

Tallinna Ülikool
Informaatika Instituut

Android operatsioonisüsteem ja rakendused

Seminaritöö

Autor: Raul Liinev

Juhendaja: Jaagup Kippar

Autor.....,.....,2012

Juhendaja.....,.....,2012

Instituudi direktor.....,.....,2012

Tallinn 2012

Autorideklaratsioon:

Deklareerin, et käesolev bakalaureusetöö on minu töö tulemus ja seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

.....
(kuupäev)

.....
(autor)

Sisukord

Sissejuhatus	3
1. Mis on Android?.....	4
1.1. Rakendused	5
1.2. Rakenduste raamistik	5
1.3. Teegid	5
1.4. Androidi käituskeskkond	6
1.5. Linux Kernel	7
2. Platvormi versioonid	8
2.1. Androidi versioonid	8
2.1.1. Android 2.2	9
2.1.2. Android 2.3	9
2.1.3. Android 4.0	10
2.2. Uued Android seadmed	11
2.3. Värsked andmed	11
3. Rakenduste arendus.....	13
3.1. Rakenduse ülesehitus	14
3.2. Rakenduse installifail.....	15
4. Rakendus <i>helloworld</i>	16
5. Seminaritöö kuvamine läbi rakenduse	18
5.1. Ülesehitus.....	18
5.2. Rõõmud ja mured rakendusega.....	19
Kokkuvõte	21
LISAD	22
Mõistete ja lühendite tõlked ja kirjeldused.....	23

QR koodid seminaritöös koostatud rahenduste installifailide juurde.....	24
Kasutatud kirjandus	25

Sissejuhatus

Käesoleva dokumendi eesmärk on uurida Androidi operatsioonisüsteemi ülesehitust ja süsteemi võimalusi. Teada saada, kuidas sellisele platvormile arendada rakendusi ja millised vahendid selleks olemas on. Töö on sissejuhatuseks suurema rakenduse arendamisele, mille plaanin vormistada bakalaureusetööks.

Seminaritöös annan ülevaate Androidi platvormist ja põhilistest Androidi versioonidest, mis hetkel kasutusel on. Tutvustan Androidi rakenduse ülesehitust ja teen valmis ise esimese rakenduse. Kirjeldan, mis on vajalik, et rakendust arendama hakata.

1. Mis on Android?

Androidi ametliku arendajate lehe kohaselt on Android mobiiliseadmetele mõeldud tarkvarakogu, mis hõlmab endas operatsioonisüsteemi, vahetarkvara ja olulisemaid rakendusi.

(Android(1), 2012)

Androidi ajalugu sai alguse Android beta versiooniga 2007 aasta novembris. Esimene kommertsversioon, Android 1.0, sai alguse Septembril 2008. Android on välja töötatud Google ja Open Handset Alliance poolt. Androidi operatsioonisüsteem on läbinud mitmeid uuendusi esimesest versioonist alates. Need uuendused on vigade parandamiseks ja uuenduste lisamiseks. Alates 2009 aasta aprillist on Androidi versioone arendatud koodnimede all ja on väljastatud alfabeedilises järjestuses: *Cupcake*, *Donut*, *Eclair*, *Froyo*, *Gingerbread*, *Honeycomb*, *Ice Cream Sandwich* ja *Jelly Bean*. Kõige hiljutisem suur uuendus oli *Jelly Bean* v4.1, mis avalikustati Juuni kuus 2012 ja kasutajate jaoks oli kättesaadav kuu aega hiljem.



Joonis 1. Android'i ülesehitus. See diagramm näitab Androidi operatsioonisüsteemi tähtsamaid komponente.

1.1. Rakendused

Algselt on Android platvormiga kaasas e-post'i klient , sms programm, kalender, kaardid, brauser, telefoniraamat ja teised programmid. Kõik need rakendused on kirjutatud Java programmeerimiskeele abil.

1.2. Rakenduste raamistik

Android on avatud lähtekoodiga platvorm, mistõttu on arendajatel võimalus luua äärmiselt innovaatilisi ja rikkalikke rakendusi. Arendajad saavad kasutada seadmete riistavara, pääseda ligi asukoha informatsioonile, jooksutada tasutaprogramme, seadistada alarme, lisada teavitustelekuribal ja palju muud.

Arendajad pääsevad täielikult ligi samadele raamistiku rakendusliidestele, mille abil on ehitatud operatsioonisüsteemi põhiaplikatsioonid. Rakenduste arhitektuur on tehtud nii, et selle komponente oleks võimalikult lihtne uuesti kasutada. Igaüks võib avalikustada enda rakenduse uudseid võimalusi ja iga teine võib neid rakenduse võimalusi ka kasutada. See sama mehhanism võimaldab kasutajal komponente ka asendada.

Kõikide rakenduste alusteks on järgnevad teenused ja süsteemid:

- Rikas ja laiendatav vaadete komplekt, mida saab kasutada rakenduste ehitamiseks. Lisaks loendid, väljad, teksti kastid, nupud ja isegi integreeritav interneti sirvija.
- Sisu lisaja, mis võimaldab rakendustel ligipääsu teistele rakendustele või jagada enda andmeid teistele.
- Ressursihaldur võimaldab ligipääsu kodeerimata ressursidele nagu lokaliseeritud stringid , graafika ja kujundusfailid.
- Teadetehaldur, mis võimaldab kõikidel rakendustel kuvada teateid olekuribal.
- Tegumihaldur, mis haldab rakenduste elutsüklit ja tagab navigatsiooni töö.

1.3. Teegid

Android hõlmab endas C ja C++ teeke, mida erinevad Androidi süsteemid kasutavad. Neid võimalusi saavad arendajad kasutada ka Androidi rakenduste juures. Mõned olulisemad teegid on järgmised:

- Süsteemi C teek – standardsest C süsteemi teegist (libc) implementeeritud BSD'st (*Berkley Software Distribution*) tuletatud ning kohandatud Linux'i põhjale loodud seadmetele.
- Meedia teegid – teegid on tehtud *Packet Video Open CORE* põhjal. Teegid toetavad mitmete levinumate heli , video formaatide ning staatiliste pildifailide taasesitust ja salvestust. Nendeks on *MPEG4, H.264, MP3, AAC, AMR, JPG* ja *PNG*.
- Pinnahaldur – haldab ekraani alamsüsteemile ligipääsu ja märkamatuult liidab *2D* ja *3D* graafika kihid erinevatelt rakendustelt kokku.
- *LibWebCore*– kaasaegne brauseri mootor, mis jooksub Androidi brauseri ja interneti vaate implementatsiooni tööd.
- *SGL* – *2D* graafika alusmootor.
- *3D* teegid – realisatsioon, mis põhineb *OpenGL ES 1.0* rakendusliidestel. Teegid kasutavad riistvara *3D* kiirendeid (kui võimalik) või hästi optimeeritud *3D* tarkvara.
- *FreeType* – rasteri või vektori kirjatüübi renderdus.
- *SQLite* – võimas ja kerge relatsiooni andmebaasi mootor, mis on kasutamiseks kõikidele rakendustele.

1.4. Androidi käituskeskkond

Android hõlmab endas tuumik teekide kogu, mis tagab enamuse funktsionaalsuse ning on saadaval Java programmeerimiskeele põhilistes teekides.

Iga Androidi rakendus töötab oma protsessis ning enda *Dalvik* virtuaalmasina koopias. *Dalvik* on kirjutatud nii, et seade on võimeline jooksumata mitmeid virtuaalmasinaid efektiivselt. *Dalviki* virtuaalmasin käivitab failid *Dalvik Executable (.dex)* formaadis: see on optimeeritud nii, et ta kasutaks minimaalselt mälu. Virtuaalmasin on registripõhine ja käitab klasse, mis on Java programmeerimiskeele kompilaatori poolt kompileeritud ja muudetud *.dex* formaadiks kompilaatoris oleva dx tööriista abil.

Dalvik virtuaalmasina alusfunktsioonid, lõimumised ja madala taseme mäluhaldus, toetuvad Linux kernelil.

1.5. Linux Kernel

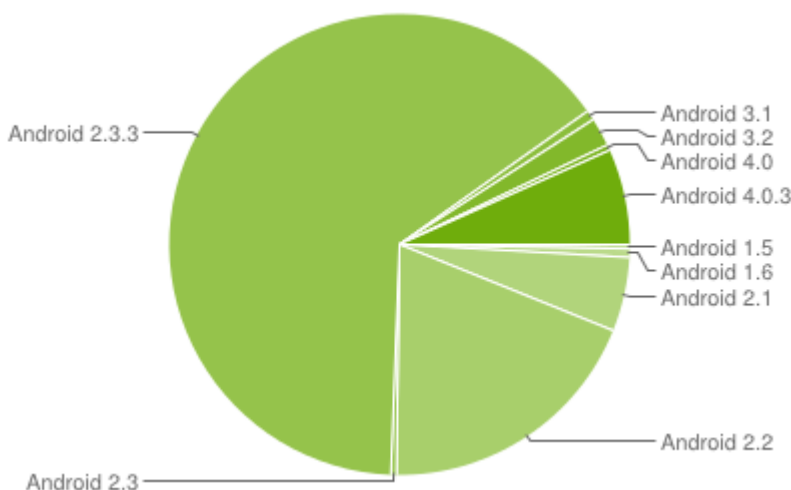
Androidi põhi süsteemi teenused, turvalisus, mäluhaldus, protsessihaldus, draiverite mudel, võrgu komponendid, toetuvad Linuxi 2.6 versioonile. Kernel kujutab endast ka abstartkset kihti riistvara ja ülejäänud tarkvara hulga vahel.

(Android app market, 2012)

2. Platvormi versioonid

Peatükk annab ülevaate Androidi versioonidest ja nende erinevustest. Millised omadused on enamlevinud versioonidel, mida on paremaks muudetud ja millele on rõhku pandud uuenduste käigus. Samuti saab ülevaate Androidi versioonide hetke osakaalust turul. Kõik eelnevad Androidi rakendusliidesed on ühilduvad järgmiste versioonidega.

2.1. Androidi versioonid



Joonis 2. Joonis põhineb GooglePlay keskkonda sisenenud seadmete hulgal, mis on kogutud 14 päeva jooksul, lõppkuupäevaga 1. juuni 2012.

(Android(2), 2012)

Kõige suurem turuosakaal on Android 2.3.3 (Gingerbread, Rakendusliidese tase 9) käes (64,6 %). Järgmisena tuleb Android 2.2 (Froyo, Rakendusliidese tase 8) (19,1%) ja kolmandana Android 4.0.3 (Rakendusliidese tase 15) (6,7%). Kui tahetakse teha rakendust kõikidele versioonidele, siis tuleb kasutada Android 1.5 rakendusliidest. See omakorda kitsendab arusaadavalt rakenduse ehitamisel võimalusi, sest uute rakendusliideste uusi omadusi kasutada sel juhul ei saa.

2.1.1. Android 2.2

Android 2.2 versioon sai alguse 20. mail 2010 . Rakendusliidesele on pandud numberidentifikaator 8, mis on süsteemi endasse salvestatud. Seda identifikaatorit kutsutakse rakendusliidese astmeks. Süsteem teeb selle identifikaatori järgi kindlaks, kas rakendus on vastava versiooniga ühilduv, enne rakenduse installeerimist.

Platvormi tehnoloogiate meedia raamistiku uuendusteks võib lugeda järgmisi:

- Uus meedia raamistik (*Stagefright*), mis toetab lokaalset faili taasesitust ja *HTTP* progressiivset voogesitust.
- *OpenCore* raamistiku tuge jätkatakse Android 2.2 platvormil.

Sinihamba uuendused:

- Häälvalimine sinihamba vahendusel.
- Võimalus jagada kontakte teiste telefonidega.
- Tugi auto ja laua dokkidele, millel on sinihammas.
- Uuendatud ühilduvusloogika auto seadmetega ja peakomplektidega.

2.6.32 kernel uuendus:

- *HIGHMEM* tugi.
- *SDIO* planeerimine ja sinihamba allsüsteemi uuendused.

HIGHMEM võimaldab seadmele lisada rohkem mälu kui 256mb. *SDIO* planeerimine vähendab ajakulu andmepakettide saatmisel riistvarale.

(Android(3), 2012)

2.1.2. Android 2.3

Android 2.3 versioon valmis 6. detsembriks 2010 aastal. Rakendusliidese numberidentifikaatoriks on 10. Nimetatakse seda versiooni koodnimega *Gingerbread*. Meedia raamistikule on tehtud järgmised uuendused:

- Uus meedia raamistik asendab täielikult *OpenCore* raamisiku, jättes alles kõik eelnevad koodekite/konteinerite toe kodeerimiseks ja dekodeerimiseks.

- Integreeritud tugi *VP8* avatud video pakendatud formaadile ja *WebM* avatud konteiner formaadile.
- Lisandub *AAC* kodeering ja *AMR* lairiba kodeering.

Linux kernel on uuendatud versioonile 2.6.35. Uuendusi on tehtud ka telefoni võrgule. Lisatud on *NFC* (*near field communication*) tugi, *SIP* pinu ja *BlueZ* pinu(sinihamba protokollipiinu). *Dalvik* käitusajas on uuendusi tehtud *Dalvik* virtuaalmasinale ja tuumik teekidele.

(Android(4), 2012)

Diagrammilt on näha, et kõige rohkem kasutatakse Android 2.3.3 versiooni. Selle versiooniga uusi lisavõimalusi ei tulnud, kuid palju parandati olemasolevate funktsionaalsusi ja tehti neid paremaks. See versioon oli mõeldud ka *API* vigade parandamiseks. Android 2.3.3 on niivõrd paljude kasutatavate seadmete peal ilmselt seetõttu, et sellel hetkel kui Androidil oli kõige stabiilsem ja uusim versioon 2.3.3 tehti hulgaliselt seadmeid, mis turul ka edu saavutasid.

2.1.3. Android 4.0

Seadmeid, mis töötasid Android 4.0 versiooni peal tutvustati esimest korda 19. oktoobril 2011. Rakendusliidese numberidentifikaatoriks on 14. Seda Androidi versiooni kutsutakse nimega *Ice Cream Sandwich*. Platvormi uuendusteks võib välja tuua järgmised omadused:

- *Android Beam* põhineb *NFC* tehnoloogial ja see loob kahe Androidi seadme vahel *NFC* ühenduse, kui nad on paari sentimeetri kaugusel teineteisest ja näitab kasutajaliidest, mis on andmevahetuseks mõeldud.
- Android 4.0 ühine kasutajaliides võimaldab luua telefonidele, tahvelarvutitele ja teistele seadmetele samu rakendusi.
- *Wi-Fi Direct* võimaldab seadmeid ühendada omavahel ilma interneti ja *hotspot* ühenduseta.
- Näotuvastus võimaldab enne telefoni kasutamist kindlaks teha näo järgi, kas tegemist on mobiili omanikuga.
- *Bluetooth HDP (Health Device Profile)* võimaldab seadmeid ühendada juhtmevabalt meditsiiniliste seadmetega ja sensoritega, mida kasutatakse haiglates või treeningsaalides.

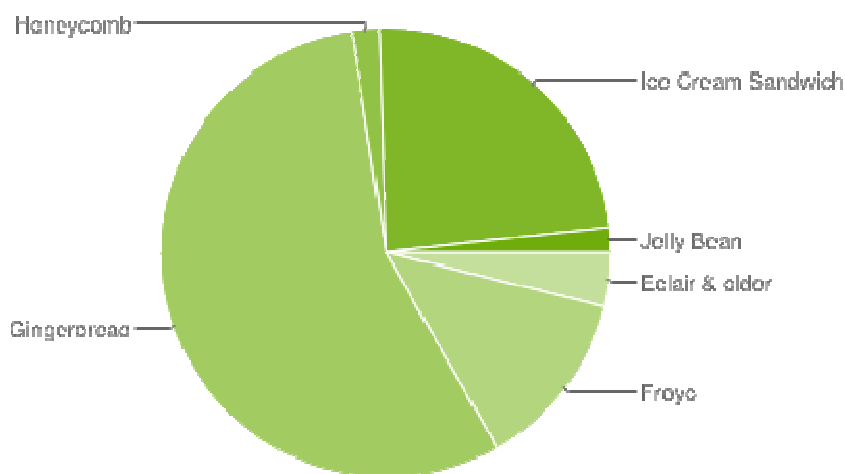
(Android(5), 2012)

2.2. Uued Android seadmed

Android 4 versioon on põhiliseks operatsioonisüsteemiks uutele nutitelefonidele ja tahvelarvutitele. Uusimateks ja populaarsemateks telefonideks, mis jooksevad selle versiooni peal on *Samsung Galaxy S III* ja *HTC one X +*. Mõlemad kasutavad Android *Jelly Bean* (versioon 4.1.x) versiooni. Nende kahe telefoni kasutamiskogemuse teeb erinevaks telefonide kasutajaliidesed, mis on tootjate poolt välja arendatud. HTC telefonid kasutavad *HTC Sense* kasutajaliidest ja Samsung *TouchWiz* kasutajaliidest. Kasutajaliidesed ongi ühtedeks põhilisteks telefonitootjate vahenditeks, (lisaks riistvarale ja disainile), millega üritatakse üksteist üle trumbata kasutusmugavuse ja kiiruse poolest. Tahvelarvutite ja nutitelefonide põhiline erinevus on ekraani suurus. Mõlemad seadmed kasutavad Android operatsioonisüsteeme, kuid natuke erinevaid kasutajaliideseid.

2.3. Värsked andmed

Algselt kogutud andmed on seisuga 1. juuni. Nüüd lisatakse andmed 1. oktoobri seisuga, et oleks näha, kas andmed on selle ajaga muutunud ja kui on siis kuidas.



Joonis 3. Joonis põhineb GooglePlay keskkonda sisenenud seadmete hulgal, mis on kogutud 14 päeva jooksul, lõppkuupäevaga 1. oktoober 2012.

Vaatluse all on taas kolm põhilist ja domineerivamat Android versiooni turul:

- Android 2.3.x - *Gingerbread*

Kõige domineerivam on endiselt Androidi *Gingerbread* versioon. See versioon on olnud peal nendel seadmetel, mis 14 päeva jooksul on sisenenud Google Play keskkonda. Protsentides on selliste seadmete osakaal 55,8%.

- Android 4.0.x - *Ice Cream Sandwich*

Järgmisena on kasutatavuse poolest üks olulisemaid versioone *Ice Cream Sandwich*. Protsentide järgi osakaal 23,7%.

- Android 2.2 - *Froyo*

Kolmandale kohale jääb versioon, mis kannab nime *Froyo*. Protsentides kasutatavuse osakaaluga 12,9%.

(Android(2), 2012)

Võrreldes esialgsete andmetega, siis esikoha positsioonid versioonide kasutatavuse osas ei ole muutunud. Küll aga on *Ice Cream Sandwich* edastanud *Froyo* versiooni. Päril suure tõusu on teinud Android 4 versioon. Seda ilmselt saab põhjendada uute seadmete müügiletulekuga, mis kasutavad Androidi kõige viimast versiooni. Inimesed vahetavad vanad mobiilid uute vastu ja sisenevad *Google Play* keskkonda juba uute seadmetega. Seda kajastab ka diagramm.

3. Rakenduste arendus

Selleks, et Android operatsioonisüsteemi peale arendama hakata, on tarvis Android *SDK* (*Software development kit*) tarkvara. See tuleb alla laadida lehelt developer.android.com/sdk/index.html. Installeeriti kõige uuemad Android rakendusliidesed ja kasutatud on arendamiseks Eclipse (*eclipse-jee-indigo-SR2-win32*) arendusplatvormi. Eclipse programmiga Android rakenduste arendamiseks on vajalik Eclipse keskkonda paigaldada *ADT* (*Android development tools*) liides. Samuti on tarvis arvutisse paigaldada *jre* (*java runtime environment*), et oleks võimalik java faile kompileerida.

Kui installida Eclipse keskkonna kaudu (abi -> paigalda uut tarkvara) Android'i arendustööriistu, siis kasutades hüperlinki <http://dl-ssl.google.com/android/eclipse/>, antakse valida, kas tahetakse installida *Developer tools* või *NDK Plugins*. Viimane neist sisaldab endas Androidi pardatööriistu, mis lubab arendada kiiremaid rakendusi. Rakenduse ülesehitus jääb siiski samaks ning rakendus pakendatakse samuti *.apk* formaati nagu seda tehakse Eclipse *Developer tools* liidest kasutades.

Kui vajalikud arendustööriistad on Eclipse keskkonda paigaldatud, ilmuvad üles tööriistaribale kaks Android platvormiga seonduvat ikooni.

Selleks, et oleks võimalik Eclipse arenduskeskkonnas tehtud rakendusi testida telefoni peal on tarvis arvutisse laadida *Google USB driver*. Samuti tuleb korra käivitada fail mis asub *C:\Users\<(kasutajakaust)\android-sdks\platform-tools\adb.exe*, et eclipse lõplikult aru saaks, et Android seade on arvutiga ühendatud usb kaabli abil.

Võimalus on ka kasutada Androidi virtuaalset seadet, millel tuleb eelnevalt paika panna Androidi versioon. Kui hakata rakendust testimise läbi Eclipse keskkonna, siis ta annab algselt valida, kas kasutaja soovib kasutada virtuaalset seadet või kui on telefon ühendatud usb kaabliga arvuti külge, siis on võimalus kasutada testimiseks seda. Vastasel juhul, kui virtuaalne seade on ainuke variant ja see on ka eelnevalt seadistatud, siis hakkab rakendus kohe seal jooksuma kui vajutada ülevalt nuppu käita rakendus (*Run application*). Autor eelistas testida rakendusi telefonil, kuna rakenduse installeerimine seadmesse on kiirem ja virtuaalse seadme üleslaadimine ajakulukam.

3.1. Rakenduse ülesehitus

Kui teha uus rakenduse projekt Eclipse platvormi peal, siis tekitatakse uus Android rakenduse failide struktuur ehk raamistik. Projekti struktuuri kõige ülemisel tasemel on projekt ise. Sellele saab nime anda uue projekti tekitamisel ja vajadusel on võimalik seda nime ka hiljem muuta. Järgmisel tasemel on kaust, mis kannab nime *src* (source) ehk allikas. See kaust sisaldab Java lähtefaile ja neid faile organiseeritakse vastavalt paketi struktuurile. Järgmine kaust kannab nime *gen* (*Generated Java Files*) ehk genereeritud Java failid, mis on samuti allika kaust, kuid need failid erinevalt *src* kaustast kuhu saab ise faile juurde teha, tekitatakse *gen* kausta failid automaatselt Androidi platvormi poolt. Edasi tuleb Androidi kaust koos versiooni numbriga. See sisaldab endas projektis vajaminevaid teeke. Eelviimane kaust on *res* (resources) ehk ressursid või vahendid, mis sisaldab endas välise päritoluga faile. *Res* kaust koosneb kolmest alamkaustast:

- *res/drawable*

Kaust sisaldab kõiki pilte. Kui on Androidi rakenduses vaja kasutada ikooni või pilti, siis tuleks see paigutada siia.

- *res/layout*

Sisaldab endas kasutajaliidese kujundusi, mida projektis kasutatakse. Need kujundused talletatakse *XML* formaati.

- *res/Values*

Sarnaselt eelmise kaustaga, sisaldab ka see *XML* faile, mis koosnevad olulistest väärtustepaaridest, millele viidatakse rakenduses. Need *XML* failid tekitavad massivid, värvid, moodsused, sõned jne. Väärtused on eraldi *XML* failis, et neid saaks kasutada lokaalselt ilma, et tuleks lähtekoodi muuta. Näiteks sõnum rakenduses võib olla erinevates keeltes.

Kaustas *assets* on samuti välise päritoluga failid, mida kasutatakse rakenduses. Peamine erinevus *res* kaustaga on see, et nad on talletatud niiöelda toorelt kausta ning on programmeerimise poolelt vaadatuna ainult loetavad (*read only*).

Otse projektikaustas on fail nimega *AndroidManifest.xml*. Selles failis on meta informatsioon vastava Androidi rakenduse kohta ning on iga rakenduse juures oluline. See *XML* fail sisaldab endas informatsiooni erinevate tegevuste, vaadete, teenuste ja paljude muude asjade kohta.

Selles sisaldub ka kasutajate õigused, mis on selle Androidi rakenduse jooksutamise jaoks vajalikud.

(Sudarmuthu, 2012)

3.2. Rakenduse installifail

Kui rakenduse testimine on edukalt kulgenud, siis oleks järgmiseks sammuks rakenduse *.apk* installifaili tegemine, millest Android operatsioonisüsteem aru saaks. Selleks tuleb Eclipse keskkonnast eksportida rakenduse failstruktuur ehk parema hiireklahviga rakenduse peale vajutada ja valida *Export*. Seejärel uuest ilmunud aknast valida *Export Android application* ja järgida juhiseid, kuni on võimalik salvestada *.apk* fail. Selliselt installifail tehes luuakse rakendusele niiöelda *self-signed* sertifikaat. See, et Androidi rakendusel on arendaja poolt tehtud sertifikaat on enamjaolt tavaline. Ühtegi rakendust, millel ei ole sertifikaati, ei ole võimalik installeerida ühelegi Androidi operatsioonisüsteemiga telefonile.

Kui rakendus on valmis kasutamiseks ka teistele inimestele, tuleb rakendusele teha *release mode* sertifikaat, mille kohta saab täpsemat informatsiooni lehelt: <http://developer.android.com/tools/publishing/app-signing.html> ning sealt tuleks otsida osa pealkirjaga *Signing in Release Mode*.

(Android(6), 2012)

4. Rakendus *helloworld*

Selleks, et hakata ükskõik, millist rakendust arendama on esialgu vaja teada, kuidas saada telefonis rakenduse pilt ette ja mingisugune kirje sinna kuvama. Seeläbi saab ka teada, et struktuur rakendusel on õige ning kõik failid seal toimivad ning failidel ei ole vigu, mis kompileerimist takistaks. Eclipse keskkonda kasutades on sellise rakenduse ehitamine sama lihtne kui uue projekti tekitamine arenduskeskkonda. Rakenduse struktuur tehakse Eclipse ja Android tööriistade poolt ise valmis. Vaikeväärtusena on valmis seatud üks muutuja, mille väärtus on sõne *Hello World*. See kuvatakse rakenduse kohesel käivitamisel ka telefoni ekraanil, kui rakendus on installeeritud telefoni peale.

Ülesande raskemaks osaks on ilmselt kõikide arendamiseks vajalike programmide installeerimine ja tööle saamine. Selleks, et see etapp edukalt läheks on tarvis järgida samme, mida kirjeldati peatükis rakenduse arendus.

Tähtsamate ülesannete järjestus tuuakse välja järgnevas loetelus:

- Esmalt tuleb alla laadida Oracle kodulehelt *jre* (java runtime environment) ja see arvutisse installeerida (*Windows x86 offline* versioon soovitatav). Selleks, et testida, kas arvuti peal on *jre* juba olemas saab kasutada järgmist java lehekülge <http://www.java.com/en/download/testjava.jsp>.
- Seejärel tuleks arvutisse tõmmata ja lahti pakendada Eclipse arenduskeskkond (versioon *eclipse-jee-indigo-SR2-win32*).
- Järgmiseks kasutades Eclipse arenduskeskkonna võimalusi installeerime peale Androidi operatsioonisüsteemile arendamiseks vajalikud Eclipse lisavahendid. *Help - > Install new software* võimaldab meil uusi lisaprogramme installeerida. Sisestades avanenud aknasse lingi <http://dl-ssl.google.com/android/eclipse/> saame installeerida *Developer tools* arendusvahendid.
- Eclipse annab teada, et uuenduste tõttu on tarvis programmile teha taaskäivitus ja peale seda võimaldab lisa aken installeerida Android *SDK*. Võimalus on valida, kas kasutada kõige uuemat Android versiooni või versiooni Android 2.2, mille rakendused töötavad 93% hetkel kasutusel olevatest telefonide ja tahvelarvutite peal.
- *New -> Project* ja avanenud aknast valides *Android -> Android Application Project* avab meile akna, kus saame alustada tööd rakenduse seadetega.

- Nüüd kui rakendus on valmis ja struktuur nähtav vasakul pool Eclipse arenduskeskkonnas tuleks vajutada rakenduse kõige ülemise kausta peale ja siis Eclipse ülevalt tööriistaribalt *run*. kui tahetakse jooksutada rakendus telefoni peal, siis tuleks eelnevalt *usb* juhtmega telefon ühendada arvuti külge. Vastasel juhul on võimalik rakendust testida ka virtuaalse seadme kaudu, mis tuli Android *sdk* arendustööriistadega kaasa.

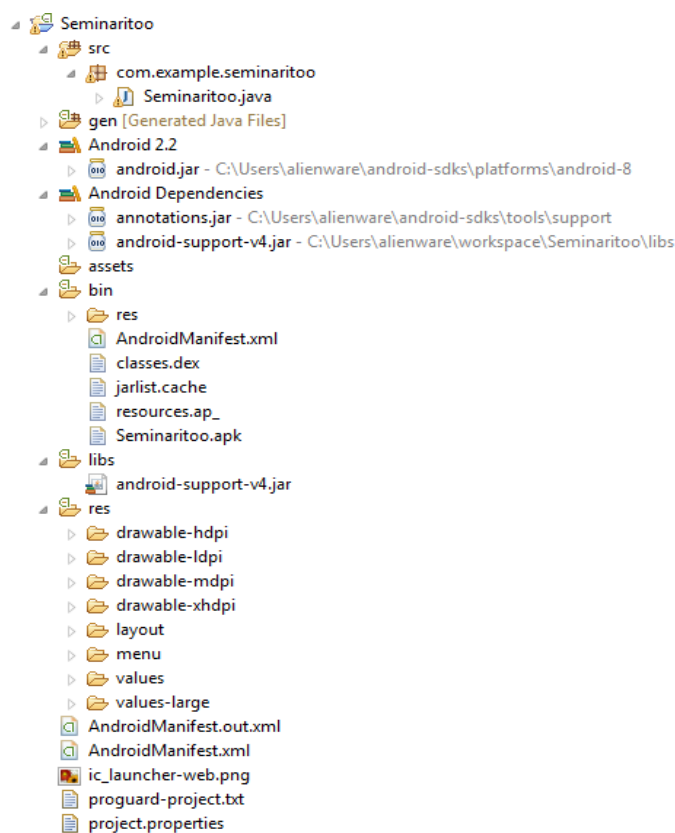
Kui kõik õnnestus edukalt peaks nüüd mobiiltelefoni ekraanil (või virtuaalses seadmes arvutis) olema nähtav kiri *Hello World*.

5. Seminaritöö kuvamine läbi rakenduse

Selles peatükis tutvustan rakendust Semestritöö.apk. Tegemist on autori poolt tehtud Androidi rakendusega ja seletatakse lahti, kuidas seda arendati ja millised olid põhilised probleemid, selle ehitamise käigus.

Eesmärk oli valmis teha rakendus, mis suudaks seminaritööd kuvada läbi Android rakenduse telefonis ja läbi internetivahenduse, et tegemist oleks dokumendiga, mis on kõige uuem versioon. Rakenduses on olemas teated kasutaja jaoks ning nupp, mis viib interneti olemasolul semestritöö *pdf* faili juurde. Läbi *google pdf* lugeja lahenduse on see kasutajale ka nähtav ning loetav. Seal saab lehekülgi vahetada, lehekülgi suurendada ja vähendada lugemise mugavuse eesmärgil.

5.1. Ülesehitus



Joonis 4. Seminaritöö.apk struktuur

Rakenduse kõige ülemkaustaks on Seminaritöö, kus kõik teised olulised alamkataloogid asuvad. Seminaritoo.java failis asuvad vajalikud funktsioonid ehk rakenduse käitamiseks vajaminevad toimingud java programmeerimiskeeles. Failstruktuuris on näha ka, mis Androidi versiooni peale on rakendus tehtud (Android 2.2) ja versiooni numberidentifikaator (android-8).

AndroidManifest.xml faili tuli ära märkida, et rakendus tohib kasutada interneti ühendust. *AndroidManifest.out.xml* fail luuakse sel juhul, kui *AndroidManifest.xml* fail on paremal pool lahti ja rakendust proovitakse käivitada samal ajal. Rakenduse käivitamisel luuakse igast lahtiolevast *xml* failist Eclipse keskkonnas *.out.xml* fail, mida loetakse Eclipse arenduskeskkonna veaks. Kui see häirib rakenduse installeerimist telefoni peale või virtuaalseadmele, siis võib selle lihtsalt eemaldada, kuna see ei ole oluline fail rakenduse jaoks.

Kaustas *res* on kaust *layout*, milles asub *activity_seminaritoo.xml*. Seda faili saab Eclipse'is vaadata graafilises vaates (*Graphical Layout*) ja *xml* vaates. Graafilises vaates näeb ära selle rakenduse visuaalse poole ja kuidas see välja näeb telefonis. Paigutuse tegemisel on seda mugav kasutada, kuna ei pea rakendust koguaeg telefoni ümber installima, et realselt rakendust näha. Graafilises vaates on võimalik lisada tekstialasid, nuppe, pilte ja palju muud vasakul olevast menüüst lohistades. Kõiki samu asju saab kirjutada koodina *xml* vaates. Kuna tegemist on taas *xml* failiga, siis tuleks ka see fail ennem rakenduse käivitamist sulgeda (vastasel juhul võib Eclipse tekitada *out.xml* faili, mis takistab rakenduse installimist).

Res kaustas on veel kaust nimega *values*, kuhu lisati *strings.xml* fail ja sinna sisse muutujad, mida kasutati. Näiteks kirjeldati seal ära sõne tu, mille väärtus on Tallinna Ülikool.

5.2. Rõõmud ja mured rakendusega

Kui autor arenduskeskkonda üles seadsin, siis algselt prooviti seda üles panna Ubuntu 10.04 peale. Eclipse on Ubuntu peal täiesti olemas ning olemas on ka Android *sdk*, mis peaks Eclipse'i peale minema. Paraku kõik nii ei läinud nagu oleks võinud arvata. Autoril ei õnnestunud Eclipse arenduskeskkonna peale Android *sdk* lisa panna ja sinnapaika see ka jäi. Idee, miks panna kogu arendus just Linux Ubuntu peale, tuli sellest, et Android kasutab Linux Kernelit ja arvati, et ühilduvus on nende kahe operatsioonisüsteemi vahel parem just seal. Paraku katsetused sellist tulemust ei andnud.

Eclipse keskkonna ja Android arenduskeskkonna installimisel *Windows 7 Home Premium* peale alustati Eclipse allalaadimisega Oracle kodulehelt. Kuna kasutusel oli Windows 64 bit'ine, siis tõmmati ka sellele vastav versiooni. Hiljem selgus, et *eclipse-jee-indigo-SR2-win64* versioon ei ühildu Android *sdk* arenduskeskkonnaga ning tuli hoopis kasutada *eclipse-jee-indigo-SR2-win32* versiooni.

Lisaks igasugustele pisematele süntaksi vigadele koodis tekkis rakenduse käivitamisel korra probleem *R.java* failiga, mida genereeritakse automaatselt. Seda parandama hakata on ilmselt ülemäära tülikas ning kui see fail annab veateateid, siis tuleks valida Eclipse keskkonnas ülevalt tööriistaribalt *Project* ja rippmenüüst *Clean*. See tekitab uue *R.java* faili, mis peaks ka veateate selle faili kohta eemaldama ja võimaldab projekti uuesti pakendada *.apk* faili.

Kokkuvõte

Androidi operatsioonisüsteem on üks enamlevinumaid telefonide operatsioonisüsteeme. Seadmeid, mis just Androidi peal jooksevad on üpris palju (nutitelefonid, tahvelarvutid, väiksemad sülearvutid jne). Seetõttu on minu arvates oluline neid uurida ja neist rohkem teada.

Uurimuste kohaselt on kõige levinum Androidi versioon 2.3 ning pidades meeles, et ajaga kaasas käimine on vajalik, kui tahta jätkusuutlikuid rakendusi luua, siis hetkel tuleks keskenduda versioonile, mida kõige rohkem kasutatakse. Kuna rakendused, mis on tehtud varasemate versioonide peal, töötavad ka uuemate peal, siis usun, et Androidi 2.3 versiooni peale rakendust tehes on tulemus kasutatavuse osas kõige parem. Ise ma kasutasin *Seminaritoo.apk* rakendust tehes küll versiooni 2.2, kuid sellest madalamale ei tasu versiooniga minna, kuna vanemate versioonide turuosa on liialt väike.

Rakenduste arenduse meetodi kohapealt usun, et Eclipse on kõige parem tasuta arenduskeskkond, mille abil saab neid luua. Samuti on selline viis ka kõige levinum ning ilmselt ka seetõttu, et paremat varianti lihtsalt pole. Millist arendamise meetodi kasutada on muidugi harjumuse küsimus.

Rakenduse ülesehitus oli minu jaoks lõpuks loogiline ja igati arusaadav. Natuke oli võõras Java programmeerimiskeel, kuna igapäevaselt kasutan tööl php keelt ja javascript'i. Sarnasusi leidis Java keelest Actionscript keelega. Minu arusaamist mööda Adobe Flash'i enam edasi ei arendata ning samuti ka Actionscript 3 võib jääda viimaseks versiooniks. Androidi jaoks flash'i ei eksisteeri, kuna Android Chrome veebilehitseja jaoks flash tugi puudub ja seda ei tulegi (Arstechnica, 2012). Positiivne külg on see, et kuna me koolis actionscripti õppisime ei jookse see mööda külge siiski maha, kuna neid teadmisi saab mõneti kasutada java keele juures.

LISAD

Mõistete ja lühendite tõlked ja kirjeldused

Library - teek

GooglePlay - keskkond, kus saab Android rakendusi osta ja alla laadida. (Nüüd nimetatakse seda *PlayStore*)

API (Application Programming Interface)– rakendusliides

Thread– lõim (Lõimed on informaatikas operatsioonisüsteemi või programmi omadus jaotuda mitmeks protsessiks, mis eeldab, et protsess operatsioonisüsteemis võib koosneda mitmest voost, mis täidetakse paralleelselt).

Progressive streaming – progressiivne voogesitus (fail salvestatakse kohalikku mäluseadmesse ja alles siis mängitakse kasutajale ette).

Stack - pinu

QR koodid seminaritöös koostatud rahenduste installifailide juurde

Siit leiab seminaritöö raames valminud rakenduste installifailid. Selleks tehti QR koodidega otseteed failideni. QR koodide skanneerimiseks kasutas autor rakendust *Barcode Scanner*. QR koodid on genereeritud aadressil <http://qrcode.kaywa.com/>. Installifailid asuvad autori enda serveris.

Mõlemad installifailid on kättesaadavad aadressil: <http://raul.matu.pri.ee/Android/>.

Hello World rakendus (*helloworld.apk*).



Seminaritöö rakendus (*Seminaritoo.apk*).



Kasutatud kirjandus

Android(1). (2012). What is Android?

<http://developer.android.com/guide/basics/what-is-android.html>

Android app market. Android architecture.

<http://www.android-app-market.com/android-architecture.html>

Android(2).(2012). Platform Versions.

<http://developer.android.com/resources/dashboard/platform-versions.html>

Android(3). (2012). Android 2.2 Platform Highlights.

<http://developer.android.com/about/versions/android-2.2-highlights.html>

Android(4). (2012). Gingerbread.

<http://developer.android.com/sdk/android-2.3-highlights.html>

Android(5).(2012). IceCreamSandwich.

<http://developer.android.com/sdk/android-4.0-highlights.html>

Android(6).(2012). Signing your application.

<http://developer.android.com/tools/publishing/app-signing.html>

Sudarmuthu.(2012). The structure of an Android project

<http://sudarmuthu.com/blog/the-structure-of-an-android-project>

Arstechnica.(2012). Adobe lays out the future for Flash: a platform for the next 5-10 years.

<http://arstechnica.com/business/2012/02/adobe-lays-out-the-future-for-flash-a-platform-for-the-next-5-10-years/>