

CHALMERS



GÖTEBORGS UNIVERSITET



NetPaint

- Androidapplikation för att måla tillsammans

Kandidatarbete vid Data- och Informationsteknik

Markus Aronsson

Emma Bogren

Thomas Mattsson

Andreas Berggren

Erik Bogren

Sebastian Pukki

Institutionen för Data- och informationsteknik

CHALMERS TEKNISKA HÖGSKOLA

GÖTEBORGS UNIVERSITET

Göteborg, Sverige 2012

Kandidatarbete/rapport nr 2012:010

Sammanfattning

Det här projektet har gått ut på att ta fram en applikation för Android där en eller flera spelare kan måla ihop. Projektets huvudsakliga utmaningar har varit att skapa en interaktiv rityta och trådlös kommunikation mellan Androidtelefoner.

Utmaningen med skapandet av ritytan låg i att modellera och abstrahera elementen kopplade till ritrörelser samt metoderna för åtkomst till dessa, så att verktygen kopplade till utmålning kunde integreras. Det låg även ritytan att skapa och hantera nätverksmeddelanden under en flerspelar-session, för att upprätthålla en identisk bild över alla Androidtelefoner som målar tillsammans. Svårigheterna med kommunikationen mellan Androidtelefonerna var att få trådlös kommunikation via antingen TCP/IP- eller Bluetooth-protokollet och att få de meddelanden som skickas mellan telefonerna att synkroniseras utan några fel. Ett annat moment innefattade att designa ett användargränssnitt med ett tema kopplat till applikationens syfte och som var så användarvänligt som möjligt.

Gruppen har arbetat i två mindre delgrupper för att kunna nå resultat snabbare, detta eftersom projektet har haft tre omfattande arbetsområden: användargränssnitt, rityta samt nätverkskommunikation.

Resultatet är en applikation som når alla de huvudsakliga mål som var uppsatta av gruppen då projektet började.

Abstract

This project was about developing an Android application where either one or several players can paint together. The main challenges with the project has been the construction of an interactive drawing surface and the implementation of wireless communication between the Android phones.

The challenge of designing a drawing surface was to model and abstract elements linked to drawing motions and the methods for accessing them, so that the tools linked to drawing could be integrated. It also fell on the drawing surface to create and manage network messages during a multiplayer session, so that an identical image across all Android phones could be maintained. The problems surrounding the communication between Android phones was to provide wireless communication through the use of either the TCP/IP or Bluetooth protocol and to get these messages sent between the Android phones in-sync. Another challenge was to implement a graphical user interface with a theme connected to the purpose of the application, and which was as user-friendly as possible.

The group was divided into two smaller groups in order to achieve faster results, with these two groups handling one of each of the main challenges of the project; the graphical user interface, the drawing surface and the network communication.

The result of all this is an application that fulfills all of the initial main goals that were set up by the group at the start of the project.

Ordlista

3G	3rd generation mobile telecommunications. Standarder för mobiltelefoner och tjänster för mobil kommunikation
Android	Operativsystem för mobila enheter som är utvecklat av Google
Avatar	Grafisk representation av en användares alter ego
Bluetooth	Teknisk standard för öppen trådlös kommunikation av data över korta avstånd
Betatestning	En någorlunda färdig produkt som funktionstestas utav utvecklare eller utomstående. Sällan buggfri och mycket av syftet är att hitta just buggarna
Datagram	Meddelandepaket som används i UDP protokollet
Datapak	Meddelandepaket som används i TCP protokollet - den mest grundläggande enhet som skickas över ett paketförmedlande nätverk
Repository	Förvaring som tillåter användare att ladda upp sin kod och hanteras av versionshantering
Google Play	En marknad för att köpa och ladda ner bland annat applikationer till sin Android telefon. Tidigare känt som Android Market
GPS	Global Positioning System. Satellitstyrt navigations system som förser användaren med information om plats och tid överallt på jorden
GUI	Graphical User Interface. Ett användargränssnitt baserat på grafik snarare än textkommandon
IP-adress	Internet Protocol-adress. Ett nummer tilldelat alla enheter som är kopplade till ett nätverk som använder protokollet Internet Protocol för kommunikation
Java	Objektorienterat programspråk
Open Source	Källkoden är gratis och tillgänglig för alla
Peer	På svenska jämlike, som i detta fall syftar på en användare

Peer-to-Peer	En uppkoppling direkt mellan flera användare
PNG	Portable Network Graphics . Ett förlustfritt format som passar bäst för enklare grafik, då detta ger bäst kompressionsgrad
Relief	Upphöjning av text eller grafik, som ger en tredimensionell effekt
SDK	Software Development Kit . En uppsättning verktyg för mjukvaruutveckling som möjliggör att man kan skapa applikationer för exempelvis ett visst operativsystem eller mjukvarupaket
Socket	Dörr för applikationens datapaket, eller med finare ord; Nätverksknytpunkt av kommunikationsflödet för interna processer över ett nätverk
SVG	Scalable Vector Graphic . Grafik som ritas upp med hjälp av matematiska formler. Sådan grafik kan förstöras eller förminskas utan att förlora kvalitet
TCP	Transmission Control Protocol . Ett protokoll som de flesta Internet-applikationer använder sig av. Ger pålitlig och ordnad leverans från en dator till en annan
Tråd	Möjliggör att en applikation kan utföra flera uppgifter samtidigt
UDP	User Datagram Protocol . Ett transportprotokoll som används på Internet. Ger en snabb men opålitlig leverans av datagram
XML	Extensible Markup Language . Ett märkspråk som definierar en uppsättning regler för kodning av dokument i ett format som är både lättläst och maskinläsbart

Innehåll

1	Inledning	8
1.1	Bakgrund	8
1.2	Syfte	8
1.3	Mål	9
1.4	Problembeskrivning	9
1.5	Avgränsningar	10
1.6	Disposition	10
2	Förstudie	12
2.1	Val av plattform	12
2.2	Val av version	13
2.3	Android i Eclipse	14
2.4	Versionshantering - SVN	14
2.5	Relaterade applikationer	15
2.6	Analys av kommunikationsval	15
2.6.1	Val av transportprotokoll	15
2.6.2	Server eller peer-to-peer	16
3	Arbetsmetod	17
3.1	Arbetsupplägg	17
3.2	Arbetsmetodik för utvecklingen av applikationen	17
3.2.1	Planeringsfasen	18
3.2.2	Fördjupningsfasen	18
3.2.3	Konstruktionsfasen	18
3.2.4	Överlämningsfasen	18
3.3	Kodstandard	19
3.4	Testning av kod	19
3.5	Kravspecifikation	19
3.6	Användartester för grafiskt gränssnitt	19
4	Design och implementation	21
4.1	Applikationsflöde	21
4.1.1	Enspelarläge	21
4.1.2	Flerspelarläge	21
4.1.3	Spara/Ladda bild i applikationen	22
4.2	Uppbyggnad	22
4.2.1	Enspelarläge - Användare och Enhet	22
4.2.2	Flerspelarläge - klient-server och Bluetooth	22

4.3	Nätverks- och Bluetooth-kommunikation	24
4.3.1	Tillhörande server för applikationen	24
4.3.2	Applikationens ramverk	25
4.3.3	Dataformatet MessagePacket	27
4.3.4	Test av nätverkslager med hjälp av ClientApp	28
4.3.5	Kommunikation via Bluetooth	28
4.3.6	Test av Bluetoothkommunikation med hjälp av BluetoothClient	30
4.4	Trådar och processer	30
4.5	Ritytan och dess komponenter	32
4.5.1	Ritrörelsen och de objekt som bygger upp den	32
4.5.2	Verktysraden i ritytan	33
4.5.3	Dialogrutor till verktygen	33
4.5.4	Skillnaden mellan enspelarläget och flerspelarläget	34
4.6	Zoomfunktion	35
4.6.1	Uppbyggnad av zoomfunktionen	36
4.6.2	Dynamik i zoomverktyget och dess animationer	37
4.6.3	Interaktioner med zoomverktyget	38
4.7	Utveckling av användargränssnitt	38
4.7.1	Initiellt användargränssnitt	38
4.7.2	Andra versionen av användargränssnittet	40
4.7.3	Tredje versionen av användargränssnittet	41
4.7.4	Slutversionen av användargränssnittet	42
4.8	Filhantering	43
4.8.1	Filformat	43
4.9	Felhantering	44
4.9.1	Ingen nätverksåtkomst	44
4.9.2	Anslutningen bröts	44
4.9.3	Uppkoppling misslyckades	44
4.9.4	Inga enheter funna	45
4.9.5	Försök att ansluta simultant	45
5	Resultat i form av implementerade krav	46
5.1	Implementerade krav	46
5.1.1	Väsentliga implementerade krav av prioritet 1	46
5.1.2	Väsentliga implementerade krav av prioritet 2	47
5.1.3	Krav som inte implementerats	47
6	Diskussion av arbetsmetod och problem under utveckling	48
6.1	Arbetsmetod	48
6.2	Problem under implementationen	48

6.2.1	Utvecklingen av nätverkslagret	48
6.2.2	Utvecklingen av Bluetooth	49
6.2.3	Problem vid utvecklingen av ritytan	50
7	Slutsats	51
8	Framtida arbete	52
A	Användarmanual	57
B	Instruktioner för användartester	69
B.1	Användartester I: Enspelarläge & Huvudmeny - Instruktioner	69
B.1.1	Observation	69
B.1.2	Intervjufrågor	69
B.1.3	Renskrivning	70
B.2	Användartester II: Enspelarläge & Huvudmeny - Instruktioner	71
B.2.1	Observation	71
B.2.2	Intervjufrågor	71
B.3	Användartester III: Enspelarläge, flerspelarläge & huvudmeny - Instruktioner	72
B.3.1	Observation	72
B.3.2	Intervjufrågor	73
C	Kravspecifikation	74
C.1	Prioritet 1	74
C.2	Prioritet 2	75
C.3	Prioritet 3	76
C.4	Plattform	77
C.4.1	Applikationen skall stödja Android 2.1 och uppåt . . .	77
C.5	Nätverk	77
C.5.1	Kommunikation genom 3G/WiFi	77
C.5.2	Kommunikation genom TCP	78
C.5.3	Kommunikation via Bluetooth	78
C.5.4	Kommunikation mellan två enheter	79
C.5.5	Kommunikation mellan två eller tre enheter	79
C.5.6	Data sänds under beröring av ritytan	80
C.5.7	GPS för att välja motspelare	80
C.6	Gränssnitt och interaktion	81
C.6.1	Touchfunktion för ritning och menynavigering	81
C.6.2	Notifieringar vid fel	81
C.6.3	Lättförståeligt användargränssnitt	82

C.6.4	Egendesignat användargränssnitt	82
C.7	Spel	82
C.7.1	Applikationen startar	82
C.7.2	Applikationen avslutas	83
C.7.3	Möjlighet att spela ensam	83
C.7.4	Möjlighet att spela två spelare	84
C.7.5	Möjlighet att spela två eller tre spelare	84
C.7.6	Val av färg	85
C.7.7	Val av effekt	85
C.7.8	Val av storlek	86
C.7.9	Avslutning av ritsession	86
C.7.10	Användaren skall ha ett användarnamn	87
C.7.11	Användaren skall ha en avatar	87
C.7.12	Användarna skall delas in i rum	88
C.7.13	Slumpmässig tilldelning av rum	88
C.7.14	Möjlighet att zooma	89
C.7.15	Panorering av bild	89
C.7.16	Möjlighet att sudda	90
C.7.17	Möjlighet att ångra det man ritat	90
C.7.18	Möjlighet att göra om det man ångrat	91
C.7.19	Möjlighet att få ljud	91
C.7.20	Facebook integrering	92
C.7.21	Spel i realtid	92
C.8	Filhantering	93
C.8.1	Spara bilder från applikationen	93
C.8.2	Ladda bilder till applikationen	93
C.8.3	Bilder skall sparas i formatet .png	94
C.9	Språk	94
C.9.1	Svenskt språk	94
C.9.2	Lätt att lägga till fler språk	94
C.10	Kod	95
C.10.1	Väldokumenterad kod	95
C.10.2	Effektiv nätverkskod	95
D	MessagePacket	96
D.1	DataMessage	96
D.2	BrushMessage	96
D.3	UndoMessage	97
D.4	RedoMessage	97
D.5	InfoMessage	97
D.5.1	PLAYER_JOINED	97

D.5.2	PLAYER_QUIT	98
D.5.3	JOIN_SUCCESS	98
D.6	QUIT_MESSAGE	98
D.7	HistoryMessage	98

1 Inledning

NetPaint är en applikation som möjliggör för användare att måla tillsammans i realtid. Genom ett intuitivt användargränssnitt, hög prestanda och möjligheten till olika inställningar är målet att skapa en applikation som ger användaren en gemytlig ritupplevelse att dela med andra.

1.1 Bakgrund

Mobilapplikationer är något som under de senaste åren blivit mycket vanliga i och med att allt fler köper enheter som dessa utvecklas för (IDC, 2011) – bland annat smarttelefoner och surfplattor. Marknaden och efterfrågan efter dessa fortsätter att växa och tillgängliga applikationer har en mängd olika syften i form av bland annat underhållning, utbildning, nyheter och webbplatser i applikationsformat. Dessa kan hämtas ner till en enhet genom ett flertal Internet-butiker, så som **App Store** och **Google Play**.

Applikationer för just underhållning - till exempel olika typer av spel - är i dag populära, då de kan fungera som tidsfördriv i brist på annan underhållning. Fördelen med att ha dem på mobila enheter är att de går att använda nästan helt oberoende av var man befinner sig och närhelst man känner för det. Det finns både applikationer där användare endast interagerar med deras egna enhet och där användarna på något sätt kommunicerar med varandra över ett nätverk. Den senare typen ger möjligheten att vara mer social och spela tillsammans med vänner eller kanske lära känna nya människor, något som blir allt mer utbrett (Müller-Veerse, 2011).

Att skapa applikationer till smarttelefoner gör att användarna kan spela på ett smidigt sätt, då de troligtvis oftast har dessa med sig och de tar relativt liten plats. Skall en sådan applikation tas fram kan det även vara en god idé att haka på trenden med flerspelarläge och låta användarna uppleva en social gemenskap, samtidigt som man också har ett enspelarläge så att applikationen kan användas i många olika situationer.

Utvecklingen av spelapplikationer för smarttelefoner kan göras för en eller flera plattformar för mobila enheter, där **Android**, **Apple iOS** och **Windows Phone** är några av de mest utbredda (Pettey, 2011).

1.2 Syfte

Syftet med detta projekt är att inrikta sig mot marknaden för spelapplikationer som den ser ut idag och utveckla en applikation av det slag som anses vara eftertraktad. Tanken är att skapa en ritapplikation med både en- och flerspelarläge, och som riktar sig till en bred publik samt är enkel och intuitiv

att använda. Flerspelarläget skall innebära att användaren är uppkopplad och kommunicerar med andra, kända eller okända människor. Applikationen, som den skulle fungera i flerspelarläge, finns i skrivande stund inte på marknaden – dock finns vanliga ritapplikationer. Syftet är alltså att använda det populära konceptet av ritapplikationer och utveckla det till något nytt som enkelt skulle kunna tas emot av dagens användare av spelapplikationer för smarttelefoner.

Syftet är även att gruppmedlemmarna skall lära sig mer om programmering av applikationer till smarttelefoner och att samarbeta i grupp.

1.3 Mål

Det mer specifika målet med arbetet är att ta fram en applikation för plattformen Android, där den huvudsakliga idén är att möjliggöra för flera användare att kommunicera grafiskt över Internet.

Applikationen som skall tas fram är menad att användas främst i underhållningssyfte och skall gå ut på att flera användare -som kanske inte känner varandra- över nätverk skall kunna rita tillsammans i realtid. Det skall också vara möjligt att måla ensam.

Mycket fokus skall läggas på prestanda, detta för att åstadkomma en produkt med snabb respons och med minimal nätverksfördröjning. Målet är också att rikta in applikationen till svensktalade användare i åldrarna 12 till 20, speciellt de med viss vana av använda applikationer för smarttelefoner. Därför kommer den att vara implementerad på ett sätt som förenklar användning för just denna grupp. Detta innebär att användargränssnitt, språk, ljud samt design kommer att implementeras med detta i åtanke. Möjligheten att använda och uppskatta applikationen skall dock finnas även för dem utanför målgruppen.

Utöver en applikation skall även en användarmanual till denna tas fram.

1.4 Problembeskrivning

Arbetet att ta fram den applikation som står som mål för detta projekt kan delas in i fyra delar; förstudie och planering, implementering, sammankoppling samt testning och dokumentation.

Innan implementationen av applikationen kan inledas måste en förstudie göras gällande vilken plattformsversion som skall användas, vilka verktyg som passar för projektet samt hur kommunikationen och arbetsfördelningen skall ske inom gruppen. Hur applikationen i stort skall vara uppbyggd och vilken funktionalitet den skall erbjuda måste hela gruppen vara överens om

innan arbetet tar fart. En grov skiss och planering av detta skall därför tas fram.

Under implementeringsfasen kommer arbetet att delas in i två huvuddelar, en nätverksdel och en del för den visuella delen av applikationen. Nätverksdelen kommer att bestå av dels en serverimplementering och dels ett ramverk som klienten använder sig av för kommunikation mellan användare. Den visuella delen kommer bestå av ett användargränssnitt, en rityta i vilken målningen sker, verktyg för hantering av utmålning och nätverksmeddelanden samt mycket av logiken i applikationen. Sedan skall även felhantering implementeras för båda delarna.

Då de båda delarna – nätverk och grafiskt gränssnitt – fungerar var för sig skall dessa sättas ihop till en enda applikation som fungerar enligt målen och kraven. Sammankopplingen görs genom implementationen av ett ramverk, som specificerar hur kommunikation till och från enheter skall ske.

Testning skall göras av de båda huvuddelarna separat, så att dessa fungerar som de skall innan de sätts ihop. Testning kommer även att göras löpande under arbetets gång och ett större test kommer att genomföras i slutet, då applikationen anses fungera på det sätt som angivits i målen och delarna är sammanfogade.

1.5 Avgränsningar

Den applikation som skall tas fram i detta arbete är inte menad att inkludera avancerade funktioner för bildredigering. Endast ett begränsat antal verktyg och effekter kommer att vara tillgängliga. Anledningen till detta är att målet med projektet är en social applikation där användarna enkelt skall kunna måla själva eller tillsammans med andra, vilket gör att ett intuitivt gränssnitt och snabb kommunikation mellan användarna är det som får fokus.

Applikationen är tänkt att tillåta totalt tre personer att samtidigt måla på samma bild. Andra avgränsningar som fått göras är att endast använda en server för applikationen, positionerad i Sverige. Det grafiska gränssnittet för applikationen kommer inte att vara optimerat för alla skärmstorlekar – de som kommer att ha testats ligger mellan 2.8 tum och 4.3 tum.

1.6 Disposition

I kapitel 2 presenteras de nödvändiga uppgifterna som utfördes innan arbetet kunde påbörjas. Här presenteras även avgörande val av utvecklingsvertyg, och plattform och version att utveckla för diskuteras.

Efterföljande kapitel 3 behandlar arbetsmetodiken gruppen adapterade för projektet och utformningen av den kodstandard, testningsmodell och

kravspecifikation som skapades.

Kapitel 4 går igenom den slutgiltiga designen och implementationen av applikationen och dess applikationsflöde. En djupare insikt om applikationens funktioner skapas även genom en redogörelse för de verktyg och trådar som används.

Kapitel 5 tar upp resultaten som uppnåddes under projektet.

I kapitel 6 diskuteras resultaten och presenterar de större problem som påträffades under utvecklingen.

Kapitel 7 redovisar slutsatser kring projektet och kapitel 8 tar upp vilka möjligheter till framtida utveckling som finns.

2 Förstudie

Innan arbetet med att implementera applikationen kunde påbörjas var en del förstudier nödvändiga. Gruppen var tvungen att gemensamt diskutera hur vissa specifika delar av applikationen skulle hanteras och hur den i grova drag skulle vara uppbyggd. Frågor gruppen fick ställa sig var ifall Android var lämpligt som utvecklingsplattform, och om så var fallet, vilken version och utvecklingsverktyg som skulle användas. För att finna svaren på dessa frågeställningar företogs en förstudie innan arbetat kunde påbörjas.

2.1 Val av plattform

Gruppen var överens om att utveckla en spelapplikation för smarttelefoner. För att avgöra vilken plattform som skulle användas togs först och främst statistik från olika pålitliga källor till hjälp.

Tabell 1: *Worldwide SmartPhone Operating System 2011 and 2015 Market Share and 2011-2015 Compound Annual Growth Rate*

Operating System	Market Share '11	Market Share '15	Unit CAGR '11-'15
Android	38.90%	43.80%	23.70%
Blackberry OS	14.20%	13.40%	18.23%
Symbian	20.60%	0.10%	-68.60%
iOS	18.20%	16.90%	17.90%
Windows Phone 7	3.80%	20.30%	82.30%
Others	4.30%	5.50%	27.60%
Total	100%	100%	100%

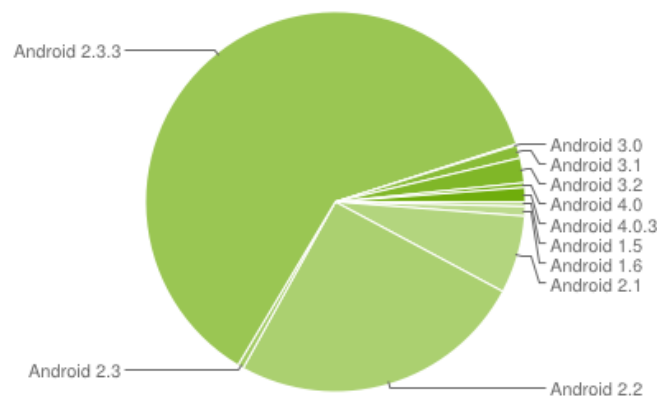
IDC förutspår att marknaden för smarttelefoner kommer att växa sig allt större (IDC, 2012), och om man ser till de olika operativsystemen för dessa kommer Android att vara det vanligaste bland användarna i dag och fram till år 2015 enligt tabell 1. Även tidningen Datormagazin (2012: 13) rapporterar att Android idag är det mest populära operativsystemet och kan vara på väg att bli ledande också i framtiden. Där skriver de att 700,000 nya Android-enheter aktiveras per dag, vilket kan jämföras med de cirka 250,000-300,000 enheter med iOS som aktiveras per dag - enligt Apples rapporter.

”Enligt många av marknadens bedömare är Android på väg att bli det de facto ledande operativsystemet för mobila enheter, på samma sätt som Windows för datorer.” (Datormagazin, 2012: 13)

I valet av utvecklingsplattform togs det även hänsyn till att Android är open-source, vilket innebär att det finns ett stort urval av exempel att tillgå. Android har även en officiell utvecklingssida (Google, 2012) med ytterligare exempel och hjälp för installation av utvecklingsmiljöer, olika sorters programmeringsramverk och beskrivningar av designprinciper. Det stora utbudet av tillgängliga exempel tillsammans med den hjälp som finns till hands på den officiella sidan skulle underlätta arbetet substansiellt, då en del gruppmedlemmar aldrig har utvecklat för smarttelefoner tidigare. Med dessa argument till grund, motiverades valet av Android som plattform.

2.2 Val av version

Valet av version att utveckla applikationen för gjordes med målet att få en så stor publik som möjligt. Därför följdes Googles råd om att alltid använda den lägsta möjliga versionen hos plattformen som applikationen kunde stödja (Google, 2012). Detta var dock något som inte gick att förutse innan applikationen var klar, då det var oklart vilka funktioner som skulle behöva användas. Därför valdes att även undersöka statistiken över vilka versioner som var vanligast, vilket enligt Google var 2.3.3, 2.2 samt 2.1 (se figur 1.1).



Figur 1.1: Andel enheter av de olika versionerna som använt tjänsten “Google Play” över en 14 dagars period som slutade 5 mars 2012

I enlighet med målet om att applikationen skall kunna nå ut till så många som möjligt, men även undvika att välja en för låg version som inte skulle klara av applikationen, valdes att utveckla i version 2.1. Valet av versionen

innebär även att alla senare versioner av Android skulle stödja applikationen, med vilka en omfattande del av alla enheter har stöd.

2.3 Android i Eclipse

Valet av utvecklingsmiljö baserades främst på vilka erfarenheter som fanns inom gruppen, samt vad som rekommenderades att använda för programmering i Android av Google – vilka skapade operativsystemet Android (Google, 2012).

Samtliga gruppmedlemmar hade tidigare arbetat med utvecklingsprogrammet Eclipse och några även med Android Development Tools (ADT), insticksprogrammet till detta. ADT är det verktyg som Google rekommenderar för att framställa en applikation i Android (Google, 2012). Android erbjuder dessutom utförliga guider på sin hemsida angående installation och användning av ADT tillsammans med Eclipse. Det var på grund av gruppens tidigare erfarenheter med att programmera i Eclipse, tillsammans med hjälpen tillgänglig via hemsidan för Android, som motiverade valet av utveckling via Eclipse.

2.4 Versionshantering - SVN

Då arbete med projektet skall vara möjligt att utföra oberoende av plats och tid bestämdes att ett versionshanteringsprogram skulle användas i utvecklingsprocessen, så att arbetsdokumenten kan synkroniseras näst intill automatiskt.

De två programmen som gruppen tidigare hade haft erfarenhet med att använda var `Git` och Subversion (SVN). Dessa var även enligt Eclipse Foundations undersökning (Eclipse, 2011) de vanligaste versionshanterarna, där 51.3% använde `SVN` som versionshanteringstjänst och 12.8% använde `Git`. Gruppens tidigare erfarenhet av `SVN`, tillsammans med det faktum att det fanns en riklig mängd information tillgänglig om användningen av det (Subversion, 2012), ledde att `SVN` valdes som versionshanterare.

Gruppen bestämde att källkoden skulle vara open-source, eftersom detta innebar att andra open-source exempel kunde användas som stöd till applikationen. Därför valdes applikationens förråd att läggas upp på Google Code, som är Googles egna server för att hantera olika typer av programspråk och program för versionshantering.

2.5 Relaterade applikationer

En del av inspirationen till applikationens funktioner kom från existerande applikationer – huvudsakligen två program kallade **Omegle** och **Fingerpaint**.

Omegle är ett anonymt kommunikationsprogram där främmande människor kopplas samman och får möjlighet chatta med varandra (Omegle, 2012). Detta koncept ville gruppen vidareutveckla genom att i stället erbjuda användare av applikationen att måla tillsammans, antingen genom ett anonymt nätverk eller över ett icke-anonymt lokalt nätverk. Möjligheten till anonymitet kan vara önskvärd eftersom kommunikationen sker mellan för varandra okända människor som kanske inte vill dela med sig av sina identiteter.

Fingerpaint är en applikation för Android som tillhör **The Android Open Source Project** (Google, 2012). Programmet är i huvudsak ett exempel med metoder för hur användarens interaktioner med smarttelefonens skärm kan översättas till rörelser som kan användas av programmet. Vid utformningen av applikationen skulle dessa metoder vidareutvecklas och anpassas till verktygen för utmålning som skulle tas fram i detta arbete.

2.6 Analys av kommunikationsval

En viktig del av applikationen är, som tidigare beskrivits, hur kommunikationen mellan användarna skall utformas för att nå målen om att vara snabb och effektiv i batteri- och nätverkskonsumtion. Kommunikationen skall även vara pålitlig, med minimalt antal förluster av meddelanden, för att den delade bilden mellan användare av applikationens flerspelarläge skall kunna hållas identisk för alla som målar på den.

2.6.1 Val av transportprotokoll

Innan arbetet med designen av kommunikationen över nätverket kunde påbörjas var ett val av transportprotokoll nödvändigt. I valet av protokoll övervägdes två olika typer av transportprotokoll, **TCP** och **UDP**, vilka båda tillåter snabb sändning av data över ett nätverk.

TCP är ett förbindelseorienterad protokoll vilket garanterar att informationen som skickas över nätverket når sin mottagare – så länge uppkopplingen upprätthålls – och att denna information mottas i samma ordning som den skickades (Network Sorcery, 2012). Att leveransen av informationen som skickas över nätverket kan garanteras är nödvändigt för det tidigare nämnda kravet av att upprätthålla en identisk bild under sessioner i flerspelarläget.

Dessa garantier finns inte i **UDP**, som i stället är ett meddelandebaserat och förbindelseöst protokoll (Network Sorcery, 2012). **TCP** innebär dock en

generellt långsammare överföring och är mer stressande för internetanslutningen, på grund av den extra information som skickas för att upprätthålla dess garantier. Avvägningen gjordes att de garantier TCP gav vägde tyngre än hastigheten UDP skulle medföra, och därför gjordes valet av TCP som transportprotokoll i stället för UDP.

2.6.2 Server eller peer-to-peer

Då ett av kraven för kommunikationen i flerspelarläget var anonymitet var det inte en möjlighet att låta användarna koppla upp sig mot varandra, genom så kallade **peer-to-peer** nätverk. Detta eftersom information om avsändare skulle paketeras med i sändningen då en uppkopplad enhet skickade data till en annan, likt avsändaren på ordinär post. Denna adress skickas med delvis för att kunna meddela avsändaren om något problem uppkommit under överföringen, men också för att mottagande sida skall kunna skicka tillbaka ett svar till avsändaren. På så sätt avslöjas avsändarens adress och kravet på anonymitet bryts.

Användning av en server som mellanhand i kommunikationen mellan användare möjliggör däremot anonymitet, då spelarna enbart ser informationen som skickas till och från servern. Den synliga adressen blir då endast den som tillhör servern och anonymiteten användarna sinsemellan kan bibehållas. Användningen av en server skulle även avlasta klienterna avsevärt gällande mängden datatrafik som skulle behöva skickas. Detta eftersom användarna endast kommunicerar med servern, i stället för med alla användare anslutna till samma flerspelarsession. Eftersom resurserna på en mobil enhet är något begränsade bidrar detta positivt till strömförbrukningen, då informationen som måste skickas över nätverket är mindre.

3 Arbetsmetod

Då gruppen bestod av sex medlemmar insågs i ett tidigt skede att en väl genomtänkt arbetsmetod var nödvändig för att involvera samtliga i projektet och skapa ett effektivt arbetsflöde.

3.1 Arbetsupplägg

Gruppmedlemmarna delades in i två grupper. De två grupperna arbetade parallellt, för att senare tillsammans arbeta med implementeringen av ramverket som skulle knyta ihop nätverksdelen och gränssnittsdelen.

Funktioner som skulle implementeras för applikationen dokumenterades i form av en kravspecifikation (Appendix C). Kraven i denna delades in i tre olika prioriteter, där de krav som låg under prioritet ett ansågs nödvändiga att implementeras för att få applikationen att fungera som tänkt. Dessa uppdaterades också under projektets gång – en del lades till eller togs bort och andra fick byta prioritet. Kraven sorterades på sådant sätt att projektet skulle ha möjlighet att fortskrida på ett så effektivt sätt som möjligt, då det var viktigt att undvika att arbetet stagnerade på grund av förseningar med vissa delar.

3.2 Arbetsmetodik för utvecklingen av applikationen

Under utvecklingen av applikationen som tagits fram i detta projekt har delar av en viss utvecklingsprocess för mjukvara använts - RUP (Rational Unified Process). Detta är en iterativ process som utvecklats av Rational Software Corporation och har använts av IBM.

“...it is not possible to sequentially first define the entire problem, design the entire solution, build the software and then test the product at the end.” (IBM, 2001)

Rational Software Corporation anser att utvecklare, genom en upprepad process, kan öka förståelsen av ett problem genom successiva ändringar och få fram en effektiv lösning genom flera iterationer. Under de olika iterationerna blir det även lättare att uppdatera de krav som existerar. För utvecklingen av denna applikation har en sådan process använts och den kravspecifikation som tidigare nämndes har anpassats till detta genom olika prioriteter.

Utvecklingen vid användandet av RUP består av fyra faser. Dessa faser är planeringsfasen, fördjupningsfasen, konstruktionsfasen och överlämningsfasen (IBM, 2001). De steg som ingår i dessa har till stor del följts under framtagningen av applikationen.

3.2.1 Planeringsfasen

Det primära målet med planeringsfasen var att göra en tillräckligt utförlig undersökning av systemet som skulle tas fram. Undersökningen av systemet bestod i att göra den förstudie som beskrevs i föregående kapitel samt att komma fram till vilka grundläggande funktioner applikationen skulle innehålla.

3.2.2 Fördjupningsfasen

I nästa fas var målet att ta fram en kravspecifikation (se avsnitt 3.5) utifrån gruppens undersökningar och diskussioner från föregående fas, samt en planeringsrapport. Dessutom gjordes en problemanalys för att eliminera de större riskerna och problemen som kunde uppkomma senare i projektet. Enligt RUP skall också en arkitekturbeskrivning tas fram, vilket gjordes i grova drag, medan större delen av den utvecklats efter hand genom de iterativa cyklerna som RUP beskriver. I denna fas togs även prototyper på ett GUI fram.

3.2.3 Konstruktionsfasen

Den tredje fasen var konstruktionsfasen, och med denna var målet en implementation av applikationen. Under denna fas utvecklades systemets komponenter och funktioner på ett iterativt sätt för att göra arbetet överskådligt och mer hanterbart. Till en början togs grunden till systemet fram, för att sedan bygga funktionaliteten iterativt utifrån den. Tack vare den kravspecifikation som togs fram i fas två kunde gruppen enkelt veta vilka funktioner som behövde implementeras först, där flera av dem kunde utvecklas samtidigt under en viss cykel. Efter varje iteration har en ny prototyp med påvisbar funktionalitet producerats och användartester har kunnat genomföras på de implementerade funktionerna. Då denna fas var över hade vi en stabil produkt med de nödvändiga kraven, samt en användarmanual för denna.

3.2.4 Överlämningsfasen

Målet med den sista fasen, överlämningsfasen, var leverans av produkten till slutanvändaren. Slutanvändaren är i det här fallet gruppens handledare, examinator samt den tänkta målgruppen. Innan dess skall en så kallad betatestning av systemet genomföras, vilket åstadkoms genom att låta en testgrupp använda och kommentera applikationen. Anledningen till detta var att se om den nådde upp till deras förväntningar och om den uppfyller sitt syfte, samt för att undersöka hur väl funktionerna har implementerats och hur användarvänligt resultatet blev. Produkten och produkttesterna jämfördes med de krav som ställts under tidigare faser. Den slutgiltiga designen av

användargränssnittet implementerades under betatestningen och de fel som upptäcktes korrigerades.

3.3 Kodstandard

För att behålla en homogen kodstil i applikationen valdes en kodstandard specificerad av dataföretaget Oracle (Oracle, 1999). Deras konventioner följdes inte strikt, men de allmänna regler om till exempel indentering, strukturering av funktionsanrop och kommentering som definierades tillämpades. Dessutom infogades beskrivande kommentarer kontinuerligt till alla applikationens funktioner. Detta gjordes för att det enligt Android Developers skulle underlätta framtida underhållsarbete samt sänka tröskeln för personer som behöver sätta sig in i koden (Google, 2012).

3.4 Testning av kod

Testning utfördes under utvecklingsprocessen och denna underlättades tack vare adaptionen av utvecklingsprocessen RUP. De funktioner som implementerats under varje iteration i konstruktionsfasen testades, vilket sedan gjorde att nya funktioner lades till på en redan fungerande grund och det förenklade eventuell felsökning. För att ytterligare effektivisera testningen kommenterades koden noggrant. Detta för att felsökning skulle kunna utföras av en medlem i gruppen som tidigare inte varit insatt i det hela.

3.5 Kravspecifikation

Målen med projektet formulerades, under den fördjupningsfas som beskrevs i föregående kapitel, som en mängd olika krav i en kravspecifikation (Appendix C). Dessa beskrev hur applikationen skulle bete sig och vilka funktioner som skulle finnas tillgängliga. På detta sätt kunde gruppen få en mall över hur delar skulle implementeras och det gjorde att alla medlemmar kunde få en gemensam bild av hur applikationen skulle vara utformad. Kraven gjorde att arbetet med specifika funktioner kunde delas upp mellan medlemmarna och underlättade även vid testning – då man kunde testa om varje funktion fungerade enligt kravet.

3.6 Användartester för grafiskt gränssnitt

För att få en så användarvänlig applikation som möjligt gjordes flera iterationer då användartester genomfördes och gränssnittet designades om. Dessa

tester innefattade observationer och intervjuer med en testgrupp, vilken bestod av fem personer med olika teknisk bakgrund i åldrarna 7 till 50. Flertalet testare hade åldrar som låg inom den tänkta målgruppen och hade tidigare erfarenhet av mobilapplikationer. Observationerna genomfördes genom användningen av **Direkt inspektion** (Prince, 2011) och användarna bads tillämpa en teknik kallad **Think Aloud**. **Think Aloud-tekniken** går ut på att testaren hela tiden skall motivera sina handlingar och berätta om sina intryck för observatören (Prince, 2011). Intervjuer gjordes efter varje observation för att få respons på vad testaren fått för intryck av de olika delarna av applikationen.

4 Design och implementation

Detta kapitel behandlar först designen av applikationen, dess flöde och upplägg, och går sedan in på själva implementationen av de olika delarna som den består av.

4.1 Applikationsflöde

I detta avsnitt beskrivs flödet för applikationen både i enspelarläge och i flerspelarläge.

4.1.1 Enspelarläge

Då användaren startar applikationen och vill måla ensam väljs alternativet **En spelare** i huvudmenyn. Användaren kommer då till en tom rityta och kan måla på denna så länge han eller hon vill, för att sedan gå tillbaka till huvudmenyn eller börja måla på en ny bild.

Under ritytan finns ett verktygsfält som spelaren kan använda för att byta storlek, färg eller effekt på penseln, sudda, ångra eller återställa den senast målade bilden samt förstora upp en del av bilden.

4.1.2 Flerspelarläge

Om användaren vill måla tillsammans med andra spelare väljs alternativet **Flera spelare**. Kommunikationen som skall användas i detta läge väljs av användaren under menyvalet **Alternativ**, där kommunikationsprotokollen för TCP/IP samt Bluetooth återfinns. Om en TCP/IP- uppkoppling valts, kommer användaren bli uppkopplad till servern och tilldelad ett slumpmässigt valt rum om det finns en ledig plats i något av de existerande sådana. Om det inte finns tillgängligt, så skapas ett nytt rum dit användaren kopplas. Då uppkoppling valts via Bluetooth kan användaren välja en enhet i närheten att kopplas ihop med, och all nätverkstrafik kommer sedan endast att gå mellan dem, utan att passera servern. Användaren tilldelas då också ett rum men det existerar bara ett rum och ingen slumpmässig tilldelning sker. Detta rum består av en rityta där upp till två spelare målar tillsammans samtidigt och delar på informationen på ritytan.

Verktygsraden finns tillgänglig, precis som i enspelarläge. En spelare kan när som helst välja att lämna rummet eller byta rum genom spelmenyn.

4.1.3 Spara/Ladda bild i applikationen

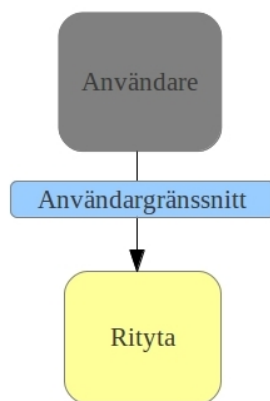
Att spara en bild ansågs vara en funktion som skulle vara inkluderad i applikationen. Detta har därför implementerats som funktioner man kan komma åt via enhetens menyknapp i både enspelarläge och flerspelarläge. Via knappen **Spara** kommer man till en dialogruta där man ges möjlighet att skriva in ett önskat filnamn, varefter filen sparas på enhetens diskminne. En **Ladda**-knapp återfinns också, där man når dialogrutan som listar de tidigare sparade filerna i enheten.

4.2 Uppbyggnad

I detta avsnitt ges en överblick på hur applikationen är uppbyggd. De lägen som är möjliga att välja, vad de innebär och hur kommunikationen sker förklaras i grova drag.

4.2.1 Enspelarläge - Användare och Enhet

Applikationen i enspelarläge består huvudsakligen av en rityta som spelaren kan måla på genom interaktion med ett användargränssnitt (figur 4.1), och för att göra detta medföljer också ett antal verktyg. I enspelarläge sker ingen kommunikation med andra enheter – spelaren ritar ensam på sin rityta.



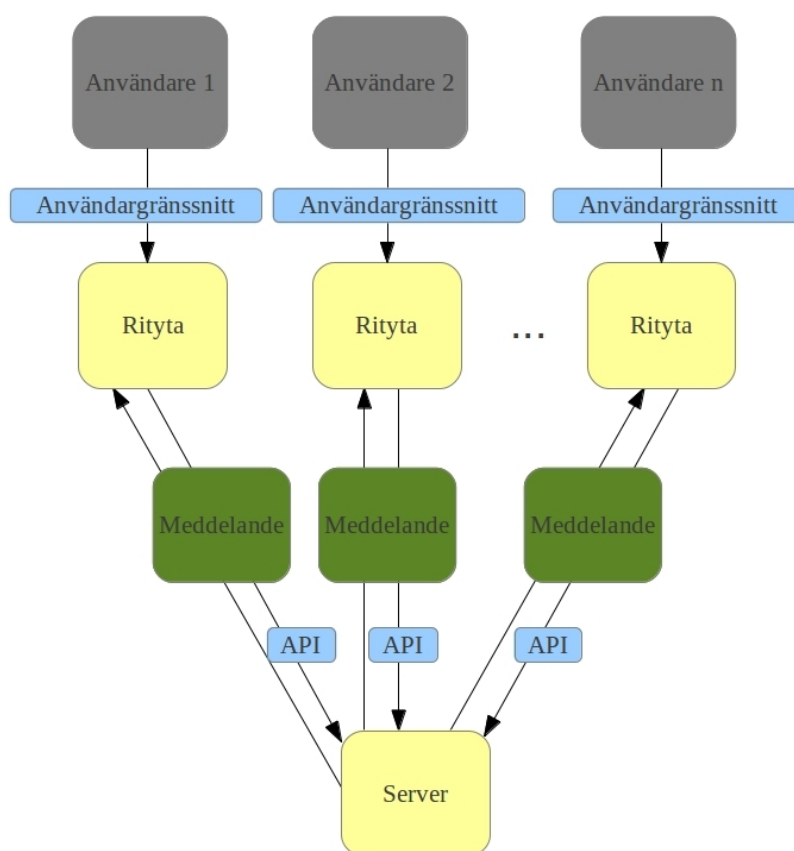
Figur 4.1 - Enspelarläge

4.2.2 Flerspelarläge - klient-server och Bluetooth

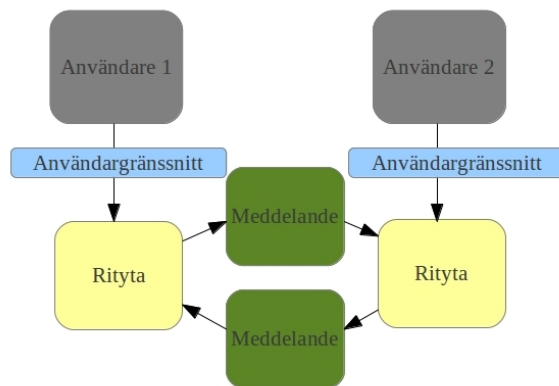
Då användaren väljer att spela i flerspelarläge finns två alternativ: att låta kommunikationen ske genom att data utväxlas via en server, eller genom Bluetooth. I flerspelarläge ritar flera användare tillsammans – de målar på sin egen rityta, vilken samtidigt uppdateras med vad andra användare målar.

Då användaren väljer klient-server-alternativet interagerar varje användare med sin egen rityta genom ett användargränssnitt och meddelanden skickas och tas emot till och från en server (Figur 4.2). Dessa meddelanden innehåller bland annat information om vad varje spelare ritar på sin egen rityta. De andra spelarna blir informerade om detta så att deras ritytor kan uppdateras. Genom detta kommer alla användare att se samma bild på sina ritytor.

Om i stället **Bluetooth** väljs för kommunikation skickas meddelandena direkt till respektive användare, utan att passera någon server (Figur 4.3).



Figur 4.2 - Flerspelarläge - Klient-Server



Figur 4.3 - Flerspelarläge - Bluetooth

4.3 Nätverks- och Bluetooth-kommunikation

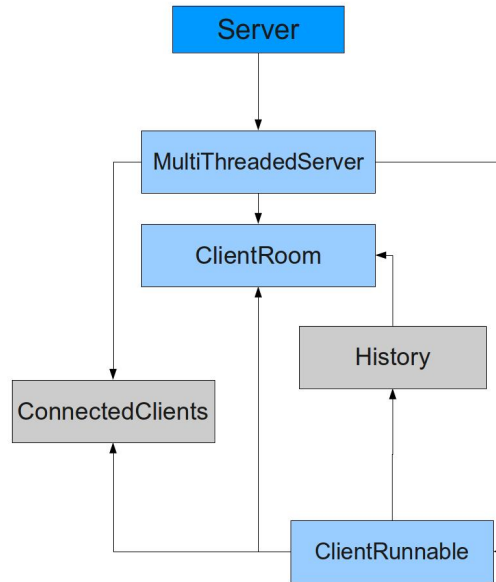
För att möjliggöra för anonym kommunikation mellan två enheter via TCP/IP konstruerades både ett ramverk och en server enligt klient-server-modellen. För att kunna måla med andra användare i sin närhet utvecklades möjligheten för kommunikation via Bluetooth.

4.3.1 Tillhörande server för applikationen

Servern som applikationen kommer att kommunicera med är en flertrådad server, det vill säga att den kan hantera flera anslutningar samtidigt. Servern väntar oupphörligt på en anslutning. När en anslutning har etablerats kontrolleras klientens IP-adress, då endast en uppkoppling från samma IP-adress skall vara tillåten.

När klienten har accepterats kommer det att skapas en separat tråd för hantering av denna klients kommunikation, och samtidigt kommer även en slumpmässig sökning ske efter ett ledigt rum för klienten så att uppkoppling mellan ett fixt antal användare kan skapas. Rummen har även i uppgift att vidarebefordra meddelanden som den får in från en klient till övriga klienter som befinner sig i samma rum.

Det finns även en påbörjad implementering av att spara historik över paket innehållande information om penselinställningar och koordinater. Ett UML-diagram för servern visas i figur 4.4.



Figur 4.4 - UML-diagram för server

4.3.2 Applikationens ramverk

Klienten är försedd med ett ramverk som innehåller standardmetoder för att kunna kommunicera med servern. Dessa metoder är till exempel att etablera en uppkoppling till servern och att utföra vissa operationer som att läsa och skriva till servern och avsluta uppkopplingen. Se tabell 2 och 3 för en mer detaljerad beskrivning av konstruktörer och metoder.

Klienten har i uppgift att tolka de olika inkommande meddelande från servern. Inläsning från servern sker i en egen tråd, av anledningen att applikationen skall fungera i full-duplex-läge, det vill säga att kommunikationen är möjlig åt båda hållen simultant. Värt att notera är att metoden för inläsning från servern är blockerande.

NetworkHandling(Handle handle) Konstruerar ett NetworkHandling objekt och tar en Handle som inparameter för att hantera uppdateringar till tråden som hanterar gränssnittet.
ListenThread(ObjectInputStream in) Konstruerar ett ListenThread objekt som är en inre klass i NetworkHandling, används för att skapa full-duplex uppkoppling mot servern. Tar en ObjectInputStream som inparameter för inkommande meddelanden. Skapar en ny tråd.

Tabell 2: *Konstruktor och dess sammanfattning*

void	establishConnection() Skapar en anslutning mot servern med förvald IP-adress och portnummer.
void	establishConnection(String server_ip, int server_port) Skapar en anslutning mot servern med en bestämd IP-adress och portnummer.
bool	writeToServer(Object msg) Sänder ett meddelande av typen Object till servern, returnerar true om sändningen var lyckad annars false.
void	disconnect() Stänger ner anslutningen mot servern.

bool	isConnected() Kollar status på anslutningen mot servern, returnerar true ifall man fortfarande är ansluten mot servern annars false.
Handler	getHandler() Returnerar den Handler som hanterar uppdateringar till den tråd som sköter gränssnittet.
void	run() Används i den inre klassen ListenThread för inläsning av meddelanden från servern, körs tills nedkoppling från servern sker. Inläsningen är blockerande.

Tabell 3: *Metoder och dess sammanfattning*

4.3.3 Dataformatet MessagePacket

MessagePacket är ett dataformat som innehåller en paketidentifierare och ett objekt. Detta paket skickas och tolkas mellan klienterna. Beroende på vilken typ av identifierare meddelandet har kommer det tolkas på olika sätt. Det vanligaste meddelandet som skickas är ett datameddelande. Ett sådant paket innehåller ett objekt av typen DataMessage som i sin tur innehåller en lista av koordinater som mottagaren kan måla upp på skärmen. Dessutom finns information om vilken spelare som meddelandet skickades ifrån, ett unikt id för det givna ritmönstret, samt en tidstämpel för att kunna sortera dem i rätt ordning.

Ett penselmeddelande används varje gång en användare ändrar sina inställningar på penseln. Då skapas ett objekt av typen BrushMessage som innehåller parametervärden för färg, penseleffekt och penselstorlek för att visa det givna mönstret korrekt på skärmen. Det skickas även med en parameter som beskriver vilken spelare som ändrade på penselinställningarna.

Vill användaren ångra eller återställa ett ritmönster finns det separata paket även för detta ändamål. Ett UndoMessage eller RedoMessage kan skapas och skickas till servern och resterande användare i rummet. Detta för att informera om att en spelare ångrade, eller återställde sin senaste ritrörelse, med hjälp av spelaridentifieraren och den unika identifieraren för det givna ritmönstret.

Det finns även ett InfoMessage-objekt som kan ta tre olika typer av meddelandeidentifierare. Den kan ta en identifierare som skapar ett meddelande för att indikera för spelarna i rummet att en ny person har gått med i rummet. För att indikera att en person har lämnat rummet, skapas ett meddelande för att notifiera resten av spelarna i rummet.

Till sist finns det en meddelandeidentifierare för att indikera att man har lyckats att gå med i ett rum. Dessa meddelanden innehåller en parameter som anger antalet personer i det aktuella rummet och en parameter som är till för att identifiera spelare som utfört en viss handling.

För att avsluta en pågående session skapas ett paket med en meddelandeidentifierare som indikerar att klienten har begärt att gå ur rummet. Detta för att meddela servern om att en nedkoppling har begärts och den aktuella klienten kan kopplas ned på ett smidigt sätt.

Identifieraren för rumshistorik är till för att skapa eller identifiera ett HistoryMessage-objekt. Den innehåller en lista av tidigare skickade MessagePacket-objekt, men endast datameddelanden och penselmeddelanden. Denna lista mottas om användaren går med i ett rum där målningar redan har skett, detta för att kunna se den befintliga bilden i rummet. Listan med historiken kan modifieras på serversidan ifall någon klient har ångrat eller gjort om sitt senaste ritmönster. Historiken är som tidigare nämnt en påbörjad implementering som inte har integrerats fullt ut, utan är till för fortsatt utveckling av applikationen. Se Appendix D för mer detaljerad information angående dataformatet MessagePacket.

4.3.4 Test av nätverkslager med hjälp av ClientApp

För att kunna testa servern och ramverkets metoder samt dataformatet under utvecklingsfasen, togs en simpel Androidapplikation fram. Syftet med applikationen var enbart att skicka en textsträng från en klient till en annan via servern.

I början av utvecklingen användes applikationen endast för att skapa en anslutning mot servern, därefter testades serverns funktioner, som att hitta ett tomt rum för användaren och att spelarens identifierare skapades korrekt. När servern hade testats, påbörjades utveckling av kommunikation mellan två klienter där strängar skickades. När väl detta fungerade krävdes inget större arbete för att gå över till dataformatet MessagePacket.

4.3.5 Kommunikation via Bluetooth

Bluetoothkommunikation togs fram som ett alternativ till TCP/IP-uppkoppling då användaren istället vill spela med någon i sin närhet. Bluetoothkommunikation är också ett bra alternativ då TCP/IP-uppkoppling inte finns tillgänglig - exempelvis krävs en trådlös accesspunkt med tillgång till Internet eller ett 3G-abonnemang i detta avseende, vilket inte alla användare har tillgång till.

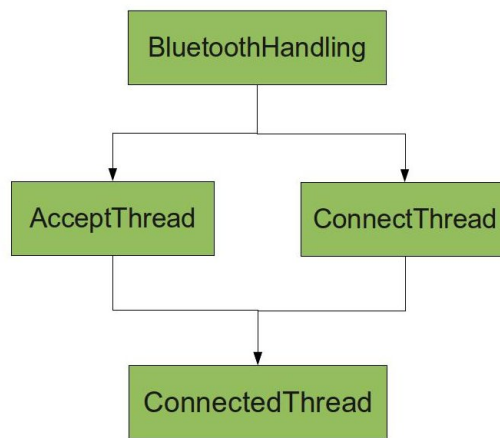
Bluetoothkommunikationen kan endast hantera kommunikation mellan

två enheter i ett **master/slave** förhållande eftersom det inte finns någon mellanliggande enhet som sköter handskakningen. Detta innebär även att varje enhet behöver agera både server och klient för att uppkoppling skall bli möjlig. Innan uppkopplingen inleds måste vissa krav uppfyllas, till exempel så måste enheten stödja och ha **Bluetooth** aktiverat. Till skillnad från TCP/IP-implementeringen krävs här lite mer involvering från användaren. Då **Bluetooth** valts för att sköta kommunikation och flerspelarläge väljs öppnas ett nytt fönster där enheter som har parats ihop sedan tidigare visas. Användaren kan här välja att söka efter fler enheter som är synliga.

Om användaren sökt efter andra aktiva enheter kommer dessa att visas i en särskild lista i fönstret. När den hittills okända enheten valts kommer systemet att leda användaren genom ett antal steg för att utbyta nycklar med den andra enheten, som är till för att upprätta en säker koppling. När detta steg har slutförts är enheterna parade och kommer att komma ihåg varandra för nästa tillfälle då applikationen används.

Rent programmeringsmässigt är Androids klasser för **Bluetooth** mycket lika de som finns i Java för TCP/IP, vilket har gjort arbetet med detta enklare. För att förloppet som tidigare har beskrivits skall fungera korrekt måste varje enhet upprätta en `BluetoothServerSocket` (Google, 2012) som ligger och lyssnar efter förfrågningar om anslutning i en egen tråd. Om användaren på enheten är den som initierar uppkopplingen kommer tråden med en öppen `BluetoothServerSocket` att avslutas. När uppkoppling har upprättats tar en annan tråd över, vars uppgift är att hålla i anslutningen. I denna skapas både in- och utströmmar från en `BluetoothSocket` (Google, 2012) som enheten nu kan ta emot och skicka dataformatet `MessagePacket`.

Figur 4.5 visar ett UML-diagram för hur **Bluetooth** implementerats.



Figur 4.5 - UML-diagram för Bluetooth

4.3.6 Test av Bluetoothkommunikation med hjälp av Bluetooth-Client

För att kunna testa funktionaliteten hos **Bluetooth** skapades en testapplikation i stil med **ClientApp**, vilken beskrevs i avsnitt 4.3.4, som användes då **TCP/IP** testades. Till en början gjordes ett försök med att göra **BluetoothClient** snarlik **NetPaint** för att förenkla arbetet med implementering av Bluetooth-delen. Detta genom att införa en knapp för alternativ där man skulle kunna välja mellan **TCP/IP** och **Bluetooth**. Tyvärr visade sig detta vara svårt att genomföra i ett såpass tidigt skede av Bluetooth-utvecklingen då inte ens kommunikationen fungerade som den skulle.

Beslutet togs att förenkla applikationen så mycket som möjligt för att komma till roten med problemen som fanns. När problemen väl lösts var det inga vidare svårigheter att ersätta strängar med **MessagePackets**. Sedan kunde **BluetoothClient** användas som ett exempel för de GUI-specifika delarna som behövdes, som att begära att användaren aktiverar Bluetooth och att presentera en lista med tillgängliga enheter.

4.4 Trådar och processer

När **Bluetooth** är valt som kommunikationsalternativ används ett antal trådar för att upprätta en uppkoppling mellan enheterna. De används i totalt tre olika faser som är; lyssnar-, anslutnings- och kopplad-fasen. Dessa trådar behövs för att applikationen skall kunna lyssna efter förfrågningar om anslutning samtidigt som användaren har möjlighet att själv initiera anslutningen.

Vi följer den allmänna principen om att endast hantera uppdateringar av gränssnittet i dess egna tråd och resterande logik i övriga trådar. Om

inte principen följs kommer det spegla sig i att applikationen förefaller icke responsiv.

Kommunikationen inleds av att enheten skapar en lyssnartråd som krävs för att andra användare skall kunna koppla upp sig mot telefonen via **Bluetooth**. Tråden körs då användaren har valt **Bluetooth** som uppkopplingsmetod i inställningsmenyn och därefter i huvudmenyn väljer att spela i flerspelarläge. I tråden skapas en `BluetoothServerSocket` (Google, 2012) som lyssnar efter inkommande förfrågningar om uppkoppling. I det fall att en annan användare skickar en begäran om att koppla upp sig mot telefonen kommer det att skapas en vanlig `BluetoothSocket` (Google, 2012) och tråden kommer att skicka denna vidare till den tråd som hanterar uppkopplingen då denna är fastställd. Om däremot användaren är den som initierar uppkoppling kommer den blockerande lyssnar-metoden få ett undantag och tråden kommer då att avslutas.

I denna tråd som hanterar uppkopplingen skapas två objektströmmar utifrån den `BluetoothSocket` som tidigare skapats - en ut- och en inström. Utströmmen kommer att användas då data i form av `MessagePacket` skall skickas till den andra telefonen. Inströmmen används för att lyssna efter inkommande `MessagePacket` som sen skickas vidare till huvudprogrammet. Tråden kommer enbart att avslutas då uppkopplingen avbryts av olika skäl, exempelvis då användaren väljer att avsluta sessionen.

Den sista tråden att nämnas körs när användaren är den som initierar anslutningen genom att välja en annan enhet att ansluta mot i listan av tillgängliga enheter. Då startas tråden med en `BluetoothDevice` (Google, 2012) varifrån det skapas en `BluetoothSocket` som sedan blir indata till tråden för att hantera uppkopplingen. När detta har gjorts har tråden uppfyllt sitt syfte och tas då bort.

På samma sätt används trådar när kommunikationen sker via **TCP/IP** bortsett från det att en tråd som lyssnar efter anslutningar inte behövs då den i detta fallet körs på servern. Dessa trådar diskuteras i följande stycken.

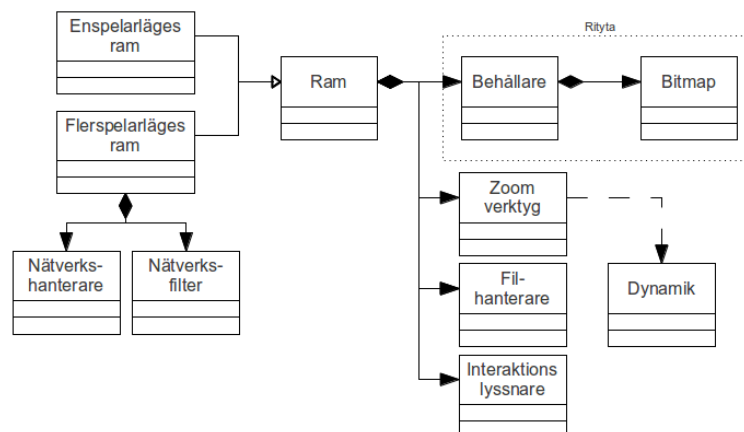
Först och främst krävs en tråd för att lyssna efter meddelanden som skickas till telefonen. Tråden får in en ström som inparameter där den lyssnar efter `MessagePacket` tills dess att anslutningen bryts. För att simultant kunna rita ut flera användares linjer skapas en tråd för varje motståndare som tar emot koordinater för linjer som den sedan ritar ut.

Utöver tidigare nämna trådar tillhörande **TCP/IP** används en tråd för att skapa rätt typ av uppkoppling och eftersom det kan ta en stund innan anslutningen har upprättats kan detta inte ske i gränssnittstråden av tidigare nämnda skäl.

4.5 Ritytan och dess komponenter

Ritytan är ett centralt objekt i applikationen. Det är här själva ritandet sker och det som användaren ser när ett spel körs. Ytan har funktioner för att rita, sudda, zooma och panorera över ytan, och hanterar interaktion från användaren – både ritrörelser på ytan och knapptryckningar på menyn.

Arkitekturen för ritytan består av tre lager. Det yttersta lagret fungerar som en ram åt de lägre lagren, och innehåller metoder för säker åtkomst till dem. Ramen innehåller dessutom de objekt som extraherades från ritytan, vilka hanterar funktionaliteten runt denna utöver själva målandet och listan över alla sparade ritrörelser. Listan hålls sorterad efter ordningen de ritades i. De två inre lagren är dels en yta som utmålningarna skall ske på och dels en behållare till den. Behållaren innehåller, utöver metoderna för att komma åt ytan, de verktyg som krävs för att skapa ritrörelserna som skall målas ut till den. Lagerindelningen gjordes för att göra utmålningen till ritytan mer överskådlig. I figur 4.5 visar ritytans olika delar och hur dessa är sammankopplade.



Figur 4.6 - Ritytans komponenter och relationer

4.5.1 Ritrörelsen och de objekt som bygger upp den

För att förenkla processen av att måla till ritytan skapades ett objekt som representerar de ritrörelser som används vid utmålning till ritytan. Detta innehåller information om vem som skapade ritrörelsen, när den skapades, koordinaterna rörelsen går igenom, ett unikt identifikationsnummer samt en pensel. Informationen visar sig användbar först när flerspelarläget är valt som speltyp, då den används för att koppla ritrörelsen till den spelare som skapade den.

Penseln, vilken utgör en av de tre delar som bygger upp en ritrörelse, innehåller information om penselns färg, storlek och aktiva effekter. Det är denna pensel som används för att färgsätta samt specificera storleken och effekterna på de ritrörelser som målas ut. Färgen i sig består av fyra separata delar: värden för mängden av färgerna röd, grön och blå samt ett alfavärde. Detta kallas ARGB-färger.

Då ritrörelser skall läggas till i listan över alla sparade ritrörelser körs först ett test på det sista elementet i listan. Ifall den ritrörelse som skall läggas till inte skapades efter den som ligger på sista platsen i listan kommer den behöva sorteras in. Om så är fallet måste även ritytan målas om, vilket görs genom att rensa ritytan, genomlöpa listan och måla om alla ritrörelser. Detta är dock endast möjligt i flerspelarläge, då flera spelare samtidigt skapar ritrörelser som pågår under olika långa tidsintervall och skickar dessa över ett nätverk med olika tidsfördröjning.

4.5.2 Verkttygsraden i ritytan

Ritytan innehåller verktyg som sköter ett flertal funktionaliteter kring interaktion med ritytan, såsom dess måla-, sudd-, zoom-, ångra- och "gör om"-verktyg. Dessa alternativ lades i en verkttygsrad under ritytan, vilken innehåller en knapp för varje verktyg. Andra alternativ som finns tillgängliga för användaren är att spara och ladda bilder, samt att i ett aktivt spel kunna påbörja ett nytt.

Alternativen för att spara och ladda bilder, samt skapa ett nytt spel, lades i Androids alternativmeny. Denna meny är kopplad till **Menu**-knappen på framsidan av telefonen. När meny visas lägger den sig över applikationen som ett lätt genomskinligt lager längs botten av skärmen, och innehåller de ovan nämnda alternativen.

Av de knappar som finns på verkttygsraden är knapparna för rit-, sudd-, och zoom-verktygen designade som radioknappar. Detta då endast ett av dessa verktyg är tillåtet att användas i taget. Verkttygen för att ångra och återställa ritrörelser placerades i stället som vanliga knappar, då funktionen hos dem körs endast en gång och deras funktion kan, aktiverade, köras tillsammans med andra verktyg.

4.5.3 Dialogrutor till verktygen

Verktygen för att rita, sudda och zooma kräver extra funktionalitet från sina knappar. Utöver möjligheten att aktivera de olika verktygen skall deras egenskaper gå att ändra. För att ändra inställningar om aktuell pensel och sudd implementerades dialogrutor med alternativ för dessa. Verkttyget aktiveras

så fort en nedtryckning av dess knapp sker, men fortsätter nedtryckningen under en viss tid aktiveras även verktygets extra-funktion.

Ritverktyget använder sig av en pensel och dess dialogruta innehåller därför ett alternativ för att ändra penselns färg, storlek och effekter, vilket görs genom olika formulärelement. Alternativen för att ändra penselns storlek och aktiva effekter görs genom redan existerande formulärelement. Penselns storlek ändras genom en enkel rullista som går mellan värdena 1 till 50, där värdet motsvarar diametern på penseln. För att se vilken storlek som är vald till penseln, utöver positionen av markören på rullistan, visas storleken i en text ovanför rullistan.

Effekterna väljs från en lista med en kort text som förklarar deras funktion och en radioknapp för att aktivera effekten. **Relief**, **Oskärpa** och **Skugga** kopplas till effekterna för att skapa reliefer, diffusa kanter och transparenta ritrörelser. Listan innehåller även ett alternativ för att inaktivera effekter med texten **Normal**, som även den kopplas till en radioknapp.

För att göra valet av penselns färg intuitiv implementerades ett nytt system - en färgcirkel med de tillgängliga färgerna och en mindre cirkel i mitten av denna som visar den valda färgen. Färgerna i den yttre cirkeln har tre punkter för färgerna röd, grön och blå jämnt utspridda. Mellan varje par av grundfärgerna i cirkeln finns deras möjliga färgkombinationer. Färgerna svart och vitt placerades för sig själva i rutor under färgcirkeln.

Sudd-verktyget har endast en varierbar storlek. För att ändra storleken skapades därför en mindre dialogruta, i vilken användaren har möjligheten att ändra storleken genom en rullista. Zoom-verktyget har en extra funktion som återställer förstoringen och panoreringen av ritytan. Den funktionen kräver ingen dialogruta, utan aktiveras direkt.

4.5.4 Skillnaden mellan enspelarläget och flerspelarläget

Då flerspelarläget fungerar som en utbyggnad till enspelarläget användes samma rityta och ritverktyg som grund, med funktionaliteten för att stödja flera spelare implementerades som ett tillägg ovanpå den ursprungliga ritytan. Den tillagda funktionalitet innehåller nya filter och verktyg för att hantera utmålning av flera användares ritrörelser samtidigt på ritytan, samt hantering av den kommunikation som krävs för att alla spelare skall veta vad deras medspelare målar.

Verktygen för nätverkshanteringen skapades för att förbättra abstraktionen av informationsflödet. De lyssnar på inkommande meddelanden från den aktuella nätverkshanteraren och vidarebefordrar dessa, genom ett filter, till ritytan. Ritytan plockar sedan isär meddelanden och återskapar den handling som meddelandet motsvarar.

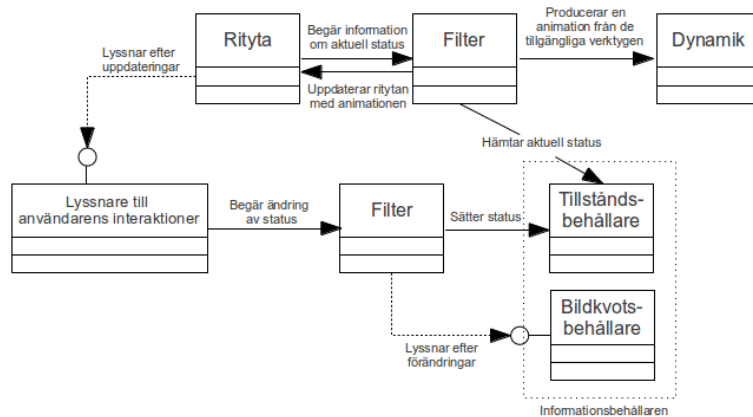
Nya filter infördes för att hantera det flöde av meddelanden som sker mellan klienterna och deras ritytor. Ett filter lades mellan ritytan och inkommande meddelanden, och dess uppgift är att filtrera inkommande datameddelanden så att deras innehåll kan rekonstrueras. Det andra lades mellan ritytan och utgående meddelanden. Detta gjordes för att rensa meddelanden från skräpdata innan de skickas på ett överskådligt sätt. De båda filtren fungerar även som temporära förvaringsutrymmen för meddelanden i deras respektive riktning, då meddelandena måste passera och behandlas av filtren innan de skickas vidare.

För att skapa de meddelanden som skall skickas vidare användes de redan implementerade metoderna i ritytan. Lyssnare placerades vid de metoder som var ansvariga för skapandet av ritrörelser till ritytan, där de punkter som används vid konstruktionen av ritrörelsen fångas upp och sparas ner i block av punkter. Dessa skickades senare vidare som meddelanden tillsammans med information om avsändaren.

4.6 Zoomfunktion

En av de mer avancerade funktioner som implementerades var den tidigare nämnda zoomfunktionen, som finns tillgänglig för användarna i verktygsfältet under ritytan. Funktionen erbjuder metoder för att förstora och panorera ritytan – inom vissa gränser, och krävdes för att möjliggöra spel mellan användare med skärmar av olika dimensioner.

Arkitekturen hos zoomfunktionaliteten består av en lyssnare för användarens interaktion med applikationen och den rityta som funktionaliteten var ämnad för. Till dessa skapades även nya filter och funktioner för tillämpningen av de fysikaliska lagar som styr animationerna vid panorering av ritytan. För att få bättre abstraktion av informationsflödet mellan lyssnaren och ritytan så extraherades viss information från dessa. Information som extraherades rör den aktuella graden av förstoring och panorering av ritytan, vilket sparades ner i en ny extern behållare. Figur 4.7 visar hur zoomfunktionens olika delar relateras till varandra.



Figur 4.7 - Zoomfunktionen och dess komponenter

4.6.1 Uppbyggnad av zoomfunktionen

Filter infördes för att underlätta hanteringen av informationsflödet som sker mellan olika objekt som försöker läsa av eller skriva till behållaren, vilket var anledningen till att två filter skapades. Det första filtret skapades för att begränsa användarens interaktioner, vilket är informationen som sparas i behållaren. Det andra filtret infördes för att kunna kontrollera flödet från behållaren till ritytan. Detta för att kunna skapa dynamiska animationer i zoomfunktionaliteten från informationsflödet. Dynamiken innefattar lagar för dämpade rörelser och friktion, vilka används för att göra förstoring och panorering av ritytan naturligare och mer intuitivt genom diverse animationer. Kommunikationen mellan olika objekt hanteras genom implementationen av designmönstret **Observer**.

Själva zoomfunktionaliteten implementerar en behållare av typen

Observable. Ritytan, samt filtret som kontrollerar informationsflödet mellan ritytan och behållaren, implementerar typen **Observer**. Implementationen av detta designmönster förbättrar abstraktionen av informationsflödet. Lyssnaren kan, genom ett filter, uppdatera den sparade informationen om förstoring och panorering, vilket sedan kan avläsas av andra objekt genom en notifiering om ändringarna från behållaren. Det sker aldrig någon direkt kommunikation mellan lyssnaren och ritytan.

Informationsbehållaren innehåller värden för den aktuella graden av förstoring, förflyttning av ritytan i x- och y-led och information om tidigare interaktioner såsom deras hastigheter, riktning och acceleration samt skillnaden i bildkvot mellan skärmen och ritytan. Graden av förstoring bestämmer hur stor del av ritytan som skall visas på skärmen. Informationen om förflyttningen av ritytan i x- och y-led används för att bestämma panorering av ritytan. Informationen om skillnaden i bildkvot mellan ritytan och skärmen används

för att separera skillnad i förstoring som krävs i x-led respektive y-led för att få rätt uppskalning av ritytan.

Informationen om förstoringen och förflyttningen i de två leden möjliggör beräkningen av dimensionerna på den del av ritytan som skall visas. För att sedan skala upp den synliga delen av ritytan till skärmen behövs även informationen om skillnaden mellan ytornas bildkvoter, så att skärmen fylls ut ordentligt.

4.6.2 Dynamik i zoomverktyget och dess animationer

Dynamiken i systemet, och dess animationer, använder sig av konstanta värden för en friktionsfaktor och fjäderkonstant. Friktionsfaktorn bestämmer hur hastigheten av panoreringen skall dö ut, då detta sker inom ramen för ritytan, enligt de fysikaliska lagarna för dämpade system. Utanför ramen beter sig i stället ritytan som ett fjädersystem, där fjäderkonstanten används för att räkna ut kraften med vilken ritytan skall dras tillbaka in till skärmens centrum.

För att kunna implementera animationerna måste även rörelseekvationerna beräknas för de rörelser som skapas genom användarens interaktioner med skärmen, vilket görs numeriskt genom **Eulers stegmetod**. Precisionen på metoden är låg, och kan visa sig vara instabil, men i implementation behövs detta inte tas hänsyn till då inga beräkningar som kräver stor precision utförs.

De dynamiska rörelser som skapas vid panorering, under vilka ritytan temporärt kan lämna den begränsade ytan av skärmen, måste kontrolleras så att den uppskalade delen av ritytan aldrig lämnar skärmen, utan bara ger en illusion av det.

Illusionen av att ritytan lämnar skärmen skapas genom att låta den målade bilden i ritytan bete sig som ett pappersark som ligger ovanpå en grå yta. Denna gråa yta kan visas tillfälligt på skärmen då användaren panorerar utanför ritytans gränser. När detta händer startas även en animation som drar ritytan tillbaka mot skärmens centrum. För att få animationen så realistisk som möjligt används **Hookes lag**.

Hookes lag beskriver den återställande kraft som verkar i fjädersystem där kraften, i första approximationen, är proportionell mot avvikelsen från systemets jämviktsläge. Lagen appliceras på systemet med fjäderkonstanten som den proportionella faktorn och positionen av ritytans som systemets jämviktsläge. När sedan användaren panorerar utanför ritytans gränser beräknas kraften, som återigen drar bilden av ritytan inåt, med den avvikelse bilden har från ritytans position.

Den påtvingande kraften, som drar ritytan tillbaka mot skärmens mitt

tillsammans med den grå bakgrundsytan, hjälper användaren att avgöra var ritytans kanter finns.

4.6.3 Interaktioner med zoomverktyget

För att styra zoomfunktionaliteten gjordes en distinktion mellan de fall att användaren håller nere fingret på skärmen eller drar det över ytan. Detta, tillsammans med begränsningen att endast tillåta en typ av interaktion åt gången (panorering eller förstoring), gjorde interaktioner med funktionaliteten mer intuitiv.

De två operationerna inom zoomfunktionaliteten kopplades till var sin typ av interaktion. När användaren håller fingret nedtryckt på skärmen under en tid aktiveras förstoringsläget, vilket signaleras genom en kort stunds vibration från telefonen. Om användaren i stället trycker ner fingret, för att sedan dra det över skärmen, aktiveras panoreringsläget. Detta för att kunna skilja mellan dessa två olika typer av händelser och en tidtagare.

Tidtagaren startas när interaktionen först registreras. Den används för att sätta en tidsgräns på hur länge användaren måste hålla sitt finger nedtryckt inom ett gränsområde, innan förstoringsläget aktiveras. Går rörelsen utanför gränsområdet innan en viss tid har passerat stoppas tidtagaren och panoreringsläget aktiveras istället.

4.7 Utveckling av användargränssnitt

I detta avsnitt presenteras slutversionen och stegen till det användargränssnitt som tagits fram för applikationen. Under designprocessen användes Googles riktlinjer (Google, 2012) för att få fram ett gränssnitt som passade Android-plattformen, och ett flertal användartester genomfördes (se avsnitt 3.6). De instruktioner som följdes vid dessa tester hittas i Appendix B.

4.7.1 Initiellt användargränssnitt

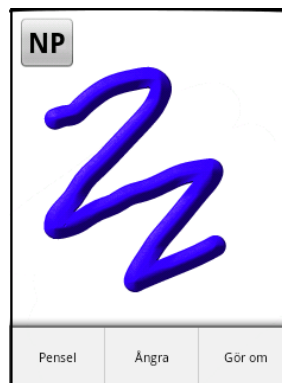
Designen av användargränssnittet inleddes tidigt under projektets gång genom att en pappersprototyp konstruerades som mötte de krav som tagits fram. Prototypen ledde till en implementation enligt figur 4.8 - 4.11. Figur 4.9 visar den rityta som användaren kan måla på, och den meny som visas längst ner på bilden dyker upp då telefonens menyknapp trycks ner. I denna meny placerades implementerade verktyg, och knappen längst upp i vänstra hörnet märkt NP öppnar upp ytterligare en meny med alternativ för att spara eller ladda en bild. Genom att använda två olika menyer kunde en uppdelning av funktionerna åstadkommas.

Figur 4.10 visar den dialogruta för penselinställningar som öppnas då man väljer alternativet **Pensel** i den lägre menyn. Designen på denna ändrades inte nämnvärt mellan den initiella implementationen och slutversionen, eftersom testgruppen i allmänhet var nöjd med den. Figur 4.11 visar ett tidigt förslag till val av rum i flerspelarläge.

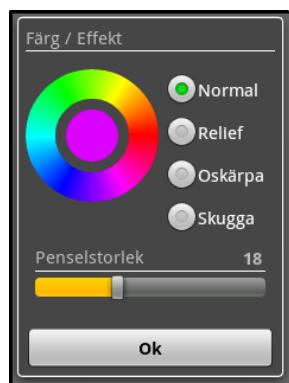
Googles råd om att stödja olika upplösningar och skärmstorlekar hos telefoner (Google, 2012) användes och en minsta storlek som skulle accepteras sattes. Gränssnittet designades så att det blev användbart för denna minsta storlek, men också så att större skärmar skulle dra nytta av sin storlek. Google nämner även att det är bra att designa användargränssnittet så att det fungerar både i landskapsvy och porträttvy, men då det av gruppen ansågs för komplicerat och tidskrävande att designa, stängdes landskapsvyn av och mer tid kunde läggas på att optimera utseendet i porträttvy.



figur 4.8 - Huvudmeny



figur 4.9 - Rityta



figur 4.10 - Penseldialog



figur 4.11 - Flera spelare

4.7.2 Andra versionen av användargränssnittet

Då första versionen av användargränssnittet var implementerad gjordes de första användartesten. I dessa uteslöts flerspelarläget, då detta inte fanns tillgängligt ännu. Resultaten från dessa visade att användarna tyckte att applikationen hittills var lätt att förstå, att de tyckte om penseldialogen och menyn som kom fram då menyknappen trycktes ner. Däremot var det flera testare som kritiserade knappen märkt NP och dess placering, så i andra versionen av gränssnittet kom gruppen därför fram till en annan design på ritytan.

Alla verktyg lades in i en fast meny längst ner på skärmen (figur 4.13) med motiveringen att Google (Google, 2012) påstår att det är en god designprincip för att göra viktiga verktyg snabba att nå. Under denna iteration började också kravet för en egenframställd design, med ett tema kopplat till applikationens syfte, att implementeras. Detta resulterade i en prototyp för huvudmenyn enligt figur 4.12, och ett verktygsfält enligt figur 4.13.

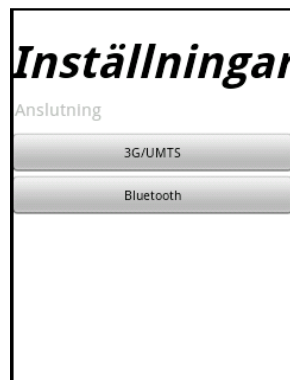
En funktion för att skapa en ny bild hade nu också blivit möjlig och en ikon för detta lades till i verktygsfältet. Genom att klicka på diskettikonen i verktygsfältet fick man fram samma meny som tidigare öppnats med hjälp av knappen märkt NP. En första version av strukturen för fönstret *Inställningar* (figur 4.15) togs också fram, men utan den egenframställda designen.



figur 4.12 - Huvudmeny



figur 4.13 - Rityta



figur 4.14 - Penseldialog *figur 4.15 - Inställningar*

4.7.3 Tredje versionen av användargränssnittet

Resultatet från de användartester som gjordes på andra versionen av gränssnittet indikerade att designen var för barnslig, men också att den statiska verktygsraden uppskattades. Dessutom gav användarna exempel på ytterligare funktioner de önskade skulle vara möjliga i applikationen - till exempel suddgummi och en zoomfunktion, vilka båda implementerades i version tre.

Designen gjordes om för att bli mer stilren, men grundidén med att använda ett statiskt verktygsfält behölls. Alternativen för att skapa en ny bild, ladda och spara bilder togs bort och lades in i en meny som visas då användaren trycker på telefonens menyknapp. På så sätt anpassades gränssnittet till Googles princip om att endast de funktioner som är viktiga för sammanhanget skall vara direkt synliga (Google, 2012).

Knapparna i applikationen designades om för att i högre grad ge användarna respons på sin interaktion, och i verktygsfältet (figur 4.17) fick dessa ramar runt om sig för att ge en indikation på vilket verktyg som var valt för tillfället. Ikonerna för knapparna baserades huvudsakligen på bilder tagna från Google (Google, 2012), vars syfte just är att fungera som mallar vid design. Tipsrutor lades också till (figur 4.18) för att användaren lättare skulle förstå de olika verktygen, och nya bakgrunder lades till i huvudmenyn (figur 4.16) samt fönstret för alternativ (4.21).

Version tre innefattar även flerspelarläget, men användargränssnittet för detta skiljer sig inte emot det för enspelarläget mer än en dialogruta som visas då anslutning via **Bluetooth** skall ske. Denna dialog var dock standard för denna funktion i Android, så ingen egen design behövdes tas fram.

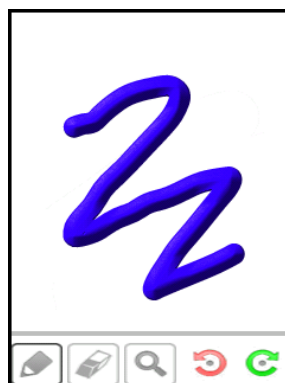
4.7.4 Slutversionen av användargränssnittet

Då nästa användartest hade genomförts var testgruppen nästan helt nöjd, och endast några få detaljer återstod att förbättras. Bland annat gjordes en ny design till fönstret för alternativ (figur 4.21), och förklaringarna till valet av kommunikationstyp förenklades för att kunna förstås av både yngre och mindre tekniskt kunniga användare.

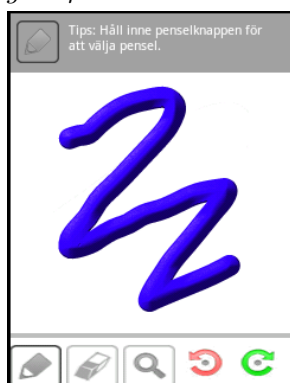
Då dessa ändringar gjorts hade testarna inget särskilt att anmärka på, och de problem som tidigare funnits ansågs lösta. Kommentarer till användargränssnittet som det såg ut nu var att det var lätt att förstå, och att dess enkelhet gjorde att man inte blev förvirrad - vilket enligt Android Developers är väldigt viktigt (Google, 2012). Slutresultatet för gränssnittet visas i figur 4.16 - 4.21.



figur 4.16 - Huvudmeny



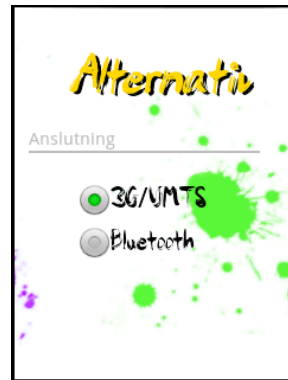
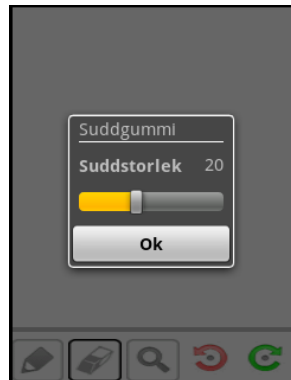
figur 4.17 - Rityta



figur 4.18 - Tipsruta



figur 4.19 - Meny



figur 4.20 - Suddgummi figur 4.21 - Alternativ

4.8 Filhantering

När applikationen installeras på en telefon skapas automatiskt en mapp för dess inställningar. Här lagras diverse inställningar som systemet sparar och mappen kan även komma åt via filhanterare. Sökvägen är placerad på enhetens externa lagring, vilket vanligtvis innebär ett minneskort (Google, 2012). Det krävs kontroll innan skrivning görs för att säkerställa behörigheter, att tillräckligt ledigt utrymme finns samt att sökvägen är monterad och åtkomlig, vilka utförs av applikationen. När detta är gjort tas ritytans tillstånd och skrivs till en bildfil, och ett meddelande visas för användaren. Skulle denne i stället vilja läsa in en tidigare sparad bild så används ovan nämnda sökväg av applikationen för att lista alla bildfiler i mappen.

4.8.1 Filformat

Android förespråkar användning av bilder med PNG-format, vilket står för Portable Network Graphics. Det är en känd standard (G. Randers-Pehrson, et. al. 1999) som stöds av alla stora webbläsare samt Android-versioner och efterträder det föråldrade GIF-formatet. Det nyare formatet PNG är lämpligast vid enklare grafik (se figur 4.22) då dess förlustfria kompression ger störst effekt då förhållandevis stora områden i bilden har samma färg. Det passar därför bra på bärbara enheter där exempelvis programikoner ofta är vektor-baserad och i formatet SVG. Då grafiken som applikationen som tagits fram under detta projekt producerar är förhållandevis simpel, och på grund av det något begränsade utbudet av andra bildformat, valdes PNG-formatet.



figur 4.22 - Exempel på programikoner med enkel grafik

4.9 Felhantering

Alla program som skapas bör hantera fel för att undvika olyckliga situationer. Säker programmering är ofta en nödvändighet och programspråk erbjuder många möjligheter att kontrollera exekvering. Eftersom användarna är människor med analogt och ibland oförutsägbart beteende kommer eventuella brister till slut att visa sig. Nedan följer några av de händelser som beaktats.

4.9.1 Ingen nätverksåtkomst

Om Androidtelefonen saknar åtkomst till Internet eller Bluetooth kommer ett fel ske då en socket skall skapas, användaren blir då varse om detta då ett meddelande som lyder “Fel vid anslutning” visas på skärmen.

4.9.2 Anslutningen bröts

Om anslutningen bryts under ett pågående nätverksspel får användaren lite olika meddelanden beroende på hurvida TCP eller Bluetooth används. För Bluetooth är meddelandet “Den andra användaren är inte längre tillgänglig” och för TCP är det kvarvarande spelare som upplyses genom ett meddelandet som lyder “En spelare lämnade rummet. Antal spelare #”.

4.9.3 Uppkoppling misslyckades

Då användaren försöker ansluta sig med Bluetooth och uppkoppling inte lyckas kommer ett meddelande om “Uppkoppling misslyckades” att visas.

4.9.4 Inga enheter funna

När Bluetooth är valt som uppkopplings metod och användaren väljer att spela i flerspelarläget finns alternativet att söka efter nya enheter. Om inga nya enheter hittades kommer ett meddelande i fönstret som listar andra enheter att visas, det lyder "Inga enheter funna".

4.9.5 Försök att ansluta simultant

Om två enheter i samma tidpunkt försöker ansluta till varandra via Bluetooth kommer enheterna att fortsätta vidare till ritytan men strax därpå få flera meddelanden. I vissa fall kommer dessa meddelanden att förefalla förvirrande då enheterna kan få ett meddelande om att anslutningen lyckades men direkt efteråt få ett meddelande om att anslutningen bröts. Detta beror på att det här felet uppdagades relativt sent under utvecklingen och genererade till en början en krasch av applikationen, den lösning som nu gjorts undviker denna krasch men lider av nämnda bi-effekter.

5 Resultat i form av implementerade krav

I följande kapitel diskuteras till vilken grad kravspecifikationen (Appendix C) har följts, och vad detta innebär för applikationen. Arbetet har dock även resulterat i en användarmanual (Appendix A).

5.1 Implementerade krav

I slutet av detta arbete hade alla krav av prioritet ett i kravspecifikationen implementerats, vilket av gruppen ansågs tillräckligt för att få en fungerande applikation och uppnå syftet med denna. Dessutom har en stor del av kraven under prioritet två också uppnåtts, vilka innefattade mer avancerade funktioner och funktioner som gör användarupplevelsen bättre. Det finns också en del krav som inte är implementerade, men som skulle kunna bli det i framtiden.

5.1.1 Väsentliga implementerade krav av prioritet 1

- Stöd för Android 2.1 och uppåt
- Kommunikation genom 3G/WiFi
- Användning av TCP vid kommunikation
- Sändning av data under beröring av ritytan
- Touchfunktion för ritning och menynavigering
- Notifieringar vid fel
- Lättförståeligt användargränssnitt
- Möjlighet att spela ensam
- Möjlighet att spela två spelare
- Penselfärg kan väljas
- Spelet sker i realtid
- Applikationen är på svenska
- Det är lätt att lägga till fler språk
- Vældokumenterad kod
- Effektiv nätverkskod

5.1.2 Väsentliga implementerade krav av prioritet 2

- Kommunikation via Bluetooth
- Egendesignat användargränssnitt
- Möjlighet att spela två eller tre spelare
- Penseleffekt kan väljas
- Penselstorlek kan väljas
- Användarna delas in i rum i flerspelarläge
- Slumpmässig tilldelning av rum
- Möjlighet att zooma
- Möjlighet att panorering inzoomad bild
- Möjlighet att sudda
- Möjlighet att ångra det man ritat
- Möjlighet att göra om det man ångrat
- Spara bilder från applikationen
- Ladda bilder till applikationen

5.1.3 Krav som inte implementerats

- Användaren skall ha ett användarnamn
- Användaren skall ha en avatar
- GPS för att välja motspelare
- Möjlighet att spela upp ljud
- Facebookintegrering

6 Diskussion av arbetsmetod och problem under utveckling

På det hela taget är gruppen nöjd med den applikation som tagits fram, då den uppfyller de nödvändiga krav som sattes på denna och en större del av de lägre prioriterade kraven. I detta kapitel diskuteras den arbetsmetod som användes under projektets gång och hur väl denna har fungerat, samt de större problem som uppstått under implementationen.

6.1 Arbetsmetod

I början av applikationens utveckling var det svårt att ta fram en design som beskrev hur applikationen skulle fungera i detalj. Detta eftersom det för de flesta medlemmar i gruppen var en ny typ av erfarenhet att jobba i ett större projekt och förutspå hur applikationen skulle fungera på olika lager. Därför gjordes en överskådlig skiss om hur vi hade tänkt oss att applikationen skulle fungera och vilka funktioner som var önskvärda att inkludera. Dessa dokumenterades som krav i den tidigare diskuterade kravspecifikationen.

Gruppen delades som tidigare nämnt upp i två mindre grupper - en som skulle hantera nätverksutvecklingen och en som skulle hantera utvecklingen av ritytan. Detta visades sig vara ett bra sätt att dela upp arbetet på eftersom de olika kraven då enkelt kunde fördelas, och det blev enkelt att samarbeta i mindre grupper. När de båda grupperna hade implementerat sina respektive delar, arbetade de gemensamt med att sammanfoga de båda delarna. Nätverksgruppen förklarade hur kommunikation skedde mellan server och klient, och gruppen för ritytan sammanfogade delarna därefter.

Att arbeta på detta sätt var också fördelaktigt eftersom var och en visste vad som skulle göras och hur lång tid man hade på sig att utföra arbetssuppgiften. De problem som uppstod och som gjorde att en tidsgräns inte kunde hållas berodde antagligen på dålig uppskattning av komplexiteten i en arbetsuppgift, vilket i sin tur kom ur bristen på erfarenhet.

6.2 Problem under implementationen

6.2.1 Utvecklingen av nätverkslagret

Den första frågan vi stötte på var vilken typ av protokoll som kommunikationen skulle ske över - UDP eller TCP. Valet föll till en början in på UDP på grund av den hastighet protokollet erbjuder, vilket var viktigt eftersom applikationen var tänkt att fungera i realtid. En bit in i utvecklingen av nätverkslagret

upptäcktes att det dock skulle krävas omfattande arbete med felhantering av förlorade paket, eftersom UDP inte garanterar att paketen når fram till mottagare. Det ansågs viktigare att alla paket kom fram som de skulle än att applikationen fungerade i en högre hastighet, och då TCP kunde erbjuda detta ersattes UDP av detta protokoll.

När det var bestämt att TCP-protokollet skulle användas fortsatte utvecklingen av server och kommunikationsramverk för klienten. Ett problem som påträffades tidigt under serverutvecklingen var hur de rum som användarna skulle delas in i skulle implementeras. Under designfasen hade vi inte överhuvudtaget tänkt över hur de olika grupperna av användare skulle delas upp eller urskiljas - bara hur resultatet skulle fungera. Problemet låg i att skapa dessa rum på ett sätt som innebar att alla meddelanden skickade till ett sådant skulle vidarebefodras till resten av rummets medlemmar. Servern skulle behöva tillåta kommunikation mellan grupper av klienter istället för att den skedde mellan alla som var uppkopplade. Brist på kunskap och erfarenhet ledde till att detta moment tog upp en betydande del av implementeringstiden, men tillslut kom vi fram till en fungerande lösning.

Dataformatet MessagePacket var något som ständigt var tvunget att modifieras, eftersom vi till en början inte visste exakt vad det skulle innehålla och ha för funktioner. Mot slutet av dess utveckling upptäcktes att allokeringen av minne för detta objekt inte fungerade som vi föreställt oss. Istället för att objektet hade varierande storlek beroende på vilket slags meddelande som skickades var storleken alltid konstant. Detta berodde på att alla de olika typerna av datameddelanden låg i samma objekt. Problemet löstes genom att använda MessagePacket som ett basobjekt och sedan låta de olika meddelandetyperna ärva dess egenskaper samt bygga ut den. Då kunde allokeringen av minnet göras dynamiskt beroende på meddelandetypen.

6.2.2 Utvecklingen av Bluetooth

Master/slave-implementationen för Bluetooth innebär att båda enheterna kan initiera uppkopplingen till en session, men den skapar också problem då båda användarna försöker att göra detta samtidigt. En analys av detta problem försvårades av att de olika sockets som används under kommunikationen skapas i trådar som existerar samtidigt. Det går inte att veta exakt när dessa trådar får tillåtelse att arbeta av systemet, och därför kan problemen då visa sig på olika sätt. Det som händer är att två förbindelser upprättas samtidigt när det i själva verket bara skall finnas en enda. Vi implementerade dock felhantering som fångar upp detta och bryter förbindelsen mellan enheterna, men vid det laget har spelaren redan gått vidare in till ritytan. Det skulle vara mer önskvärt att användaren fortfarande skulle befinna sig i

huvudmenyn då denne fick reda på att uppkopplingen misslyckades.

Problemet mildras av att användarna befinner sig väldigt nära varandra då de kopplar ihop sig via Bluetooth, och de kan därför prata ihop sig om vem som skall starta uppkopplingen. Om en klient-server-modell valts i stället hade denna problematik aldrig uppstått, men då hade applikationen inte kunnat bibehålla sin enkelhet. Tidsbrist ledde till en ogenomtänkt lösning - den fungerar, men det finns som sagt vissa brister med den.

6.2.3 Problem vid utvecklingen av ritytan

Under implementationen av ritytan och dess verktyg uppstod huvuddelen av alla problem vid sammankopplingen av dessa. Delarna fungerade i allmänhet som separata delar, men inte efter de slagits ihop med varandra.

Ett av de verktyg som orsakade instabiliteter var zoom-verktyget. Då detta skapade ett nytt koordinatsystem för ritytan var de andra verktygen tvungna att anpassas efter detta. Bland de fel som uppstod fanns felaktig transformering av koordinater mellan koordinatsystemet i smarttelefonens skärm och koordinatsystemet i ritytan, inkorrekt skalning av figurer mellan dessa två system och därmed också fel på de animationer som långsamt skulle måla ut ritrörelser på ritytan.

Problemen som rörde transformering och skalning mellan de två koordinatsystemen låg i att det inte fanns något enhetligt tillvägagångssätt för detta. Vissa verktyg transformerade inte alla koordinater till det nya koordinatsystem som zoom-verktyget skapade. Dessa problem åtgärdades genom införandet av metoder för sådan transformering av koordinater, tillsammans med att direkt åtkomst till ritytan förbjöds. Rittrörelser placerades nu rätt på ritytan och figurer skalades korrekt. Lösningen gjorde att utmålning nu kunde anpassas efter förstoringen av ritytan. De nya metoderna fungerade dock inte för panorering av ritytan - ritrörelserna målades fortfarande ut fel då detta verktyg användes.

Animationerna för ritrörelserna gick igenom flera utvecklingscykler. Vi undersökte länge vilken typ av trådat objekt som skulle kunna schemalägga utmålningar effektivt. På grund av att panoreringen inte fungerade korrekt uppstod problem då animationerna för utmålning skulle köras samtidigt som ritytan panorerades, eftersom de ritrörelser som då ritades ut inte följde med i panoreringen. Implementeringen av den extra logik som behövdes för hantering av panorering samtidigt som animationen för utmålning kördes prioriterades bort till fördel för problem med kommunikationsdelen hos applikationen. Därför togs animationerna bort, och ritrörelserna ritas ut med en gång i stället för att få en illusion av att de långsamt ritas ut på skärmen.

7 Slutsats

Gruppen har uppnått majoriteten av målen som sattes och har skapat en applikation för Android där upp till tre spelare i realtid kan rita tillsammans via en nätverksuppkoppling. En applikation som är enkel att komma igång med och som efter testning resulterat i ett väl utformat användargränssnitt, vilket enligt användartester är enkelt att lära sig och använda.

Under projektets gång har ett antal problem uppstått, men de allra flesta av dessa har kunnat lösas inom rimlig tid. De som inte ansågs hinnas med prioriterades om eller togs bort.

Samtliga gruppmedlemmar har lärt sig mycket om olika arbetsprocesser inom projekt, men också om Android-programmering och programmering i allmänhet. Olika individer har haft varierande tankar och åsikter, vilket gruppen fått anpassa sig efter. Detta har förbättrat gruppens sociala förmåga för att behålla en god arbetsmiljö och samhörighet.

8 Framtida arbete

I det fall att projektet fortsätter i en eller annan form bör en diskussion föras om värdet i vissa av de utelämnade funktionerna i kravspecifikationen. Att integrera stöd för Facebook i applikationen behöver inte nödvändigtvis vara särskilt tidskrävande då det finns färdiga hjälpmedel att hämta från Google och Facebook själva för detta ändamål. Andra idéer, som att visa en avatar och ett användarnamn i ett pågående spel, kan visa sig vara en större utmaning då man förmodligen behöver implementera en databas i mjukvaran för servern. På samma sätt försvåras tanken om GPS stöd i applikationen, då man med största sannolikhet inte kan upptäcka andra enheter såvida man inte redan hade dessa lagrade i sin smarttelefon. Detta skulle inte fungera särskilt väl med kraven på anonymitet. Man skulle då kunna behöva lägga till fler alternativ för flerspelarläge och implementera en lista med tillgängliga enheter om användaren skall kunna spela anonymt eller inte.

Något som även skulle kunna implementeras i framtiden är att man tillsammans kan ladda en bild och måla tillsammans på denna. I dagsläget kan man ladda en bild till ritytan, men de andra användarna kommer inte att se detta. Däremot är ljud - musik och ljudeffekter - något som inte påverkar applikationens funktionalitet eller utformning i särskilt stor grad och skulle kunna läggas till utan större svårigheter.

I skrivande stund kan man inte heller radera filer man sparar på ett enkelt sätt. Filerna är små till storlek men kommer ofrånkomligt att fylla upp diskminnet i enheter som används över en längre tid. En implementering av denna funktion är önskvärd om applikationen skall användas i större skala.

Det som ligger närmast till hands är att städa upp bland de buggar som fortfarande finns, samt att slutföra implementeringen av historikhanteringen. Vissa buggar är bieffekter från vissa funktioner, till exempel animering av linjer - vilket tills vidare har lämnats inaktiverad. Arbetet med att fixa lösa de betydande buggarna är också ett krav om gruppen vill lägga upp applikationen på till exempel Google Play, för att dela med sig av den till andra.

Referenser

(1999) 700000 nya androider per dag. Datormagazin, nr 2, ss. 13.

About (2012). Google [online]. Tillgänglig via: <<http://www.android.com/about/>> [Hämtad 2012-03-26].

ADT Plugin for Eclipse (2012). Google [online]. Tillgänglig via: <<http://developer.android.com/sdk/eclipse-adt.html>> [Hämtad 2012-03-26].

Android API Levels (2012). Google [online]. Tillgänglig via: <<http://developer.android.com/guide/appendix/api-levels.html>> [Hämtad 2012-03-26].

Android BluetoothDevice (2012). Google [online]. Tillgänglig via: <<http://developer.android.com/reference/android/bluetooth/BluetoothDevice.html>> [Hämtad 2012-05-14].

Android BluetoothServerSocket (2012). Google [online]. Tillgänglig via: <<https://developer.android.com/reference/android/bluetooth/BluetoothServerSocket.html>> [Hämtad 2012-05-12].

Android BluetoothSocket (2012). Google [online]. Tillgänglig via: <<https://developer.android.com/reference/android/bluetooth/BluetoothSocket.html>> [Hämtad 2012-05-11].

Android Design Principles (2012). Google [online]. Tillgänglig via: <<http://developer.android.com/design/get-started/principles.html>> [Hämtad 2012-05-04].

Android Data Storage (2012). Google [online]. Tillgänglig via: <<http://developer.android.com/guide/topics/data/data-storage.html>> [Hämtad 2012-05-14].

Android Devices and Displays (2012). Google [online]. Tillgänglig via: <<http://developer.android.com/design/style/devices-displays.html>> [Hämtad 2012-05-04].

Android Fingerprint (2012). Google [online]. Tillgänglig via <<http://developer.android.com/design/style/fingerprint.html>> [Hämtad 2012-05-04].

[//www.developer.android.com/resources/samples/ApiDemos/src/com/example/android/apis/graphics/FingerPaint.html](http://www.developer.android.com/resources/samples/ApiDemos/src/com/example/android/apis/graphics/FingerPaint.html)> [Hämtad 2012-05-14].

Android Principles (2012). Google [online]. Tillgänglig via: <<http://developer.android.com/design/get-started/principles.html>> [Hämtad 2012-05-14].

Android Templates pack, (2012). Google [online]. Tillgänglig via: <http://developer.android.com/guide/practices/ui_guidelines/icon_design.html#templatespack> [Hämtad 2012-05-14].

Android UI Guidelines (2012). Google [online]. Tillgänglig via: <http://developer.android.com/guide/practices/ui_guidelines/index.html/> [Hämtad 2012-05-14].

Android User Interface Guidelines (2012). Google [online]. Tillgänglig via: <http://developer.android.com/guide/practices/ui_guidelines/index.html> [Hämtad 2012-05-04].

Android code style guidelines (2012). Google [online]. Tillgänglig via: <<http://source.android.com/source/code-style.html>> [Hämtad 2012-05-14].

Code Conventions for the Java Programming Language (1999). Oracle [online]. Tillgänglig via: <<http://www.oracle.com/technetwork/java/javase/documentation/codeconvtoc-136057.html>> [Hämtad 2012-02-05].

G. Randers-Pehrson, et. al. (1999), PNG (Portable Network Graphics) Specification, Version 1.2. Scientificcommons [online]. Tillgänglig via: <<http://en.scientificcommons.org/42872441>> [Hämtad 2012-05-09].

IDC Worldwide Quarterly Mobile Phone Tracker. (2011) World-wide Smartphone Market Expected to Grow 55Tillgänglig via: <<http://www.idc.com/getdoc.jsp?containerId=prUS22871611>> [Hämtad 2012-05-29].

Index (2012). Omegle [online]. Tillgänglig via: <<http://www.omegle>.

com> [Hämtad 2012-05-14].

Index (2012). Google [online]. Tillgänglig via: <<http://developer.android.com/>> [Hämtad 2012-03-26].

Müller-Veerse F., et al (2011) Online, Social, and Mobile: The future of the video games industry. Cartagena-capital [online]. Tillgänglig via: <<http://www.cartagena-capital.com/news-and-events/news/256-online-social-and-mobile-the-future-of-the-video-games-industry>> [Hämtad 2012-03-26].

Pettey, C. (2011) Gartner Says Sales of Mobile Devices Grew 5.6 Percent in Third Quarter of 2011; Smartphone Sales Increased 42 Percent. Gartner [online]. Tillgänglig via: <<http://www.gartner.com/it/page.jsp?id=1848514>> [Hämtad 2012-03-26].

Prince, J., Rogers, Y. och Sharp, H. (2011) Interaction Design - beyond human-computer interaction. 3:e uppl. United Kingdom: John Wiley & Sons Ltd.

Ramon, L. T., Stofega W. (2012) Worldwide Smartphone 2012–2016 Forecast and Analysis. IDC. [Online] Tillgänglig via: <<http://www.idc.com/getdoc.jsp?containerId=233553>> [Hämtad 2011-05-29].

Rational Software White Paper (2001). IBM [online]. Tillgänglig via: <http://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf> [Hämtad 2012-04-30].

Subversive (2012). Eclipse [online]. Tillgänglig via: <(http://www.eclipse.org/subversive/)> [Hämtad 2012-05-14].

TCP, Transmission Control Protocol. Network Sorcery [online]. Tillgänglig via: <<http://www.networksorcery.com/enp/protocol/tcp.htm>> [Hämtad 2012-03-26].

The Source Developer Report. Eclipse [online]. Tillgänglig via: <http://www.eclipse.org/org/community_survey/Eclipse_Survey_2011_Report.pdf> [Hämtad 2012-05-29].

UDP, User Datagram Protocol. Network Sorcery [online]. Tillgänglig via: [<http://www.networksorcery.com/enp/protocol/udp.htm>](http://www.networksorcery.com/enp/protocol/udp.htm) [Hämtad 2012-03-26].

A Användarmanual

NetPaint är en ritapplikation som riktar sig till dem som vill ha en social nöjesapplikation, eller som vill rita på sin telefon som tidsfördriv. Den som vill kan rita för sig själv, annars kan applikationen användas för att rita med andra personer.

För kommunikation finns både **Bluetooth** och **WiFi/3G** som alternativ. Väljer man **Bluetooth** kan man koppla ihop sig med någon man själv väljer i sin närhet och måla tillsammans med, och väljer man **WiFi/3G** spelar man över Internet med spelare som är okända.

De verktyg som erbjuds är olika storlek, färg och effekter på sin pensel, zoom, sudd, ångra och gör om. Då man spelar flera spelare över Internet får man ett "rum" tilldelat sig, vilket alla som ritat på samma rityta tillhör. Den information man själv ritat på sin skärm skickas till alla andra spelare i samma "rum" som man själv, så att alla kan se det som ritat. När man väljer att spela över Internet kan max tre spelare rita i ett rum samtidigt. Till skillnad från andra applikationer på marknaden är detta inte bara en ritapplikation, utan tillåter användare att tillsammans bygga upp en bild.

Innehåll

1	Installation	2
1.1	Systemkrav	2
1.2	Installationsanvisningar	2
2	Fönster och menyer	3
2.1	Huvudmeny	3
2.2	Alternativmeny	3
2.3	Sätt på Bluetooth	4
2.4	Anslutning via Bluetooth	5
2.5	Rityta	6
2.6	Penseldialog	8
2.7	Suddgummi	9
2.8	Menyknapp	9
3	Felsökning	11

1 Installation

1.1 Systemkrav

- Tillgång till 3G/WiFi eller Bluetooth (för flerspelarläge)
- Android 2.1 eller senare installerat
- Skärmupplösning på minst 240 x 320 pixlar
- Minst 1 MB tillgängligt utrymme på telefonen

1.2 Installationsanvisningar

1. Se till att din telefon tillåter installation av applikationer från okända källor.
2. Skriv in URL:en `http://code.google.com/p/project-bengt/downloads/list` i din telefons webbläsare.
3. Klicka på filen med namnet "Drawing.apk".
4. Ladda ner "Drawing.apk".
5. Då nedladdningen är färdig, öppna telefonens statusfält, och klicka på namnet "Drawing".
6. Välj att installera applikationen.
7. Då installationen är klar visas en dialogruta där du kan välja att starta applikationen direkt eller gå vidare.
8. NetPaint ligger nu redo att använda under Applikationer på din telefon.

2 Fönster och menyer

2.1 Huvudmeny

Huvudmenyn är det första man ser när man startar applikationen. Det är härifrån man kan gå vidare till olika spelarlägen och inställningar (Se figur 1).



Figur 1 - Huvudmeny

1. Enspelarläge

Detta alternativ tar användaren vidare till en rityta där denne ensam kan rita utan att någon uppkoppling görs.

2. Flerspelarläge

Om användaren valt att kommunikation skall ske över Internet kommer denne genom att välja detta alternativ in i ett rum där det eventuellt finns andra spelare inne. Om Bluetooth har valt kommer ett fönster upp där denne får välja vilken enhet man vill ansluta till (se figur 3).

3. Alternativ

Alternativ dirigerar användaren till en sida där inställningar kan väljas.

4. Avsluta

Telefonens tillbaka-knapp används för att stänga av applikationen, och alternativet textitJa väljs i den dialogruta som kommer upp när man trycker på den.

2.2 Alternativmeny

Alternativmenyn finns för att spelaren skall kunna välja olika inställningar för applikationen. I denna version kan alternativ för hur kommunikation skall ske väljas.



Figur 2 - Alternativmeny

1. **3G/WiFi**

Då denna knapp trycks ner sker kommunikation över Internet då man stänger ner fönstret. Då alternativet är valt indikeras detta av att cirkeln i mitten markeras grön.

2. **Bluetooth**

Då denna knapp trycks ner sker kommunikation via Bluetooth då man stänger ner fönstret.

3. **Gå tillbaka**

Tillbaka-knappen på telefonen tar användaren tillbaka till huvudmenyn.

2.3 Sätt på Bluetooth

Då användaren för första gången väljer att kommunicera via Bluetooth i Alternativ-fönstret dyker en dialogruta upp (se figur 3). Här skall användaren tillåta eller neka att Bluetooth på telefonen sätts på och att enheten går att upptäcka under en viss tid.



Figur 3 - Sätt på Bluetooth

1. Tillåt Bluetooth

Acceptera att Bluetooth sätts på och att den egna enheten går att upptäckas av andra enheter under en viss tid.

2. Neka Bluetooth

Om användaren inte godkänner att Bluetooth sätts på stängs dialogrutan ner, men det fungerar sedan inte att välja *Flera spelare* i huvudmenyn. Istället får man välja att spela över Internet eller som en spelare.

3. Neka Bluetooth

Då telefonens tillbaka-knapp trycks ner stängs dialogrutan och Bluetooth blir inte vald. Se 2. *Neka Bluetooth*.

2.4 Anslutning via Bluetooth

Då Bluetooth är inställt som kommunikation och användaren väljer *Flera spelare* i huvudmenyn kommer en dialogruta upp på skärmen (se figur 4).



Figur 4 - Anslutning via Bluetooth

1. Val av spelare

Då användaren trycker på namnet på en enhet i denna lista, startas ett nytt spel där de två kan måla tillsammans. Endast en av spelarna får välja enhet att rita med.

2. Val av spelare

Då användaren trycker på namnet på en enhet i denna lista kommer en förfrågan skickas till den andra enheten om denna vill koppla ihop sin enhet med användarens. Användaren får också en förfrågan, och svarar de båda ja startas en ny rityta som de båda kan måla på. Väljer någon av dem nej startas inte ett nytt spel, och användaren får välja en annan enhet i listan eller skicka en ny förfrågan. Endast en av spelarna får välja en enhet att kopplas ihop med.

3. Ihopkopplade enheter

Denna lista innehåller enheter man tidigare godkänt att de kopplas ihop med den egna enheten. Någon av dessa kan väljas för att direkt starta nytt spel om den andra enheten finns tillgänglig.

4. Tillgängliga enheter

Då dialogrutan för valbara enheter öppnas finns en knapp "Sök efter enheter". Då denna trycks ner söker programmet efter andra enheter som är i närheten och listar dem här. När en av dessa väljs måste båda enheter godkänna ihopkopplingen för att kunna rita med varandra.

2.5 Rityta

Då användaren kommer in i antingen flerspelarläge eller enspelarläge visas ritytan med tillhörande verktygsfält (se figur 5).



Figur 5 - Rityta

1. Rityta

Då användaren drar fingret över ritytan målas streck med den valda penseln. I flerspelarläge visas här även de andra spelarnas streck.

2. Pensel

Då användaren trycker på denna knapp kan han/hon måla på ritytan. För att ändra pensel hålls denna knapp in tills en dialogruta dyker upp (se figur 6). Detta läge är valt då en svart ram markerar knappen.

3. Sudd

Användaren trycker ner denna knapp för att välja sudd, som sedan kan användas på ritytan. Hålls denna knapp inne en längre stund dyker en dialogruta upp (se figur 7). Detta läge är valt då en svart ram markerar knappen.

4. Zoom

Efter att användaren tryckt ner denna knapp kan han/hon sätta fingret på en plats på ritytan som hon/han vill zooma ut eller in på. När telefonen ger ifrån sig en vibration kan användaren föra fingret uppåt för att zooma in eller nedåt för att zooma ut. Om knappen hålls inne en stund kommer bilden att återgå till normal storlek. Detta läge är valt då en svart ram markerar knappen.

5. Ångra

Denna knapp kan tryckas på för att ångra det senast målade strecket. Det försvinner då från användarens rityta, och i flerspelarläge även från de andra användarnas ritytor.

6. Gör om

Användaren kan trycka på denna knapp efter att ha tryck på 5. *Ångra*, och då ritas det borttagna strecket ut på ritytan igen. (Lägg till när man inte kan göra detta). I flerspelarläge dyker strecket även upp på de andra spelarnas ritytor igen.

7. Meny

Då menyknappen på telefonen trycks ner kommer en meny fram, där man i enspelarläge kan skapa en ny bild, ladda och spara bild. I flerspelarläge kan man gå ur spelet, ladda och spara bild.

8. Avsluta

För att avsluta trycks telefonens bakåt-knapp ner, och *Ja* väljs i dialogrutan som då dyker upp.

2.6 Penseldialog

Då användaren håller inne 2. *Pensel* (se figur 5) dyker en penseldialog upp (se figur 6). Denna är till för att sätta önskade egenskaper för sin pensel.



Figur 6 - Penseldialog

1. Val av färg

Användaren kan välja en färg i färgcirkeln genom att sätta ner sitt finger på den och lyfta upp det då det ligger över den färg man vill ha. Vald färg visas då i den inre cirkeln.

2. Vald färg

Då dialogen öppnas visas den färg som penseln för tillfället har. Då användaren ändrar färg genom att föra fingret över färgcirkeln sätts denna inre cirkel till vald färg. Färgen sparas bara då 4. *Ok* trycks ner efteråt.

3. Svart/vit färg

För att välja färgerna svart eller vitt klickar användaren på respektive ruta. Dessa två färger finns inte med i färgcirkeln, men valet av färg sker på samma sätt.

4. OK

OK-knappen trycks ner då användaren vill spara de ändringar han/hon nyss gjort på penselns effekter.

5. Effekter

Det finns fyra olika effekter användaren kan sätta på sin pensel, men bara en i taget kan väljas. Vald effekt markeras med en grön cirkel.

6. Penselstorlek

Då dialogen precis öppnats visas här aktuell penselstorlek. Då storleken ändras i penseldialogen kommer även detta värde att uppdateras.

7. Val av storlek

Genom att föra denna slider åt vänster får penseln en mindre storlek, och åt höger en större.

8. Avbryt

Om användaren trycker på telefonens tillbaka-knapp stängs dialogfönstret och inga nya ändringar på penseln görs.

2.7 Suddgummi

Då 3. *Sudd* (se figur 5) trycks ner en stund dyker en dialog upp (se figur 7). Denna är till för att användaren skall kunna bestämma storlek på suddgummit.



Figur 7 - Suddgummi

1. OK

Då användaren trycker på denna knapp sparas den storlek på suddgummit som valts.

2. Suddstorlek

Då användaren precis öppnat dialogen visas här den aktuella suddstorleken. När ett nytt värde väljs uppdateras denna.

3. Val av storlek

Denna slider används för att sätta en ny storlek på suddgummit. Då den dras åt höger blir den större, och åt vänster mindre.

4. Avbryt

Då telefonens tillbaka-knapp trycks ner stängs dialogen, och inga värden uppdateras. Suddstorleken är lika stor som innan fönstret öppnades.

2.8 Menyknapp

Då telefonens menyknapp 7. *Meny* (se figur 5) trycks ner kommer en meny med tre alternativ fram.



Figur 8 - Menyknapp

1. Nytt spel

I enspelarläge får användaren en förfrågan om denne vill rensa ritytan. Väljs *Ja* så kommer hela ritytan att tömmas. I flerspelarläge i denna version kommer användaren till huvudmenyn och kan åter igen välja *Flera spelare* eller *En spelare*.

2. Ladda bild

Då detta alternativ väljs i menyn kommer en lista fram på alla bilder man tidigare målat i NetPaint. Klicka på en av dessa för att sätta som bakgrund på ritytan.

3. Spara bild Då detta alternativ väljs kommer en dialogruta upp, där användaren kan döpa bilden och sedan trycka på *Spara*. Bilden kan sedan laddas in på ritytan igen genom *2. Ladda bild*.

4. Stäng meny

Då användaren trycker på menyknappen kommer menyn att stängas.

5. Stäng meny

För att stänga menyn kan även telefonens tillbaka-knapp användas. Då återvänder spelaren till ritytan.

3 Felsökning

- *Min bild återställs inte korrekt när jag håller inne zoom-knappen*
 1. Håll inne zoom-knappen
 2. Upprepa om felet kvarstår
- *Vi har valt Bluetooth och en ny rityta visas på båda enheterna. Men när vi målar kommer ingenting fram till motspelaren*
 1. Stäng av och låt båda klicka på *Flera spelare*
 2. Endast en av er klickar på den andras enhet för att starta ett nytt spel
 3. Vänta tills ritytan dyker upp igen
- *Strecken kommer inte fram*
 1. Spara, om du inte vill förlora din bild
 2. Starta om spelet
 3. Upprepas detta fel ofta och snabbt när spelet har startat - försök undvika att måla streck utanför den vita ritytan
- *När jag suddar blir det inte helt vitt*
 1. Välj en pensel med vit färg i stället
 2. Starta om spelet i fall felet kvarstår
- *Jag kan inte zooma in/ut*
 1. Se till att du håller fingret på ritytan där du vill zooma tills du känner en svag vibration från telefonen
 2. Håll kvar fingret och dra det uppåt/nedåt

B Instruktioner för användartester

B.1 Användartester I: Enspelarläge & Huvudmeny - Instruktioner

B.1.1 Observation

1. Informera testaren om att
 - Testaren skall utföra åtta uppgifter
 - Uppgifterna skall utföras i applikation på en Android-telefon
 - Testaren skall utföra uppgifterna på ett effektivt sätt
 - Testaren skall använda "Think-Aloud" tekniken – och förklara vad denna innebär (att användaren skall säga allt den tänker och varför personen gör sina val)
2. Presentera följande uppgifter – en i taget – för testaren
 - (a) Starta applikationen och stäng av den igen
 - (b) Sätt igång applikationen igen och starta nytt spel för en spelare
 - (c) Rita ett blått streck med effekten "Skugga"
 - (d) Rita ett streck med penselstorlek 30
 - (e) Rita ett streck, ångra, gör om
 - (f) Spara bilden som "apa"
 - (g) Ladda bilden "apa"
 - (h) Gå tillbaka till huvudmenyn

En uppgift är färdig då testaren antingen gett upp eller utfört angivna instruktioner. Ingen hjälp får ges till testaren. Under observationen skall anteckningar föras över hur personen går till väga för att påbörja uppgiften, vad personen säger genom "Think-Aloud"-tekniken, om det går att avgöra om personen uttrycker tankar eller känslor genom kroppsspråk samt om personen verkar ha problem med en viss uppgift eller funktion hos applikationen.

B.1.2 Intervjufrågor

Intervjun skall hållas direkt efter att observationen ägt rum. Efter att en fråga har ställts kan ytterligare frågor, som inte finns med i instruktionerna,

ställas. Detta fortsätter tills ingen mer information relaterad till just denna fråga tycks fås fram.

Exempel på frågor är:

1. Vad är ditt första intryck av applikationen?
2. Vad tyckte du om strukturen på huvudmenyn?
3. Vad tyckte du om utseendet på ritytan i enspelarläge?
4. Var menyn man kunde få upp längst ner intuitiv att använda och bra placerad?
5. Var tyckte du om knappen längst upp i vänstra hörnet?
6. Förstod du hur du skulle spara och ladda bilder?
7. Tycker du att det var enkelt att byta färg, effekt och storlek på penseln?
8. Verkade applikationen rolig att använda i enspelarläge?
9. Var det någon funktion som var speciellt svår att använda?
10. Vad är ditt sammanfattande intryck av strukturen på applikationen?

B.1.3 Renskrivning

Anteckningar och svar på intervjufrågor bör renskrivas så snart efter intervjun och observationen som möjligt.

B.2 Användartester II: Enspelarläge & Huvudmeny - Instruktioner

B.2.1 Observation

1. Informera testaren om att
 - Detta test är en fortsättning på det test som tidigare gjordes
 - Testaren skall utföra åtta uppgifter
 - Testaren skall utföra uppgifterna på ett effektivt sätt
 - Testaren skall använda "Think-Aloud" tekniken igen
2. Presentera följande uppgifter – en i taget – för testaren
 - (a) Starta applikationen och stäng av den igen
 - (b) Sätt igång applikationen igen och starta nytt spel för en spelare
 - (c) Rita ett blått streck med effekten "Skugga"
 - (d) Rita ett streck med penselstorlek 30
 - (e) Rita ett streck, ångra, gör om
 - (f) Spara bilden som "apa"
 - (g) Skapa en ny bild och ladda sedan bilden "apa"
 - (h) Gå tillbaka till huvudmenyn

B.2.2 Intervjufrågor

1. Vad är ditt första intryck av applikationen?
2. Vad tyckte du om utseendet på ritytan i enspelarläge?
3. Vad tycker du om verktygsfältet som är placerat längst ner på ritytan?
4. Var alla ikoner lätta att förstå?
5. Förstod du hur du skulle spara och ladda bilder?
6. Var det någon funktion som var speciellt svår att använda?
7. Var det någon funktion du tyckte saknades?
8. Vad är ditt sammanfattande intryck av designen på applikationen?
9. Tycker du att applikationen har förbättrats sedan förra testet?

B.3 Användartester III: Enspelarläge, flerspelarläge & huvudmeny - Instruktioner

B.3.1 Observation

1. Informera testaren om att
 - Detta test är en fortsättning på det test som tidigare gjordes
 - Testaren skall utföra 18 uppgifter
 - Testaren skall utföra uppgifterna på ett effektivt sätt
 - Testaren skall använda "Think-Aloud" tekniken igen
 - Den som utför testerna använder en annan telefon då flerspelarläget skall testas, så att testaren får någon att måla tillsammans med
2. Presentera följande uppgifter – en i taget – för testaren
 - (a) Starta applikationen
 - (b) Starta nytt spel för en spelare
 - (c) Rita ett blått streck med effekten "Skugga"
 - (d) Rita ett streck med penselstorlek 30
 - (e) Rita ett streck, ångra, gör om
 - (f) Spara bilden som "apa"
 - (g) Skapa en ny bild och ladda sedan bilden "apa"
 - (h) Gå tillbaka till huvudmenyn
 - (i) Välj alternativ, och markera att du vill spela över Internet
 - (j) Starta ett nytt spel i flerspelarläge
 - (k) Rita ett grönt streck med effekten "Relief"
 - (l) Vänta på att din motspelare skall måla något, och se hur detta också dyker upp på din egen bild
 - (m) Gå tillbaka till huvudmenyn
 - (n) Välj att spela över Bluetooth
 - (o) Starta ett spel med din motspelare som befinner sig i närheten
 - (p) Rita ett grönt streck med effekten "Relief"
 - (q) Vänta på att din motspelare skall måla något, och se hur detta också dyker upp på din egen bild
 - (r) Gå tillbaka till huvudmenyn

B.3.2 Intervjufrågor

1. Vad är ditt första intryck av applikationen?
2. Vad tyckte du om utseendet på ritytan i enspelarläge?
3. Vad tycker du om den nya designen verktygsfältet längst ner på ritytan?
4. Var alla ikoner lätta att förstå?
5. Förstod du hur du skulle spara och ladda bilder?
6. Var det någon funktion som var speciellt svår att använda?
7. Vad tycker du om de tipsrutor som dyker upp när du trycker på en knapp första gången?
8. Vad tycker du om designen på menyn för alternativ?
9. Var det lätt att starta ett nytt spel över Internet?
10. Var det lätt att starta ett nytt spel över Bluetooth?
11. Var flerspelarläget intuitivt att använda?
12. Vad är ditt sammanfattande intryck av designen på applikationen?
13. Tycker du att applikationen har förbättrats sedan förra testet?

C Kravspecifikation

C.1 Prioritet 1

Följande krav gavs prioritet ett:

- C.4.1 - Applikationen skall stödja Android 2.1 och uppåt
- C.5.1 - Kommunikation genom 3G/WiFi
- C.5.2 - Kommunikation genom TCP
- C.5.4 - Kommunikation mellan två enheter
- C.5.6 - Data sänds under beröring av ritytan
- C.6.1 - Touchfunktion för ritning och menynavigering
- C.6.2 - Notifieringar vid fel
- C.6.3 - Lättförståeligt användargränssnitt
- C.7.1 - Applikationen startar
- C.7.2 - Applikationen avslutas
- C.7.3 - Möjlighet att spela ensam
- C.7.4 - Möjlighet att spela två spelare
- C.7.6 - Val av färg
- C.7.9 - Avslutning av ritsession
- C.7.21 - Spel i realtid
- C.9.1 - Svenskt språk
- C.9.2 - Lätt att lägga till fler språk
- C.10.1 - Vældokumenterad kod
- C.10.2 - Effektiv nätverkskod

C.2 Prioritet 2

Följande krav gavs prioritet två:

- C.5.3 - Kommunikation via Bluetooth
- C.5.5 - Kommunikation mellan två eller tre enheter
- C.6.4 - Egendesignat användargränssnitt
- C.7.5 - Möjlighet att spela två eller tre spelare
- C.7.7 - Val av effekt
- C.7.8 - Val av storlek
- C.7.10 - Användaren skall ha ett användarnamn
- C.7.11 - Användaren skall ha en avatar
- C.7.12 - Användarna skall delas in i rum
- C.7.13 - Slumpmässig tilldelning av rum
- C.7.14 - Möjlighet att zooma
- C.7.15 - Panorering av bild
- C.7.16 - Möjlighet att sudda
- C.7.17 - Möjlighet att ångra det man ritat
- C.7.18 - Möjlighet att göra om det man ångrat
- C.8.1 - Spara bilder från applikationen
- C.8.2 - Ladda bilder till applikationen
- C.8.3 - Bilder skall sparas i formatet .png

C.3 Prioritet 3

Följande krav gavs prioritet tre:

- C.5.7 - GPS för att välja motspelare
- C.7.19 - Möjlighet att få ljud
- C.7.20 - Facebook integrering

C.4 Plattform

C.4.1 Applikationen skall stödja Android 2.1 och uppåt

ID	C.4.1
Användningsfall/Scenario	Applikationen skall stödja Android 2.1 och senare versioner
Utlösare	-
Förhandsvillkor	Applikationen är kodad för Android 2.1
Standardväg	-
Undantagsväg	-
Slutligt tillstånd	Applikationen stödjer Android 2.1 och senare versioner.

C.5 Nätverk

C.5.1 Kommunikation genom 3G/WiFi

ID	C.5.1
Användningsfall/Scenario	Kommunikation genom 3G/WiFi
Utlösare	Användaren vill spela i flerspelarläge
Förhandsvillkor	Användaren har valt 3G/WiFi för kommunikation
Standardväg	En enhet väljer att skicka information till en annan enhet.
Undantagsväg	3G/WiFi finns inte tillgängligt.
Slutligt tillstånd	Användaren kommunicerar med andra enheter genom 3G/WiFi.

C.5.2 Kommunikation genom TCP

ID	C.5.2
Användningsfall/Scenario	Kommunikation genom TCP
Utlösare	Enheter vill kommunicera med varandra
Förhandsvillkor	Enheten skall ha nätverkstillgång

Standardväg

En enhet väljer att skicka information till en annan enhet.

Undantagsväg

-

Slutligt tillstånd

Information mellan kommunicerande enheter skickas med hjälp av TCP-paket.

C.5.3 Kommunikation via Bluetooth

ID	C.5.3
Användningsfall/Scenario	Kommunikation via Bluetooth
Utlösare	Användaren vill spela i flerspelarläge
Förhandsvillkor	Enheten skall ha nätverkstillgång

Standardväg

En enhet väljer att skicka information till en annan enhet.

Undantagsväg

Inget stöd för Bluetooth. Den andra enheten befinner sig på för långt avstånd ifrån den egna.

Slutligt tillstånd

Användaren kommunicerar med andra enheter genom Bluetooth.

C.5.4 Kommunikation mellan två enheter

ID	C.5.4
Användningsfall/Scenario	Två enheter kan kommunicera med varandra
Utlösare	En användare vill kommunicera med en annan användare
Förhandsvillkor	Användaren har valt att kommunicera via 3G/WiFi (Req C.5.1) Användaren har valt att kommunicera via Bluetooth (Req C.5.3)

Standardväg

En enhet försöker kommunicera med en annan enhet.

Undantagsväg

-

Slutligt tillstånd

Två enheter kommunicerar med varandra.

C.5.5 Kommunikation mellan två eller tre enheter

ID	C.5.5
Användningsfall/Scenario	Två eller tre enheter kan kommunicera med varandra
Utlösare	En användare vill starta flerspelarläge
Förhandsvillkor	Användaren har valt att kommunicera via 3G/WiFi (Req C.5.1)

Standardväg

En enhet försöker kommunicera med andra enheter.

Undantagsväg

3G/WiFi finns inte tillgängligt.

Slutligt tillstånd

Två eller flera enheter kommunicerar med varandra.

C.5.6 Data sänds under beröring av ritytan

ID	C.5.6
Användningsfall/Scenario	Data skall sändas då användaren rör vid ritytan
Utlösare	Användaren sätter fingret på ritytan
Förhandsvillkor	Kommunikation mellan två eller tre enheter (Req C.5.5)

Standardväg

Användaren sätter fingret på ritytan.

Undantagsväg

-

Slutligt tillstånd

Data skickas till motspelare när användaren rör ritytan.

C.5.7 GPS för att välja motspelare

ID	C.5.7
Användningsfall/Scenario	GPS skall kunna användas för att visa och välja motspelare i flerspelarläge
Utlösare	Användaren väljer att leta efter motspelare genom GPS
Förhandsvillkor	Användaren har valt att kommunicera via 3G/WiFi (Req C.5.1)

Standardväg

Användaren väljer att leta efter motspelare genom GPS som visas på en karta.

Undantagsväg

-

Slutligt tillstånd

Användaren kan se och välja motspelare på en karta genom GPS, och sedan rita med dessa.

C.6 Gränssnitt och interaktion

C.6.1 Touchfunktion för ritning och menynavigering

ID	C.6.1
Användningsfall/Scenario	Ritning och menynavigering skall ske genom touchfunktion hos plattformen
Utlösare	Användaren rör vid skärmen
Förhandsvillkor	Användaren skall rita eller navigera i menyerna
Standardväg	Användaren försöker navigera eller rita med hjälp av sina fingrar.
Undantagsväg	-
Slutligt tillstånd	Användaren kan rita och navigera i menyerna genom enhetens touchfunktion.

C.6.2 Notifieringar vid fel

ID	C.6.2
Användningsfall/Scenario	Användaren skall notifieras vid fel hos applikationen eller kommunikationen
Utlösare	Ett fel uppstår hos applikationen eller under kommunikationen med andra enheter
Förhandsvillkor	-
Standardväg	Ett fel uppstår då applikationen är igång
Undantagsväg	-
Slutligt tillstånd	Applikationen notifierar användaren om att ett fel har uppstått

C.6.3 Lättförståeligt användargränssnitt

ID	C.6.3
Krav	Användargränssnittet skall vara lätt för användaren att förstå och interagera med

C.6.4 Egendesignat användargränssnitt

ID	C.6.4
Krav	Användargränssnittet skall vara egendesignad, med applikationens syfte i åtanke

C.7 Spel

C.7.1 Applikationen startar

ID	C.7.1
Användningsfall/Scenario	Användaren skall kunna starta applikationen
Utlösare	Användaren vill starta applikationen
Förhandsvillkor	Applikationen är inte igång
Standardväg	
Användaren startar applikationen.	
Undantagsväg	
-	
Slutligt tillstånd	
Applikationen startas.	

C.7.2 Applikationen avslutas

ID	C.7.2
Användningsfall/Scenario	Användaren skall kunna stänga av applikationen
Utlösare	Användaren vill stänga av applikationen
Förhandsvillkor	Användaren befinner sig i rätt meny i applikationen för att kunna avsluta
Standardväg	
Användaren väljer att avsluta	
Undantagsväg	
-	
Slutligt tillstånd	
Applikationen avslutas	

C.7.3 Möjlighet att spela ensam

ID	C.7.3
Användningsfall/Scenario	En användare skall kunna välja att rita ensam
Utlösare	Användaren väljer att spela ensam
Förhandsvillkor	Användaren vill spela ensam
Standardväg	
Användaren väljer att spela ensam.	
Undantagsväg	
-	
Slutligt tillstånd	
Användaren kan måla ensam utan att kommunikation sker.	

C.7.4 Möjlighet att spela två spelare

ID	C.7.4
Användningsfall/Scenario	En användare skall kunna välja att spela med en annan användare
Utlösare	Användaren väljer att spela flera spelare
Förhandsvillkor	Användaren vill spela med en annan användare

Standardväg

Användaren väljer att spela flera spelare

Undantagsväg

-

Slutligt tillstånd

Användaren ritar tillsammans med en annan användare - antingen via 3G/WiFi eller Bluetooth.

C.7.5 Möjlighet att spela två eller tre spelare

ID	C.7.5
Användningsfall/Scenario	En användare skall kunna välja att spela tillsammans med två eller tre spelare
Utlösare	Användaren väljer att spela flera spelare
Förhandsvillkor	Användaren vill spela med andra användare

Standardväg

Användaren väljer att spela flera spelare

Undantagsväg

Alla rum är fulla och användaren måste gå in i ett nytt rum.

Slutligt tillstånd

Användaren ritar tillsammans med en eller två spelare

C.7.6 Val av färg

ID	C.7.6
Användningsfall/Scenario	Användaren skall ändra färg att måla med till valfri RGB-färg
Utlösare	Användaren vill måla med en annan färg
Förhandsvillkor	Användaren håller på och målar

Standardväg

Användaren väljer att måla med en annan färg.

Undantagsväg

-

Slutligt tillstånd

Användaren målar med den önskade RGB-färgen.

C.7.7 Val av effekt

ID	C.7.7
Användningsfall/Scenario	Användaren skall kunna byta effekt på penseln
Utlösare	Användaren vill måla med en annan effekt
Förhandsvillkor	Användaren håller på och målar

Standardväg

Användaren väljer att måla med en annan effekt.

Undantagsväg

-

Slutligt tillstånd

Användaren målar med den önskade effekten.

C.7.8 Val av storlek

ID	C.7.8
Användningsfall/Scenario	Användaren skall kunna byta effekt på penseln
Utlösare	Användaren vill måla med en annan effekt
Förhandsvillkor	Användaren håller på och målar

Standardväg

Användaren väljer att måla med en annan effekt.

Undantagsväg

-

Slutligt tillstånd

Användaren målar med den önskade effekten.

C.7.9 Avslutning av ritsession

ID	C.7.9
Användningsfall/Scenario	Användaren skall kunna avsluta en ritsession
Utlösare	Användaren önskar avsluta en ritsession
Förhandsvillkor	Användaren håller på att rita

Standardväg

Användaren väljer att avsluta en ritsession.

Undantagsväg

Det finns ingen aktiv ritsession

Slutligt tillstånd

Ritsessionen avslutas

C.7.10 Användaren skall ha ett användarnamn

ID	C.7.10
Användningsfall/Scenario	Användaren skall ha ett användarnamn
Utlösare	En användare skapas i applikationen
Förhandsvillkor	-

Standardväg

En användare skapas vid första uppstart av applikationen.

Undantagsväg

En användare är redan skapad i applikationen.

Slutligt tillstånd

Användaren får välja ett användarnamn som denne kommer att identifieras med

C.7.11 Användaren skall ha en avatar

ID	C.7.11
Användningsfall/Scenario	Användaren skall ha en avatar
Utlösare	En användare skapas i applikationen
Förhandsvillkor	-

Standardväg

En användare skapas vid första uppstart av applikationen.

Undantagsväg

En användare är redan skapad i applikationen.

Slutligt tillstånd

Användaren får välja en avatar som denne kommer att identifieras med

C.7.12 Användarna skall delas in i rum

ID	C.7.12
Användningsfall/Scenario	Användarna skall delas in i rum
Utlösare	En användare väljer att spela i flerspelarläge
Förhandsvillkor	-

Standardväg

Användaren blir tilldelad ett rum, som max kan innehålla tre spelare vid kommunikation med 3G/WiFi och max två vid Bluetooth

Undantagsväg

-

Slutligt tillstånd

Användaren kommer in i ett rum där denne kan måla

C.7.13 Slumpmässig tilldelning av rum

ID	C.7.13
Användningsfall/Scenario	Användarna skall slumpmässigt placeras i rum med plats
Utlösare	En användare väljer att spela i flerspelarläge
Förhandsvillkor	-

Standardväg

Användaren blir slumpmässigt tilldelat ett utav de rum där det finns plats för minst en till spelare

Undantagsväg

Det finns inga lediga rum - vilket leder till att användaren tilldelas ett tomt rum

Slutligt tillstånd

Användaren kommer in i ett rum där denne kan måla

C.7.14 Möjlighet att zooma

ID	C.7.14
Användningsfall/Scenario	Användaren skall kunna zooma in och ut på sin bild
Utlösare	Användaren väljer att zooma
Förhandsvillkor	Användaren håller på att måla

Standardväg

Användaren väljer att zooma in eller ut

Undantagsväg

Användaren har redan zoomat ut/in så långt det går

Slutligt tillstånd

Bilden blir in- eller utzoomad

C.7.15 Panorering av bild

ID	C.7.15
Användningsfall/Scenario	Användaren skall kunna panorera för att se en annan del av bilden
Utlösare	Användaren väljer att panorera
Förhandsvillkor	Användaren har zoomat in så inte hela bilden visas på skärmen

Standardväg

Användaren väljer att panorera bilden

Undantagsväg

-

Slutligt tillstånd

Den del av bilden som användaren panorerat till kommer att synas på skärmen

C.7.16 Möjlighet att sudda

ID	C.7.16
Användningsfall/Scenario	Användaren skall kunna sudda på sin bild
Utlösare	Användaren väljer att sudda
Förhandsvillkor	Användaren har tidigare målat något

Standardväg

Användaren väljer att sudda på bilden

Undantagsväg

-

Slutligt tillstånd

Det som användaren suddar färgas vitt

C.7.17 Möjlighet att ångra det man ritat

ID	C.7.17
Användningsfall/Scenario	Användaren skall kunna ångra något som denne ritat tidigare
Utlösare	Användaren väljer att ångra det som denne senast ritade
Förhandsvillkor	Användaren har ritat något

Standardväg

Användaren väljer att ångra det senast ritade

Undantagsväg

-

Slutligt tillstånd

Det användaren senast ritade på bilden tas bort

C.7.18 Möjlighet att göra om det man ångrat

ID	C.7.18
Användningsfall/Scenario	Användaren skall kunna göra om något denne målat och sedan ångrat
Utlösare	Användaren väljer att göra om något den tidigare ritat
Förhandsvillkor	Användaren har ångrat något denne har ritat

Standardväg

Användaren väljer att göra om något denne tidigare ångrat

Undantagsväg

-

Slutligt tillstånd

Det som användaren senast ångrade ritas åter ut på bilden

C.7.19 Möjlighet att få ljud

ID	C.7.19
Användningsfall/Scenario	Användaren skall kunna välja att sätta på ljud
Utlösare	Användaren väljer att sätta på ljud på applikationen
Förhandsvillkor	Applikationen körs

Standardväg

Användaren väljer att sätta på ljud för applikationen

Undantagsväg

-

Slutligt tillstånd

Bakgrundsmusik spelas upp och ljudeffekter hörs då användaren interagerar med applikationen

C.7.20 Facebook integrering

ID	C.7.20
Användningsfall/Scenario	Användaren skall kunna lägga upp sina bilder på Facebook
Utlösare	Användaren väljer att lägga upp en bild på Facebook
Förhandsvillkor	Användaren har målat en bild - ensam eller tillsammans med andra

Standardväg

Användaren väljer att dela ritad bild på Facebook

Undantagsväg

Användaren har inget Facebook-konto

Slutligt tillstånd

Användarens bild läggs upp på Facebook så att andra kan se den

C.7.21 Spel i realtid

ID	C.7.21
Krav	Användaren skall kunna rita med andra spelare i realtid

C.8 Filhantering

C.8.1 Spara bilder från applikationen

ID	C.8.1
Användningsfall/Scenario	Användaren skall kunna spara bilder från applikationen
Utlösare	Användaren vill spara en bild från applikationen
Förhandsvillkor	Användaren håller på att rita

Standardväg

Användaren väljer att spara en bild från applikationen.

Undantagsväg

-

Slutligt tillstånd

Användaren sparar den önskade bilden lokalt i enheten.

C.8.2 Ladda bilder till applikationen

ID	C.8.2
Användningsfall/Scenario	Användaren skall kunna ladda bilder från enheten till applikationen
Utlösare	Användaren vill ladda en bild till applikationen
Förhandsvillkor	Användaren håller på att rita

Standardväg

Användaren väljer att ladda en bild till applikationen.

Undantagsväg

Det finns inga bilder på enheten att ladda.

Slutligt tillstånd

Användaren laddar en bild som denne sedan kan måla på.

C.8.3 Bilder skall sparas i formatet .png

ID	C.8.3
Användningsfall/Scenario	Bilder från applikationen skall sparas i formatet .png
Utlösare	Användaren vill spara en bild
Förhandsvillkor	Användaren håller på att rita Spara bilder från applikationen(Req C.8.1)

Standardväg

Användaren sparar en bild från applikationen

Undantagsväg

-

Slutligt tillstånd

Bilden sparas lokalt i enheten, i formatet .png.

C.9 Språk

C.9.1 Svenskt språk

ID	C.9.1
Krav	Applikationen använder sig av det svenska språket

C.9.2 Lätt att lägga till fler språk

ID	C.9.2
Krav	Koden för applikationen skall vara skriven på ett sätt som underlättar för att senare lägga till fler språk än svenska

C.10 Kod

C.10.1 Vældokumenterad kod

ID

C.10.1

Krav

Koden för applikationen skall vara vældokumenterad, med kommentarer för alla metoder och förklaringar till klasser

C.10.2 Effektiv nätverkskod

ID

C.10.2

Krav

Applikationen skall ha effektiv nätverkskod för att få snabb respons och för att den skall fungera bra i realtid.

D MessagePacket

Ett dataformat för kommunikationen mellan inblandade parter fick tas fram. Den information som skickas mellan dessa kapslas in i ett så kallat "MessagePacket". Detta meddelande analyseras vid mottagande och programmet agerar utefter diverse parametrar som skickats med. Det är av största vikt att dessa paket som skickas innehåller så lite information som möjligt inom ramen för funktionaliteten. Detta för att de mobila enheterna, med deras begränsade prestanda, skall påverkas i minsta möjliga mån.

D.1 DataMessage

Vanligt datapaket för att skapa en ritrörelse, skapas genom att ge en IdCode av typen DATA_MESSAGE som msg_id.

Innehåller:

byte player_id = Identifierare på spelaren som skickade meddelandet.

short path_id = En unik path identifierare.

short coordinates[] = Lista av koordinater att måla upp.

long timeStamp = Tidstämpel på när meddelandet skapades i millisekunder.

D.2 BrushMessage

Meddelande som innehåller pensel-information, skapas genom att ge en IdCode av typen BRUSH_MESSAGE som msg_id.

Innehåller:

byte player_id = Identifierare på spelaren som ändrade pensel-inställningar.

byte r = Röd färg värde.

byte g = Grön färg värde.

byte b = Blå färg värde.

byte alpha = Alpha värde.

byte brush_size = Tjockleken på pensel.

boolean[4] effects = Lista av boolean för att indikera vilken effekt som är aktiv.

D.3 UndoMessage

Meddelande ifall en specifik path skall tas bort, skapas genom att ge en Id-Code av typen UNDO_PATH som msg_id.

netpaint netpaint **Innehåller:**

short path_id = Unikt path id på vilken path som skall tas bort.

D.4 RedoMessage

Meddelande av att återställa den senaste pathen, skapas genom att ge en IdCode av typen REDO_PATH som msg_id.

Innehåller:

byte player_id = Id på spelaren som ändrade pensel-inställningar.

byte r = Röd färg värde

byte g = Grön färg värde

byte b = Blå färg värde

byte alpha = Alpha värde

byte brush_size = Tjockleken på pensel

boolean[4] effects = Array av boolean för att indikera vilken effekt som är aktiv.

short path_id = En unik path identifierare.

short coordinates[] = Lista av koordinater att måla upp.

long timeStamp = Tidstämpel på när meddelandet skapades i millisekunder.

D.5 InfoMessage

InfoMessage kan vara av tre olika typer av meddelanden, för att ge information till spelaren.

Se nedanför:

D.5.1 PLAYER_JOINED

Meddelande för att indikera att en spelare har gått med i rummet, skapas genom att ge en IdCode av typen PLAYER_JOINED som msg_id.

Innehåller:

byte player_id = Id på spelaren som gick med i rummet.

byte members = Nuvarande antal spelare i rummet.

D.5.2 PLAYER_QUIT

Meddelande för att indikera att en spelare har lämnat rummet, skapas genom att ge en IdCode av typen PLAYER_QUIT som msg_id.

Innehåller:

byte player_id = Id på spelaren som gick ut ur rummet.

byte members = Nuvarande antal spelare i rummet.

D.5.3 JOIN_SUCCESS

Meddelande för att indikera att du har lyckats att gå med i rummet, skapas genom att ge en IdCode av typen JOIN_SUCCESS som msg_id.

Innehåller:

byte player_id = Id på spelaren som gick ut ur rummet.

byte members = Din unika spelar identifierare skapad av servern.

D.6 QUIT_MESSAGE

Meddelande till server från klienten att nedkoppling skall ske, skapas genom att ge en IdCode av typen QUIT_MESSAGE som msg_id.

D.7 HistoryMessage

Ett meddelande innehållande tidigare skickade datapaket i det aktuella rummet, skapas genom att ge en IdCode av typen ROOM_HISOTRY som msg_id.

Innehåller:

ArrayList history = En arraylist av DATA_MESSAGE och BRUSH_MESSAGE