

Institutionen för datavetenskap
Department of Computer and Information Science

Examensarbete

Google Maps som spelmotor för mobila plattformar

av

Ante Wall

LIU-IDA... -----

2015-06-03



Linköpings universitet

Examensarbete

Google Maps som spelmotor för mobila plattformar

av

Ante Wall

LIU-IDA/...

2015-06-03

Handledare: Anders Fröberg & Erik Berglund

Examinator: Johan Åberg

Google Maps som spelmotor för mobila plattformar

Ante Wall
antewall@gmail.com

SAMMANFATTNING

Spel kan utvecklas på flera olika sätt med flera olika utmaningar. I detta arbete undersöks det hur lämpligt det är att utveckla och implementera ett spel tillsammans med Google Maps. Undersökningen testar hur det går att implementera vanliga element som förekommer i spel tillsammans med Google Maps. Dessutom undersöks hur mycket dataanvändning som Google Maps använder för olika typer av kartor som den erbjuder. Detta har genomförts genom att implementera ett spel baserat på Google Maps som grundmoment i spelet och undersöker därifrån resultatet av implementationen.

Även förflyttning över kartan undersöks och hur mer traditionella spelanspassade kontroller som joysticks anpassar sig till Google Maps jämfört med dess normala interaktion undersöks. Undersökningen resulterar i att Google Maps erbjuder goda möjligheter att tillämpa vanliga spelelement. Dessutom visar undersökningen av datanvändningen att den normala karttypen är mest effektivast medans terrängkartan använder sig av mest nätversdata. Google Maps vanliga touchinteraktion visas lämpas sig mest för precisionsförflyttningar och ger användare lättare att förstå interaktionen men andra typer av kartförflyttningar kan också tillämpas.

INLEDNING

Den största tiden man spenderar på smartphones är med att spela mobilspel. Hela 32 % [10] av den totala tiden som användare sitter vid telefonen spelar dem något slags spel. Detta har gjort mobilspelsindustrin till en stor marknad de senaste åren.

Nya spel använder sig ofta av en redan färdig spelmotor för att göra det enkelt att implementera spel. Men ibland kan det vara intressant att skriva en egen spelmotor ifall den behöver funktioner som inte andra spelmotorer klarar av eller om man behöver få tillgång till data från andra tjänster.

Problemet med spel som använder sig av data som måste laddas ner till telefonen under körning är att det kan påverka prestandan för mobilen samt att data inte kan hämtas och läsas i tid vilket leder till att användarens upplevelse påverkas.

De flesta spel använder sig ofta av samma kontroller även över flera plattformar, kontroller som D-pad och joysticks har varit populära de senaste femton åren och även förstås över i många spel även på de mobila plattformarna. Detta visar dock inte på att det är de bästa lösningarna på spelkontroller för mobila plattformar utan kan finnas metoder som är bättre.

Denna rapport kommer att undersöka om det är möjligt att designa och utveckla ett spel för Android med Google Maps

som en spelmotor för ett där utmaningar i spelet är baserat på användarens geografiska position där förflyttning och kontroll av spelaren är viktigt.

Google Maps är en kart och satellitfototjänst från Google som används i många kartapplikationer och är även standardkarta i Android enheter.

Syfte

Målet med examensarbetet är att undersöka hur det fungerar att designa och implementera ett action/pussel spel för Android enheter med Google Maps som spelmotor.

Resultatet i arbetet ska kunna ge kunskap till vidareutveckling av spelet som utvecklats i arbetet men även ge information angående implementation av spel tillsammans med Google Maps för framtida arbeten.

Spelet ska vara körbart på flera Android enheter där prestanda är viktigt för att skapa en bra upplevelse för användaren under spelets gång oavsett vilken typ av enhet man kör spelar på eller vilken typ av internetanslutning användaren har. Men även spelaren interaktion är viktig gentemot kartan, kan man designa egna rörelsemetoder tillsammans med Google Maps eller ska man använda sig av Google Maps färdiga touchmetoder.

Spelet ska även uppmuntra social delning mellan vänner via turbaseerade utmaningar mellan varandra.

Det här examensarbetet kommer använda sig av Google Maps API ¹ för att hämta data till spelet.

Frågeställning

Vad för fördelar eller nackdelar finns det med att designa ett action/pussel spel med hjälp av Google Maps som spelmotor med avseende på:

- Design och implementation av spelkomponenter i eller tillsammans med Google Maps

Där huvudfokus ligger i att testa och utvärdera en implementation av en virtuell joystick kontra Google Maps inbyggda touchinteraktion.

- Mäta och utvärdera dataanvändning av Google Maps för det implementerade spelet

Avgränsningar

Detta examensarbete ska inte visa det optimala sättet som man kan implementera Google Maps som en spelmotor eller

¹Google Maps API <http://developers.google.com/maps/> 2015-02-24

komponenter tillsammans med det utan mer som ett exempel på hur det kan implementeras. Inte heller ska det ses som det mest optimala sättet att få ut data från Google Maps utan tar upp viktiga aspekter att tänka på samt sätt som det går att lösa problemen på.

Enheterna kommer även att limiteras till Android enheter då rapporten har valt att endast implementera applikationen till Android i nuläget

Google Maps touchinteraktion

I denna rapport kommer orden Google Maps touchinteraktion att användas mycket. Detta syftar på de rörelsefunktioner som finns inbyggda och aktiverade från början i Google Maps för Android. Det vill säga de vanliga interaktionerna för zoom, förflyttning med hjälp av pinch och fingerrörelse.

TEORI

Spelet

Spelet som ska utvecklas är ett äventyrs/actionspel. Spelet går ut på att man väljer en bana som man vill köra, sedan kommer banan att laddas. Man kommer då få en position man ska ta sig till, tex Linköping eller London. När man är tillräckligt nära inom en specifikt satt radie räknas punkten som träffad och man får en ny plats att ta sig till så snabbt som möjligt. Spelet är slut när man har tagit alla punkterna på ett bana. Utöver detta ska man kunna utmana vänner och bekanta på att få bästa tiden på banor i spelet.

Application Programming Interface

Ett Application Programming Interface även kallat API är ett gränssnitt som kan användas för att kommunicera med annan programvara. Det är ett gränssnitt mellan två olika system, oftast via webbtjänster. API används för att få tillgång till data från andra tjänster på ett smidigt sätt genom anrop till olika funktioner som returnerar data som sedan kan användas för att visa användaren eller för att kalla på andra funktioner beroende på datatypen. [11]

Google Maps API

Google Maps har ett antal olika API:er att välja mellan [12] för att lätt kunna erbjuda olika tjänster. Några exempel är API för att ta reda på snabbaste turer mellan olika positioner, en annan hur man kan ändra utseende på markörer och andra objekt som finns i Google Maps.

I Google Maps kan man till exempel ställa ut en förbestämd väg, detta kan implementeras i spel som spelkomponenter då ifall man följer vägen får man bonusar men det kanske inte alltid är den snabbaste vägen, detta ger spelaren valmöjligheter och replay value vilket är någonting som är viktigt inom speldesign [4].

Google Maps Style API ger en friheten att ändra stilar och utseende på markörer och andra komponenter i Google Maps. Det kan användas för att skapa egna komponenter i spelet som saker som ger spelaren bonusar eller annat som kan påverka spelaren och spelet i sig.

Den här rapporten kommer att använda sig mestadels av Google Maps Android API v2 ² Det är API:et som är byggd att användas med Android för att lätt implementera Google Maps komponenter i egna applikationer på Android enheter. Den ger full tillgång till Android grafiska bibliotek och även andra tjänster såsom GPS och GSM för att kunna komma åt enhetens nuvarande position [7].

Det är dock viktigt att notera att även om Google Maps är gratis att använda så har de en maxgräns på 100 000 request per tjugofyra timmar. ³. Vill man utöka gränsen krävs speciellt tillgång från Google som isåfall kostar pengar.

Design av spel och mobilspelet

Det finns flera olika saker man bör tänka på när man designar och utvecklar ett mobilspelet eller spel i allmänhet. Att kunna definiera vad spelet ska kategoriseras som kan vara viktigt då personer som gillar vissa typer av spel som action, äventyrsspel eller pusselspel gillar olika saker i sina spel. Detta kommer även hjälpa att utveckla mot sin målgrupp men även för att kunna förbättra designen och spelkänslan för spelarna [6].

Studier visar att det är viktigt att spelaren känner att man har kontroll över spelet samt att göra viktiga funktioner i spelet lätta att använda ökar chansen att spelaren fortsätter att spela spelet [2]. Detta riktar sig mycket mot kontrollen av spelaren i spelet.

Det finns även flera olika saker som gör mobila spel populära och ökar chansen att användare spelar dem. Bland det vanligaste är att mobilspelet är ofta sociala och ökar ofta försäljning och antalet användare som spelar spelet ifall den har sociala aspekter i sig [1, 4]. Man kan se att många av de spel som ligger högst upp på nerladdningslistorna har någon social funktion som att man kan utmana sina vänner eller be om hjälp. Även topplistor är en viktig komponent i många av dessa spel [4] då det ger en mer utmaning åt spelarna och replay value.

Design av spelmotor

Det är viktigt att definiera och planera hur man bör designa sin spelmotor, detta kan leda till stora förbättringar på de områden man fokuserar på som till exempel att reducera minnesanvändning eller batterianvändning [8].

Viktiga aspekter att kolla på är hur utvecklaren ska använda sig av motorn samt hur man lätt paketerar och skickar data inom sin motor. Prestanda, minnesanvändning och batterianvändning är också viktiga aspekter inom mobilspelet samt hur spelet och spelmotorn kan anpassa sig till olika enheter med olika storlekar då Android har stöd många olika enheter.

Mobil nätverkstrafik

Den mobila nätverkstrafiken har blivit en stor del av världens nätverkstrafik, bara 2014 ökade den mobila nätverkstrafiken med 69% från tidigare år. Dessutom finns nästan en halv

²Google Maps Android API v2 <http://developers.google.com/maps/documentation/android/2015-02-10>

³Google API Quota, 2015-05-12 <https://developers.google.com/maps/documentation/business/webservices/quota>

miljard mobila enheter som har tillgång till internet. Därför kan mobila nätverk ses som en nödvändighet för de flesta användare.

Mobil nätverksdata är något som har växt och kommer fortsätta växa, redan 2019 räknas en normal smartphone att använda sig av ypp till fyra gigabyte av data i månaden, det är en ökning på 500%. Nya tjänster som 4G visar även på en trend av ökande nätverkstrafik jämfört med likande tjänster.

Därför är det viktigt att försöka att undvika att öka mängden data i onödan då många datacenter och annan utrustning måste uppgraderas för att kunna klara den typen av datamängd och antalet mobila enheter som är uppkopplade till dem. De kommande åren kommer därmed att vara kritiska för den mobila marknaden[3].

Navigering och Rörelse

Tidigare forskning har gjorts på rörelse för kartor där man använt en joystick som kan röra sig i fyra led. Upp, ner, höger och vänster och visar att en fördel med joysticks när det kommer till kraft, smidighet och noggrannhet när man försöker att navigera till punkter på en digital karta. [9]

Andra källor visar att när man utvecklar mobila applikationer vill spelare ofta ha nya unika spel men samtidigt vara välkända i formen, detta gäller även för navigering och rörelse i vanliga använda applikationer som Google Maps. Ifall man ändrar för mycket i vad användare är vana att använda sig av kan det vara negativt för spelet.

Dessutom visar en trend på att den traditionella D-padden är på väg att försvinna då man hellre vill använda de interaktiva kontrollerna på mobila enheter som accelerometern, gyroskopet eller vanliga touchfunktioner som finns på dagens smartphones.[1]

Man har även observerat att när man utforskar okända områden man tidigare inte är bekant med försöker man ofta hitta landmärken eller vägar för att kunna orientera sig kring.[9] Det är då viktigt att försöka leda spelarna rätt i spelet med hjälp av spelupplägget.

METOD

Android

Android⁴ är ett operativsystem utvecklat av Google som främst är inriktad på mobila enheter såsom smartphones och tablets. Android kan till skillnad från till exempel iOS utvecklas på flera olika operativsystem såsom Windows, Mac OS och Linux vilket ger den lite fördelar jämfört med iOS [5].

Android applikationer utvecklas oftast i Java med Androids SDK⁵ som lätt ger dig tillgång till alla funktioner i Android operativsystem som till exempel GPS, internet och den inbyggda kameran. Andra alternativet är att använda Native Android som kodas i C/C++. Google Maps SDK fungerar dock inte lika smidigt med Native Android. Det är dessutom lätt att implementera få tillgång till Google Maps och dess

⁴Android <http://www.android.com> 2015-02-09

⁵Android SDK <http://developers.android.com/sdk/> 2015-02-09

versioner i Android som man kan se i kodexempel 1. Därför kommer denna rapport använda sig av Android SDK.

```
OnMapReadyCallback {  
    @Override  
    protected void onCreate(Bundle  
        savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.map_activity);  
        MapFragment mapFragment = (MapFragment)  
            getSupportFragmentManager()  
                .findFragmentById(R.id.map);  
  
        mapFragment.getMapAsync(this);  
    }  
    @Override  
    public void onMapReady(GoogleMap map) {  
    }  
}
```

Utvecklingsplattform

Det finns en mängd olika utvecklingsplattformar att använda sig av. Eclipse, Netbeans, IntelliJ är några av de största.

Google erbjuder en utvecklingsplattform som heter Android Studio som är byggd på IntelliJ. Android Studio är designad av Google och för att lätt och smidigt kunna utveckla emot Android med hjälp av Android SDK. Android Studio kom därmed att användas som utvecklingsplattform i denna rapport.

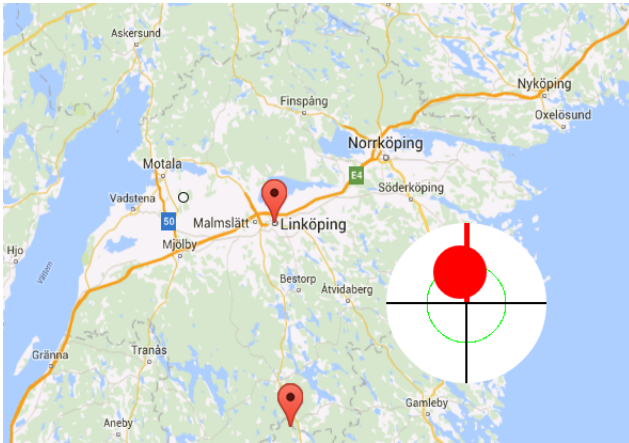
Implementation

Arbetet utfördes Agilt, dock med inslag utifrån kundens önskemål. En gång i veckan planerades ett möte med kund där vi gick igenom den nuvarande builden samt diskuterade framtida förändringar och tillägg till applikationen. Till sammans med kund tog vi fram ett antal komponenter som ska implementeras in i spelet och Google Maps. Dessa punkter förändrades under arbetets gång. Funktioner och komponenterna valdes ut av kund i första hand som skulle implementeras baserat på deras tankar runt spelet. Förslag och förändringar diskuterades gentemellan parterna för att få ett bra resultat. Nedanför visas en lista på de saker som valdes som prioriterade och implementerades i spelet.

Spelkomponenterna som skulle implementeras som togs fram tillsammans med kund var följande:

- Banor placerade på olika platser och höjdnivåer
- Testning av joystickförflyttning eller Google Maps touchinteraktion
- Achievements
- Turbaserade utmaningar mellan spelare
- Leaderboards med bästa tiderna

Utifrån dessa punkter undersöktes det hur komponenterna skulle kunna tänkas implementeras på bästa sätt i ett spel samt vad Google Maps har för alternativ för att lätt och smidigt kunna implementera det. Först undersöktes det ifall det fanns komponenter inbyggt i Google Maps som går att använda för att implementera uppgiften. Antingen en komponent som går



Figur 1. Bild på den virtuella joysticken under körning i applikationen.

att direkt använda eller modifiera. Sedan kollades det upp hur det skulle gå att implementera själv in i Google Maps. Ifall ingen bra lösning hittades undersöktes det ifall det fanns Android komponenter som gick att implementeras tillsammans med Google Maps.

Enheter

De Androidenheterna som används för testning är:

- LG Google Nexus 5
- Samsung Galaxy Tab 3 10.1

Kartförflyttning

Google maps inbyggda system för rörelse använder zoom, pitch och fingerprecision för att röra kartan. I en mer traditionell spelmiljö brukar en Gamepad eller joystick användas för att förflytta spelaren omkring i spelvärlden.

Genom att skapa en virtuell joystick i Android kan vi utvärdera och analysera hur det fungerar gentemot Google Maps interaktion.

Joysticken byggdes med hjälp av ett färdig JoystickView skri- ven av Aj Alves⁶. Rörelsen över kartan förflyttas i pixlar exponentiellt gentemot hur mycket man drar i joysticken, från 0 pixlar upp till 400 pixlar. Joysticken kommer fram där man sätter ner fingret på kartan och förflyttar sig i den riktningen som man drar fingret åt efter den ursprungliga positionen. Detta gjordes för att fungera på personer med olika storlekar av händer samt på både höger och vänsterhänta. Se figur 1 för en visuell presentation.

Zoomfunktion lades till för joysticken genom att klicka en gång på kartan så zoomar den in, är den inzoomad kan man klicka igen för att zooma ut.

Två rörelsealternativ kom att utvärderas genom att låta en testgrupp på tio personer testa spelet. Fem personer fick testa att använda sig av Google maps vanliga rörelse och touchinteraktioner. De resterande fem personerna fick använda sig av den virtuella joysticken som byggdes i arbetet. De fick

⁶JoystickView, 2015-05-08 <https://github.com/zerokol/JoystickView>

inga instruktioner om hur spelet fungerade eller vilket mål spelet hade. Inte heller fick de någon information angående hur rörelsen i spelet fungerade. Efter de hade fått köra spelet i ca fem till tio minuter fick de efteråt svara på 5 frågor angående vad de ansåg sig tycka om rörelsen i spelet. Frågorna togs fram av författaren av arbetet där frågorna var indirekt avsedda att få kommentarer på vad folk tyckte var svårt i spelet och sedan mer direkta frågor angående förflyttning över kartan. Detta kom sedan att evalueras för att få fram vilket sätt som passar bäst till spelet. Detta undersöktes då det är viktigt att spelaren ska känna att den har kontroll över spelet vilket är viktigt i design av spel [2].

Testgrupp

Testgruppen togs ut slumpmässigt av personer som var intresserade och tillgängliga att testa applikationen. Testgruppen bestod av både män och kvinnor i åldern 20-49 där majoriteten var studenter inom något typ av datavetenskap, men även några frivilliga från andra grupper. Sammanlagt testade tio personer rörelsen i applikationen. Se tabell 1 för mer detaljerad information.

Ålder	Kön	Sysselsättning
27	Man	Studerande
22	Man	Studernade
22	Man	Studerande
24	Man	Studerande
21	Man	Studerande
20	Man	Studerande
25	Man	Snickare
24	Kvinna	Studerande
25	Kvinna	Grafisk Designer
49	Kvinna	Ekonomiassistent

Table 1. Testgrupp för rörelseinteraktion

Spelets rutter

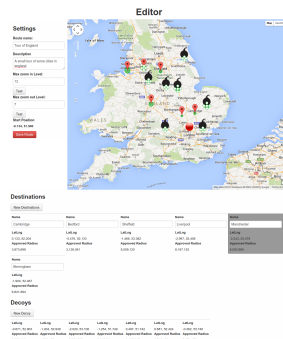
Banorna i spelets fick komma att kalla sig Routes under implementation. Dessa är ett antal olika koordinater specificerat med longitude och latitude samt en radie på hur nära en spelare kan vara punkten för att den ska godkännas som träffad.

För att lätt kunna testa olika typer av Routes för att undersöka spelvärdet samt datahämtning skrevs en lätt editor i Ruby on Rails tillsammans med Angular som kopplats ihop med Google Maps Javascript API v3⁷. Med editorn kan man lätt sätta ut startposition, olika mål och decoys samt ställa in dess träffradie och spara ruten för testning.

Datanvändning

För att kunna mäta datan som används av Google maps så kopplades Android enheterna ihop med Android Device Manager, ett program som inkluderas tillsammans med Android Studio där man kan mäta olika typer av data ifrån den inkopplade enheten, antingen via hela enheten eller specifika applikationer. Android Device Manager har en funktion som

⁷Google Maps Javascript API v3, 2015-05-13 <https://developers.google.com/maps/documentation/javascript/>



Figur 2. Editorn med en rutt utsatt inklusive start punkt, mål och decoys

låter en mäta antalet bytes som skickas och tas emot av programmet som körs på enheten samt antalet paket som skickas över nätverket för den givna applikationen.

Ett test sattes upp på två olika Routes som hade olika längder och olika inzoomningsnivåer på kartan. Test nummer ett var uppsatt över en liten stad med sex stycken stopplatser. Andra testet var inställt på att hitta fem stycken huvudstäder. Data mättes sedan på de olika banorna under körning. Dessutom mättes olika karttyper, Google maps tillåter fem olika typer av karttyper⁸.

- Normal - Grundläggande kartan
- Hybrid - Satellit kartor med en genomsnittligt lager med namn av stora gator eller städer.
- None - Inga kartrutor
- Satellit - Satellit kartor utan märkningar
- Terräng - Terräng kartor.

För varje test gjordes även fyra tester per karttyp. Första testet gjordes med att rensa cachén i Android applikationen och sedan köra Routen. Andra testet kördes direkt efter första utan att rensa cachén. Tredje testet rensades även cachén på applikationen, dock så zoomade man in så mycket det gick när man kom fram till en stopplats och sedan zoomade ut efteråt för att kunna mäta datan ifall datan ökade när man växlade mellan olika inzoomningslägen. Fjärde testet utfördes precis efter tredje testet och gjordes på samma sätt som test nummer tre dock utan att rensa cachén mellan körningen. Datan sammanställdes sedan för att se vilken karttyp som sammanlagt använde mest nätverksdata under alla fyra testen.

Dessa tester utfördes endast på Samsung Galaxy Tab 3 10.1

RESULTAT

Implementation av spelkomponenter

Nedan kommer resultat på hur de olika komponenterna som togs fram av kund implementerades i spelet.

⁸Google Maps karttyper, 2015-05-06

https://developers.google.com/maps/documentation/android/map#map_types

Routes

Routes valdes banorna i spelet att kallas under implementationen och implementerades med hjälp av Google Maps markörer. Markörerna kan sättas ner på en plats baserat på longitude och latitude. Därefter beräknas avståndet mellan spelarens position och markörer. Ifall spelaren är tillräckligt nära en radie representeras av en cirkel som är en klass i Google Maps API. Med hjälp av denna radie räknas spelaren som att han har träffat positionen ifall han är inom cirkeln.

Google Maps har mycket funktioner för att representera visuell data på sina kartor som fungerar även vid ändring av zoomdjup.

Turbaserade utmaningar och Achievements

Google Maps är inte byggt för spel och har därmed inte egna lösningar för system som utmaningar och Achievements. Google erbjuder dock andra tjänster till Android som erbjuder dessa typer av tjänster och är skapat för att kunna köras i bakgrunden av alla typer av Android applikationer.

Turbaserade utmaningar och Achievements blev därmed implementerat med hjälp av Google Game Service⁹ som ligger i bakgrunden av applikationen. Detta valdes för att slippa bygga en bakomliggande service men även för att det är byggt för att implementeras i Android spel och har stöd för både turbaserade utmaningar och Achievements.

Ett flertal problem uppstod dock under implementationen av Google Game Service då viss felhantering var problematiskt att undersöka. Dessutom måste all data skickas som en byte array vilket gjorde att man var tvungen att skapa speciella klasser som man tillät att serialisera.

Leaderboards

Google Game Service har stöds för Leaderboards och diskuterades att användas men valdes bort då maxantalet leaderboards man får skapa är 70 stycken. Spelet som utvecklades till detta arbetet ska ha stöd för fler än 70 Routes vilket gjorde att andra metoder undersöktes.

Leaderboards implementerades tillslut med hjälp av SQLite3 vilket har inbyggt stöd i Android SDK. Leaderboards implementerades bara lokalt då det inte fanns tid att skapa en bakomliggande tjänst för att spara datan inom arbetets tidsram.

Google Game Services visade sig vara mycket bra lämpad för dessa ändamål och fungerade bra tillsammans med Google Maps.

Rörelseinteraktion

Joystick

Joysticken gick igenom flera stadier av utveckling under arbetet innan den testades på folk. Först förflyttade den med en fast satt hastighet över kartan och saktades ner när man kom nära sitt mål. Detta valdes sedan att tas bort och ersattes istället med att exponentiellt öka hastigheten på joysticken ju högre som användaren drar i joysticken för att kunna ge mer precision över kontrollerna. Sista som lades till var att flytta

⁹Google Game Service, 2015-08-12, <https://developers.google.com/games/services/>

från en statisk position av joystickens placering till att utgå ifrån punkten där spelaren sätter ner sitt finger.

För att testa den bästa rörelseinteraktionen med spelet gjordes tester med tio personer vilket gav följande resultat.

Testare - Joystick

Personerna som blev utvalda att testa joystick implementationen kände att de hade mer utmaningar än de som använder Google maps touchinteraktion. Utöver att de tyckte att det var en utmaning i spelet att hitta positionerna var det även en utmaning att kontrollera spelaren i spelet. Många hade dessutom svårt att förstå vad joysticken var för någonting eller hur den fungerade. Ett flertal ville kunna använda sig av pitch funktion och vanliga rörelser och blev förvirrade och irriterade på joysticken. Några tyckte även att det var svårt i början av spelet att använda joysticken med blev lättare med tiden och att det var att frustration och förvirring var på grund av ovanhet med den typen av kontroll för Google Maps som var problemet.

Observationer under körnings visade även på att joysticken gjorde det svårare att träffa sina mål då spelarna ofta åkte förbi målen för de rörde sig för snabbt.

Desto längre tid spelarna spelade spelet desto mer precisa och lättare kunde de manövrera rörelsen och ta sig runt på kartan. Detta visar på att folk lär sig själva hur de ska klara av utmaningarna med rörelsen och att uteslutande av denna typ av rörelse inte aktuell utan fortfarande fungerar.

Testare - Google Maps touchinteraktion

De personer som testade med Google maps touchinteraktion tyckte att den enda utmaningen var att det var svårt att hitta platser som man inte hört talas om innan. Ingen klagade på rörelsen utan kände att dom redan från start hade full kontroll över inzoomning, rörelse och precision.

Datan visar därmed att användare som får använda sig av Google Maps inbygga rörelse har lättare att sätta sig in i hur rörelsen i spelet fungerar och fokuserar därmed mer på utmaningen i spelet.

Dataanvändning

Resultatet visar på att för en Route använder sig Google Maps någonstans mellan 0,5 megabyte upp till 17 megabyte beroende på rutt, kartyp och mängden inzoomningar. Caching utförs dock automatiskt och reducerar antalet megabytes till mellan cirka 0,05 - 4,5 megabytes. Se bilagor för mer detaljerad data.

Karttyp - Normal

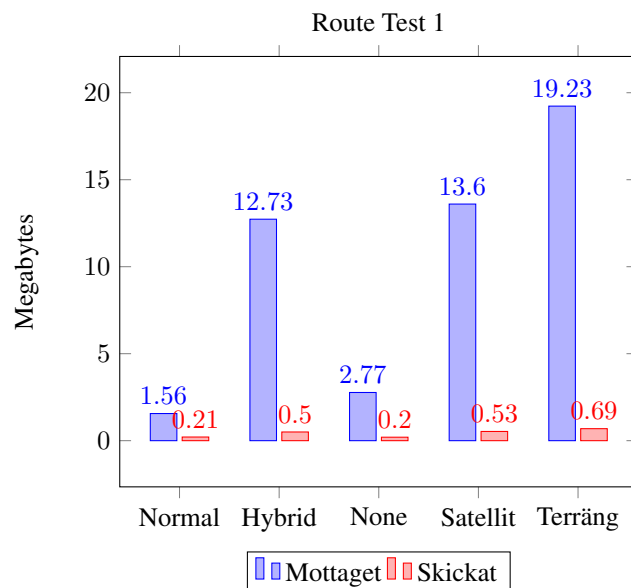
Den normala kartypen gav minst dataanvändning och verkade vara mest optimerat även när det gällde dataanvändning.

Karttyp - None

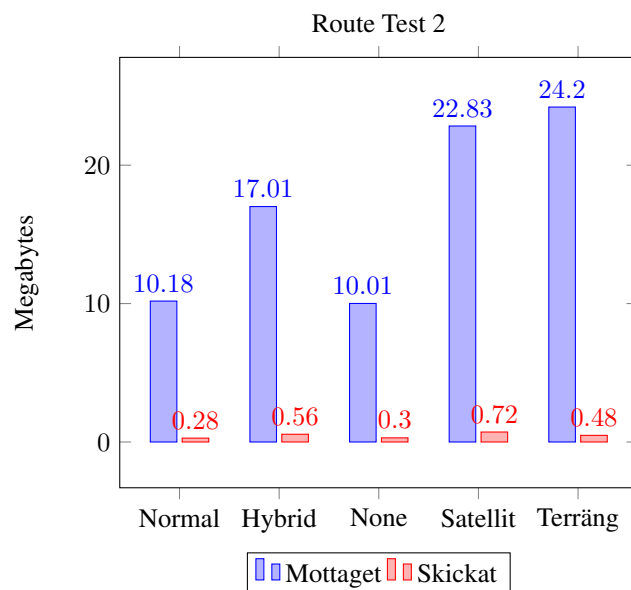
Trots att denna kartyp inte renderar någon data så skickar den ungefär lika mycket data som den normala kartypen.

Karttyp - Satellit & Hybrid

Resultatet mellan dessa två kartyper var mycket lika varandra. Satellit använde sig dock av mer data under test två när zoomningen testades mycket. Både karttyperna gav stora



Figur 3. Sammanlagda dataanvändningen för alla tester per kartyp under Route test nummer ett, se bilagor för mer detaljerad data



Figur 4. Sammanlagda dataanvändningen för alla tester per kartyp under Route test nummer två, se bilagor för mer detaljerad data

förbättringar när cachening hade gjorts av Google maps och reducerade dataanvändningen marginellt. Se figur 5.

Karttyp - Terräng

Denna karttyp använde sig av mest data under båda testerna. Testningen av hämtning under zoomning gav stora resultat på test två medans under test ett lade den såg nära Sattelit karttypen. Även under körning verkade kartan ta längre tid att ladda färdig relativt till de andra karttyper.

Cachening

Google Maps användaravtal säger att man inte får cachea datan som kommer från Google Maps. Dock visar resultatet av testningen att cachning används automatiskt av Google Maps och därmed reducerade marginellt antalet bytes som skickades ifall man körde samma Route en gång till vid samma inzoomningsdjup. Se figur 5

DISKUSSION

Method

Implementation gick ut på att skapa ett spel för Android med hjälp av Google Maps. Undersöka hur man kunde implementera vanliga lösningar som används i spel och se vilka typ av verktyg som redan fanns i Google Maps för att implementera något liknande.

Mycket av designvalen i spelet diskuterades med kund direkt efter implementation och evaluerades och itererades flera gånger. Dessutom kom många av punkterna som implementerades upp under själva arbetet med spelet. Detta ledde även till att koden blev väldigt ostrukturerat ett flertal gånger och behövdes omstruktureras vilket tog mycket tid. Dessutom ändrades mycket av prioriteringen mellan de olika komponenterna under implementationen då nya idéer kom och blev aktuellt att undersökas. Tidigare forskning visar att planering av spelmotorn är viktigt [8] och kunde ha planerats bättre innan och under arbetets gång.

Vad som kunde ha förbättras är att striktare och mer noggrant använt sig av Scrum liknande metod för implementationen. Mycket ändrades i planeringen under arbetets gång vilket bidrog. Detta för att vid olika tidpunkter kunde arbetet bli lite ostrukturerat.

Arbetet gick tillslut ihop på ett bra och smidigt sätt men förbättringar i strukturen av arbetet kunde ha gjorts.

Spelkomponenter

För att implementera de olika spelkomponenterna valdes i första hand lösningar som fanns i Google Maps API såsom markörer, cirklar och annat. Google Maps har mycket som gör att man kan manipulera kartan och dess utseedende men komponenter som behövs utöver sånt har den väldigt lite av.

Komponenter som UI och annat valdes därför att implementeras med hjälp av så mycket Android insperat som möjligt. Eftersom Google Maps är ett fragment i Android så tillåter SDK att man implementerar alla andra Androidfunktioner över fragmentet om det önskas vilket gav mycket hjälp. UI

och menyer skapades därmed med hjälp av vanliga implementationer som används i Android. I detta arbete implementerades menyn med hjälp av en navigation drawer som kommer ut från höger sida över kartan för att inte få bort spelaren från spelupplevelsen.

Google Game Service blev ett klar fördel när det kom till implementation av Turbaserade utmaningar och Achievements då den gav mycket gratis.

Dock gav de många småproblem då det var småklurigt att implementera på ett bra sätt så den fortsatte vara med i flera olika aktiviteter samt att det inte fanns mycket hjälp utöver dokumentationen av den. Felhanteringen var inte optimal från tjänsten och många krasher skapades vilket tog mycket tid av arbetet på grund av detta. Delvis för att sätta upp API att tillåtas att användas i applikationen men även när request och inloggning skulle göras emot tjänsten.

Även lite problem uppstod med att spara datan då man var tvungen att göra om datan till bytearrays vilket gjorde implementationen lite krångligare. Detta löstes med att göra två klasser som innehöll datan som skulle skickas och serialisera den när den skulle skickas och deserialisera den när den togs emot.

Dessutom gav vissa object från Google Game Services ibland läckage vilket krashade applikationen när man läste datan. Det fanns ingen dokumenterat att man var tvungen att stänga vissa buffers som användes. När detta dock var löst fungerade det felfritt och förenklade implementationen av dessa delar avsevärt.

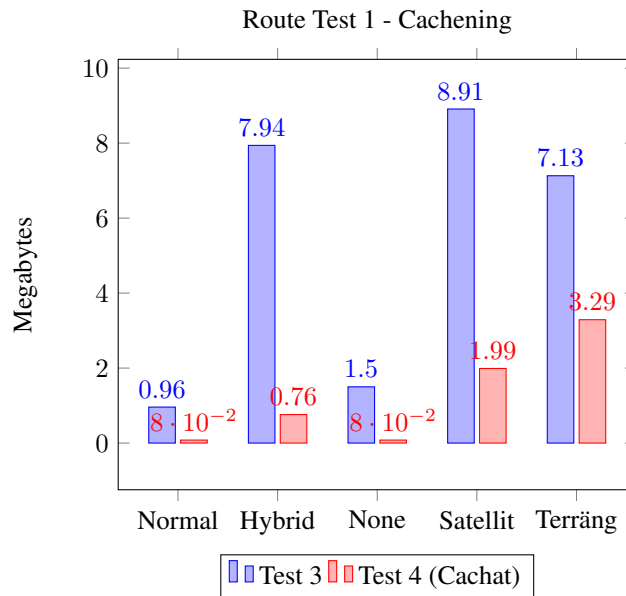
Google Game Service erbjuder även Realtime multiplayer som diskuterades att använda under implementationen av spelet. Detta prioriterades dock ner och lades tillslut på hyllan på grund av tidsbegränsning.

Leaderboards skulle varit smidigt att kunna implementera tillsammans med Google Game Service då den har stöd för det samt att det skulle vara smidigt att använda det fullt ut. Dock så var maxantalet 70st leaderboards per applikation vilket då inte blev aktuellt för applikationen som byggdes för arbetet men kan vara intressant att kolla upp mot andra spel som kan komma att implementera Google Maps. I detta arbetet valdes därför att implementera en egen lokal databas i Android applikationen då det fanns stöd för det redan i Android SDK samt att det gick snabbt och smidigt.

Google Game Services är ett bra verktyg i allmänhet för spel-design i Android och limiteras inte till Google Maps. Dokumentation och felhantering kan utvecklas men detta kan komma att lösas sig med tiden ifall fler användare börjar att använda sig av tjänsten och information och guider blir fler och lättare att hitta.

Joystick

Joystickimplementationen grundades på ett redan skrivit program och underlättade mycket implementationen av den. Dock var det en mycket enkel implementation av en joystick och gav därmed inte mycket utrymme till ändring. Problem uppstod när man skulle få joystickens att utgå från utgångspunkten där fingret satts på skärmen. Kartan och joys-



Figur 5. Resultat för test 3 och 4 för varje rutt för att visa cachningen effekt under Route test 1

ticken är varsin aktivitet och hade problem med att skicka vidare touch eventent mellan dem. Slutligen löste det sig dock så är den inte helt centerat utan ligger cirka 1-2 centimeter över fingrets position. Detta kan påverkat hur användare tyckte om joysticken då den inte ger full den effekt som man förväntar sig av en joystick.

Joysticken blev inte helt felfritt fungerade och tappade ibland toucheventen som gjordes ifall spelaren inte var så tydlig med rörelsen över skärmen, även zoomfunktionaliteten fungerade inte helt felfritt under testningen och kan därmed också ha påverkat resultatet. Mer effektiva metoder för implementation för joystick tillsammans med Google maps bör finnas som inte ger samma typ av negativa effekter.

Testning av Dataanvändning

Testerna utfördes av ett och samma program vilket gör att resultatet mellan de olika karttyperna bör bli liknande och inget annat påverkas av programmet. Dock utfördes testerna förhand med försök att replikera tidigare rörelser mellan testerna, detta på grund av tidsbrist. Det är dock omöjligt att göra exakt lika rörelser förhand mellan de olika testerna och datan kan därmed blivit smått fel mellan de olika testerna. För att kunna få mer tillförlitlig data både mer tester görs samt mer exakta tester mellan varandra. Möjligtvis genom att en datorn förflyttar sig då man lätt kan replikera detta då mellan testerna, dessutom skulle man kunde köra fler tester på en kortare tid och därmed få mer specifik data.

Resultat - Rörelseinteraktion

Testning

Datan från testningen av rörelseinteraktionen visar på att mest precision hade användarna som använde sig av Google Maps touchinteraktion och inte personerna som använde joystick. Några tyckte att joysticksrörelsen blev lättare desto längre de spelare men kände aldrig att de hade full kontroll över sin

rörelse. De som använde Google Maps inbyggda system tyckte dock att de hade full kontroll och fokuserade endast på att försöka snabbast hitta de platser som de sökte efter.

Datan visar tydligt på att folk tycker det är bättre med att använda sig av kontroller som de är vana att använda vilket det finns tidigare studier på.[1]. Det blir svårt och krångligt med att använda en joystick och ger personerna mer utmaningar i spelet än vad som kan vara tänkt. Många personer hade problem med att använda sig av pitch zoom tillsammans med joystick samt att de höll inte enheten som det var utvecklat för utan ofta lade enheten i knät och spelade med ett finger. Google Maps tillåter att man kan spela på båda sätten med joysticken har en fördel om man håller den upp och spelar med tummarna som implementaionen var tänkt.

Dock kunde man se att desto längre tid som folk satt med joysticken desto mer lärde de sig att kontrollera sig. Man kan då dra slutsatsen att joystick är ett alternativ som absolut kan fungera till Google maps ifall själva utmaningen i spelet skulle vara rörelsen. Spelet som utvecklades i arbetet var dock utmaningen inte rörelsen och valdes därmed bort som alternativ och vidareutvecklas med Google maps inbygga rörelse. Detta för att det är viktigt att spelare har kontroll över spelet och spelaren[2].

Testgrupp

Testningen av rörelseinteraktionen över kartorna gjordes på tio personer utan tidigare erfarenhet av spelet. Gruppen har en majoritet på folk i tjugoårsåldern och som läser datavetenskapliga ämnen. Detta kan ha påverkat hur de interagerar med implementationen och en större testgrupp borde nog göras på fler personer och med personer med mer olika bakgrunder och åldrar. Barn och äldre personer kan använda smartphones på annorlunda sätt som kan ge annorlunda resultat vilket också bör undersökas. Detta skulle då kunna visa på att andra me-

toder är bättre riktade mot olika åldersgrupper vilket inte tas upp i denna rapport.

Resultat - Dataanvändning

Dataanvändningen mellan de olika karttyperna i Google Maps gav intressant resultat. Normala karttypen visade sig använda sig av minst data under körning någonting som förväntades då hypotesen var att minst data används när ingen karta används som i karttypen None. Detta beror antagligen på hur Google Maps API är implementerat och antagligen så returnerar den samma data som Normala karttypen fast gömmer kartan inställt för att rendera den, detta antagande togs då resultatet var mycket nära varandra. Mer tester skulle behövas undersökades för att kunna dra en slutsats angående ämnet. Avikelser mellan datan på dessa karttyper kan också bero på hur testerna kördes.

Högst data visades Terräng karttypen ge vilket innan var hypotesen att Satellit karttypen skulle använda sig av mest data då den hämtar ner riktiga fotobilder. Men eftersom Terrängtypen visar höjdskillnader på kartan kan detta bero på att det skickas data i tre olika riktningar, både x,y och z för att kunna skapa en höjdskillnad på kartan.

Hybridläget gav även mindre data än satellitläget men var i test ett väldigt nära varandra och kan därmed bero på hur testningen utfördes som påverkat resultatet.

Speltiden för de olika testerna varierade men hade en genomsnittstid på ungefär 20 till 30 sekunder. Under denna tid kunde Google Maps ladda ner uppmot 20 megabytes av data beroende på kartyp. Detta kan medföra stora mängder nätverksdata. Mobil nätverksdata är ofta begränsad till en viss mängd beroende på abonnemang och användare bör därför kanske varnas vid användning av spel med Google maps om mängden data som kan komma att användas och därmed rekommendera att köra spelet över WiFi istället för mobila nätverk.

Replikerbarhet

Rapporten har noggrant gått igenom de olika spelkomponenterna som implementerades tillsammans med Google Maps samt hur de implementerades för att underlätta replikerbarheten för studien. Android enheter har specificeras för att kunna replikera datan på de olika enheterna.

Testerna har förklarats hur de fungerade och bilagor finns över resultatet och ruttinformation för att lätt kunna återskapa och replikera nätverkstesterna.

På det stora hela anses undersökningen vara replikerbar, författaren har försökt att dela med sig av förhållande som kom under arbetets gång.

Reliabilitet

Studiens pålitlighet varierar på olika områden. I allmänhet anses den vara hög och viktiga slutsatser och jämförelser har gjorts. Dataanvändningstesterna som tidigare nämnts i rapporten har brister och noggrannare tester bör göras för att kunna få mer allmän och specifik data. Dock anses den vara tillräckligt utförd för att kunna se relevanta skillnader och jämförelser. Möjligheten för liknande resultat är fortfarande hög för få liknande resultatet vid replikering av testerna.

Validitet

Som tidigare nämnt vore en testmiljö med datastyrda förflyttningar över kartan varit optimalt testning för dataanvändningen, därmed har inte dataanvändningen mellan de olika karttyperna kunnat jämföras optimalt dels av praktiska skäl. Resultatet ska dock vara tillräckligt applicerbart för att se jämförbara skillnader mellan karttyper.

Etiska konsekvenser

Om spel skulle börja använda sig av mycket data från tjänster som Google Maps kan de påverka samhällets datakonsumtion. Mängden data som används vid denna typ av spel är mycket mer än mer traditionella spel till mobilplattformen där nerladdningen görs en gång oftast och sedan är färdig.

Mängden data som skickas och redovisas i rapporten ser man en stor ökning jämfört med traditionella spel. Data överfört på mobila nätverk där fasta gränser för mängden mobildata som får tas emot och skickas kan påverka negativt av detta. Detta medför till ökade kostnader för konsumenter och kan leda till ökade abonnemangskostnader från nätverksleverantörerna då behovet av datamängden som behövs skickas ökas.

Datatrafiken i det svenska mobilnätet är ungefär 901 terrabytes om dagen¹⁰. En användarbas på 100 000 användare som spelar ett likande spel som utvecklasts i det här arbetet där mängden data som tas emot och skickas på en dag är 100 mb per användare skulle öka datatrafiken dagligen i Sverige med ungefär 10 Terrabyte. Siffran kan se marginell ut med vad som den totala datatrafiken om dagen är men skulle utgöra att ett spel motsvaras av cirka 1.1% av den totala mobildata som skickas i Sverige vilket har en negativ effekt för konsumenter och nätverksleverantörer, detta tillsammans med den ökade mängden data som skickas över mobila enheter[3].

Källkritik

De källor som används i rapporten har valts ut till kända och väl utvalda källor som Google officiella källor samt välkända konferanser och artiklar. Det är svårt att verifiera alla källor men arbetet har försökt att använda sig av information som är tagen ifrån mer än en källa.

SLUTSATS

Syftet med arbetet var att undersöka vad det finns för nackdelar eller fördelar att använda Google Maps för att skapa spel i, vilket arbetet har gjort.

Google Maps är ett intressant API med mycket användbarhet och tillåter att man kan modifiera mycket i den befintliga koden. Detta är en klar fördel och hjälper mycket för att implementera egna moment och komponenter tillsammans med det. Detta är både viktigt och bra då det ska användas som spelmotor.

Rekommenderad kartyp att använda är den förinställda karttypen då minst data används ifall applikationen inte behöver använda annan kartyp. Ifall Google öppnar upp mer i Google Maps API och tillåter som exempel cachning av data kan på ett bättre och snyggare reducera dataanvändningen för

¹⁰Statistikportalen - PTS, 2015-05-12 <http://statistik.pts.se/pts1h2014/>

användarna. Mängden data som används av vissa karttyper kan bli mycket höga i vissa fall och bör tänkas på under implementation av spel tillsammans med Google Maps. Flera metoder finns att reducera mängden data som används genom att till exempel använda sig av rätt typer av kartor som man kan se i detta arbete.

Android och Google Maps tillåter att man modifierar mycket och gör Google Maps till en lämpligt system att utveckla spel i om man är villig att modifiera och komma på intressanta lösningar på saker man vill implementera. Saker som Achievements, Leaderboards och andra vanliga implementationer i spel finns inget direkt i Google Maps som går att använda men med hjälp av Android SDK och flexibiliteten av Google Maps går det lätt att använda sig av Androids vanliga element, funktionaliteter samt andra tjänster tillsammans för att kunna implementera dem tillsammans med Google Maps som då interagerar smidigt med varandra.

Med hjälp av den modifikation som finns i Google Maps finns det även möjlighet att utveckla andra rörelsemotoder för att styra kartan. Google Maps touchinteraktion ger mycket bra precision och personer har det lätt att direkt sätta sig in och förstå hur kartan manipuleras tack vare detta och visar på att folk gärna vill ha likande koncept dom är vana vid [1]. Detta är en klar fördel att använda sig av, dock uteslutar dock inte att andra rörelsemotoder är sämre, arbetet visar att ifall användaren får använda sig av en joystick kan den lära sig att manipulera kartan också. Detta utgör dock fler utmaningar i spelet och spelets huvudfokus kan då försvinna i en sådan implementation. Därför är det viktigt att undersöka vad man vill att utmaningen i sin speldesign ska vara vid implementation av andra rörelsemotoder i Google Maps.

Framtida arbeten

Rörelseinteraktion

Vidare forskning på rörelse i Google Maps kan vara intressant då den virtuella joystickens gick att implementera och använda sig av i Google Maps. Användare som använde sig av den var negativa i början men lärde sig relativt fort hur de skulle anpassa sig. Vidare forskning kan göras med andra typer av rörelse som rörelse med till exempel gyroskop.

Dataanvändning

Eftersom datamätningen från Google Maps gav intressanta resultat kan det vara intressant att noggrannare mäta datan som skickas för att kunna få en noggrannare bild på vad som skickas och tas emot för att få en noggrannare bild på vad. Är det intressant även hur detta påverkas av cachningen?

REFERENCES

1. Au, W. J. *Game Design Secrets*. John Wiley & Sons Inc, 2012.

2. Chou, J., Hung, C., and Hung, Y. Design factors of mobile game for increasing gamer's flow experience. In *Management of Innovation and Technology (ICMIT), 2014 IEEE International Conference on* (Sept 2014), 137–139.
3. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2014–2019
. https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.pdf, 2015-05-15.
4. Filho, V., Moreira, A., and Ramalho, G. Deepening the understanding of mobile game. In *Computer Games and Digital Entertainment (SBGAMES), 2014 Brazilian Symposium on* (Nov 2014), 183–192.
5. Goadrich, M. H., and Rogers, M. P. Smart smartphone development: Ios versus android. In *Proceedings of the 42Nd ACM Technical Symposium on Computer Science Education, SIGCSE '11*, ACM (New York, NY, USA, 2011), 607–612.
6. Hunicke, R., Leblanc, M., and Zubek, R. Mda: A formal approach to game design and game research. In *In Proceedings of the Challenges in Games AI Workshop, Nineteenth National Conference of Artificial Intelligence*, Press (2004), 1–5.
7. Kumar, S., Qadeer, M., and Gupta, A. Location based services using android (Ibsoid). In *Internet Multimedia Services Architecture and Applications (IMSAA), 2009 IEEE International Conference on* (Dec 2009), 1–5.
8. Peker, A., and Can, T. A design goal and design pattern based approach for development of game engines for mobile platforms. In *Computer Games (CGAMES), 2011 16th International Conference on* (July 2011), 114–120.
9. Rohs, M., Schöning, J., Raubal, M., Essl, G., and Krüger, A. Map navigation with mobile devices: Virtual versus physical movement with and without visual context. In *Proceedings of the 9th International Conference on Multimodal Interfaces, ICMI '07*, ACM (New York, NY, USA, 2007), 146–153.
10. State of Mobile 2013. <http://supermonitoring.com/blog/state-of-mobile-2013>, 2015-01-27.
11. What is an API? <http://www.3scale.net/wp-content/uploads/2012/06/What-is-an-API-1.0.pdf>, 2015-03-13.
12. Which API do I Need? - Google Developers. <http://developers.google.com/maps/documentation/api-picker>, 2015-02-10.

BILAGA 1

[illegible]

BILAGA 2

[illegible]

Frågeformulär

1. Vad ansåg du var utmaningen eller utmaningarna i spelet?

2. Vad gjorde du för att komma över utmaningen eller utmaningarna?

3. Vad var det svåraste att göra?

4. På en skala ett till tio, hur svårt var det att förflytta sig över kartan.

5. Vilket var det största problemet med förflyttningen över kartan.