lyhyesti, minkälaista teoriaa kurssi sisältää ja minkälaisia tenttejä siihen tehtiin.

7.1 Teorian sisältö

Kurssissa on yhteensä viisitoista lukua. Kurssi aloitettiin Androidin kehitysympäristön asentamisella. Ensimmäisessä luvussa käydään läpi, mistä kehitysympäristön voi ladata, kuinka se asennetaan ja mitä muita kehitysvälineitä pitää olla asennettuna. Kun kehitysympäristö on saatu asennettua, luvussa käydään läpi, kuinka tehdään ensimmäinen yksinkertainen sovellus. Samalla käydään läpi, kuinka luodaan Android-projekti ja virtuaalilaite, jonka avulla voidaan testata sovellusta. Luvun lopussa käsitellään sovelluksen testaaminen oikeassa Android-laitteessa.

Toinen luku käsittelee Android-sovelluksen perusteita. Luvussa opitaan, minkälainen on Android-sovelluksen rakenne ja mitä se sisältää, esimerkiksi mitä aktiviteetti tai palvelu tekee sovelluksessa. Luku käy myös läpi Android-projektin manifestitiedoston ja hakemistorakenteen, joka näkyy kuvassa 4. Luvussa opitaan myös Android-versioiden yhteensopivuudesta ja miten se pitää ottaa huomioon, sekä lokikirjoittamisesta. Lokikirjoittaminen helpottaa sovelluksen virheenjäljitystä kehitysvaiheessa.



Kuva 4. Esimerkkiohjelman hakemistorakenne.

Kolmas luku käsittelee aktiviteettien ulkoasua. Luvussa käydään läpi Näkymän asettelu, yleisimmät asettelumallit ja niiden käyttäminen, näyttöä suurempi asettelu ja yleisimmät käyttöliittymäkomponentit. Näiden asioiden oppiminen mahdollistaa sen, että sovellus saadaan näkymään laitteen ruudulla oikean näköisenä.

Neljäs luku käsittelee valikoita. Luvussa käydään läpi Androidin valikkorakenteet, valikkojen määrittäminen, niiden käyttäminen ohjelmakoodissa ja alavalikoiden luominen ja käyttäminen. Valikkojen avulla voidaan tehdä valinta ja antaa syötteitä ohjelmassa.

Viides luku käsittelee esimerkkisovelluksen runkoa. Luvussa käydään läpi, mitä hakemistoja ja tiedostoja sovellus tarvitsee. Luvussa opitaan, kuinka aiempien lukujen ohjelmakoodia käytetään sovelluksessa.

Kuudes luku käsittelee tapahtumia ja tapahtumakäsittelijöitä. Tapahtumakäsittelijöiden avulla sovellus voi reagoida käyttäjän toimintoihin, esimerkiksi, kun käyttäjä klikkaa sovelluksessa painiketta. Luvussa käydään myös läpi tapahtumien merkitseminen käsitellyiksi ja muut tapahtumatyypit.

Seitsemäs luku käsittelee dialogeja. Dialogi on valintaikkuna, joka aukeaa avoinna olevan näkymän päälle. Käyttäjä ei yleensä pysty tekemään muuta ennen kuin dialogi on suljettu. Dialogit ovat yleensä valintaikkunoita, joissa on joku pieni viesti ja OK- ja Peruuta-painikkeet. On myös dialogeja, joiden avulla käyttäjä valitsee päivämäärän tai kellonajan.

Kahdeksas luku käsittelee resurssitiedostoja. Luvussa opitaan, kuinka resurssitiedostoihin viitataan ohjelmakoodissa id-arvolla. Luvussa käydään myös läpi tärkeimmät resurssialihakemistot. Alihakemistot on jaoteltu resurssien tiedostotyyppien mukaan. Tuki sovelluksen monikielisyydelle saadaan toteutettua myös resurssitiedostojen avulla. Esimerkiksi jokaiselle kielelle luodaan tietyn niminen hakemisto, johon tallennetaan sen kielen merkkijonot tiedostoon strings.xml. Jokaiseen strings.xml tiedostoon luodaan samannimiset merkkijonot, jotka saavat arvoksi hakemiston kielen mukaisen merkkijonon.

Yhdeksäs luku käsittelee täydennettyä esimerkkiohjelmaa. Tässä luvussa lisätään aiemmassa luvussa tehtyyn esimerkkisovelluksen runkoon luvuissa kuusi, seitsemän ja kahdeksan opitut asiat. Esimerkkiohjelmalle tehdään tarvittavat resurssitiedostot ja ohjelmakoodiin lisätään tapahtumakuuntelijat.

Luvussa kymmenen käsitellään fragmentteja. Luvussa käydään läpi, kuinka fragmentteja käytetään ja mikä on niiden elinkaari sovelluksessa. Luvussa tehdään myös esimerkki, jossa yksi esimerkkisovelluksen aktiviteetti muutetaan fragmentiksi.

Luvussa yksitoista käsitellään asetustiedostoja ja –näkymiä. Luvussa käydään läpi, kuinka tietoa tallennetaan pysyvästi laitteen muistiin, niin että se säilyy sovelluksen käyttökertojen välillä. Luvussa opitaan, kuinka tietoa luetaan asetustiedostosta ja kirjoitetaan asetustiedostoihin. Luvussa opitaan myös käyttöoikeuksista, asetusaktiviteettien luomisesta ja asetusten ryhmittelystä.

Luvussa kaksitoista käsitellään tiedon lataamista verkosta. Luvussa käydään läpi verkon käyttöoikeudet, HTTP-protokollan käyttöä ja kuinka tietoa tallenne-

taan verkkoon. Käyttäjän tulee antaa sovellukselle verkonkäyttöoikeudet, ennen kuin se voi ladata tai tallentaa verkosta.

Luku kolmetoista käsittelee tukea erilaisille näytöille. Androidia käyttävät hyvin monet erilaiset laitteet, joilla on erikokoisia näyttöjä ja jotka käyttävät eri resoluutioita. Luvussa opitaan, kuinka resurssitiedostoja käytetään hyväksi tässä asiassa ja kuinka tuetut näytöt määritellään.

Luku neljätoista käsittelee sovelluksen tietoturvaa. Luvussa käydään läpi käyttöoikeuksien pyytäminen, käyttöjärjestelmätason ja sovellustason turvallisuus, sovellusten allekirjoittaminen ja kryptografia. Sovelluksen allekirjoittamisella varmistetaan, että asennettava sovellus on tosiaan oikean kehittäjän kehittämä. Kryptografiaa tarvitaan, jos sovelluksen tietoja halutaan salata.

Luku viisitoista käsittelee sovelluksen julkaisua. Luvussa käydään läpi, mitä toimenpiteitä tarvitsee tehdä ennen julkaisua, kuinka luodaan julkaisupaketti, sovelluksen testaaminen ja julkaisu Google Play -kaupassa. Google Play -kauppa on isoin ja yleisimmin käytetty sovelluskauppa, mutta se ei ole kuitenkaan ainoa julkaisupaikka.

7.2 Tentit

Jokaisen luvun loppuun tehtiin pieni tentti, jonka tarkoitus oli auttaa opiskelijaa testaamaan oppimistaan. Tentit on toteutettu monivalinta- ja totta vai tarua -tehtävillä. Heti tentin tehtyään opiskelija näkee, mitkä tehtävät menivät oikein ja mitkä eivät. Opiskelija näkee myös, mitkä vastaukset olisivat olleet oikeita väärin menneissä kysymyksissä.

Tentit on toteutettu kysymyspankin avulla. Kurssin tekijä voi tehdä kysymyspankkiin useita kysymyksiä, jotka kone arpoo tenttiin. Tämä tarkoittaa sitä, että opiskelijoilla ei välttämättä ole samoja kysymyksiä tai ne eivät ole samassa järjestyksessä. Tämä vaikeuttaa hieman huijaamista, koska toisen opiskelijan vastauslista ei ole samanlainen toisen opiskelijan kanssa. Tosin huijaamisen esto ei ole tämän kurssin tapauksessa kovin tärkeää, koska tästä kurssista ei saa arvosanaa.

8 Moodlen lisäosat

Kurssia tehdessä tuli vastaan pari vaihtoehtoa, joilla voisi lisätä kurssiin erityyppisiä tehtäviä. Valitettavasti tämän projektin aikana niitä ei ehditty toteuttaa kurssiin. Toinen vaihtoehto on Moodlen lisäosa nimeltä Virtual Programming Lab ja toinen on suomalaisen Viope-yhtiön tekemä työkalu, joka voidaan liittää Moodleen.

8.1 Viope

Viope on suomalainen yritys, joka on keskittynyt digitaalisten oppimisratkaisujen kehittämiseen. Viope on kehittänyt web-pohjaisen työkalun matematiikan ja ohjelmoinnin opetukseen. Viopen työkaluja on käytetty yli 70 maassa erilaisissa oppilaitoksissa. (Viopen kotisivut.)

Viopen työkalu on mahdollista integroida Moodleen. Tästä on koululle monia hyötyjä. Esimerkiksi käyttäjien autentikointi tehdään Moodle-tilien avulla eikä Viopeen tarvita erillisiä käyttäjätilejä. Myös Viopessa olevat opiskelutiedot ja saavutukset olisivat nähtävissä Moodlessa. (Viope.)

Viopen yhdistämiseen Moodleen tarvitaan aktiivinen Viope-lisenssi ja ylläpitäjän oikeudet. Tämän jälkeen pitää ottaa käyttöön Moodle-integraatio-ominaisuus Viopesta. Myös Moodleen tarvitaan opettajan tai ylläpitäjän oikeudet. (Viope.)

Viopeen kirjaudutaan ylläpitäjän tilillä sisään ja painetaan Manage-nappia. Seuraavaksi valitaan Integration-välilehti, jolta painetaan Moodle-nappi Ontilaan. Kun tämä on tehty, näyttää Viope nettisivun osoitteen ja kaksi koodiavainta, joita Viope käyttää Moodle-integraatiossa. Nämä lisätään Moodlen asetuksiin, jonka jälkeen Moodle-kursseille voidaan lisätä Viope-aktiviteettejä. (Viope.)

Viope tarjoaa opettajille työkalut kurssien tekemiseen. Opettaja voi luoda kurssille kokeita, jotka tarkastetaan ja pisteytetään automaattisesti. Opettaja näkee oppilaiden vastaukset ja miten he ovat ratkaisseet tehtävän. Opettaja pystyy helposti testaamaan oppilaan koodia tarvittaessa. Opettaja voi lisätä kurssille ohjelmointitehtäviä, monivalintatehtäviä ja avoimia tehtäviä, jotka voivat olla esseitä tai tehtäviä, joissa pitää lähettää tiedosto. (Viopen kotisivut.)

Viopen kurssieditorilla voidaan tuottaa kurssille interaktiivista materiaalia. Kurssille voi lisätä video- tai tekstipohjaista teoriaa ja siihen pystyy liittämään e-kirjoja tunnetuilta julkaisijoilta. Tarkastusta varten Viopessa on raportointiominaisuus. Tuloksista saa tehtyä erilaisia kuvaajia, joiden avulla pystyy analysoimaan tuloksia. Tulokset voidaan myös muuttaa HTML-muotoon, jolloin ne on helpompi siirtää muualle. Viopessa on kommunikointityökalu, jota voi käyttää oppilaiden kanssa viestittelyyn. Työkalun avulla voi tallentaa ja toistaa luentoja, jakaa tietokoneen työpöydän ja pitää Voice Over IP –konferensseja (VOIP). (Viopen kotisivut.)

Olen itse suorittanut muutaman Viopen ohjelmointikurssin. Ne olivat itseopiskelukursseja. Ne koostuivat luettavasta teoriasta, monivalintatehtävistä ja ohjelmointitehtävistä. Kurssit oli jaettu lukuihin, jotka käsittelivät tiettyjä aiheita. Seuraavaan lukuun pääsi vasta, kun oli suorittanut edellisen luvun kaikki tehtävät. Monivalintatehtävissä saattoi olla yksi tai useampi oikea vastaus. Jokaisen tehtävän jälkeen näki heti, menikö se oikein vai ei.

Ohjelmointitehtävissä piti tehdä yksinkertainen ohjelma, jonka Viope tarkasti. Tarkastus tapahtui yleensä ajamalla ohjelma monta kertaa läpi erilaisilla syötteillä ja tarkastamalla, että ohjelma palautti niillä oikeat arvot takaisin. Jos ohjelma ei toiminut oikein, niin Viope palautti virheilmoituksen. Virheilmoitus yritti näyttää, missä kohtaa ohjelmaa virhe todennäköisesti oli, ja se toimikin yleensä aika hyvin. Kyseessä oli kuitenkin automoitu prosessi, joten se ei aina löytänyt oikeaa syytä siihen, miksi ohjelma ei toiminut. Tästä johtuen myös ohjelmien tulosteiden piti olla juuri samanlaiset mitä malliesimerkeissä oli annettu. Jos tulosteissa oli kirjoitusvirhe, niin ohjelma ei mennyt tarkastuksesta läpi.

8.2 Virtual Programming Lab

Moodleen on saatavilla lisäosa nimeltä Virtual Programming Lab. Virtual Programming Lab ajaa hyvin pitkälti saman asian kuin Viope. Se on tarkoitettu helpottamaan ohjelmoinnin opetusta Moodlessa. Se antaa oppilaille mahdollisuuden muokata lähdekoodia selaimessa. Lisäosa sisältää myös kääntäjän, joten oppilaat voivat myös ajaa lähdekoodia suoraan selaimessa. Lähdekoodin editori näyttää myös ohjelmointikielen lauseopin korostukset, jotta lähdekoodia on helpompi lukea. (VPL.)

Virtual Programmin Lab sisältää tuen useille eri ohjelmointikielille, esimerkiksi C, C++, C#, Java, Matlab, Perl, PHP, Python, Ruby ja SQL. Ohjelmat voidaan ajaa tekstikonsolissa, jossa voidaan antaa syötteitä ja katsoa, että ohjelma tulostaa oikeat vastaukset. Opettaja voi tehdä testejä ohjelmille, jotka varmistavat että ohjelma toimii oikein. (VPL.)

Virtual Programming Lab antaa myös mahdollisuuden etsiä mahdollisia huijauksia. Opettaja voi etsiä samanlaisia tiedostoja ja lisätä vesileiman ladattuun koodiin. Opettaja voi myös rajoittaa editointimahdollisuuksia ja estää ulkopuolisen koodin kopioinnin ja liittämisen. (VPL.)

En pystynyt asentamaan lisäosaa Saimaan ammattikorkeakoulun Moodleen, koska se pitäisi testata hyvin ennen kuin sen ottaa käyttöön koko koulun järjestelmässä. Tämä johtuu siitä, että lisäosa saattaisi sekoittaa Moodlen. Todennäköisyys sille, että tämä tapahtuu, on pieni. Koska tämä kurssi on sen verran pieni, ei kannata ottaa riskiä, että koko koulun järjestelmä kaatuu.

9 Yhteenveto

Yhteenveto-osiossa mietitään, mitä projektissa opittiin, miten projekti onnistui ja mitä voisi parantaa.

9.1 Moodlen soveltuminen ohjelmoinnin opettamiseen

Moodle-opetusalusta soveltuu kohtalaisesti ohjelmoinnin opettamiseen. Se sopii siihen paremmin, jos kurssilla on opettaja. Moodleen on helppo palauttaa tiedostoja, jotka opettaja voi myöhemmin tarkastaa. Moodle myös antaa mahdollisuuden ajoittaa tehtävät, jolloin ne pitää olla palautettuna oikeaan aikaan, eikä niitä voi palauttaa enää myöhässä. Moodlen tentteihin voisi myös tehdä useampia erilaisia tehtäviä, jos kurssilla on opettaja, joka tarkistaisi ne. Moodlessa valmiina olevat tehtävävaihtoehdot ovat melko rajalliset.

Jos kurssilla ei ole opettajaa, muodostuu ongelmaksi tehtävien ja tenttien tarkastaminen. Tenteissä käytännössä ainoa tehtävävaihtoehto on monivalintakysymykset, joista Moodle pystyy suoraan antamaan oikean vastauksen ja arvosanan. Lisäosien, kuten Virtual Programming Lab ja Viope, avulla kurssil-

le pystyisi tekemään ohjelmointitehtäviä, jotka olisi mahdollista tarkastaa ilman opettajaa. Mutta lisäosien käytössä olisi myös omat ongelmansa, kuten maksullisuus tai kurssin tekemiseen menevä aika.

9.2 Projektin onnistuminen

Projekti saatiin päätökseen ja se onnistui hyvin. Kurssiin tuli viisitoista lukua. Oppilas saa kurssilta tarvittavat perustiedot Android-ohjelmointiin, joka oli tämän projektin tavoite.

Projektin raporttia olisi kannattanut alkaa tehdä aikaisemmin muun työn ohella, eikä jättää sitä projektin loppuun. Aikataulu venyi suunnitellusta, koska kurssi ja raportti vaativat enemmän työtä kuin olin alun perin suunnitellut.

Kurssia olisi voinut olla tekemässä useampi henkilö. Tällöin sisällöstä olisi saatu kattavampi ja olisi ollut parempi mahdollisuus tutkia Moodlen lisäosia ja mahdollisesti toteuttaa ne kurssiin.

9.3 Kuinka kurssia voisi kehittää tulevaisuudessa

Kurssia voisi mielestäni kehittää lisäämällä kurssiin Virtual Programming Lab tai Viope -lisäosa. Ne lisäisivät kurssiin ohjelmointitehtäviä, joista saisi suoran palautteen. Viope olisi helpompi ratkaisu, koska se sisältää kursseja ja materiaalia jo valmiiksi. Viopen kanssa ei tarvitsisi tehdä muuta kuin liittää se jo olemassa olevaan Moodle-kurssiin. Viopen huono puoli on sen maksullisuus. Kurssille pitäisi saada paljon osallistujia, jotta kurssi kannattaisi liittää Viopeen.

Toinen vaihtoehto olisi tehdä kurssiin ohjelmointitehtäviä Virtual Programming Lab -lisäosalla. Sen hyvä puoli on se, että se on ilmainen. Huonona puolena on taas valmiin materiaalin puute. Kurssin suunnitteluun ja valmisteluun pitäisi käyttää paljon aikaa, koska kurssille pitäisi tehdä ohjelmointitehtävät ja myös tehtävät tarkastavat skriptit. Toisaalta, kun kurssin on kerran saanut valmiiksi, niin sitten se ei tarvitsisi enää opettajan osallistumista.

9.4 Projektin oppimistavoitteet

Oma tavoitteeni oli oppia Android-ohjelmointia, jotta voisin itse tehdä Android-sovelluksia tulevaisuudessa. Ohjelmoinnista minulla oli jo kokemusta muilla ohjelmointikielillä, mutta minulla ei ollut kokemusta sovelluskehityksestä mobii-

lilaitteille. Opin myös käyttämään Android Studio -kehitysalustaa, josta minulla ei myöskään ollut kokemusta. Android Studio on minun mielestäni hyvä kehitysalusta. Se tarjosi tarvittavat työkalut helposti käytettäväksi ja siinä oli monia ohjelmointia helpottavia ominaisuuksia.

Myös Moodlen käyttö opettajan roolissa oli minulle uusi asia. En ole koskaan toteuttanut tämän tyyppistä kurssia. Opin paljon virtuaalisen itseopiskelukurssin suunnittelusta ja toteutuksesta.

Kuvaluettelo

Kuva 1. Moodlen kurssisivu. s 9

Kuva 2. Android Studion ulkoasu työkalu. s 10

Kuva 3. Android Studion ohjelmointityökalu. s 11

Kuva 4. Esimerkkiohjelman hakemistorakenne. s 14

Lähteet

Androidcentral 2016, Androidin historia. http://www.androidcentral.com/android-history Luettu 5.1.2016

Android Developers 2014, Google julkaisee ensimmäisen vakaan version Android Studiosta. http://android-developers.blogspot.fi/2014/12/android-studio-10.html Luettu 5.1.2016

Android Developers 2015, Google vaihtaa Android Studion viralliseksi kehitysympäristöksi. http://android-developers.blogspot.fi/2015/06/an-update-on-eclipse-android-developer.html Luettu 18.12.2015

Android Studio 2016, Android Studion kotisivu http://developer.android.com/sdk/index.html Luettu .18.12.2015

Ars technical 2014, Androidin historia. http://arstechnica.com/gadgets/2014/06/building-android-a-40000-word-history-of-googles-mobile-os/ Luettu 18.12.2015

Cnet 2014, Androidin historia. http://www.cnet.com/news/history-of-android/ Luettu 18.12.2015

Harju, J. 2013 Android-ohjelmoinnin perusteet. Helsinki, Books on Demand GmbH

Moodle tilastot 2016, Tilastoja Moodlen käytöstä. https://moodle.net/stats/ Luettu 6.1.2016

Moodle VPL 2016, Virtual Programming Lab –lisäosan lataussivu. https://moodle.org/plugins/mod_vpl Luettu 5.2.2016

Moodle yhteisö 2016, Tietoa Moodlesta. https://docs.moodle.org/30/en/About_Moodle Luettu 4.1.2016

NBAndroid 2015, Netbeansin Android lisäosa. http://nbandroid.org/ Luettu 18.12.2015

Tekeye 2014, Lista Androidin kehitysympäristöistä. http://tekeye.biz/2014/list-of-android-app-development-ides Luettu 6.1.2016

Viopen kotisivut 2016, Tietoa Viopesta. https://www.viope.com/#!/ Luettu 4.2.2016

Viope 2015, Viopen liittäminen Moodleen. https://www.viope.com/resources/Viope%20Moodle%20setup%20guide.pdf Luettu 5.2.2016

VPL 2016, Virtual Programming Lab –lisäosan kotisivut. http://vpl.dis.ulpgc.es/ Luettu 5.2.2016

ZDNet 2014, Google julkaisee Android Studion. http://www.zdnet.com/article/google-releases-android-studio-kills-off-eclipse-adt-plugin/ Luettu 6.1.2016