

**Ville Isopahkala**

**MAGIC: THE GATHERING APUSOVELLUS ANDROIDILLE**

**Opinnäytetyö  
CENTRIA-AMMATTIKORKEAKOULU  
Tietotekniikan koulutusohjelma  
Tammikuu 2017**

**TIIVISTELMÄ OPINNÄYTETYÖSTÄ**

<b>Centria ammattikorkeakoulu</b>	<b>Aika</b> Tammikuu 2017	<b>Tekijä/tekijät</b> Ville Isopahkala
<b>Koulutusohjelma</b> Tietotekniikka		
<b>Työn nimi</b> Magic: The Gathering apusovellus androidille		
<b>Työn ohjaaja</b> Kauko Kolehmainen	<b>Sivumäärä</b> 25 + 2	
<b>Työelämäohjaaja</b> Kauko Kolehmainen		
<p>Opinnäytetyönä oli omavalintainen android-sovellus Magic: The Gathering –korttipelille. Tavoitteena oli toteuttaa akkuystävällinen apusovellus kyseistä peliä pelaaville käyttäen android studiota. Työssä tutustutaan javaan, androidiin sekä android studioon, niiden historiaan sekä ominaisuuksiin. Magic: The Gathering:iin tutustutaan perustasolla. Opinnäytetyö keskittyy sovellukseen, sen luomiseen, koodauskieleen sekä alustaan. Tarkoituksena ei ole opettaa pelaamaan Magic: The Gatheringiä.</p>		

<b>Asiasanat</b> Android, Android Studio, Java, Magic: The Gathering, Mobiiliohjelmointi, käyttöjärjestelmä
--

## ABSTRACT

<b>Centria University of Applied Sciences</b>	<b>Date</b> June 2016	<b>Author</b> Ville Isopahkala
<b>Degree programme</b> Information Technology		
<b>Name of thesis</b> Magic: The Gathering assistance software for Android		
<b>Instructor</b> Kauko Kolehmainen	<b>Pages</b> 25 + 2	
<b>Supervisor</b> Kauko Kolehmainen		
<p>The thesis work was a self-selected android application for Magic: The Gathering –card game. The Objective was to implement a battery-friendly assistance application using android studio. The application is for anyone who plays Magic: The Gathering and has an android device. The thesis introduces Java, android and Android Studio, their history and features. The thesis has a basic level introduction for Magic: The Gathering. The thesis focuses on the application, its creation, programming and platform. The purpose of the thesis is not to teach how to play Magic: The Gathering.</p>		

<p><b>Key words</b> Android, Android Studio, Java, Magic: The Gathering, Mobile software developing, operating system</p>
---

## KÄSITTEIDEN MÄÄRITTELY

MTG	Magic: The Gathering. Peli, jolle tämä sovellus on kehitetty
SDK	Software Development Kit / Sovelluskehitys työkalu
API	Application Programming Interface / Ohjelmointirajapinta
ABI	Application Binary Interface / Ohjelmointi binäärirajapinta
IDE	Integrated Development Environments / Ohjelmointiympäristö
JVM	Java Virtual Machine / Java Virtuaali Kone
JDK	Java Development kit
JSE	Java Standard Edition / Java Standardi Versio
WWW	World Wide Web
AWT	Abstract Window Toolkit
GUI	Graphical User Interface / Graafinen käyttöliittymä
POSIX	Portable Operating System interface for UNIX
Java RMI	Java Remote Method Invocation
JIT	Just-In-Time
IDL	Interface Description Language
JCF	Java collections framework
JNDI	Java Naming and Directory Interface
JPDA	Java Platform Debugger Architecture
JDBC	Java Database Connectivity
NIO	Non-blocking Input / Output
GB	GigaByte / GigaBitti
RAM	Random Access Memory
WotC	Wizards of the Coast / Magic: The Gathering yritys
REL	Rules Enforcment Level

**TIIVISTELMÄ**  
**ABSTRACT**  
**KÄSITTEIDEN MÄÄRITTELY**  
**SISÄLLYS**

<b>1 JOHDANTO .....</b>	<b>4</b>
<b>2 JAVA .....</b>	<b>5</b>
2.1 Historia .....	5
2.2 Javan luomisen päätaivoitteet.....	6
2.3 Javan versiot .....	7
2.4 Javan merkittävimmät ominaisuudet.....	9
<b>3 ANDROID.....</b>	<b>10</b>
3.1 Historia .....	11
3.2 Android SDK .....	12
3.3 Androidin kehitys .....	12
<b>4 ANDROID STUDIO .....</b>	<b>14</b>
4.1 Android Studion ominaisuuksia .....	14
4.2 Android studion asennus .....	14
4.3 Laitteistovaatimukset.....	15
4.3.1 Windows .....	15
4.3.2 Linux .....	15
4.3.3 Mac.....	15
<b>5 MAGIC: THE GATHERING.....</b>	<b>16</b>
5.1 Korttityypit .....	16
5.2 Pelin kulku .....	18
5.3 Vuoron kulku.....	19
5.4 Pelin päättymisen.....	20
5.5 Peliformaatit .....	20
5.5.1 Limited .....	20
5.5.2 Constructed.....	21
5.5.3 Commander .....	21
5.5.4 Tiny Leaders .....	21
5.6 Pelitasot .....	21
5.6.1 Kasuaali.....	22
5.6.2 FNM.....	22
5.6.3 Competitive REL.....	22
5.6.4 Pro Tour.....	22
<b>6 POHDINTA .....</b>	<b>23</b>
<b>7 YHTEENVETO .....</b>	<b>25</b>
<b>LÄHTEET .....</b>	<b>26</b>
<b>LIITTEET .....</b>	<b>27</b>

**TAULUKOT**

TAULUKKO 1. Androidin käyttöjärjestelmiä.....	11
---	----

**KUVAT**

KUVA 1. Javan suunnitteluvaiheen tavoitteet.....	9
KUVA 2. Android-käyttöjärjestelmän pääkomponentit.....	12
KUVA 3. Activated Ability.....	17
KUVA 4. Static Ability.....	17
KUVA 5. Triggered Ability.....	17
KUVA 6. Keskenräinen pelitilanne.....	18
KUVA 7. Magic: The Gathering Platinum Angel.....	20
KUVA 8. Turnauspelaamista.....	22
KUVA 9. Oman sovelluksen aloitusruutu.....	23
KUVA 10. Oman sovelluksen 1 vs 1.....	24
KUVA 11. Oman sovelluksen 4 pelaajan EDH.....	24

## 1 JOHDANTO

Työssä käydään läpi Java-ohjelmointikieltä, sen merkittävimpiä ominaisuuksia, alkuvaiheen kehityksen tavoitteita sekä historiaa. Työssä tutustutaan myös androidiin, sen historiaan, androidin käyttöjärjestelmiin ja niiden eroavaisuuksiin toisistaan. Androidin sovelluskehitykseen sekä sovelluskehityspakkeihin ja virtuaalitodellisuuteen tutustutaan. Työssä tutustutaan Android Studioon, sen historiaan, kehitysalustaan, asennukseen, tuettuihin koodauskieliin, ominaisuuksiin sekä Magic: the Gatheringiin perustasolla. Android Studion asentamisen ohjeet ovat myös työssä.

Sovelluksella saadaan helpotusta pelitilanteisiin, esim. pitämään kirjaa kaikkien elämäpisteistä samalla ruudulla, nopanheittosimulaattorilla voi korvata oikeat nopat, kolikonheitolla fyysiset kolikot. Sovellus ei ole välttämätön, mutta saatavilla halukkaille, jotka eivät halua tai eivät omista kyseisiä noppia tai tarvikkeita.

## 2 JAVA

Java on yleiskäyttöön tarkoitettu ohjelmointikieli, joka on suunniteltu tarvitsemaan mahdollisimman vähän täytäntöönpano riippuvuuksia. Kerran kirjoitettu Java-koodi voidaan ajaa kaikilla alustoilla ilman muuttamisen tarvetta. Java on suunniteltu yksinkertaiseksi, jotta mahdollisimman moni ohjelmoija saavuttaisi sujuvuuden sen kanssa. Javassa on useita samankaltaisuuksia C ja C++ ohjelmointikieliin sekä muutama ominaisuus muista kielistä. Koonti-aika virheet tulee huomata koontivaiheessa ja ajovirheet ajon aikana, ja Javan vahva ja staattinen kirjoitus mahdollistavat näiden erottelun. Javassa on oma toimintonsa kokoamiselle sekä ajolle, kun jotkin kielet suorittavat molemmat toiminnot samasta nappulasta.

Koontiaika muodostuu ohjelman kääntämisestä laitteistoriippumattomaksi bitti-koodiksi. Ajoaika toimintoihin lasketaan mukaan lataaminen sekä tarvittavien luokkien linkittäminen ohjelman ajamiseen. Java on korkealaatuinen ohjelmointikieli, jossa koneen edustus ei ole saatavilla ohjelmointikielen kautta. Javan kanssa tulee automaattinen tallennustilan hallinta, eli turhien tai haitallisten tiedostojen poistaja jolla vältetään tiedostojen purkamisen turvallisuusongelmat. Java-kieli ei sisällä vaarallisia konstruktioita, kuten arrayn sisäänkäyntiä tarkistamatta indeksiiä koska tällaisten turvattomien konstruktioiden käyttö aiheuttaa ohjelman käyttäytymisen epävakauden. (Java Language Specification, 2015.)

### 2.1 Historia

Java-ohjelmointikieli on kehittynyt versio vanhemmasta ohjelmointikielestä nimeltä Oak. Oak:in loi Sun Microsystems 90-luvulla James Gosling, Mike Sheridan ja Patrick Naughton alustasta riippumattomaksi ohjelmointikieleksi jonka tarkoituksena oli mahdollistaa konsolipelien ja videonauhureiden kommunikointi. Oakin piti alunperin ilmestyä digiboksina tarjoamaan videota halutessa. Juuri kun sopimukset digibokseista olivat valmistumassa, internet alkoi levittäytymään ympäri maailmaa. Kun oakin kehittäjät alkoivat tunnistaa tämän trendin, heidän keskittyminen siirtyi internetiin ja WebRunneriin, joka on oakin mahdollistava selain. Oakin nimi vaihtui Javaan ja WebRunnerista tuli HotJava internet selain. Innostus internetiin veti puoleensa ohjelmistojen myyjät, ja ohjelmistot kuten Java development tools tuli saataville monilta myyjiltä nopeasti. Sama innostus varusti monet ohjelmistokehittäjät löytämään Javan ja sen monet huikeat ominaisuudet. Vuodesta 2016



lähtien Java on tunnetuin käytössä oleva ohjelmointikieli, jolla on noin 9 miljoonaa kehittäjää. (Java Programming, 1997.)

## 2.2 Javan luomisen päätavoitteet

Javan luomisvaiheessa oli pöydällä tavoitteita, jollanen Javasta pitäisi tulla. Tärkeimpänä ajatuksena oli luoda yksinkertainen kieli jotta ohjelmoijat oppisivat sen helposti, ilman pitkää harjoittelemista. Opittuaan peruskäsitteet ohjelmoijat voivat olla tuottavia ja tehokkaita alusta alkaen. Toisena ajatuksena oli tehdä Javasta yhteensopiva nykyisten ohjelmistojen käytäntöihin. Java-ympäristö suunniteltiin olio-pohjaiseksi, joka on pitkän tutkimuksen ja kehityksen jälkeen löytänyt tiensä riippumattomaksi tieksi ohjelmoinnissa.

Internetin ja WWW:n aikakaudella on tärkeää kehittää lujatekoisia ja luotettavia sovelluksia, erityisesti sähköisen kaupankäynnin ympäristössä. Java on suunniteltu luomaan luotettavia ja lujatekoisia sovelluksia tarjoamalla kaksinkertaisen tason tarkistuksen aloittaen koontiajasta jonka jälkeen on vuorossa ajo-aika tarkastus. Tarkkaan optimoidut ja todistetut ohjelmointikielen ominaisuudet ohjaavat kehittäjiä kohti luotettavan ohjelmoinnin käytäntöjä. JVM:n ohjaama muistinhallinta pitää huolen osoittimien hallinnasta ja suojaa muistivuodoilta. Ohjelmoijat voivat kehittää Java-sovelluksia luottamuksellisesti, sillä järjestelmä hoitaa virheet nopeasti.

Yksi Javan tavoitteista oli mahdollistaa toiminta hajautetuissa ympäristöissä, joten turvallisuusnäkökohdat olivat erittäin tärkeitä. Tuloksena Javan turvallisuusominaisuudet suunniteltiin kieleen ja ajo-aika järjestelmään, joten se on turvattu perustasta lähtien. Verkkoympäristössä myös Java-sovellukset ovat turvassa tunkeutumisyrittäiltä luvattomalta koodilta jolla yritetään päästä järjestelmään tehden siitä haavoittuvaisen. (Design goals of Java, Kaushki Pal, 2016.)

Toinen Javan päätavoite oli tehdä siitä arkkitehtuurineutraali niin, että sitä voidaan ajaa ja käyttää heterogeenisillä alustoilla sekä verkkoympäristössä. Tällaisessa ympäristössä sovelluksen on pystyttävä suorittaa erilaisilla laitteistoilla ja eri käyttöjärjestelmillä. Jotta Javaa pystyisi ajaa erilaisissa käyttöympäristöissä, Java kääntäjä luo tavukoodia joka on arkkitehtuurineutraalia väliformaatissa olevaa koodia joka on suunniteltu yhteensopivaksi useaan laitteistosta ja ohjelmistosta riippumattomaksi. Tämän seurauksena Java ympäristö ratkaisee sekä binääri jakelun että versio-ongelmat.

JVM on arkkitehtuurineutraalisuuden sekä kannettavuusominaisuuden yhdistetty Java-ympäristö. Periaatteessa JVM on määrittely abstraktille koneelle, jolle Java kääntäjät voivat luoda koodia joka on ajossa ympäristössä. JVM:n eritelmä perustuu POSIX liittymämäärittelyyn joka on alan standardi kannettavalle systeemin käyttöliittymälle. JVM:n implementointi on yksinkertaista mille tahansa arkkitehtuurille, mikäli kohdejärjestelmä tukee virtuaalikoneen perusvaatimukset.

Kaikille sovelluksille, erityisesti hajautetuille sekä internetiä käyttäville suorituskyky on merkittävä ominaisuus ja aina otettava huomioon. Java-ympäristö saavuttaa korkean suorituskyvyn käyttämällä mekanismeja joka mahdollistaa sen ajon täydellä nopeudella tarkistamatta ajoaikaympäristöä. Roskien kerääjät kulkevat erillisen säikeen varmistaen oikea-ajoitteisen muistin vapautuksen parantaen suorituskykyä. (Design goals of Java, Kaushki Pal 2016.)

### 2.3 Javan versiot

Ensimmäisestä versiosta joka julkaistiin vuonna 1996 viimeisimpään versioon 8.0 joka julkaistiin 2014 on kulunut lähes 20 vuotta Javan kehitystä, ja teknologiaan on tullut monia muutoksia ja parannuksia vuosien kuluessa. Isoimmat versiot julkaistiin kahden vuoden välein, mutta Java SE -versiot 7 & 8 vaativat useamman vuoden kehittämisen. Java SE 6:sen jälkeen meni 5 vuotta kunnes Java SE 7 oli julkaisuvalmis ja Java SE 8:n kehittämiseen meni 3 vuotta.

JDK 1.0 (Tammikuu 21, 1996) kutsutaan Oak:iksi. Ensimmäistä stabiilia versiota, JDK 1.0.2, kutsutaan Java 1:seksi.

JDK 1.1 (Helmikuu 19, 1997) ei koodinimeä, suuri päivitys edelliseen versioon sisältäen laajan uudelleentyöstetyn AWT:n tapahtumamallin, sisäistetyt luokat, JavaBeans luokat, datapankkien käyttöoikeusasetuksia, Java RMI, Ajonaikainen kääntäminen, Unicode ja kansainvälistymis tuki.

J2SE 1.2 (Joulukuu 8, 1998) kutsutaan Playground:iksi, graafinen API integrointi ydinluokille, JVM ajettiin JIT-kääntäjällä ensimmäistä kertaa, Java IDL, JCF.

J2SE 1.3 (Toukokuu 8, 2000) kutsutaan Kestrel:iksi. Suorituskykypäivitys, RMI tukipäivityksiä, JNDI siirtyi lisäosasta osaksi ydinluokkia, JPDA, JavaSound, synteettiset proxy luokat.

J2SE 1.4 (Helmikuu 6, 2002) kutsutaan Merlin:iksi. Poikkeusten liittäminen, IPv6 tuki, NIO, Integroidut turvallisuus ja salakirjoitus laajennukset.

J2SE 5.0 (Syyskuu 30, 2004) kutsuttiin Tiger:iksi. Useita kieli ominaisuuksia lisätty, standardi kirjastoihin parannuksia.

Java SE 6 (Joulukuu 11, 2006) kutsutaan Mustang:iksi. Kehityksen aikana parannuksia ja bugikorjauksia julkaistiin lähes viikottain, skriptauskieletuki, suorituskykyparannuksia, JDBC tuki, monia GUI ja JVM parannuksia.

Java SE 7 (Heinäkuu 28, 2011) kutsutaan Dolphin:iksi. JVM tuki dynaamisille kielille, tiivistetyt 64-bittiset osoittimet, pieniä kieli muutoksia, automaattinen resurssien käsittely, päivityksiä XML:ään ja Unicodeen.

Java SE 8 (Maaliskuu 18, 2014) Sisältää joitain ominaisuuksia jotka suunniteltiin Java SE 7:aan mutta siirrettiin tuonnemmaksi. Java SE 8 ei aluksi tukenut Windows XP:tä, mutta päivityksen numero 25 myötä se voidaan asentaa XP:lle.

Java SE 9 (Heinäkuu, 2017) Useita lykkäyksiä, alunperin piti julkaista vuonna 2016. Tulee sisältämään paremman tuen useamman gigabitin pinoille, paremman integroinnin koodiksi sekä itsetuunautuvan JVM.

Java SE 10 (Tulevaisuudessa) Spekulointia objekteista ilman identiteetti arvoja sekä muutos kohti 64-bittisiä osoitetaulukoita (array) tukemaan suuria tietomääriä. (Java SE Versions history, 2014.)

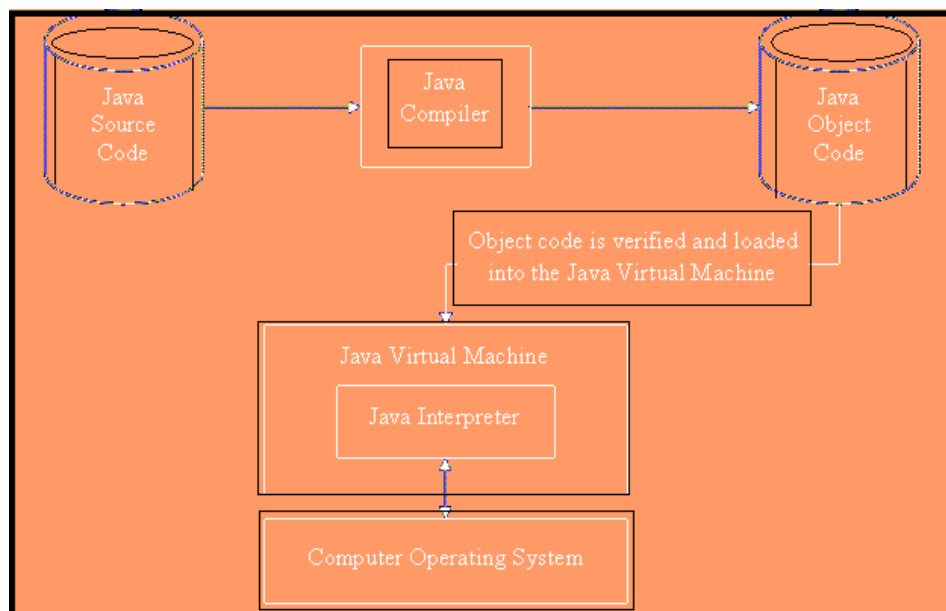
## 2.4 Javan merkittävimmät ominaisuudet

Javan merkittävimpiin ominaisuuksiin kuuluu riippumattomuus alustoista. Java-ohjelmointikielen kääntäjät tuottavat bittikoodia JVM:lle alustoille syntyperäisen kohdekoodin sijaan. Toimiakseen kunnolla tietyssä alustassa Java koodi tarvitsee vain bittikoodi tulkin simuloimaan JVM:ää. Tämä tarkoittaa sitä, että sama koottu bittikoodi toimii muuttamattomana kaikilla alustoilla joilla on Java-tuki. Java-ohjelmassa kaikki objektit ja luokat ovat olioita jotka sijoittuvat pääolio-luokan alle.

Javalla on laaja peruskirjasto, joka on yksi Javan viehättävimmistä ominaisuuksista. Java-ympäristöön kuuluu satoja luokkia ja menetelmiä kuudesta eri toiminnallisesta alueesta. Kielitukiluokat kehittyneille kielten ominaisuuksille kuten *string*, *array*, *thread* sekä poikkeuskäsittely. Apuohjelmaluokat, joihin kuuluu satunnaisgeneraattori, päivämäärä ja aika sekä säiliöluokat. Input/output luokat monien erilaisten tietojen kirjoittamiseen ja lukemiseen eri lähteistä. Verkottumisluokat, jotka mahdollistavat viestinnän tietokoneiden välillä lähiverkossa tai internetin kautta. AWT alustariippumattomien GUI-sovellusten luomiseen. Applet on luokka, jonka avulla voidaan luoda Java-ohjelmia, joita voi ladata ja ajaa asiakasselaimilla.

Javan yleinen hyväksyntä sovelluskehittäjien keskuudessa tapahtui Javan ja C++:n samankaltaisuuden vuoksi. Sovelluskehittäjien oli helppo oppia javaa kun taustalla oli C++:n tuntemus.

Java ei vaadi ohjelmoijia dynaamisesti vapauttamaan allokoitua muistia, tehden Java-koodista helpompaa kirjoittaa ja samalla vähentää muistivirheiden määrää. Tätä kutsutaan roskien keräykseksi. (Java Programming, 1997.)



KUVA 1. Javan suunnitteluvaiheen tavoitteet

### 3 ANDROID

Android on ohjelmistopino mobiililaitteille johon kuuluu käyttöjärjestelmä, väliohjelmisto ja avainsovellukset. Android on avoin kehitysalusta, mutta kaikki voi osallistua keskeneräiseen kehitykseen vaan kaikki luodaan ja päivitetään suljettujen ovien takana Googella. Androidin avoimuus lähtee siitä kun lähdekoodi on valmis ja julkaistaan maailmalle sen valmistuttua. Tällä tarkoitetaan että valmista lähdekoodia voi kuka tahansa ottaa käyttöön ja muokata kuinka haluavat. Android-sovelluksen tekemiseen sovelluskehittäjä tarvitsee Android SDK:n, jossa on mukana työkalut ja API:t. Sovelluskehittäjät usein integroivat SDK:n graafiseen käyttöympäristöön, IDE:en lyhentääkseen itse sovelluksen koodaamiseen menevää aikaa. Aloittelijoille on myös olemassa App Inventor, eli sovellussovellusten kehittämiseen joka on vapaasti saatavilla internetissä ja sovelluskaupoissa. Android SDK tarjoaa tarpeelliset työkalut ja ohjelmistorajapinnan (API) sovellusten kehittämisen aloittamiseksi android alustalla käyttäen java-ohjelmointikieltä. Androidin ominaisuuksia ovat:

- Sovellusten viitekehukset, jotka mahdollistavat komponenttien uudelleenkäytön ja korvauksen
- Dalvik virtuaalikone, joka on optimoitu mobiililaitteille
- Sisäänrakennettu selain, joka on rakennettu avoimen lähdekoodin WebKit moottorin ympärille
- Optimoidut grafiikat, jotka saavat virtansa kustomoidusta 2D grafiikkakirjastosta tai 3D grafiikat jotka on rakennettu OpenGL ES 1.0 määrittelystä jossa on laitteiston kiihdytysmahdollisuus
- SQLite, jäseneltyyn tietojen säilytykseen
- Media tuki yleisimmille ääni-, video-, ja kuva formaateille kuten MPEG4, H.264, MP3, AAC, AMR, JPG, PNG ja GIF
- GSM puhelinliikenne, laitteistoriippuvainen
- Bluetooth, Edge, 3G & WiFi, laitteistoriippuvainen Kamera, GPS, kompassi, kiihtyvyyssanturi, laitteistoriippuvainen
- Laaja kehitysympäristö, mukaanlukien laitteen emulointi, virheenkorjaustyökaluja, muistin ja suorituskyvyn profilointi ja kytkemismahdollisuus Eclipse IDE:een (Android, IBM, Gizmodo, Kanak Dodiya.)

### 3.1 Historia

Android sai alkunsa 5. Marraskuuta 2007, jonka jälkeen se on kasvanut yhdeksi maailman suosituimmista käyttöjärjestelmistä maailmassa. Android ei kuitenkaan tapahtunut hetkessä, vaan sillä on ollut myös hyvin huonoja aikoja joista se on kehittynyt käyttöjärjestelmäksi joka se on tänään. Androidin loi liittoutuma nimeltä Open Handset Alliance, johon kuuluu Google, HTC, Motorola, Texas Instruments sekä muita. Vaikka useimmat sovellukset jotka käyttävät android alustaa on kirjoitettu Java-ohjelmointikielellä, omaa Java virtuaalikonetta ei ole olemassa. Sen sijaan Java luokat kootaan ensin Dalvik Executables:iin ja ajetaan Dalvik virtuaalikoneella.

TAULUKKO 1. Androidin käyttöjärjestelmiä (Gizmodo, Android history).

Cupcake	1.5	Virtuaaliset näppäimistöt, Median lisääminen youtubeen	Samsung i5700
Donut	1.6	Universaalinen etsintä, Teksti puheeksi	Acer Liquid A1
Eclair	2.0 – 2.1	Kameran salama-tuki, digitaalinen zoomi Elävät taustakuvat	Samsung Galaxy S
Froyo	2.2 – 2.2.3	Nopeuspäivitys, käyttäjystävällisyys päivityksiä	Samsung Galaxy S2
Gingerbread	2.3 – 2.3.7	Latausten hallinta, copy-paste	Samsung Galaxy Pocket
Honeycomb	3.0 – 3.2.6	Holograafiset liitännät, intuitiivinen näppäimistö	Samsung Galaxy Tab
Ice Cream	4.0 – 4.0.4	Edelliset sovellukset –näppäin, toimintapalkki	Samsung Galaxy S3
Jelly Bean	4.1 – 4.3.1	Kustomisointi päivitys, kotiruudun uusinta, pienohjelmien lisääminen minne haluaa	Samsung Galaxy S3 Mini, Sony Xperia SP
KitKat	4.4 – 4.4.4	Käyttöjärjestelmän koon pienennys	Samsung Galaxy S5
Lollipop	5.0 – 5.1.1	Lukitusnäytön käytettävyys ja ilmoitukset, sovellusten moniajon parannuksia	Samsung Galaxy S6
Marshmallow	6.0 – 6.0.1	Uusi sovellusten lupien käsitteilytapa, sormenjälkitunnistus	Samsung Galaxy S7
Nougat	7.0-7.1	Usean sovelluksen käyttö jaetulla näytöllä (splitscreen), materiaaliparannuksia, ilmoitusten parannuksia	Nexus 9, Google Pixel, HTC 10, OnePlus 2&3 (kaikki ei varmoja)

Nimi

Android Versio Merkittävimmät päivitykset

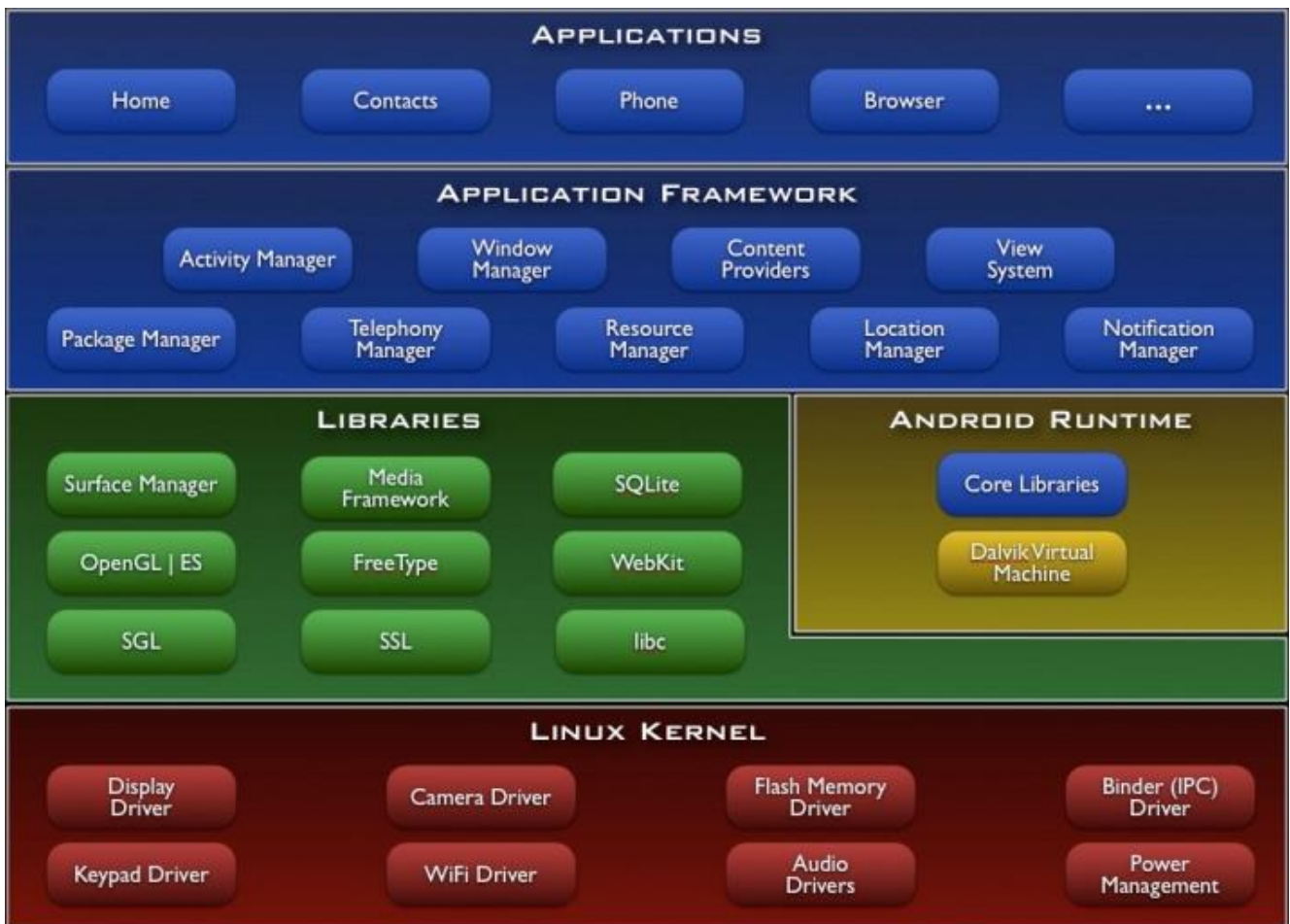
Laite

### 3.2 Android SDK

Ohjelmistokehityspakki joka mahdollistaa sovelluskehittäjien luoda sovelluksia android alustalle. SDK:n mukana tulee malliprojekteja joissa on lähdekoodi valmiina, sovelluskehitystyökaluja, android emulaattoreita sovelluksen testaamiseen sekä tarvittavat kirjastot android sovellusten kirjoittamiseen. Sovellukset kirjoitetaan Javalla käyttäen omavalintaista alustaa. (IBM, Kanaskin Dodiya.)

### 3.3 Androidin kehitys

Google pitää androidin kehittämisen poissa julkisuudesta kunnes viimeisimmät muutokset ja päivitykset on tehty, jonka jälkeen lähdekoodi julkaistaan ja sitä saa vapaasti muuttaa omiin käyttötarkoituksiin. Lähdekoodi toimii sellaisenaan vain tietyillä laitteilla ja monet laitevalmistajat joutuvat muuttamaan lähdekoodia ja tekemään omat ajurinsa jotta käyttöjärjestelmä toimisi heidän laitteillaan. (IBM, Kanaskin Dodiya.)



KUVA 2. Android-käyttöjärjestelmän pääkomponentit

Ylimpänä on applikaatiokerros eli *Applications* joka koostuu esiasennetuista sekä käyttäjän itse lataamista sovelluksista. Ohjelmistokehyksen eli *Application Framework*in parissa työskentelevät ohjelmistokehittäjät. Laitteen kaikki toiminnot ja sovellukset tarvitsevat tätä kerrosta toimiakseen. Kirjastot, eli *libraries* pitää sisällään androidin omat esiasennetut kirjastot jotka pitävät sisällään komentoja joilla varmistetaan oikea käsittelytapa kaikille datatyypeille.

Sovellusten ajo virtuaalikoneissa tuo lukuisia etuja, kuten sovellusten ”kontittaminen” eli eristys. Eristetyt sovelluksen eivät voi haitata käyttöjärjestelmää tai muita sovelluksia, eivätkä voi olla suoraan yhteydessä laitteen komponentteihin. Virtuaalikoneissa ajo tuo myös alusta-neutraalisuuden mukanaan, joten sovellukset eivät ole sidottuja tiettyyn laitteistoon.

Dalvik-virtuaalikone on luotu androidia varten googlen toimesta ja se luottaa taustalla olevaan Linux kerneliin matalan tason toiminnallisuuteen. Tämä säästää muistia ja on erityisesti suunniteltu mahdollistamaan tehokkaan useiden sovellusten toiminnan mobiililaitteen rajatuilla resursseilla.

Jotta suoritus olisi mahdollista dalvik-virtuaalikoneella, sovelluskoodi tulee muuntaa standardista Java-luokasta dalvikin omaan formaattiin joka on (.dex). Tällä formaatilla on puolet pienempi muistinkäyttö verrattuna standardiin javan tavukoodiin. Useimmiten Java-luokka tiedostot voidaan muuntaa .dex formaattiin käyttäen dx-työkalua joka tulee android SDK:n mukana. (Techopedia, 2016.)



## 4 ANDROID STUDIO

Android studio on androidin virallinen ohjelmointiympäristö jonka perustana on JetBrains:in IntelliJ koodieditori idea joka mahdollistaa nopeimman mahdollisen koodaamisen ja työtehokkuuden. Android Studion tarkoituksena on nopeuttaa android-sovellusten kehitystä ja auttaa sovelluskehittäjiä tekemään mahdollisimman hyvälaatuisia sovelluksia jokaiselle android-laitteelle. (Android studion ominaisuudet.)

### 4.1 Android Studion ominaisuuksia

Välitön koodinajo – kun sovellus on ajossa niin *Instant Run* ominaisuus ajaa koodi- ja resurssimuutokset sovellukseen lopettamatta ajoa. Tämä helpottaa työskentelyä huomattavasti, kun sovelluskehittäjä näkee koodin muutoksen vaikutukset sovellukseen samantien.

Älykäs koodieditori – Editori auttaa kirjoittamaan parempaa koodia, toimimaan nopeammin ja tuottavammin tarjoamalla koodin lopetusehdotuksia, refaktorointia ja koodin analysointia. Kun kehittäjä kirjoittaa koodia, android studio tarjoaa ehdotuksia pudotusvalikosta jotka saa lisättyä oman koodin perään painamalla TAB-näppäintä.

Nopea ja paljon ominaisuuksia sisältävä emulaattori – android emulaattori asentaa ja käynnistää sovellukset nopeammin kuin oikea laite, mahdollistaen nopean testauksen erilaisilla android-laitteilla.

Tiimityöskentely – android studio integroituu versionhallinta työkalujen kanssa kuten GitHubin, jotta tiimi voi pitää sovelluksen synkronoituna. (Android studion ominaisuudet.)

### 4.2 Android studion asennus

Ladataan asennusohjelma osoitteesta <https://developer.android.com/studio/index.html>, jonka mukana saa SDK:n sekä virtuaalilaitteen. Latauksen jälkeen asennetaan android studio sivujen ohjeen mukaan <https://developer.android.com/studio/install.html>. (Android studio ja sen asennus.)

### 4.3 Laitteistovaatimukset

Laitteistovaatimuksilla tarkoitetaan tietokoneen suorituskykyä ohjelman käyttämiseksi kunnolla. Jos tietokoneen tehot eivät riitä vähintään minimivaatimuksiin niin ohjelma hidastelee ja todennäköisesti kaatuu viimeistään koodin ajon aikana. Laitteistovaatimukset tulee täyttää ennen käyttöä mieluisan käyttökokemuksen saavuttamiseksi.

#### 4.3.1 Windows

Windowsin tuetut käyttöjärjestelmät ovat Windows 7, 8 ja 10. Android studio suosittelee käyttämään 64-bittisiä käyttöjärjestelmiä tehokkaamman ohjelmoinnin vuoksi. RAM-muistia vähintään 3 GB jos käytössä on 32-bittinen käyttöjärjestelmä, suosituksena kuitenkin 8 GB sekä 64-bittinen käyttöjärjestelmä. Vapaata levytilaa tulee olla vähintään 6 GB, joka täyttyy sovelluksesta, IDE:stä, SDK:sta sekä emulaattorista. Myös omat sovellukset vievät levytilaa joka kannattaa ottaa huomioon. Näytön resoluution tulisi olla vähintään 1280 x 800.

#### 4.3.2 Linux

Linux GNOME tai KDE käyttöjärjestelmä ja vähintään 3 GB RAM-muistia, suosituksena on 8 GB. GNU C Kirjasto (glibc) 2.11 tai uudempi versio. Vapaata levytilaa tulee olla vähintään 6 GB, joka täyttyy sovelluksesta, IDE:stä, SDK:sta sekä emulaattorista. Myös omat sovellukset vievät levytilaa joka kannattaa ottaa huomioon. Näytön resoluution vähintään 1280 x 800.

#### 4.3.3 Mac

Käyttöjärjestelmänä Mac® OS X® 10.8.5 tai uudempi, vähintään 3 GB RAM-muistia, kuitenkin mieluiten 8 GB. Vapaata levytilaa tulee olla vähintään 6 GB, joka täyttyy sovelluksesta, IDE:stä, SDK:sta sekä emulaattorista. Myös omat sovellukset vievät levytilaa joka kannattaa ottaa huomioon. Näytön resoluution tulisi olla vähintään 1280 x 800. (Android Studio.)

## 5 MAGIC: THE GATHERING

Magic: The Gathering on keräilykorttipeli jossa yhdistyvät kriittisen ajattelun strategointi sekä fantasia. Pelissä pelaajat ovat velhoja joita kutsutaan *planeswalkereiksi* jotka kutsuvat olentoja, tekevät taikoja sekä aseita tuhotakseen vastustajansa. Pelistä voi nauttia keräilykohteena tai pelaamalla muita ihmisiä vastaan sosiaalisissa tilanteissa.

Useimmiten yhdessä pelissä on vain kaksi pelaajaa, mutta moninpeli on mahdollista, mikäli kaikille on samaan peliformaattiin sopivat korttipakat. Rakennetut pelipakat koostuvat 60:stä kortista ylöspäin, ilman ylärajaa, mutta pelaajat yleensä rakentavat 60 kortin pakan oman pelistrategian onnistumisen vuoksi. *Limited* eli rajatussa pelimuodossa avataan korttipakkauksia joista rakennetaan pelipakka turnausta varten, näissä pakoissa minimi korttimäärä on 40. Pelissä pakkaa kutsutaan kirjastoksi.

Korteilla on pelaamisnopeus, joko *sorcery* tai *instant*. Sorceryjä voi pelata vain omalla vuorollaan päävaiheen aikana. Tähän kategoriaan kuuluvat olennot, ellei niillä ole *flash*-kykyä jonka avulla ne voidaan pelata vastauksena aktiivipelaajan kortteihin. Instant –taikoja voi pelata myös vastustajan vuorolla. Pelissä on 5 väriä jotka edustavat omantapaisia taikoja ja kykyjään. Nämä värit ovat valkoinen (plains), sininen (island), musta (swamp), punainen (mountain) ja vihreä (forest). Kukin maa tuottaa yhden manan omaa väriään jota käytetään korttien pelaamiseen. Värillistä manaa voi käyttää myös väritöntä manaa vaativia kortteja ja kykyjä.

Pelatuilla olennoilla on *summoning sickness* kun ne päätyvät pöydälle. Tämä estää niitä hyökkäämästä vuorolla jolla ne on pelattu, paitsi jos olennolla on *haste* –kyky.

Lähes kaikki kortit käyttävät *Stackiä* johon kaikki menevät kaikki kyvyt, taiat, olennot ennen kuin ne ratkaistaan (*resolve*). Stackin päällimmäinen eli viimeksi pelattu kortti ratkaistaan ensin, joka voi vaikuttaa aiemmin pelattuihin kortteihin tai kykyihin. Joissain korteissa on *split second*-kyky joka estää kyseiseen korttiin vastaamisen. Kortti jolla on *split second* menee siis stackin päälle ja se ratkaistaan ensin. (Magic: The Gathering perusteet, Wikihow, HubPages.)

### 5.1 Korttityypit

Pysyvä – <i>Permanent</i>	Kortti joka jää pelattaessa pöydälle. Esim. Olennot, maat ja lumoukset.
Taika – <i>Spell</i>	Kaikki pelatut kortit, poislukien maat.
Maa – <i>Land</i>	Manan tuottajia joilla voi olla jotain muitakin kykyjä.
Olento – <i>Creature</i>	Pelipöydällä olevat olennot, joilla taistellaan.
Hidas taika – <i>Sorcery</i>	Taika, joka tekee kuten kortissa lukee.
Nopea taika – <i>Instant</i>	Sama kuin sorcery, lisäyksenä mahdollisuus pelata vastustajan vuorolla
Luomous – <i>Enchantment</i>	Pöydälle jääviä taikoja joilla on erilaisia kykyjä.
Esine – <i>Artifact</i>	Samankaltaisia kuin lumoukset, erilaisia kykyjä, mukaanlukien aseet.
Planeswalker	Useita aktivoitavia Sorcery nopeuden kykyjä. (Isleofcards, 2015.)

Aktivoitu kyky – *Activated Ability*

Staattinen kyky – *Static Ability*

Aktivoituva kyky – *Triggered Ability*

Voidaan aktivoida koska vain ellei toisin sanota.

Kokoajan voimassa

Tietyn asian tapahtuessa itsestään aktivoituva kyky.



KUVA 3. Activated Ability



KUVA 4. Static Ability



KUVA 5. Triggered Ability (Isleofcards, Magic for Beginners).

## 5.2 Pelin kulku

Pelin alussa pelaajilla on useimmiten 20 elämäpistettä, mutta eri formaateissa voi elämäpisteitä olla 25 tai 40. Pelin edetessä pelaajat voivat sekä menettää että saada lisää elämäpisteitä. Jokaisen pelin alussa pelipakat sekoitetaan ja omasta kirjastosta nostetaan 7 korttia käteen, jonka jälkeen mietitään että onko tämä hyvä käsi vai ei pelin alussa voi tehdä *mulliganin* eli sekoittaa nostamansa 7 korttia takaisin pakkaan ja nostaa uudet, mutta yhden vähemmän. Mulliganin voi tehdä niin monta kertaa että kädessä on enää yksi kortti jäljellä, jolloin peli alkaa. Joka vuoron alussa nostetaan yksi kortti.

Kun pelaaja hylkää kortin (discard), olento kuolee tai taika on pelattu niin kyseinen kortti laitetaan pelaajan hautuumaalle. Hautuunmaa on yleensä kirjaston vieressä oleva korttipino josta voi katsoa pelin aikana pelatut kortit. Pelaajat tekevät vahinkoa olenoihin sekä toisiinsa joko omilla olennoilla tai taioilla. Lähes kaikki olennot ja taikat maksavat manaa jotta ne voidaan pelata. Manaa saadaan Maakorteista ja niitä voi pelata yhden omalla vuorolla. Maat tuottavat manaa kun ne käännetään sivuasentoon (*tap*) ja palautuvat uudelleen käytettäviksi seuraavalla vuorolla (*untap*). Jotkut kortit vaativat manan lisäksi jotain muuta – kuten kortin hylkäyksen, oman olennon uhraamisen tai elämäpisteiden menetystä. (WotC, isleofcards, HubPage, WikiHow, Magic pelin kulku.)



KUVA 6. Keskeneräinen pelitilanne

### 5.3 Vuoron kulku

Vuorot perustuvat vaiheisiin jotka käydään läpi tässä osiossa. Nämä vaiheet ovat yhden pelaajan vuoro.

**Aloitukset** – *Untap phase* käännetään kaikki omat kortit pystyasentoon, jonka jälkeen on mahdollista pelata instant ja flash kortteja

**Ylläpito** – *Upkeep* tässä vaiheessa kortit joiden kyvyt aktivoituvat vuoron alussa *at the beginning of your upkeep, do X*. Näihin voi vastata Instant-nopeuden korteilla ja kyvyillä.

**Nosto** – *Draw step* aktiivipelaaja nostaa vuorokortin (Aloittava pelaaja ei nosta korttia ensimmäisellä vuorollaan).

**Ensimmäinen päävaihe** – *First Main phase*. Tässä vaiheessa pelataan maa, olentoja, aseita, esineitä, lumouksia, planeswalkereita ja aktivoidaan kykyjä. Vastustaja voi pelata Instant-nopeuden kortteja vastauksena kaikkiin pelattuihin kortteihin tai aktivoituihin kykyihin.

**Taistelun alku** – *Beginning combat* peli on menossa taisteluun, vastustaja voi yrittää tuhota olentoja tai estää aktiivipelaajaa hyökkäämästä taioin ja kyvyin.

**Hyökkäys** – *Attack* aktiivipelaaja valitsee olennot joilla haluaa hyökätä. Hyökkääjä voi valita hyökkäkö vastustajan kimppuun vai mahdollisten *planeswalkerin* joka vastustajalla voi olla. Hyökkäämään lähtevät olennot käännetään sivuttain (*tap*), paitsi jos niillä on *vigilance* –kyky jolloin ne jäävät pystyasentoon.

**Puolustus** – *Blocking* pelaaja jota vastaan on hyökätty valitsee omista olennoistansa mitkä menevät taisteluun minkäkin olennon kanssa, useampi puolustava olento voi taistella yhtä hyökkääjää vastaan. Vain pystyasennossa olevat olennot (*untapped*) voivat puolustaa. Ennen seuraavaan vaiheeseen menemistä molemmat pelaajat pystyvät käyttämään Instant-nopeuden taikoja ja kykyjä vaikuttaakseen taistelun lopputulokseen.

**Taisteluvahingot** – *Combat damage* Kaikki taistelevat olennot tekevät voimansa verran vahinkoa kohteisiinsa. Jos olentoa vastaan ei puolustauduttu, tämä tekee vahinkonsa vastustavaan pelaajaan tai planeswalkeriin.

**Taistelun loppu** – *End of combat* tässä vaiheessa voi vain aktivoida kykyjä tai Instant nopeuden taikoja

**Toinen Päävaihe** – *Second Main phase* samanlainen vaihe kuin ensimmäinen, jälleen voi pelata olentoja sekä Sorcery-nopeuden taikoja tai kykyjä.

**Loppuvaihe** – *Ending phase* kortit, joiden kyvyt aktivoituvat vuoron lopussa *at the beginning of your end step, do X* joihin vastustaja voi vastata Instant nopeuden korteilla

**Siivous** – *Cleanup* kädestä hylätään kortteja kunnes korttimäärä on 7, haavoittuneet taistelusta selviytyneet olennot paranevat maksimi elämäpisteisiin. Mihinkään ei voi vastata taioilla eikä kyvyillä. (Magic for dummies, 2012.)

## 5.4 Pelin päätyminen

Pelaaja häviää pelin jos elämäpisteet ovat 0 tai alle eikä mikään kortti estä häviämistä, tai kirjasto on tyhjä kun vuoron alussa nostettava ”vuorokortti” tulisi nostaa, tai kun pelaaja saa 10 *poison counteria* joita antavat tietyt kortit. Pelissä on tuhansia sääntöjä ja säädöksiä jotka tulee ymmärtää pelataksaan arvoturnauksissa ja kilpailuhenkisissä kisoissa. (LIITE 1). Lähes jokaiselle pelin säännölle löytyy kortti, joka rikkoo jonkin säännön. Esimerkki korttina (KUVA 6) Platinum Angel joka estää häviämisen niin kauan kun se on pelattuna. (WikiHow, isleofcards, Magic for beginners.)



KUVA 7. Magic: The Gathering Platinum Angel

## 5.5 Peliformaatit

Magic: The Gatheringiin kuuluu lukuisia eri peliformaatteja, joissa kussakin on vain tiettyjen settien kortteja sallittuina. Tässä käydään läpi pelatuimmat ja yleisimmät formaatit. (Isleofcards, Magic for beginners.)

### 5.5.1 Limited

Limited on peliformaatti jossa pelaamiseen käytetään avaamattomia korttipusseja joista rakennetaan itselle pakka ja pelataan turnausmuodossa. Yleensä näihin turnauksiin tuodaan ylimääräinen korttipussi palkintoihin, jotka jaetaan sitten parhaiten pärjänneiden kesken. Limited formaatteja on *Booster Draft* jossa avataan korttipussi, otetaan yksi kortti ja vaihdetaan pussi seuraavalle pelaajalle. Jokainen pelaaja tuo yleensä 3 korttipussia turnaukseen. Toinen limited formaatti on *Sealed Deck* johon tuodaan yleensä 3 avaamatonta korttipussia joista rakennetaan pakka. Kolmas ja viimeinen formaatti on *Cube Draft* jonka joku pelaaja on rakentanut. Cube Draft on monenlaisia riippuen korteista joita on käytetty sen rakentamiseen. (Isleofcards, Magic for beginners.)

### 5.5.2 Constructed

*Constructed* formaateissa pelaajat ovat rakentaneet pakkansa valmiiksi ja pakoissa saa olla maksimissaan 4 samanlaista korttia (poislukien perusmaat), ja pakan minimikoko on 60 korttia. Constructed formaatista riippuen vain tietyt korttisetit ovat sallittuja ja osassa nämä setit vaihtuvat. Jokaisessa constructed formaatissa on myös omat *ban listit* eli liian voimakkaita kortteja sallitusta setistä on poistettu formaatista.

*Standard* on suosituin constructed formaatti jossa sallitut kortit ovat kaksi viimeisimmäksi julkaistua lisäosaa Magic: The Gatheringiin. Ongelmana on tiuhaan tahtiin vaihtuvat sallitut kortit joita tulee kokoajan ostaa uusia, vanhojen jäädessä lähes arvottomiksi.

*Modern* on laajempi formaatti, jossa sallitut kortit ovat vuodelta 2003 ja siitä eteenpäin. Modernissa sallitut setit eivät katoa, vaan formaatti laajenee jatkuvasti uusien korttien julkaisujen myötä.

*Legacy* formaatissa on kaikkien settien kortit sallittuja, mukaanlukien tietyistä erikois- ja promotaatio seteistä. Legacyssä on suunniteltu erikoinen ban list poistamaan liian voimakkaita kortteja. (Isleofcards, Magic for beginners.)

### 5.5.3 Commander

Komentajassa (toiselta nimeltään EDH – Elder Dragon Highlander) pelaajilla on legendaarinen olento komentajanaan ja loppupakan kortit saavat olla vain komentajan väreissä. Tässä peliformaatissa aloitus elämäpisteet ovat 40 ja pakan koko on tasan 100 korttia – komentaja sekä 99 muuta. Komentaja formaatissa kaikkien pakan korttien tulee olla erilaisia, poislukien perusmaat. Formaattiin kuuluu myös erikois voitto-tapa, jos yksi sama komentaja tekee pelaajaan 21 vahinkoa pelin aikana, kohdepelaaja häviää ja tippuu pelistä. Tämä on suosituin moninpeliformaatti, pelien kestot voivat vaihdella minuuteista useisiin tunteihin. (Isleofcards, Magic for beginners.)

### 5.5.4 Tiny Leaders

Komentaja variaatio, jossa mikään kortti ei saa maksaa yli 3 manaa, pelaajat aloittavat 25 elämäpisteestä ja pakan koko on 50 korttia ilman duplikaatteja. Formaatti käyttää omaa ban list:äänsä ja sen suosio kasvoi räjähdysmäisesti vuoden 2015 alusta. (Isleofcards, Magic for beginners.)

## 5.6 Pelitasot

MTG:tä voi pelata usealla eri tasolla, jotka lajitellaan neljään pääkategoriaan jotka ovat kasuaali, FNM (Friday Night Magic), kilpahenkien täytäntöönpanotaso (Competitive rules enforcement level) sekä Pro Tour eli maailman parhaiden pelaajien kisat. Kilpahenkisissä formaateissa on useita tuomareita, kamerat sekä striimauspalvelu mahdollistaen kenen tahansa katsoa pelejä internetin kautta. (Isleofcards, Magic for beginners.)



### 5.6.1 Kasuaali

Kasuaali on yleisin pelimuoto joka on kavereiden kanssa pelaamista. Tämä on sanktioimatonta peliä, eli paikalla ei ole tuomaria eikä minkäänlaista sääntöjen valvontaa pelaajien lisäksi. Kasuaali tapahtumia ja turnauksia järjestetään vaikka ne ovatkin vain hovin tai harjoituksen vuoksi. Rento ja huoleton pelimuoto eikä lainkaan kilpahenkinen. (Isleofcards, Magic for beginners.)

### 5.6.2 FNM

FNM:llä tarkoitetaan Friday Night Magic eli Perjantai illan magic pelit. Tämä on viikottainen turnaus joka järjestetään paikallisissa pelikaupoissa perjantai iltais. FNM on sanktioitu tapahtuma jossa yleensä kaupanpitäjä toimii tuomarina. Tapahtumaan on pieni sisäänpääsymaksu ja parhaiten pärjänneille on tiedossa palkintoja kuten harvinaisempia promotaatiokortteja ja korttipusseja. FNM on alimman tason kilpapelamista. (Isleofcards, Magic for beginners.)

### 5.6.3 Competitive REL

Keskitason kilpapelamista, jossa pelaajien oletetaan osaavan turnaus- ja pelisäännöt sekä turnauskäytännöt DCI:stä (Duelists' Convocation International). Pelitapahtumat tässä kategoriassa mukaanlukevat Preliminary Pro Tour Qualifiers eli alustavat karsinnat, Pro Tour Qualifiers eli karsinnat Pro Tourille, Grand Prix karsinnat sekä Grand Prix. (Isleofcards, Magic for beginners.)

### 5.6.4 Pro Tour

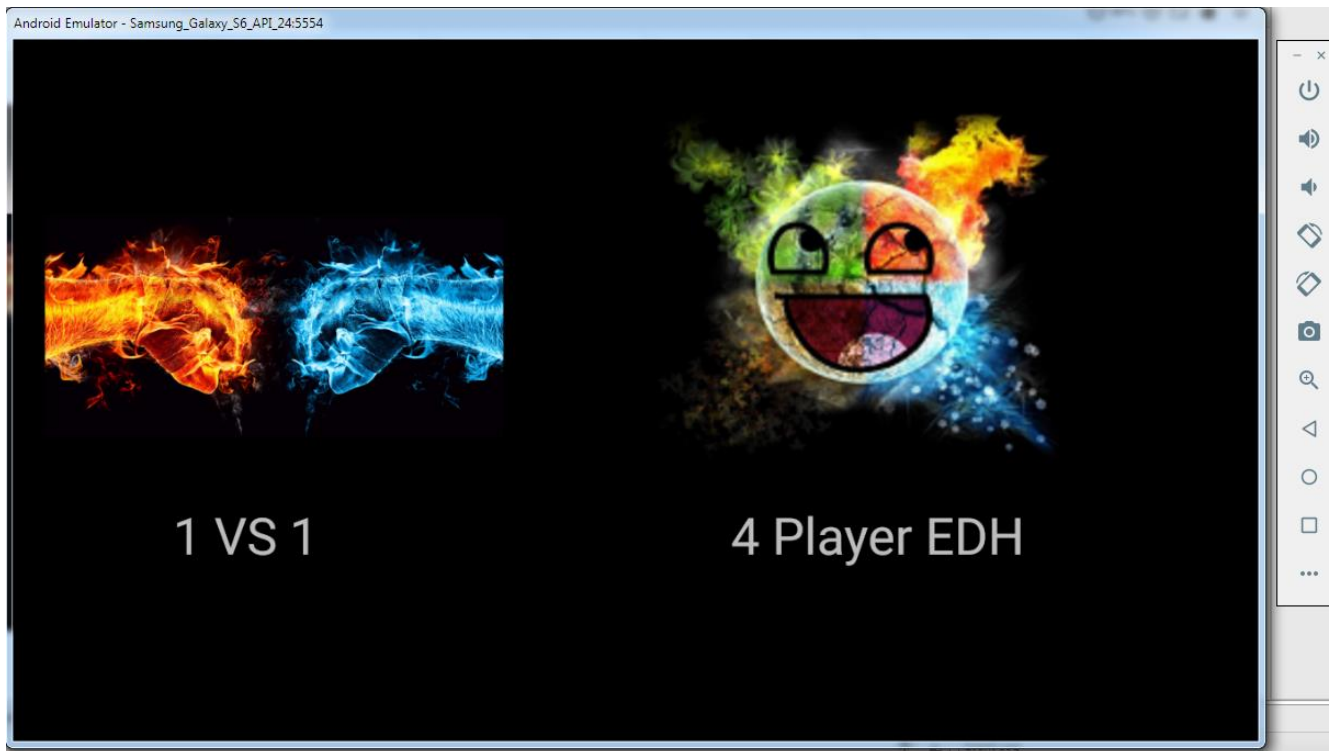
Pro Tour tapahtumat ovat korkeimman tason kilpapelamista, jossa on myös isoimmat palkinnot mitä Magic: The Gatheringin pelaamisesta voi saada. Päästäkseen pelaamaan Pro Tourille, pitää selvitä karsinnoista (Grand Prix tai Pro Tour Qualifier), tai muuten osoittaa pätevyys kuten olemalla Pro Tourin kunnia-gallerian jäsen. Parhaat MTG:n pelaajat pelaavat 4 eri vuotuista Pro Tour tapahtumaa joissa pelataan sekä limited että constructed formaatteja. Useimmat competitive REL tapahtumien pelaajien toiveena on päästä mukaan Pro Tourille. (Isleofcards, Magic for beginners.)



KUVA 8. Peliturnaus

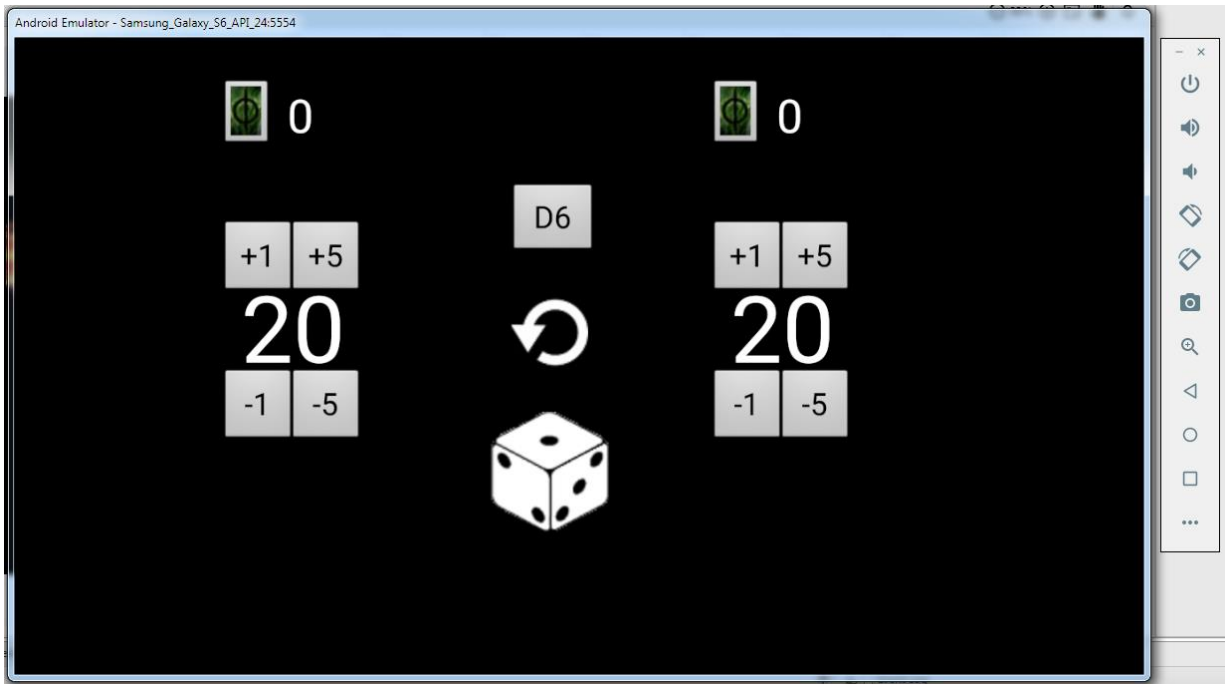
## 6 POHDINTA

Omasta sovelluksestani tuli tarkoituksieni mukainen eli helppokäyttöinen kevyt sovellus joka tukee myös usean pelaajan formaatteja. Sovelluksen käynnistyessä aukeaa ensin yksinkertainen ruutu josta valitaan kaksin- tai moninpeli. Nämä toteutin sopivilla kuva-painikkeilla antaakseni sovellukselle hieman väriä sekä muistuttaakseni että MTG:n pelaamisen tulee olla hauskaa.



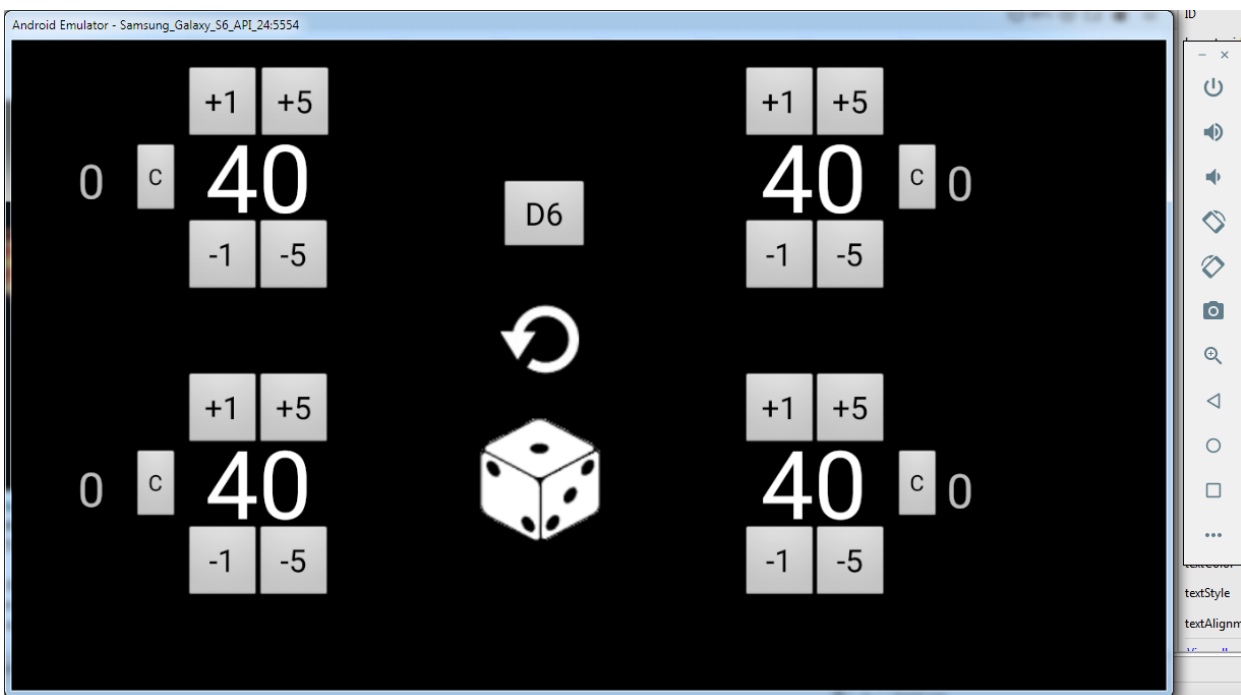
KUVA 9. Oman sovelluksen aloitusruutu.

Valittaessa kahden pelaajan pelin, aukeaa uusi eli uusi näkymä jota Android Studiossa kutsutaan *activity*:ksi. Tässä näkymässä on käytettävissä elämäpistelaskurit, kätevät plus yksi, plus 5, minus 1, minus 5 painikkeet jotka kontrolloivat pelaajien elämäpisteitä. Yläpuolella on *poisoncounter*:it, joka on lähinnä kaksinpeliformaattien mekaniikka. Tämän takia jätin kyseiset painikkeet pois moninpelimoodista. Keskellä ruutua on uudelleenasetuspainike joka palauttaa kaiken oletusaroihinsa kyseisessä näkymässä. Ylhäällä keskellä on nopanheittopainike D6 (*dice 6*), jota jotkin pelikortit tarvitsevat. Tällä voidaan myös määrittää esimerkiksi aloittava pelaaja. D6 painiketta painettaessa alhaalla keskellä oleva nopankuva muuttuu yhdeksi kuudesta nopan silmäluvusta satunnaisesti.



KUVA 10. Oman sovelluksen 1 vs 1.

Viimeisenä moninpelinäkymä, jossa on elämäpisteet neljälle pelaajalle, nopanheitto sekä uudelleenasetusnappulat sekä lisäksi EDH formaattiin tarvittava C joka tarkoittaa komentajapainiketta. Aina, kun komentaja pelataan niin painetaan kyseistä nappia kerran. Tämä lisää numeron arvoksi plus kaksi, jolla tarkoitetaan että seuraavan kerran kun komentaja pelataan, se maksaa kaksi manaa enemmän.



KUVA 11. Oman sovelluksen 4 pelaajan EDH.

## 7 YHTEENVETO

Minulla oli ennen opinnäytetyötä muutaman kurssin verran kokemusta java-kielestä sekä android studiosta, joka oli hyödyksi sovelluksen tekemisessä. Kiinnostukseni javaa kohtaan kasvoi entisestään opinnäytetyötä tehdessä. Sovelluksesta tuli samankaltainen kuin mitä olin visioinut ennen opinnäytetyön aloitusta ja saavutin tavoitteeni. Opinnäytetyön tekemisen kohdalla tein virheen kun aloitin teoriaosuudesta, nyt jälkikäteen ajatellen olisi kannattanut aloittaa ohjelmoinnista. Opinnäytetyötä tehdessäni opin java-kieltä, android studion käyttöä sekä yleistä tietoa android-laitteista. Sovelluksen jatkumahdollisuudet ovat lähes rajattomat, tosin silloin saattaa akkuystävällisyys kärsiä. Jatkumahdollisuuksia olisivat esimerkiksi korttiskanneri laitteen kameran avulla, jolla voisi kääntää vieraskieliset kortit englanniksi.

Opinnäytetyön tavoitteena oli luoda Magic: The Gathering apusovellus android laitteille, josta on hyötyä kaikille kyseistä peliä pelaaville. Sovelluksella voi korvata joitain pelitarvikkeita ja se kulkee kätevästi aina mukana. Ongelmia tuli vastaan ohjelmoinnissa sekä sopivien lähteiden löytämisessä, jotka ratkesivat opinnäytetyöhön omistautumisella sekä kertaamalla java-ohjelmointia.

## LÄHTEET

- Gizmodo 2015. History of Android. Www-dokumentti. Saatavissa:  
<http://gizmodo.com/a-history-of-android-from-cupcake-to-m-1707432419> luettu 2.4.2016 1
- IBM 2009. Android development. Www-dokumentti. Saatavissa:  
<http://www.ibm.com/developerworks/library/os-android-devel/> luettu 22.5.2016 2
- Techopedia 2016. Android Platform. Www-dokumentti. Saatavissa:  
<https://www.techopedia.com/definition/4219/android-platform> luettu 23.5.2016 3
- Java programming 1997. Www-dokumentti. Saatavissa:  
<http://groups.engin.umd.umich.edu/CIS/course.des/cis400/java/java.html> luettu 28.5.2016 4
- Kanak Dodiya 2011. Www-dokumentti. Saatavissa:  
<http://kanaksinhdodiya.blogspot.fi/2011/08/basic-information-about-android.html> luettu 25.3.2016 5
- Java Language Specification 2015. Www-dokumentti. Saatavissa:  
<https://docs.oracle.com/javase/specs/jls/se8/jls8.pdf> luettu 15.6.2016 6
- MRBool Design goals of Java, Kaushik Pal 2016. Www-dokumentti. Saatavissa:  
<http://mrbool.com/describing-the-design-goals-of-java/28023> luettu 20.10.2016 7
- Java SE versions history 2014. Www-dokumentti. Saatavissa:  
<http://www.codejava.net/java-se/java-se-versions-history> luettu 22.10.2016 8
- Techopedia 2016. An Overview of the Android Architecture. Saatavissa:  
[http://www.techotopia.com/index.php/An\\_Overview\\_of\\_the\\_Android\\_Architecture](http://www.techotopia.com/index.php/An_Overview_of_the_Android_Architecture) luettu 4.1.2017 9
- Android studio ominaisuudet. Www-dokumentti. Saatavissa:  
<https://developer.android.com/studio/features.html> luettu 5.11.2016 10
- Android studio. Www-dokumentti. Saatavissa:  
<https://developer.android.com/studio/index.html> luettu 5.11.2016 11
- Android Studiom asennus. Www-dokumentti. Saatavissa:  
<https://developer.android.com/studio/install.html> luettu 5.11.2016 12
- Magic: The Gathering, WotC. Www-dokumentti. Saatavissa:  
<http://magic.wizards.com/en/gameplay/how-to-play> luettu 7.12.2016 13
- HubPages Magic: The Gathering for dummies. Www-dokumentti. Saatavissa:  
<http://hubpages.com/games-hobbies/mtgfordummies> luettu 7.12.2016 14

- WikiHow, how to play Magic: The Gathering. Www-dokumentti. Saatavissa: <http://www.wikihow.com/Play-Magic:-The-Gathering> luettu 7.12.2016 15
- Isle of Cards, Magic for beginners 2015. Www-dokumentti. Saatavissa: <https://www.isleofcards.com/magic-the-gathering-beginner-guide> luettu 8.12.2016 16
- Efka Bladukas, 2015. Www-dokumentti. Saatavissa: <http://tradingcardgames.com/ultimate-guide-on-attending-your-first-magic-tournament/> luettu 8.12.2016 17
- Brandon, 2016. Magic: The Gathering. Www-dokumentti. Saatavissa: <https://rctom.hbs.org/submission/from-physical-to-digital-magic-the-gathering/> luettu 8.12.2016 18
- ClipartFest, 2016. Sovelluksen pienkuvat. Www-dokumentti. Saatavissa: <https://clipartfest.com/> luettu 14.12.2016 19

## LIITTEET

- LIITE 1: Magic: The Gathering Comprehensive rules (Täydet säännöt) Saatavissa: [http://media.wizards.com/2016/docs/MagicCompRules\\_20161111.pdf](http://media.wizards.com/2016/docs/MagicCompRules_20161111.pdf) luettu 14.12.2016
- LIITE 2: Magic: The Gathering Quick Start Guide (Perusteet) Saatavissa: <http://magic.wizards.com/en/gameplay/how-to-play> luettu 14.12.2016