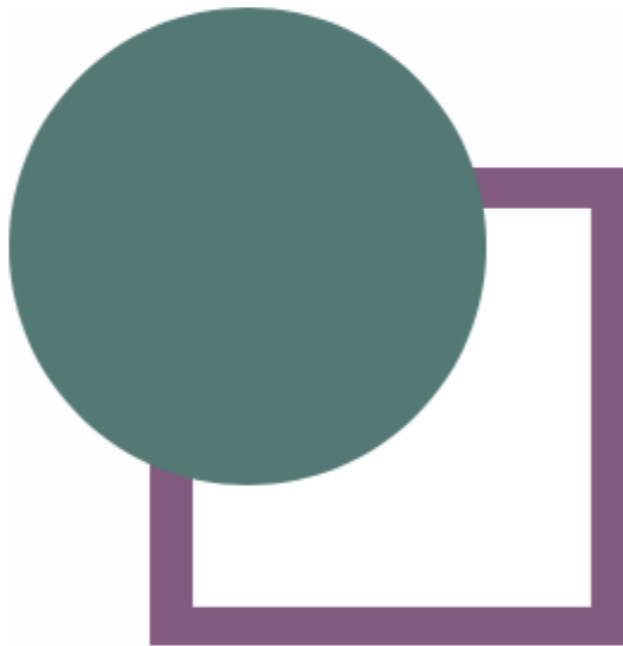




CHALMERS



CirKva

*Ett spel som sträcker sig över flera iPads,
anpassat för barn med kognitiva funktionsnedsättningar*

Kandidarbete vid data- och informationsteknik

Alexander Alvmo
Amanda Belfrage
Andreas Henriksson
Emina Hromic
Lisa Lipkin
Mikael Lönn

The Author grants to Chalmers University of Technology and University of Gothenburg the nonexclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet.

The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Author shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Author has signed a copyright agreement with a third party regarding the Work, the Author warrants hereby that he/she has obtained any necessary permission from this third party to let Chalmers University of Technology and University of Gothenburg store the Work electronically and make it accessible on the Internet.

CirKva

Ett spel som sträcker sig över flera iPads, anpassat för barn med kognitiva funktionsnedsättningar.

ALEXANDER ALVMO
AMANDA BELFRAGE
ANDREAS HENRIKSSON
EMINA HROMIC
LISA LIPKIN
MIKAEL LÖNN

© ALEXANDER ALVMO, 2016.
© AMANDA BELFRAGE, 2016.
© ANDREAS HENRIKSSON, 2016.
© EMINA HROMIC, 2016.
© LISA LIPKIN, 2016.
© MIKAEL LÖNN, 2016.

Supervisor: Olof Torgersson

Examiner: Arne Linde

Cover:

[The logo of the game CirKva, authors own image.]

Chalmers University of Technology
Department of Computer Science and Engineering
SE-412 96 Gothenburg
Telephone +46 (0)31 772 1000

Department of Computer Science and Engineering
Gothenburg, Sweden June 2016

Förord

Denna rapport beskriver ett kandidatarbete utfört av studenter på Chalmers Tekniska Högskola med antingen inriktningen Datateknik eller inriktningen Informationsteknik.

Vi vill tacka vår handledare Olof Torgersson för hans stöd och hans vägledning under hela projektet. Vi vill även tacka Peter Börjesson för hans hjälp med att få en förståelse för vilka behov vår tänkta målgrupp har och hur dessa behov ska bemötas. Vi vill dessutom tacka avdelningen för interaktionsdesign på Chalmers för utlåningen av tekniskt material. Vi vill likaså tacka alla som tagit sig tid att testa och ge respons på vårt spel genom de olika utvecklingsstadierna. Till sist vill även vi tacka de som hjälpt till med korrekturläsning av rapporten.

Abstract

According to to the Swedish National Agency for Education tablets aids with both cooperation and exploration. The Swedish National Agency for Education also indicates that children usually learn better through play, which has led to an increased use of tablets in Swedish schools. The Touch AT! project has drawn attention to the increased use of tablets in special schools and the deficit on applications to increase social and emotional capabilities of the children. As a reaction to this, the game has CirKva created in partnership with Touch AT!

This report describes the development of CirKva, a logical cooperation game that extends over multiple tablets. The purpose of CirKva is that it will be well adapted for children with cognitive impairment and give them the opportunity to practice their social and cooperative abilities.

The development began with a study of available literature on the target user group, where a deficiency of studies regarding how to adapt applications for cognitively impaired children was found. CirKva was developed iteratively, and user-tests with both a paper prototype and the actual product was conducted. The user tests were followed by semi-structured interviews to gather data for further analysis and development.

Results of game tests suggest that the game concept is on the right path but the game must be developed further and more testing must be done before we can say anything about how well it lends itself as a tool for social and emotional training for pupils in special schools.

Sammanfattning

Enligt Skolverket underlättar surfplattor både för samarbete och utforskande. Skolverket pekar dessutom på att barn vanligtvis lär sig lättare genom lek vilket har lett till en ökad användning av surfplattor i Svenska skolor. Projektet Touch AT! har uppmärksammat det ökade användandet av surfplattor i särskolan och underskottet på applikationer för att öka den sociala och emotionella förmågan hos barnen. Som en reaktion på detta har spelet CirKva skapats i samarbete med Touch AT!.

Denna rapport beskriver framtagningen och utvecklingsprocessen av CirKva, ett logiskt samarbetsspel som sträcker sig över flera surfplattor. Målet med CirKva är att det ska vara anpassat för barn med kognitiva funktionsnedsättningar och ska ge barnen möjlighet att öva på sina sociala- och samarbetsförmågor.

Utvecklingen inleddes med en litteraturstudie där en brist i underlag för riktlinjer för applikationsutveckling för barn med kognitiva funktionsnedsättningar upptäcktes. CirKva utvecklades sedan iterativt, och användartester på både pappersprototyp och produkt utfördes. Användartesterna följdes av semistrukturerade intervjuer, vilka gav underlag till analys och vidare utveckling.

Resultat från speltester föreslår att spelkonceptet är på rätt väg men spelet måste vidareutvecklas och fler tester måste göras innan man kan säga något om hur väl det lämpar sig som ett hjälpmedel för social och emotionell träning för elever i särskolan.

Definitioner

Evaluering: Tester av spelet som involverar användare oberoende av spelets utveckling.

GUI: Förkortning av Graphical User Interface (eller grafiskt användargränssnitt på svenska).

iOS: Operativsystem utvecklat för Apples mobila enheter: iPhone, iPad samt iPod Touch.

LAN: Local Area Network (eller lokalt datornät på svenska).

OS X: Operativsystem för Mac-datorer utvecklat av Apple Inc.

SpriteKit: (Spel)ramverk från Apple Inc.

TCP: Transmission Control Protocol.

UDP: User Datagram Protocol.

Innehållsförteckning

1	Inledning	1
1.1	Projektets mål	2
1.2	Rapportens syfte	2
1.3	Avgränsningar	2
2	Inledande efterforskning	3
2.1	Spelets målgrupp	3
2.2	Val av plattform	5
2.3	Undersökning av befintliga spel	5
2.4	Sammankoppling	6
3	Teknisk bakgrund för utveckling av ett flerspelarspel för operativsystemet iOS	7
3.1	Utvecklingsverktyg	7
3.2	MVC och MVP	8
3.3	Multipeer Connectivity Framework	9
3.4	SpriteKit	10
4	Utvecklingen av CirKva	12
4.1	Framtagning av spelkoncept	12
4.1.1	Inledande kravidentifiering	13
4.1.2	Fastställande av spelkoncept utifrån krav	13
4.2	Framtagning av spelets grafiska gränssnitt	14
4.2.1	Navigering av spelet	15
4.2.2	Karaktärers rörelsemönster	16
4.3	Pappersprototyp	16
4.4	Evaluering av prototyp	21
4.4.1	Evalueringsmetod	21
4.4.2	Resultat av evaluering av prototyper	22
4.5	Metod vid evaluering av slutlig design	23
4.6	Systemmodellering	23
4.7	Mjukvaruarkitektur	25
5	Resultat	27
5.1	Slutlig produkt	27
5.1.1	Spelkoncept	27
5.1.2	Design	27

5.1.3	Genomgång av spelomgång	28
5.2	Evaluering av slutlig design	31
5.2.1	Grupp 1	32
5.2.2	Grupp 2	33
5.2.3	Grupp 3	33
6	Diskussion	35
6.1	Forskningsläget	35
6.2	Plattform	36
6.3	Evalueringsprocess	36
6.4	Användning av pappersprototyp	37
6.5	Testgrupper vid evalueringar	37
6.6	Förväntningar och målsättningar	38
6.7	Målgruppsanpassning	39
6.7.1	Anpassning av gränssnittet till målgruppen	39
6.7.2	Anpassning av spelets upplägg för målgruppen	41
6.8	Framtida utveckling	43
7	Slutsats	44
A	Intervju efter testning av pappersprototyp	I
B	Intervju efter testning av slutlig produkt	II
C	Multipeer Connectivity Framework	IV
C.1	Klasser & Objekt	IV
C.2	Protokoll	V
D	Kravspecifikation	VI
D.1	Funktionella krav	VI
D.2	Icke-funktionella krav	VII
E	Användningsfall	VIII
E.1	Prioriterade användningsfall	VIII
E.2	Samtliga användningsfall	XI

1

Inledning

I takt med att utvecklingen av mobil teknologi har gått framåt har den blivit allt mer lättillgänglig. Det har lett till att det idag har blivit allt vanligare för skolor i Sverige att erbjuda sina elever varsin surfplatta eller dator som de får använda sig av under sin skolgång [1]. Tanken är att teknik ska vara en positiv bidragande faktor för undervisningen.

Enligt Skolverket underlättar surfplattor både för samarbete och utforskande. Skolverket pekar dessutom på att barn vanligtvis lär sig lättare genom lek, vilket gör att bland annat spel på surfplattorna används för att lära dem olika saker [2]. Det finns redan en rad exempel på sådana spel. Bland annat Mattebageriet, som hjälper barn att öva på addition och subtraktion [3], och Kalle Kunskap där barn från årskurs noll till två kan träna sina färdigheter inom språk och matematik [4].

Den ökade användningen av surfplattor i Svenska skolor har lett till att de idag även används inom särskolan. Det går idag ungefär 12 000 barn i den svenska särskolan, och bland dessa barn finns många varierande typer av kognitiva funktionsnedsättningar. Trots den stora variationen bland barnen finns ett återkommande mönster, vilket är svårigheten av den emotionella och sociala förmågan som barnen besitter.

Projektet Touch AT! har uppmärksammat det ökade användandet av surfplattor i särskolan och underskottet på applikationer för att öka den sociala och emotionella förmågan hos barnen. De ämnar därför bland annat att undersöka hur barn med kognitiv funktionsnedsättning kan utveckla dessa förmågor med hjälp av pekskrämsbaserade hjälpmedel[5].

I samarbete med Touch AT!, och för att utnyttja möjligheterna som finns med surfplattor och lärande genom lek, har ett logikspel utvecklats för att lyfta fram samarbetsviljan hos elever med kognitiva funktionshinder. Spelet är inte bara underhållande, utan kräver även att eleverna interagerar och diskuterar med varandra för att lösa de olika problem som spelet presenterar. Tanken med spelet är främst att öka elevernas samarbetsförmåga, men det ska även bidra till en ökad social förmåga.

1.1 Projektets mål

Då detta projekt har ett samarbete med Touch AT!-projektet medför det att spelet främst är riktat mot barn med kognitiv funktionsnedsättning. Målet med projekt är därför främst att utveckla ett spel som är anpassat för just denna målgrupp. Spelet ska främja samarbete och fungera som ett hjälpmedel för social- och emotionell träning för elever i särksolan. Det var Touch AT!-projektet som kom med förslaget till det här kandidatarbete och i projektbeskrivningen var ett av kraven att spelet skulle sträcka sig över flera enheter som tillsammans utgör en enda stor spelplan.

Projektet kan delas in i två delmål:

- Utveckla ett spel som är anpassat för barn med kognitiva funktionsnedsättningar.
- Implementera spelet så att spelplanen sträcker sig över flera enheter.

1.2 Rapportens syfte

Syftet med denna rapport är att ge läsaren en inblick i utvecklingen av spelet CirKva, med fokus på hur spelet har anpassats för barn med kognitiva funktionsnedsättningar.

1.3 Avgränsningar

Då de skolor som Touch AT!-projektet samarbetar med enbart använder sig av iPads var intresset för andra plattformar från Touch AT!s sida svalt. Således kommer spelet endast att utvecklas för iPads.

På grund av en begränsad budget i projektet har vi haft ett begränsat antal enheter att testa spelet på. I implementationen har vi strävat efter att spelet, teoretiskt sett, ska fungera med så många enheter som möjligt. Trots det har vi enbart fokuserat på att designa spelinnehåll för upp till fyra enheter. Detta för att vi ska kunna testa och garantera att allt fungerar som önskat.

I och med att projektets bakgrund och mål är nära förknippat med kommunikation och social kompetens har vi valt att i huvudsak begränsa oss till att se över teknikens möjligheter inom den sociala sfären av hållbar utveckling.

2

Inledande efterforskning

Projektet inleddes med en fördjupning i ämnesområden relevanta för projektet, där en stor mängd information samlades in. I denna fas togs grunden för spelet fram, genom att krav och begränsningar framställdes samt att en undersökning utfördes på vilka användarna av spelet är. Det var av stor vikt att undersöka vilka riktlinjer som behövde följas för att spelet skulle passa sin målgrupp, främst då de riktlinjerna utgör grunden för spelet. Baserat på riktlinjerna, studerades sedan olika typer av plattformar samt programmeringsspråk som utvecklingen kunde ske mot. För att kunna skapa ett så bra spel som möjligt, var det av stor vikt att rätt plattform och programmeringsspråk valdes. Därefter kunde undersökning och testning av marknadsbefintliga spel ske, som ledde till att en unik spelidé anpassad för målgruppen utvecklades.

Parallellt med att en fördjupning kring målgruppen samt befintliga spel utfördes, behövde även sammankopplingsmöjligheter mellan enheterna undersökas. Idag finns det flera olika sätt att få enheter att kommunicera samt skicka data mellan varandra. Därför behövdes en undersökning kring, samt jämförelse mellan, de olika typerna göras, för att på så sätt finna den lösningsmetod som var mest lämplig för detta spel.

2.1 Spelets målgrupp

Målgruppen för spelet är densamma som för projektet Touch AT!, det vill säga barn med kognitiv funktionsnedsättning. I särskolan går barn som bedöms inte kunna klara av kunskapskraven i den vanliga grundskolan på grund av en utvecklingsstörning eller en begåvningsmässig funktionsnedsättning till följd av en hjärnskada [6].

En litteraturstudie på hur applikationer kan användas för att hjälpa barn med kognitiva funktionsnedsättningar, och hur applikationer för denna målgrupp bör utformas gjordes. Forskningsläget har en stark överrepresentation när det kommer till autismspektrumet [7], vilket kan förklaras av att autismspektrumet utgör en stor del av kognitiva funktionsnedsättningar med sina 6.9 av 1000 individer i samhället [8]. Det ger en frekvens av 2 till 1 jämfört med Cerebral pares [9], och 6 till 1 jämfört med Downs syndrom [10].

2. Inledande efterforskning

Något som var väldigt tydligt inom litteraturen var att målgruppen var svårgeneraliserad på grund av att målgruppens funktionsnedsättningar visar sig på många olika sätt, men att det fanns en del riktlinjer för mjukvarudesign. Riktlinjerna hade ofta autismspektrumet som utgångspunkt [7], [11], [12], [13] och bygger på generella egenskaper som kan hittas hos personer med autismspektrumtillstånd [11]. Bland de egenskaper som riktlinjerna för gränssnitt baseras på finns bland andra:

- Tycker om kontrollerade och förutsägbara miljöer och rutiner.
- Har svårt för misslyckanden.
- Letar efter mening i detaljer, och kan ha svårt att se helheten.
- Tenderar att ha ett repetitivt beteendemönster.
- Kan vara överkänsliga för sinnesintryck.
- Är visuella tänkare och har en visuell inlärningsstil.
- Kan ha svårt för att läsa eller kommunicera verbalt.
- Kan ha svårt för att uppfatta referenser till avlägsna objekt.

Bland riktlinjerna finner vi de som Nikolay Pavlov tagit fram i sin artikel om riktlinjer för användargränssnitt för personer med autism, mest relevanta. De riktlinjer som varit mest relevanta för utveckling av applikationer på pekskärmsheter lyfts fram i följande tabell [14]:

Ja!	Nej!
Kontrast mellan text och bakgrund	Starka färger
Mjuka färger	Bilder som bakgrund
Presentera text i en enkel kolumn	Överlappa transparent bild och text
Snabb laddtid	Dialogrutor eller andra distraktioner
Enkel grafik	Element som står ut mer än de andra
Tydlig text, sans serif font	Horisontell scroll
Enkel navigering	Komplexa menyer
Indikera var användaren är	Plotter
Visuella indikatorer vid lång laddtid	Flerfärgade ikoner
Hjälpknapp	Knappar med endast ikoner
Designa för enkelheten med få element på skärmen	
Stora tydliga knappar med både ikon och text	
Korta instruktioner för varje steg	

För att få ytterligare kunskap om målgruppen, speciellt den grupp som Touch AT! samarbetar med, utfördes en intervju med Peter Börjesson, som är inblandad i Touch AT!-projektet. Nedan presenteras den viktigaste informationen som mottogs.

- Belöningsystem kan bli en låsning och upplevas som frustrerande, främst för barn med kognitiva funktionsnedsättningar. Fokus riktas därmed mer åt att få samtliga belöningar än att lösa den faktiska uppgiften.
- Moment som skapar stress kan upplevas som svåra och obehagliga. Dessutom har en del barn även problem med finmotorik.
- Bland barnen i särskolan finns det många olika mentala nivåer.
- I mångt och mycket har barnen i särskolan samma intressen som vilket annat barn som helst.
- För en del barn kan ett alltför detaljerat grafiskt gränssnitt bli en påfrestning. Det är därför bättre med en avskalad design utan de extra detaljerna för att lägga fokus på uppgiften.

I mångt och mycket konfirmerade intervjun med Börjesson de egenskaper och riktlinjer som hittats under litteraturstudien. Intervjun och de riktlinjer som fanns inom litteraturen låg till grund för de diskussioner och beslut som togs under designprocessen.

2.2 Val av plattform

För att spelet skulle bli användbart i miljön en särskola utgör, undersöktes vilken typ av plattform skolorna generellt använder. Efter att frågan tagits upp med Olof Torgersson och Peter Börjesson, som båda verkar inom Touch AT!-projektet, visade det sig att målgruppen nästan uteslutande använder sig av iPads (se avsnitt 1.3). Därför föll det sig naturligt att inkludera iOS som en plattform att utveckla mot.

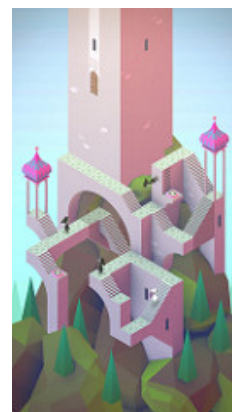
Nästa steg i den inledande efterforskningen var att besluta vilken uppsättning av utvecklingsverktyg spelet skulle kunna tänkas byggas i. Efter att ha studerat verktyg, som Corona [15] och Xamarin [16], som båda stödjer utveckling till flera plattformar, och Xcode som ger möjligheten att istället utveckla ett nativt spel, som är direkt knutet till enhetens operativsystem, valdes det sistnämnda. Valet av utvecklingsverktyg motiverades med att slutprodukten skulle ha större chans att prestera bra om den konstruerades direkt i miljön den var ämnad att användas i. Spelet skulle därmed utvecklas för iOS med programmeringsspråket Swift.

2.3 Undersökning av befintliga spel

Speltester utfördes på marknadsbefintliga spel där syftet var att skaffa inspiration och tankar som senare kunde användas i framställandet av en spelidé. Under testerna analyserades spel utifrån de kriterier som tagits fram under studien av målgruppen. Det vill säga om de till exempel hade låg stressfaktor och avskalad design.

När testerna genomförts på ett brett spektrum av spel och genrer diskuterades för- och nackdelar med olika sorters spel. Målsättningen var att plocka intressanta särdrag ur populära spel för att på så sätt senare kunna sätta ihop en produkt med många egenskaper som skulle passa i just detta projekt.

Bland annat testades spelet Monument valley, som tillhör genren logikspel. Monument valley består av en väldigt snygg grafik, se figur 2.1, med lagom svåra nivåer. Då detta var ungefär vad vi hade tänkt oss i svårighetsgrad, tog vi med oss olika idéer därifrån och hade de i åtanke vid utvecklingen av CirKvas design. Dock ansåg vi att grafiken var för avancerad, både för målgruppen och för oss som utvecklare, då vi hade en väldig tidspress.



Figur 2.1: En bana ur monument valley. Från [17] CC BY-NC 2.0.

Testning gjordes även på spelet Color Switch, vilket går ut på att leda en figur i en färg genom hinder med samma färg. Det som upplevdes positivt med det var den väldigt enkla designen, då det blev tydligt vad som var målet. Det negativa med spelet var att det inte finns olika nivåer, utan om något går fel börjar spelet om från början, vilket kan upplevas som frustrerande.

2.4 Sammankoppling

Innan utvecklingen av spelet kunde komma igång behövdes även en idé om hur de olika enheterna skulle kopplas samman för att kunna kommunicera med varandra. Flera olika metoder av sammankoppling diskuterades efter studier i hur andra spel behandlar lokal kommunikation.

Till en början utforskades metoder för att upprätta TCP- eller UDP-kontakt utan att använda yttre ramverk. Problemet med detta var att få alla enheter att hitta varandra utan tidigare kontakt. En lösning som tidigt kom fram var att upprätta kontakt via internet och en server som agerade mellanhand och introducerade enheterna för varandra. Då ett LAN var önskvärt för att ge spelet mer stabilitet så till vida att det inte skulle vara beroende av en server, förkastades detta förslag.

Vidare utforskades olika sätt att få enheterna att hitta varandra via WiFi eller Bluetooth. Det var i det här stadiet som Multipeer Connectivity Framework upptäcktes. Då Multipeer Connectivity Framework redan var utvecklat, testat och erkänt bland utvecklare, togs beslutet att använda detta istället för att utveckla en egen lösning. Detta främst då en egen lösning säkerligen inte hade fungerat lika bra som Multipeer Connectivity Framework.

3

Teknisk bakgrund för utveckling av ett flerspelarspel för operativsystemet iOS

För att utveckla en iOS-applikation behövs kunskap om utvecklingspråket samt utvecklingsmiljön som Apple använder sig av. Apple har valt att endast låta personer med datorer från deras eget fabrikat utveckla applikationer till deras samtliga enheter. Därför är grundläggande kunskap om hur operativsystemet iOS fungerar viktigt. Det är även viktigt för att få en förståelse för hur applikationer ska byggas. Spelet CirKva har skrivits i programmeringsspråket Swift, som är ett språk Apple själva har utvecklat. Swift är utvecklat med samtliga C-baserade programmeringsspråk i åtanke, vilket innebär att Swift och C-språken har snarlika egenskaper. Vid utveckling av iOS-applikationer rekommenderas det att det sker i Apples egen utvecklingsmiljö, Xcode, som erbjuder en del extra funktioner för att underlätta utvecklarens arbete.

Vid utveckling av spel och andra applikationer som kräver animation erbjuder dessutom Apple ett ramverk vid namn SpriteKit. Detta ramverk möjliggör det för utvecklaren att hantera grafik och animering vid utveckling av applikationer för iOS och OS X. Vidare, för att möjliggöra det för applikationer att ha flera enheter som kommunicerar och skickar data mellan varandra, har Apple utvecklat ett ramverk vid namn Multipeer Connectivity Framework. Båda ramverken har använts vid utvecklingen av CirKva, och utgör grunden för spelet.

3.1 Utvecklingsverktyg

Då projektets målgrupp enbart använder sig av iPads i skolorna, medförde det att det föll naturligt för spelet CirKva att utvecklas för operativsystemet iOS. iOS används i Apples mobila enheter, såsom iPhone, iPad och iPod Touch. Gränssnittet är känt för att vara väldigt användarvänligt vilket gör det attraktivt för bland annat skolor.

Vid utveckling av applikationer för iOS, finns det ett flertal olika utvecklingsmiljöer som kan väljas, såsom Xcode, Corona och Xamarin. För utvecklingen av spelet

CirKva har Xcode valts. Xcode är väldigt användarvänligt, vilket främst beror på att Apple har velat göra det så enkelt som möjligt för programmeraren genom att erbjuda funktioner utöver det vanliga [18].

För att underlätta utvecklingen av iOS-applikationer ytterligare, har Apple utvecklat ett eget programmeringsspråk, vid namnet Swift. Målet med Swift är att förse utvecklaren med ett säkert, snabbt och interaktivt programmeringsspråk. Swift är uppbyggt med Objectiv-C samt samtliga C-baserade programmeringsspråk i åtanke, och ses därför som en ersättare för dessa [19]. Det medför att Swift har en del liknande funktioner som C-språken har, såsom typ- och flödeskontroll. Orsaken till det beror på att de flesta funktioner som Swift handhar ska kunna jämföras med de funktioner som C-språken handhar.

3.2 MVC och MVP

Model View Controller (eller MVC) är ett mycket välkänt designmönster inom programmering som går ut på att koden delas upp i tre distinkta kategorier: modeller, vyer och kontroller. MVC är ett designmönster som Apple rekommenderar sina utvecklare att följa i sina projekt [20].

- **Modeller:** Enkapsulerar data och beteende från domänen.
- **Vyer:** Själva GUI:t, (grafiska användargränssnittet) som visas på skärmen.
- **Kontroller:** Agerar på användarens input.

Genom åren har det vuxit fram andra mönster som bygger vidare på MVC. I den här studien [21] jämförs några av dessa mönster, bland annat Model View Presenter (MVP). MVP är särskilt intressant för den här rapporten eftersom det har stora likheter med vad Apple kallar för MVC [20]. Leonardo Taehwan Kim menar till och med att det Apple kallar MVC är samma sak som MVP [22].

I MVP är en presentatör motsvarigheten till en kontroller i MVC. En presentatör ska fungera som en mellanhand och ansvara för att uppdatera båda vyer och modeller utifrån användarens input [21]. Figur 3.1 visar hur kommunikationen ser ut mellan de olika rollerna i MVP och MVC.

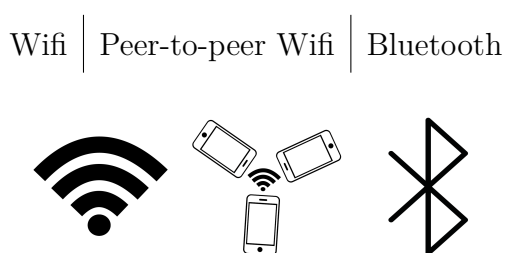


Figur 3.1: Funktionalitet för Model View Presenter, från [23] CC-BY-SA-3.0, samt Model View Controller, från [24] CC-BY-SA-3.0. Författarens egen bild CC-BY-SA-3.0.

3.3 Multipeer Connectivity Framework

För att kunna koppla samman enheter och få dem att kommunicera med varandra, används ett ramverk utvecklat av Apple vid namn Multipeer Connectivity Framework. Ramverket möjliggör för enheter att kommunicera med varandra genom att sända data. Datan som sänds mellan enheterna kan sändas i två olika tillstånd: pålitligt (reliable) eller icke-pålitligt (unreliable). Om säkerhet prioriteras, det vill säga att varje byte måste nå mottagaren för att få rätt funktionalitet, ska pålitlig dataöverföring användas. Om snabbhet å andra sidan prioriteras ska det senare tillståndet användas [25].

För att göra det möjligt för enheterna att hitta varandra och koppla upp sig till varandra, används något av följande:



Figur 3.2: Grafisk representation av de sätt som enheter kan hitta varandra i Multipeer Connectivity Framework. Adapterad från [26] CC0, [27] CC0 och [28] CC0. Författarens egen bild.

För att Multipeer Connectivity Framework ska kunna användas, måste samtliga enheter i samma session antingen vara uppkopplade på samma WiFi, eller vara anslutna via samma Bluetooth [25]. Vilket av dessa alternativ som används beror helt och hållet på vad användarna av applikationen föredrar.

Ramverket används i två olika faser, vilka är upptäcksfas samt sessionsfas. Under upptäcksfasen letar enheten efter andra enheter uppkopplade på samma nätverk. Sessionsfasen å andra sidan, börjar först när samtliga enheter har kopplats upp till varandra. Under sessionen kan enheterna sedan kommunicera med varandra [25].

Upptäcksfas

Under upptäcksfasen använder applikationen ett objekt som kontrollerar vyerna, `MCBrowserViewController`, för att visa ett gränssnitt innehållandes närliggande enheter som befinner sig på samma nätverk. Dessutom används ett "Advertiser Assistant-objekt" som heter `MCAdvertiserAssistant`. Det objekt gör det möjligt för användaren att bjuda in andra enheter till applikation. Dessutom sköter det användarnas respons på inbjudan.

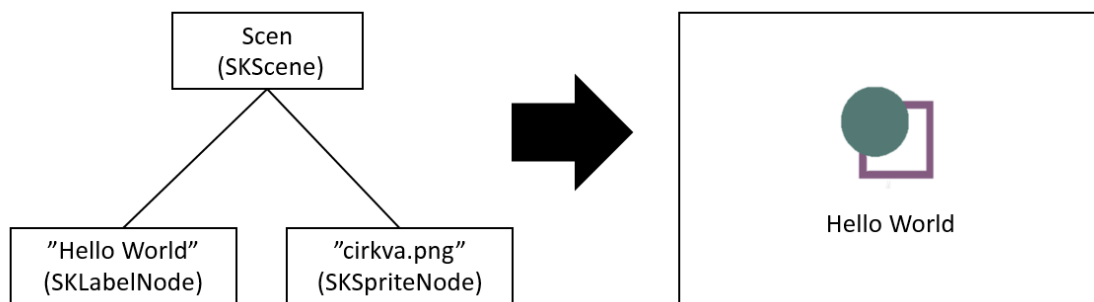
Sessionsfas

När användaren väl har valt vilka enheter som ska bjudas in till sessionen kommer de inbjudna användarna att motta en förfrågan från denna enhet som de antingen kan acceptera eller neka. Om användarna väljer att acceptera inbjudan, kommer sessionen mellan enheterna att starta. Under denna session kan enheterna kommunicera sinsemellan och skicka data. I det fall då någon av enheternas status ändras, såsom att de blir offline, kommer samtliga enheter i session att bli meddelade.

Skaparna av ramverket erbjuder två olika uppsättningar av det, beroende på vilken typ av säkerhet applikationen kräver. Utvecklaren av applikationen kan antingen välja att samtliga enheter på samma nätverk bjuds in automatiskt till sessionen, eller att en explicit inbjudan måste skickas till samtliga enheter som sedan måste accepteras innan en kommunikation mellan enheterna kan påbörjas. Den senare medför att inte vilka enheter som helst kan börja kommunicera med varandra, utan att en inbjudan måste ske först. Efter att inbjudan har accepterats kan en kommunikation inledas och upprätthållas [25]. Denna applikation använder sig av det sistnämnda fallet, vilket gör att endast det kommer tas upp i denna rapport.

3.4 SpriteKit

SpriteKit är ett ramverk från Apple som tillhandahåller funktionalitet för att hantera grafik och animering i både iOS och OS X. För att skapa bildrutor använder SpriteKit sig av en loop som tolkar och bearbetar innehållet i varje bildruta innan den skapas. Det är sedan själva spelet som bestämmer innehållet i varje scen och hur det ska förändras mellan varje bildruta. SpriteKit använder den grafiska hårdvaran för att effektivt göra bildrutor med animationer. Dessutom kan SpriteKit hantera ljud och simulera fysik som finns inbyggt i Xcode. Kombinationen av dessa funktioner, menar Apple, gör det lämpligt att använda SpriteKit för spelutveckling på deras plattformar [29]. Apple menar även att, eftersom SpriteKit hanterar allt arbete på låg nivå, så kan utvecklarna istället fokusera på att lösa designproblem på en högre nivå och därmed skapa en bra spelupplevelse. De klasser som är viktigast för utveckling av spel med SpriteKit är `SKView`, `SKScene` och `SKNode`, då det är de som är grunden för att skapa animationer och scener.



Figur 3.3: En förenkling av ett scen-träd och hur det resulterar i grafik på en skärm. Författarens egen bild.

SKNode

SpriteKit-noder, SKNode, är de grundläggande byggstenarna som används för att presentera allt innehåll i ett spel. Det finns olika typer av noder som visar olika typer av innehåll. Till exempel finns det en typ av nod som visar bilder (SKSpriteNode) och annan som visar text (SKLabelNode) [30].

SKScene

I SpriteKit delas spelinnehållet upp i en eller flera scener. SKScene är den klass som representerar en scen och det kan till exempel vara själva spelet. Det kan dock även vara huvudmenyn eller skärmen med reklam som visas efter att man har förlorat i ett spel. För programmeraren representeras varje scen av ett träd där scenen är roten och noderna är innehållet som ska visas i den scenen [31].

SKView

SKView är den klass som används för att visa en scen och dess spelinnehåll på skärmen [32].

4

Utvecklingen av CirKva

Utvecklingen och framtagningen av CirKva har varit en lång process bestående av olika delmål. Till en början togs ett första utkast av ett spelkoncept och en speldesign fram. När detta väl var färdigt, kunde pappersprototyper av spelet börja produceras. Samtliga prototyper jämfördes, för att bästa design och funktionalitet skulle fås fram.

Innan en faktiskt implementering av spelet kunde påbörjas, var det av stor vikt att testa prototyperna på oberoende användare. Genom att göra detta kunde en uppfattning skapas gällande vilka eventuella brister som fanns med spelet, samt vad som skulle vara bra att bevara. Samtliga speltester sammanställdes och ett slutligt grafiskt gränssnitt togs fram.

Parallellt med prototypframtagning och speltester, utvecklades även kravspecifikationer samt användningsfall för spelet. Kravspecifikationerna består av funktionella samt icke-funktionella krav. De funktionella kraven handlar om vad systemet ska kunna göra, medan de icke-funktionella kraven handlar om hur systemet faktiskt ska vara. Användningsfallen representerar de uppgifter en användare ska kunna utföra på spelets gränssnitt. Genom att ta fram sådana krav på systemet, underlättades implementeringen av spelet betydligt.

4.1 Framtagning av spelkoncept

Innan ett spel kunde utvecklas behövdes en generell idé om vad spelet skulle gå ut på och hur det skulle fungera. Vid framtagning av spelkonceptet behövde flera faktorer tas hänsyn till, framför allt med tanke på målgruppen. Att spelet dessutom ska användas av flera personer samtidigt ställer speciella krav på spelet, både konceptuellt och grafiskt.

Spelkonceptet togs fram iterativt med utgång från en lista av krav som behövde vara uppfyllda. Därefter analyserades det aktuella konceptet och brister som upptäcktes eliminerades. Cykeln med analys och åtgärd av brister fortsatte tills spelkonceptet var stabilt.

4.1.1 Inledande kravidentifiering

Utifrån Touch AT!s syfte, målgruppens speciella krav, undersökningen av befintliga spel samt grundkonceptet för projektet, det vill säga att flera enheter ska utgöra en spelplan, kunde ett antal krav på spelet fastställas:

- **Samarbetsfrämjande**
Spelet ska främja, eller rent utav kräva, samarbete av spelarna. Detta för att spelet ska kunna vara ett hjälpmedel för barnen, vad gäller träning av deras sociala och emotionella förmåga. Ett steg i rätt riktning är att låta enheterna ligga på ett bord så att spelarna kan sitta runt om och enkelt kunna ha kontakt med varandra samtidigt som de får en bra överblick över spelplanen.
- **Vy ovanifrån**
För att ge alla spelare en jämlik upplevelse av spelet, oberoende av var någonstans kring spelplanen spelaren sitter, är det viktigt att spelet visas ovanifrån samt att knappar, information och dylikt kan tydas från alla håll. Spelet ska ge samma intryck och spelkänsla oberoende på viken sida spelaren ser spelplanen ifrån.
- **Spelplan**
Då spelplanen ska byggas upp av flera enheter som samverkar, kan dess form sättas samman på olika sätt snarare än att alltid vara likadan. Det ska utnyttjas för att spelet ska kunna ge en unik spelupplevelse samt addera en extra dimension av samarbete när planen faktiskt ska uppföras.
- **Problemlösning**
För att kräva samarbete från spelarna behövs ett problem som måste lösas. Det ska vara ett problem som uppmuntrar fundering och diskussion och som innefattar alla aktiva spelare.

4.1.2 Fastställande av spelkoncept utifrån krav

Utifrån de krav som fastställdes bestämdes att spelet skulle vara ett slags logikspel, då ett problem att lösa kan främja diskussion och samarbete. Flera sorters speltyper presenterades och analyserades. En tungt vägande faktor i val av speltyp var att spelet skulle fungera med en vy från ovan.

Utmaningen i spelet är till viss del inspirerat av brädspelet Ricochet Robots [33] och de isbanor som finns i Pokémon-spelen [34], där karaktärerna behöver ta en viss väg genom en yta med hinder. En fördel med ett spelupplägg inspirerat av de spelen är att de båda är beprövade och fungerar mycket bra med en vy ovanifrån.

De ovan nämnda spelen har en statisk bana och ett rörelsemönster hos karaktärerna som enbart går i fyra riktningar. Det ansågs inte vara interaktivt nog eller utnyttja plattformen nog för CirKva. För att ge fler möjligheter till olika konstellationer av nivåer och öka interaktiviteten i spelet beslutades att hindren i CirKva skulle vara interaktiva. Med interaktiva menas att hindren kan påverkas av vad karaktärerna

gör på spelplanen. En ytterligare fördel med interaktiva hinder var att möjligheten till att kravet på samarbete för att klara spelet ökade markant, då ett hinder kan påverkas från en annan enhet än den hindret är på.

Det grundkoncept som vidare utveckling utgått från var ett logiskt spel där ett antal karaktärer skulle ta sig till specifika mål, i form av knappar. I vägen för karaktärerna fanns väggar som öppnas med hjälp av knapparna. Även karaktärernas placering på spelplanen var ett hinder som behövde överkommas. Karaktärerna behöver nämligen flyttas mellan enheter för att ta sig till sitt mål.

4.2 Framtagning av spelets grafiska gränssnitt

Även spelets gränssnitt togs fram iterativt och lutade sig starkt mot riktlinjerna för gränssnitt i 2.1. För varje iteration analyserades den aktuella designen och problemområden rättades till.

Utifrån den information som framkom i kapitel 2.1 visade det sig att en del av målgruppen kan bli distraherad av allt för detaljrika figurer. Istället för att försöka hitta ett upplägg med god balansgång mellan detaljrikedom och tydlighet gick valet mot en minimalistisk design. I CirKva representeras en banas delar av cirklar, rektanglar, kvadrater och trianglar i olika färger.

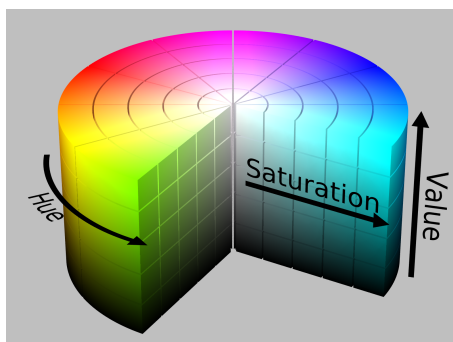
En ytterligare fördel med att ha en minimalistisk design är att de karaktärer som finns i spelet inte har några individuella karaktärsdrag utöver färgen. Då spelet är upplagt på ett sådant vis att ingen spelare äger en speciell karaktär, är förhoppningen att karaktärernas avsaknad av individualitet gör det enklare för spelarna att samsas om karaktärerna. Om karaktärerna till exempel hade kunnat väljas av spelarna vid start av spelet hade det kunnat göra att barnen haft svårare för att skicka iväg figuren till en annan enhet, då de hade kunnat anse att det var deras figur och velat fortsätta styra den.

För att spelet ändå ska se bra ut, och ge ett behagligt intryck, har mer fokus lagts på färgvalet. På grund av de riktlinjer som framkom i den inledande efterforskningen består färgskalan av färger med låg mättnad och medel till hög intensitet. Spelets grundpalett togs fram med hjälp av verktyget Paletton [35] med fastställda riktlinjer och tillgänglighet i beaktning. Paletton har en funktion för att simulera olika sorters färgblindhet, och paletten som togs fram ger synliga skillnader vid alla de sorters färgblindhet som simuleras, se figur 4.1.



Figur 4.1: Grundpalett för CriKva. Genererad med paletton [35].

Med mättnad kan mängden pigment i färgen jämföras, där hög mättnad innebär att det är mycket färg, och låg mättnad ger en gråare ton till färgen. Intensiteten är ljusheten på färgen, där hög intensitet ger en ljusare färg och en låg intensitet en mörkare färg. Låg mättnad och hög intensitet resulterar i färger som drar åt pastell, se figur 4.2.



Figur 4.2: En cylinder som visar färgrelationer i nyans-mättnad-intensitet (eng: Hue-Saturation-Value) modellen. Från [36] CC-BY-SA.

4.2.1 Navigering av spelet

Spelets menyer och navigering är även de minimalistiska till sin utformning, för att skapa enhetlighet i spelet. I enlighet med riktlinjerna används ett sans serif-typsnitt, med mörk text på ljus bakgrund, för att texten i menyerna ska vara lätt att läsa.

En diskussion som uppstod vid framtagningen av menyer handlade om på vilket sätt som sammankopplingen av enheterna skulle hanteras. Som det nämns i avsnitt 3.3 finns det två olika sätt att koppla ihop enheterna med det ramverk som används. De olika sammankopplingsätten utmynnar i två olika varianter av navigering. Det ena sättet ger en variant där en enhet sänder att den har ett spel redo, för att andra enheter sedan ska kunna ansluta till den sändande enheten. Det andra sättet resulterar i att en enhet agerar inbjudare till de andra enheterna som sedan får acceptera inbjudan.

I CirKva används den senare varianten, då den ger större kontroll över vilka enheter som ansluts till nätverket, samt minimerar risken att enheter som ej önskas finnas i nätverket hamnar där. I en miljö med många enheter på samma ställe, så som i en skolmiljö, kan det vara fördelaktigt att ha den kontrollen då det blir enkelt att koppla ihop en bestämd grupp enheter.

Ett ytterligare problem kring navigeringen i spelet, som behövde lösas, var utplaceringen av de olika enheterna så att banorna fick rätt orientering. Under spelets utveckling diskuterades flera lösningar. Först utforskades det ifall det fanns någon teknologi på enheten som skulle kunna indikera hur enheterna låg i relation till varandra. Det konstaterades dock snart att så inte var fallet. Att per automatik kunna kontrollera hur enheterna låg var således inte möjligt. Detta innebär att det

är upp till användarna att lägga sina enheter på ett korrekt vis och att låta enheterna ligga på sin plats tills dess att en nivå är avklarad.

I huvudsak utforskades två huvudkoncept för utplacering av enheterna. Det första var att använda sig av en orienteringsvy innan en spelomgång startas, och den andra var att indikera hur enheterna skulle vara placerade med hjälp av banornas utseende. På grund av den redan långa väntetiden från det att spelet startas tills det att spelarna kan börja spela ansågs det inte vara försvarbart med en orienteringsvy. Istället valdes lösningen att indikera via banans utseende hur enheterna skulle placeras.

4.2.2 Karaktärers rörelsemönster

Ett ytterligare problemområde som uppstod vid utvecklingen av spelet var hur karaktärerna skulle förflyttas. Det här problemet var tvådelat och berörde dels hur karaktärerna skulle förflyttas på en enhet, dels hur karaktärerna skulle förflyttas mellan enheter.

Karaktärernas rörelsemönster och hur användaren interagerar med karaktärerna genomgick flera olika stadier under utvecklingens gång. Många olika varianter diskuterades, allt i från att styra karaktärer med pilar till att skicka iväg dem genom ett flick med fingret. Till slut landade valet på att dra karaktärer längs banan. Det slutliga valet av interaktionssätt för karaktärerna grundade sig i minimering av stressfulla moment och maximalt utnyttjande av det medium som iPads utgör.

För att förflytta karaktärer mellan enheter diskuterades framför allt tre metoder. Den första var att dra karaktären över kanterna på de olika enheterna, den andra var att skicka karaktärerna över enheterna med en flick med fingret och den tredje metoden var att skicka karaktärerna med portaler. Valet föll på den tredje metoden, att skicka karaktärer med hjälp av portaler. Beslutet grundar sig i att det minimerade spelarnas interaktion med enheter som de inte är menade att agera med, samt tydlighet i vart karaktärerna tar vägen i otydliga korsningar.

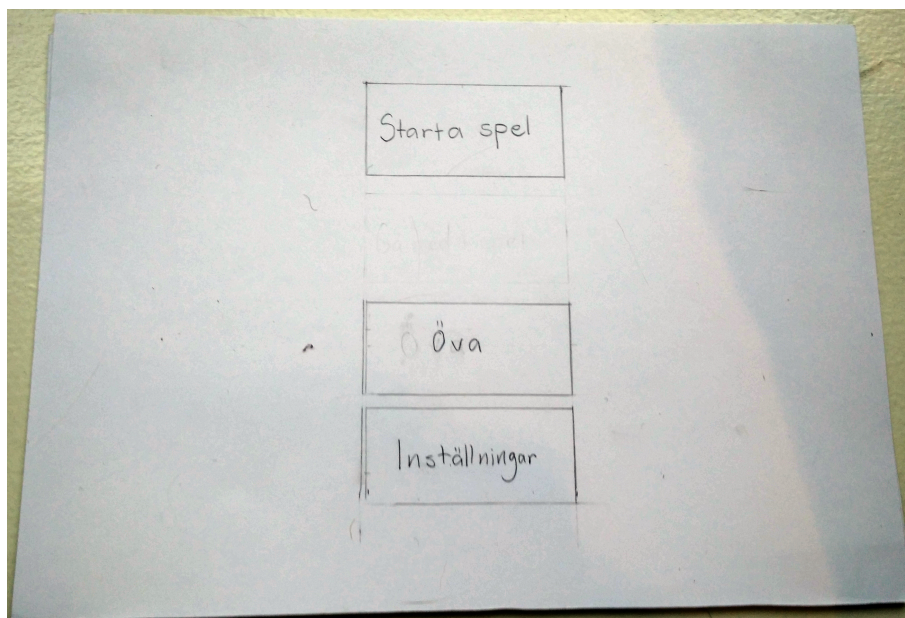
4.3 Pappersprototyp

För att skaffa förståelse för hur andra upplevde spelet innan det implementerades, utvecklades flera olika pappersprototyper. En pappersprototyp är en representation av en applikation, gjord av papper, där en person, här kallad spelare, agerar dator istället för en faktiskt dator.

När ett stabilt koncept till spelet hade tagits fram överfördes detta till en enkel pappersprototyp. Efter att den testats inom projektgruppen och konstaterats fungera skapligt bestämdes att den mer robusta pappersprototypen kunde ha ett snarlikt upplägg. Pappersprototypen togs fram i ett tidigt skede och innehåller därför inte alla de element och funktioner som diskuteras tidigare i det här kapitlet.

4. Utvecklingen av CirKva

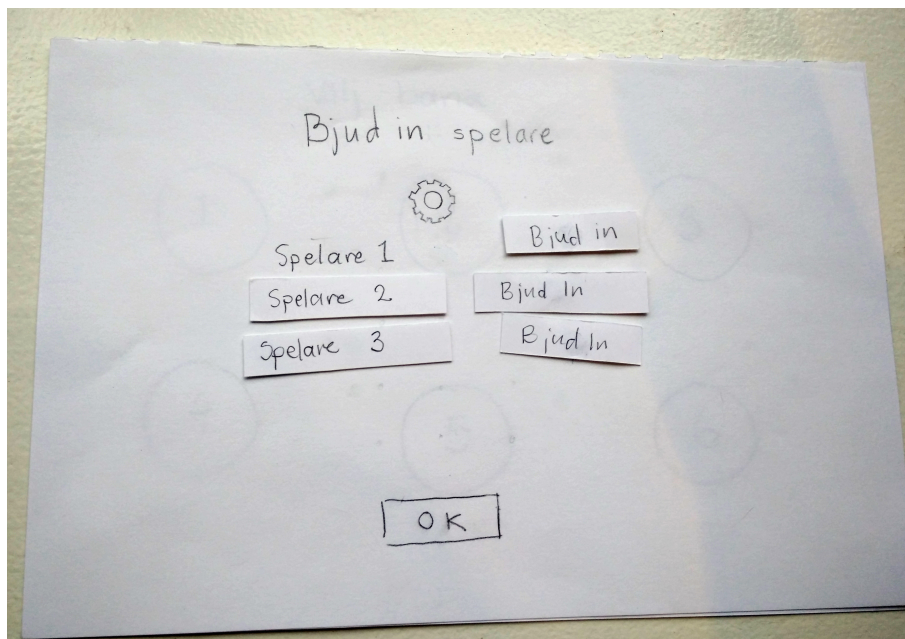
Den första vyn i papperprototypen var menyvyn. Som kan ses i figur 4.3 är menyn mycket enkel till sin utformning. I ett tidigt skede fanns även knappen “Gå med i spel”, då tanken till en början var att en av spelarna startade spelet, för att de andra senare skulle gå med i spelet. Denna knapp togs sedan bort då metoden att bjuda in spelare till ett spel valdes.



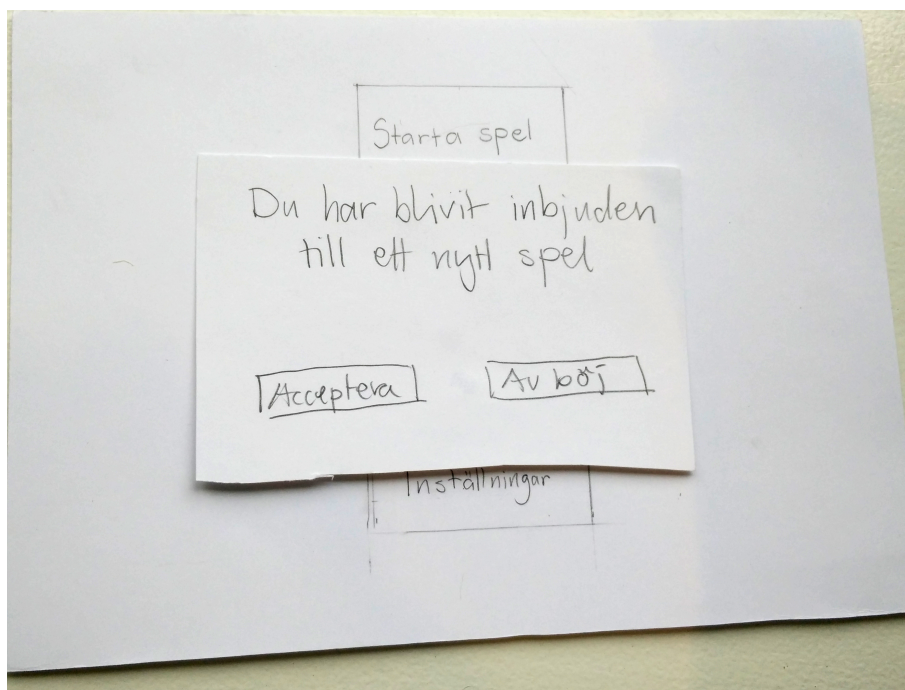
Figur 4.3: Pappersprototypens menyvy. Författarens egen bild.

Vyn för att bjuda in spelare har ett variabelt antal spelare, och knapparna där det står “Bjud in” ändras under spelets gång till att visa spelarens status, det vill säga huruvida spelaren accepterat inbjudan eller ej, se figur 4.4. Den spelare som blir inbjuden får en mycket enkel dialogruta på sin skärm där spelaren kan välja att antingen acceptera eller avböja inbjudan, se figur 4.5.

4. Utvecklingen av CirKva



Figur 4.4: Pappersprototypens inbjudningsvy. Författarens egen bild.



Figur 4.5: Pappersprototypens vy för att acceptera en inbjudan. Författarens egen bild.

4. Utvecklingen av CirKva

För att välja nivå klickar spelaren på en av de olika cirklarna. Vyn för att välja nivå, som visas i figur 4.6, visas bara på den enhet som bjudit in de andra spelarna.



Figur 4.6: Pappersprototypens vy för att välja en nivå. Författarens egen bild.

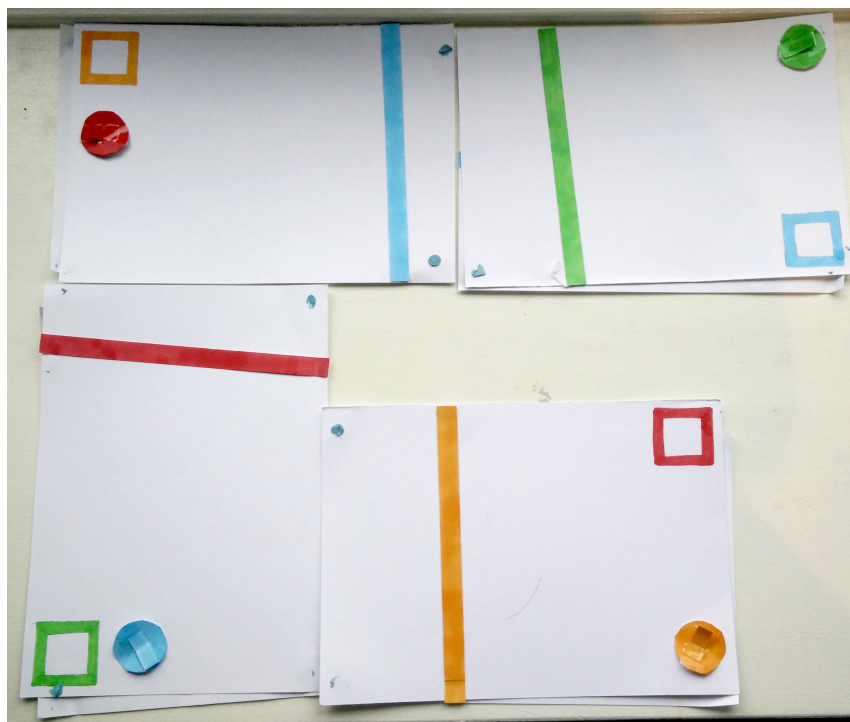
För att orientera de olika enheterna korrekt visas sedan en orienteringsvy, där likafärgade streck ska matchas ihop, se figur 4.7.



Figur 4.7: Pappersprototypens orienteringsvy. Författarens egen bild.

4. Utvecklingen av CirKva

När enheterna väl är ihoppusslade tas orienteringsvyn bort. Då kunde spelledaren lägga ut spelpjäserna och användarna börja spela. Väggarna är fästa med häftmassa för att lätt kunna öppnas och stängas. Figur 4.8 visar den första nivån som togs fram. Den är enkel och innehåller endast de grundläggande koncepten i spelet. Figur 4.9 visar istället en lite mer komplicerad nivå med fler spelmoment. På nivån i figur 4.9 finns det gömda föremål som ska användas till att bygga en bro över vattnet. Det går inte att lösa nivån utan bron.



Figur 4.8: Den första, enkla, nivån som utvecklades. Författarens egen bild.



Figur 4.9: En senare, lite svårare, nivå som utvecklades. Författarens egen bild.

4.4 Evaluering av prototyp

För att testa spelidén gjordes tester på studenter vid Chalmers Tekniska Högskola. De var mellan 20 och 25 år och hade alla stor datorvana. Då det skulle undersökas om spelet var samarbetsfrämjande observerades endast testgruppen under testningen, vilket sedan följdes upp av en semistrukturerad intervju. Resultatet från testningen och intervjun gav underlag för vad som behövdes förtydligas till implementeringen av spelet.

4.4.1 Evalueringsmetod

Innan testerna fick testgruppen information om vad som skulle ske och hur spelet fungerar, dock var denna information mycket begränsad då det skulle undersökas om spelet verkligen var samarbetsfrämjande och hur mycket initiativ spelarna tog. Den information som användarna mottog lyder enligt följande:

- Samtliga spelare ska tänka högt, och säga precis vad de gör i varje steg.
- Spelet sträcker sig över flera plattor.
- Varje spelare har kontroll över en platta.
- Vissa banor kan innehålla "nycklar" gömda under föremål. Nycklarna behövs för att kunna låsa upp delar av banan i fråga.
- Målet med spelet: Alla måste nå den knapp vars färg är densamma som karaktären.
- Även om en spelare står i mål, kan den flytta på sig om det behövs.
- Hela processen kommer att spelas in, och det kommer endast att användas av i evalueringssyfte.
- Förklaring till vad som är karaktärer och hur dessa kan flyttas.
- Knapparnas funktionalitet.

Själva testet bestod av två delar: en där spelarna skulle ansluta samtliga enheter till varandra, och en där de spelade själva spelet. För information om hur enheter ansluts till varandra, se avsnitt 3.3.

Under processen observerades främst spelarnas utförande av uppgifterna, deras ansiktsuttryck samt om något var problematiskt. Dessutom observerades samarbetet mellan spelarna. Det hela antecknades in i minsta detalj för att samtlig information skulle bli dokumenterad. Under spelets gång fick ingen av spelarna hjälpa spelarna eller avbryta dem på något vis, då ingen yttre påverkan skulle existera. Spelarna skulle tillsammans hjälpas åt för att lösa uppgifterna, men om de fastnade under en längre period, var det tillåtet att ge spelarna en liten ledtråd om hur de skulle gå vidare.

Den evalueringsmetod som användes vid testning av prototyperna kallas för Observation i laboratorium. Testerna gjordes i naturliga, något kontrollerade miljöer, så som i grupprum på Chalmers [37]. Under observationen användes en metod vid namn Tänka högt-tekniken. Det är en teknik där användarna pratar högt under hela

processen, gällande vad de gör samt hur de gör något i gränssnittet. Det möjliggör för observatörer att veta vad användarna tänker och varför de utför en uppgift på ett visst sätt [37].

I kombination med att observera användarna när de utför uppgifter i gränssnittet, gjordes även en intervju. Intervjumetoden som användes kallas för semistrukturerad intervju, vilket innebär att om intervjuaren kom på en följdfråga under intervjun hade denne möjlighet att ställa den [37]. I intervjun ställdes både slutna ja- och nejfrågor och mer öppna frågor där användarna fick möjlighet att svara mer fritt. Anledningen till intervjun var att få mer information om personerna samt deras upplevelser och åsikter kring spelet. Dessutom gav intervjun personerna en chans att ge sina samtliga åsikter, vilket är lätt att gå miste om under själva evalueringsprocessen. Frågorna som ställdes presenteras i bilaga A.

4.4.2 Resultat av evaluering av prototyper

Generellt sett tyckte testpersonerna att spelet var roligt och hade en lagom svårighetsgrad. Dock var det vissa moment som gjorde dem fundersamma. Bland annat tyckte de inte att det framgick att enheterna skulle ligga på samma plats under en hel nivå, utan de trodde att de skulle flytta själva enheterna för att uppfylla målet. De tyckte även att det i början var otydligt vad som var en karaktär och vad som var en vägg. De önskade alla att det hade funnits någon typ av genomgång av hur spelet fungerade innan de börjat spela.

Även om de tyckte att spelet var roligt så hade de svårt för att endast hålla sig till en enhet. Det var många som sträckte sig och spelade på en annan, vilket inte främjade samarbetet särskilt mycket. De önskade att det istället funnits portaler som skickade karaktärerna mellan enheterna då det hade lett till att enheterna inte hade behövt ligga ner på bordet, utan alla hade kunnat hålla i sin enhet som de velat. De ansåg att det hade ökat samarbetet då de hade varit tvungna att prata mer med varandra eftersom de inte hade kunnat se hur de andra enheterna såg ut. All denna respons togs sedan hänsyn till när implementeringen av spelet skedde.

4.5 Metod vid evaluering av slutlig design

För att undersöka om spelet var anpassat för spelets målgrupp gjordes tester på barn med kognitiva funktionsnedsättningar (se avsnitt 2.1). Även de fick information om vad som skulle ske och vad spelet gick ut på. För att testningen skulle flyta på bra och för att barnen skulle kunna fokusera på att lösa uppgiften, mottog de ganska mycket information om hur man spelade:

- Förklaring till vad som är karaktärer och hur dessa flyttas.
- Knapparnas funktionalitet.
- Att man har gått i mål när alla spelare står på en knapp med samma färg som spelaren.
- Att hinder öppnas genom att valfri spelare står på knapp i samma färg som hindret.
- Även om en spelare står i mål, kan den flytta på sig om det behövs.
- Förklaring om vad som är portaler och hur dessa används.
- Att varje spelare endast ska kontrollera en iPad.

Den evalueringsmetod som användes vid testning av slutprodukten kallas för Observation i fält. Orsaken till att denna metod valdes, beror på att testerna gjordes på projektets målgrupp, i deras naturliga miljö [37]. Testerna gjordes på Hovåskolan, beläget i södra i Göteborg.

Även under evalueringen av den slutliga designen var det tänkt att Tänka högt-tekniken skulle användas. Efter att ha konsulterat med Peter Börjesson, från Touch AT!, bestämdes att den metoden inte skulle tillämpas och att barnen enbart skulle observeras. Det beslutet härstammar från att många av barnen har svårt för att kommunicera, vilket hade gjort det svårt för dem att koncentrera sig på sina uppgifter ifall de även skulle behöva formulera hur de tänker under spelomgången. Evalueringen av den slutliga designen utfördes på de surfplattor som projektets målgrupp hade tillgång till, det vill säga iPads.

Även under denna evalueringen gjordes en intervju efter testningen och även här användes intervjumetoden semistrukturerad intervju, dock med andra frågor anpassade för barnen, se bilaga B.

4.6 Systemmodellering

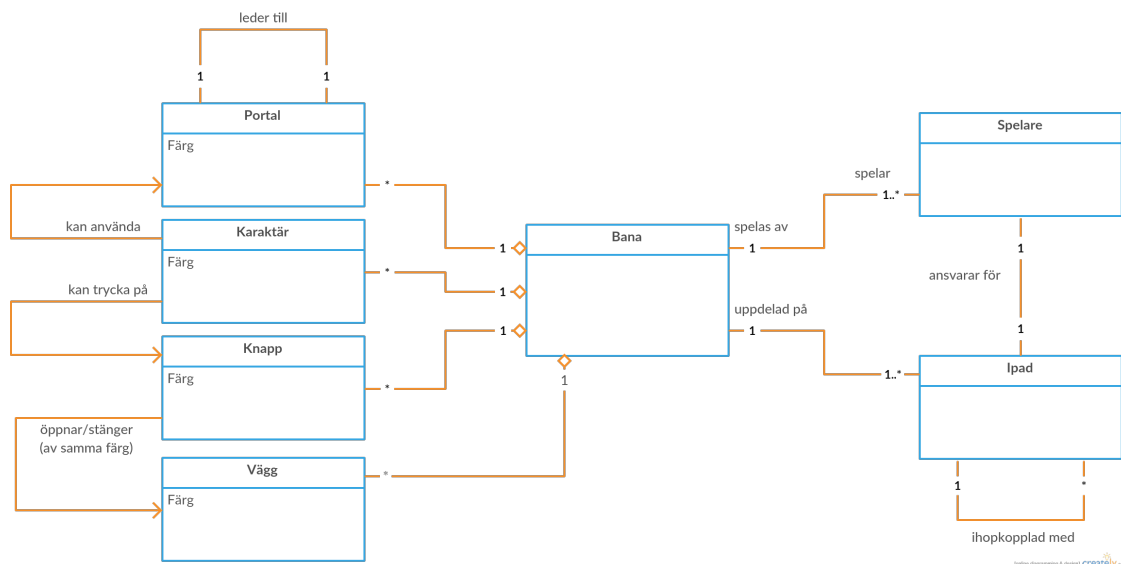
I takt med att evalueringar av prototyperna utfördes, togs även olika krav på systemet fram, sådant som var grundläggande för ett fungerande spel med ett bra gränssnitt. Först började olika kravspecifikationer tas fram, vilket representerar vad systemet ska kunna prestera. Kravspecifikationerna består av funktionella och icke-funktionella krav. De funktionella kraven handlar om vad systemet ska kunna göra, medan de icke-funktionella kraven handlar om hur systemet faktiskt ska vara, se bilaga D.

4. Utvecklingen av CirKva

För att kunna utveckla de olika delarna av spelet, behövde vissa kravspecifikationer omvandlas till användningsfall. Ett användningsfall representerar de uppgifter en användare ska kunna utföra på spelets gränssnitt, ordningen på uppgifterna, vilka fel användaren eventuellt skulle kunna göra samt hur dessa åtgärdas. Genom att ta fram användningsfall för spelet, så blir det inte bara mer verklighetsbaserat, utan det gör underlättat även de olika delarna av implementeringen, se bilaga E.

Parallellt med det togs även en domänmodell fram, som både skulle hjälpa implementationsprocessen men även skapa en förståelse för vilka beståndsdelar systemet är uppbyggt av och hur olika delar av spelet interagerar med varandra. Framtagningen av domänmodellen var en iterativ process och diagrammet växte fram allt eftersom spelkonceptet växte fram. Figur 4.10 visar hur diagrammet slutligen kom att se ut. I den figuren ser vi att:

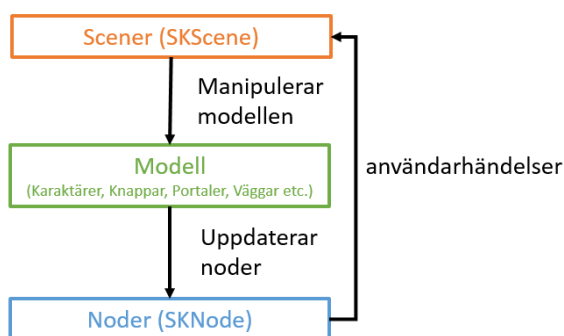
- En bana kan ha många portaler, karaktärer, knappar och väggar.
- Varje portal, karaktär, knapp och vägg har en färg.
- Varje portal leder till en annan portal.
- Karaktärer kan använda portaler.
- Karaktärer kan trycka på knappar.
- Knappar öppnar/stänger väggar av samma färg.
- En bana kan spelas av en till flera spelare.
- En bana är uppdelad på en till flera iPads.
- En ipad kan vara ihopkopplade med noll till flera iPads.
- En spelare ansvarar för en ipad.



Figur 4.10: Ett UML-diagram av den slutliga domänmodellen för CirKva.
Författarens egen bild.

4.7 Mjukvaruarkitektur

CirKva följer den traditionella implementeringen av MVC, det vill säga där en kontroller tar emot indata från en användare och manipulerar modellen utifrån den datan. Sedan ansvarar modellen för att uppdatera vyn. I CirKva är det scenerna från SpriteKit (SKScenes) som agerar kontroller och noderna från SpriteKit (SKNodes) som agerar vyer. Modellerna är rena data-klasser som härletts utifrån domänmodellen som togs fram tidigare i projektet. Figur 4.11 visar hur MVC-mönstret fungerar i CirKva.



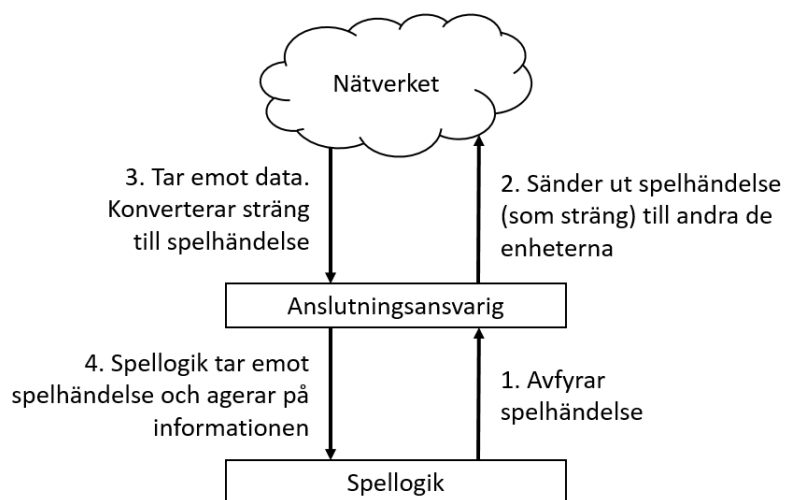
Figur 4.11: Model View Controller-mönstret i CirKva. Författarens egen bild.

Kommunikation mellan enheter

CirKva har ett kommunikationsprotokoll där enheterna kommunicerar via ett antal fördefinierade spelhändelser (Game Events) som spelet kan agera på. Det finns till exempel en speciell händelse för att skicka en karaktär till en portal.

Implementationsmässigt är spelhändelserna en uppräkningsstyp (Enum). Swift tillåter uppräkningsstyper att ha instansmetoder för utökad funktionalitet [38]. CirKva använder sig av Swifts flexibla uppräkningsstyper genom att låta spelhändelser ha funktionaliteten att konverteras till och från strängar för att enkelt kunna skickas mellan enheter över nätverket.

Som nämnts tidigare, så använder CirKva ramverket Multipeer Connectivity Framework för att sköta kommunikation över nätverket. Ramverket består av en rad olika klasser och protokoll, mer information om dessa klasser och protokoll presenteras i bilaga C. I CirKva har ramverkets funktionalitet inkapsulerats i en klass, kallad Anslutningsansvarig (ConnectivityManager), som följer designmönstret Fasad (Facade pattern) [39]. Olika enheter pratar med varandra genom varsin instans av Anslutningsansvarig. Varje Anslutningsansvarig lyssnar efter nya spelhändelser på sin enhet och skickar sedan vidare dem till de andra enheterna som bestämmer hur de ska agera. Som nämnts tidigare så kan kommunikationen antingen vara pålitlig eller icke pålitlig. CirKva använder sig av pålitlig kommunikation eftersom det är viktigt att varje spelhändelse når varje enhet.



Figur 4.12: Kommunikationsflöde mellan enheter i CirKva. Författarens egen bild.

5

Resultat

Målet med projektet var att skapa ett spel, anpassat för barn med kognitiv funktionsnedsättning, som sträcker sig över multipla surfplattor (se avsnitt 1.1). För att uppnå målet med att anpassa spelet till målgruppen, gjordes spelet så generellt som möjligt, för att på så sätt omfatta så många typer av funktionsnedsättningar som möjligt. Det har medfört att spelet inte är begränsad, utan att den kan användas i samtliga skolor för barn med kognitiv funktionsnedsättning.

5.1 Slutlig produkt

I avsnitt 1.1 presenterades de två delmål som sattes upp gällande den slutliga produkten. Det ena delmålet var att implementera ett spel som sträcker sig över flera kommunicerande enheter. I detta avsnitt visas resultatet av spelet projektet genererade. Spelet kommer förklaras och stegvis gås igenom för att ge en tydlig bild av vilken funktionalitet det besitter.

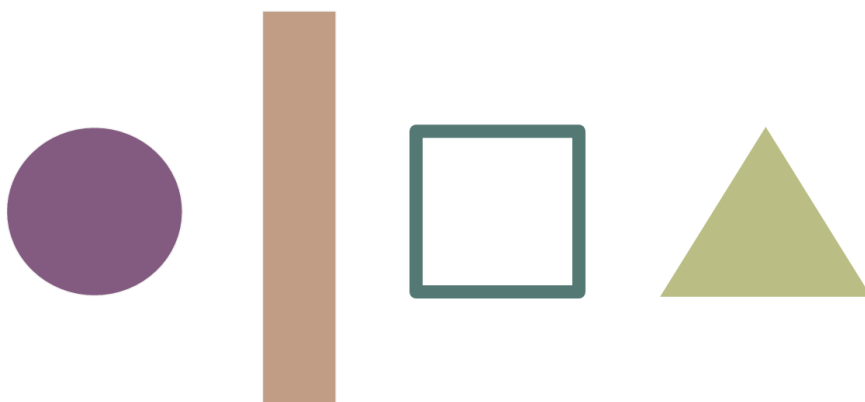
5.1.1 Spelkoncept

CirKva är ett logikspel som sträcker sig över flera iPads. Spelet innehåller karaktärer, knappar, väggar och portaler och går ut på att klara av nivåer genom att förflytta samtliga karaktärer till knappar i motsvarande färg. Förflyttningen sker av att en spelare drar en karaktär med fingret. För att klara av en nivå måste karaktärerna passera hinder i form av väggar och skickas mellan enheter med hjälp av portaler. En vägg försvinner när godtycklig karaktär står på en knapp i väggens färg. Tanken är att alla enheter ska ligga bredvid varandra på ett bord för att bilda en stor spelplan och att varje spelare endast kontrollerar en enhet. Således måste spelarna kommunicera och samarbeta med övriga spelare för att klara av en nivå.

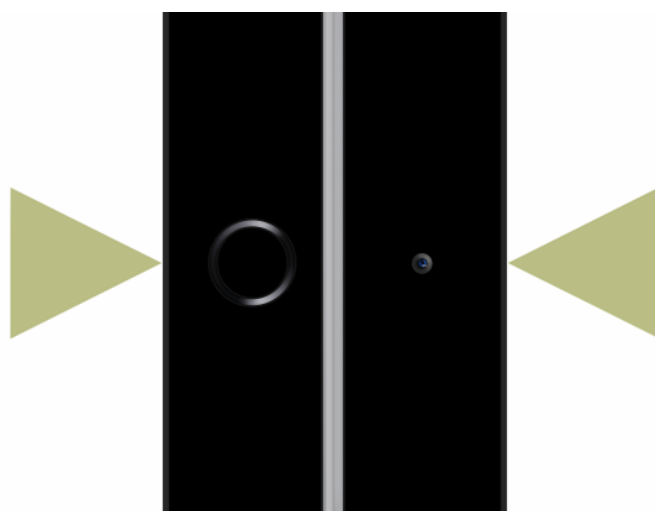
5.1.2 Design

Designen och navigeringen i CirKva är, till sin utformning, väldigt minimalistisk. Karaktärer utgörs av cirklar, knappar av ihåliga kvadrater, väggar av avlånga rektanglar och portaler av trianglar, se figur 5.1. Portalerna är placerade i kanterna av enheternas spelplan för att, utöver sin funktion som portal, indikera hur de olika

enheterna ska placeras ut. Kanten på en enhet där en portal finns ska läggas bredvid en kant med en matchande portal, se figur 5.2.



Figur 5.1: En karaktär, en knapp, en vägg och en portal. Författarens egen bild.



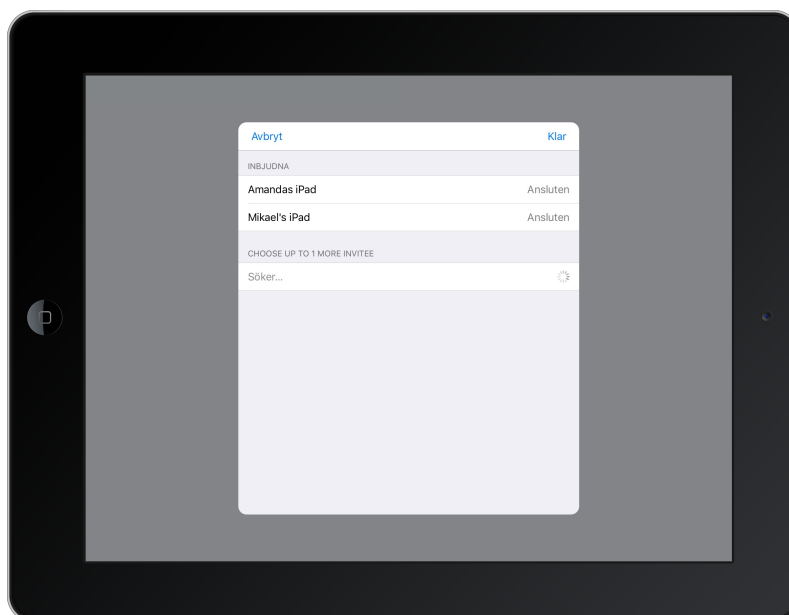
Figur 5.2: Portaler visar hur enheterna ska placeras. Författarens egen bild.

5.1.3 Genomgång av spelomgång

När spelet startas synliggörs så snabbt som möjligt skärmen som visar menyn, se figur 5.3. I menyn syns spelets logotyp och under den, knapparna “Skapa nytt spel” och “Träna själv”. Trycker spelaren på den förstnämnda öppnas ett fönster, se figur 5.4, där 1 till 3 enheter kan bjudas in till ett nytt spel.

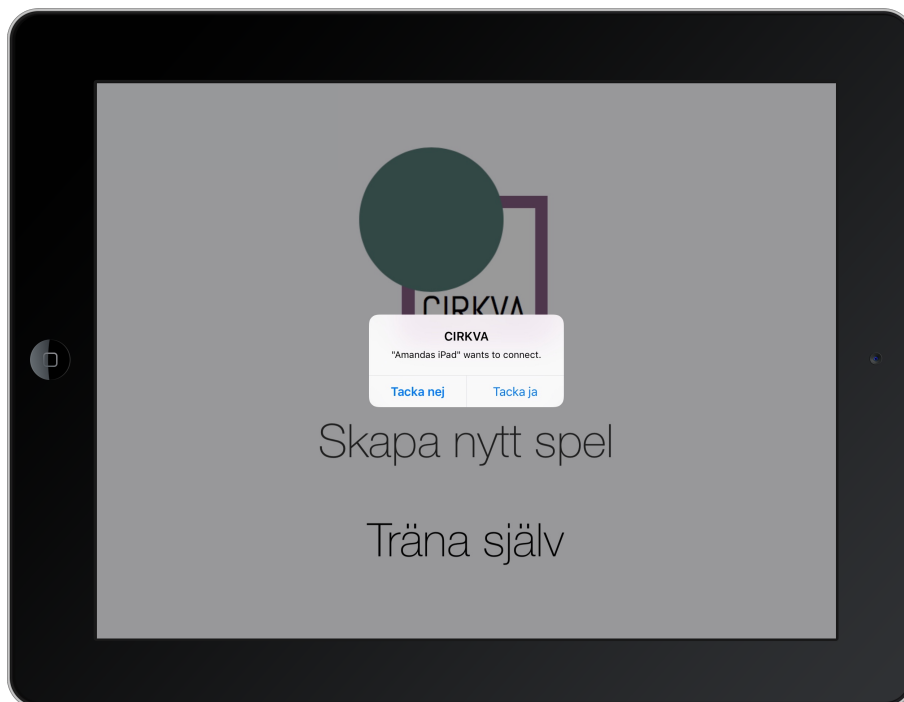


Figur 5.3: Spelets startmeny. Författarens egen bild.



Figur 5.4: Skärm där man bjuder in andra enheter till att spela. Författarens egen bild.

Väljer spelaren att bjuda in en enhet som har spelet aktivt så visas inbjudan i figur 5.5 på den enheten. När önskat antal enheter är anslutna till sessionen trycker spelaren på “Klar” för att komma till nästa skärm, i vilken man väljer nivå, som visas i figur 5.6.



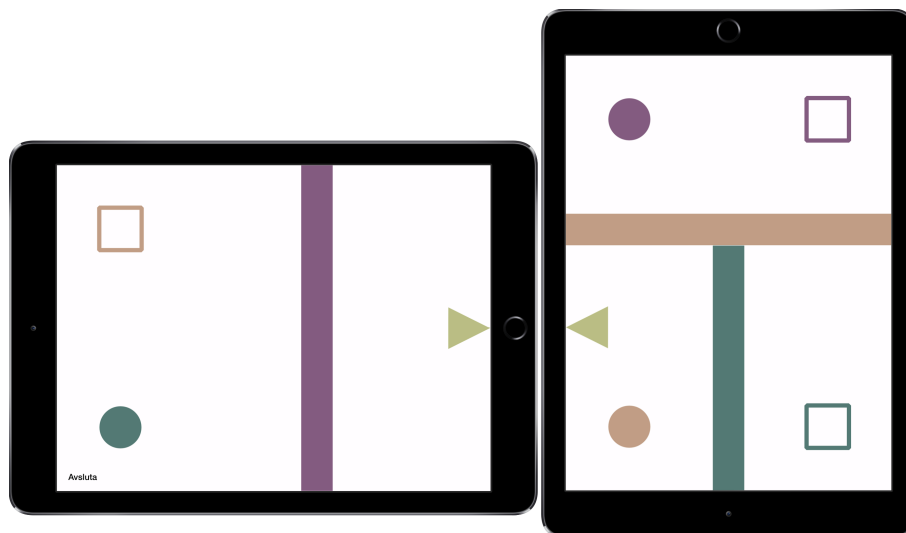
Figur 5.5: En dialogruta visas när inbjudan tas emot. Författarens egen bild.



Figur 5.6: Skärmen som visas när nivå ska väljas. Författarens egen bild.

Efter det väljs nivå som sedermera startas och då är det dags att placera ut enheterna på bordet. En nivå för två spelare visas i figur 5.7. När enheterna är på plats kan spelarna börja dra runt karaktärerna för att lösa det logiska problem en nivå utgör.

För att antyda att en karaktär har nått sitt mål kommer knappen den står på rotera 45 grader, se figur 5.8. En av spelarna kan också avsluta spelet genom att trycka på knappen "Avsluta". När en nivå avslutas eller klaras av återgår samtliga enheter till startmenyn.



Figur 5.7: En nivå för två spelare. Författarens egen bild.

En spelare kan lära sig hur spelet fungerar i ett enspelarläge som nås via knappen "Träna själv" i startmenyn, se figur 5.3. Enspelarläget består av fyra nivåer där spelets samtliga spelelement kan utforskas. Förhoppningsvis kan spelaren känna större social trygghet i flerspelarläget om den redan känner till spelets alla delar.



Figur 5.8: En karaktär som nått sitt mål.
Författarens egen bild.

5.2 Evaluering av slutlig design

I denna del av rapporten presenteras resultat från evalueringarna av den slutliga designen. Det utfördes evalueringar med tre olika grupper. För varje evaluering som gjordes presenteras först lite information om användarna som gjorde testerna och sedan presenteras deras tankar och upplevelser kring spelprototyperna.

Testförfarandet behövde modifieras något utifrån det tilltänkta utförandet för att passa målgruppen bättre. Speciellt då det kom till den information testgrupperna fick ta del av innan och under spelets gång. Barnen behövde tydliga instruktioner, speciellt innan, men ibland även under spelomgången.

5.2.1 Grupp 1

I den första gruppen fanns tre barn i nedre tonåren. För att fylla upp för att få fyra spelare var även Peter Börjesson med och spelade spelet. Barnen i den här gruppen hade relativt lite problem med att kommunicera. Barnen förstod vad spelet gick ut på och alla klarade övningsläget utan några större problem. De förstod hur alla element fungerade och hade inga problem med att kontrollera dem. Det tog dock olika lång tid för barnen att ta till sig hur spelet gick till.

Det var inte intuitivt för barnen att portalerna indikerade hur spelplanen skulle vara utlagd. De la plattorna i tänkt formation först efter att ha blivit instruerade om det, men när de väl byggt ihop spelplanen tyckte de att det gjorde det lättare att spela. Vid ett tillfälle uppstod förvirring kring en portal. En karaktär hade av misstag skickats igenom en portal och skulle skickas tillbaka igen. Ett av barnen försökte då att bara flytta karaktären inom portalområdet och förväntade sig att portalen skulle flytta över karaktären igen. Då portalen inte går att aktivera igen förrän karaktären har flyttats bort från den helt orsakade beteendet förvirring hos barnet. Efter vidare förklaring hur portalerna fungerade kunde barnet flytta karaktären dit hen ville.

Vid sammankopplingen för fyra spelare gick det över lag bra, men behövde göras några gånger. Vid första tillfället startade ett barn spelet då alla inbjudningar hade skickats och accepterats, men innan alla enheter hade hunnit anslutas. Vid efterföljande försök skickades och accepterades inbjudningar snabbt efter varandra, och enheterna kopplades inte ihop korrekt.

Vid spelomgången för fyra spelare var det ett av barnen som tänkte mycket snabbare än de andra. Hen sträckte sig ofta över till de andra spelarnas enheter, men försökte att undvika det efter tillsägelser. Ett av barnen som tog längre på sig att tänka blev märkbart frustrerad när någon annan tog över för hen.

Efter att sista spelomgången var avklarad utfördes en intervju (för mer detaljerad kring frågorna som ställdes, se bilaga B). I korthet tyckte barnen att det var ett roligt spel. Ett av barnen tyckte dessutom att det var så roligt att hen fortsatte att spela själv under intervjun. Generellt tyckte barnen att samarbetet hade gått bra, även om det var lite jobbigt när någon sträckte sig till övriga spelares enheter. De uppfattade att den som sträckte sig ville hjälpa, men att det skulle varit bättre om personen frågade om lov innan.

När det kommer till förbättringar tyckte de att det hade varit bra om det gick att få instruktioner på hur man skulle spela under spelomgången, då de ibland hade svårt att komma ihåg hur man skulle utföra ett moment i spelet. De tyckte även att det hade varit bra med fler banor, där några var svårare.

5.2.2 Grupp 2

I den andra gruppen var det tre barn som var yngre än de i den första gruppen. Även här var Peter Börjesson med för att fylla upp de fyra platserna. De tre barnen var otåliga på att få börja spela och hade svårt att ta till sig instruktionerna. De ville gärna skynda sig igenom träna själv-läget och sammankopplingen. Otåligheten ledde till att det blev mycket tekniskt strul under sammankopplingen. De ville alla bjuda in samtidigt, och alla klickade på acceptera så fort en inbjudan kom upp på skärmen. De klickade även på "Klar" trots att alla enheter inte var anslutna än. Att anslutningen gick dåligt och att momentet behövdes göras om flera gånger för att det skulle ske korrekt orsakade frustration och barnen ville inte lyssna på varandra eller observatörerna.

Under spelomgångarna tänkte två av spelarna mycket snabbare än den tredje. En av de snabbare spelarna var mycket dominant och tog över spelomgången och spelade åt de andra. De som inte fick spela blev frustrerade. Barnet som tog över ville inte lyssna på att hålla sig på en platta.

Under tvåspelarläget förstod ett par av spelarna att det gick att pussla ihop spelplanen till en större spelplan. Under omgången med fyra spelare var alla för otåliga för att tänka på att pussla ihop den. De uttryckte dock att det blev lättare att förstå hur spelplanen hängde ihop efter att observatörerna flyttat ihop spelplanen. Två av spelarna tyckte det var ett roligt spel, medan en inte tyckte så mycket om det. De tyckte att det hade varit roligare om det funnits fler och svårare banor, samt om det funnits någon slags fiende med i spelet.

Under hela evalueringssessionen var stämningen hetsig och stressad. Barnen var otåliga på att få spela och fördröjningar såsom instruktioner, tekniskt strul och att behöva vänta in de andra spelarna ökade barnens frustration. Barnen blev till slut sura på varandra.

5.2.3 Grupp 3

Den tredje gruppen bestod av fyra barn, vilket dessutom var den grupp som hade yngst barn. Med fanns även en lärare som delvis tolkade med tecken. Ett par av de här barnen hade mycket svårt för att kommunicera, därav tolkandet. En ytterligare följd av barnens kommunikationssvårigheter var att endast träna själv-läget och barnorna för två spelare testades. Barnen hade även lite svårt att förstå instruktionerna och behövde få flera moment förklarade för sig många gånger. De klarade ändå av träna själv-läget relativt självständigt. Viss hjälp från lärare och observatörer var dock nödvändig för att de skulle förstå vad det var som hände.

Det blev än en gång problem med sammankopplingen. De hade liknande problem som i den tidigare gruppen, samt hade svårt att förstå vem som skulle göra vad. De blev märkbart förvirrade när plattan inte indikerade att systemet arbetade. Även portalerna orsakade viss förvirring. Spelarna ville inte använda portalerna, utan de

5. Resultat

försökte flytta karaktärerna mellan plattorna genom att dra dem till kanten.

Under spelomgången för två spelare gick det bra med tanke på spelarnas svårigheter med att kommunicera. De försökte göra sig förstådda. Viss frustration uppstod när de inte kunde få fram vad de tänkte. När de inte kunde berätta vad de ville hände det att de sträckte sig över till den andre spelarens platta, vilket orsakade viss irritation. Trots spelarnas svårigheter under spelets gång uttryckte de ändå att de tyckte spelet var kul. De tyckte dock att det var svårt, speciellt att flytta karaktärerna.

6

Diskussion

I denna del av rapporten diskuteras det vad vi tycker om de metoder och val vi gjort under utvecklingen av spelet CirKva. Vi går in på huruvida forskningen som CriKvas gränssnitt lutat sig mot räcker och är relevant och om varför testgrupperna under de olika evalueringarna såg ut som de gjorde. Det tas även upp vilka evalueringsmetoder som används och om de fungerat bra, samt begränsningar med pappersprototyper. I övrigt diskuteras det om vi nått vårt ena mål och verkligen anpassat spelet för barn med kognitiva funktionshinder, och vilka möjligheter som finns för att utveckla spelet i framtiden.

6.1 Forskningsläget

Som det nämns i avsnitt 2.1 baseras den mesta forskningen, som framtagandet av CirKvas gränssnitt lutat sig mot, på personer med autismspektrumsyndrom. Det finns en del överlappning mellan autismspektrumet och elever i särskolan. Kognitiv funktionsnedsättning diagnostiseras ofta tillsammans med andra diagnoser, däribland de som faller under autismspektrumet [40]. Ur den synvinkeln är forskningen som den här rapporten tagit upp relevant.

Dock är det så att kognitiva funktionsnedsättningar innefattar fler personer än enbart dem inom autismspektrumet. Då alla olika diagnoser och individer har sina utmärkande egenskaper och problembilder är den forskning vi tagit del av inte nödvändigtvis representativ.

Trots stor ansträngning har projektgruppen inte lyckats hitta forskning med syfte att ta fram riktlinjer för applikationsutveckling med andra grupper än autismspektrumet som målgrupp. Det finns en del forskning kring vilken metodik som passar andra målgrupper, men ingen som kommit fram till resultat på konkreta riktlinjer. I de projekt som togs del utav under litteraturstudien var det sällan fokus på att hitta eller ta fram några riktlinjer. Det var vanligare att applikationerna som togs fram i samband med projekten skraddarsyddes till den grupp personer som de samarbetade med. På så sätt kunde de målgruppsanpassa applikationerna utan att behöva fundera på målgruppspecifika riktlinjer.

Under utvecklingen av CirKva har flera beslut tagits med forskning riktad mot autismspektrumet som bas. Resonemanget kring att luta sig mot forskning som inte är helt representativ är därför viktigt att förklara. Till att börja med är autismspektrumet inte sällan en delgrupp av kognitiva funktionsnedsättningar och därför till viss del relevant. Att följa riktlinjerna medförde således att hänsyn tagits till åtminstone en del utav målgruppen. Vidare överensstämmer de riktlinjer som fanns i mångt och mycket med den information om målgruppen som Peter Börjesson delgivit. Utifrån de underlag som fanns ansågs det att det, även om det kanske inte var helt representativt för hela målgruppen, i alla fall inte var helt fel för målgruppen heller.

6.2 Plattform

Valet av plattform var beroende av diverse andra val som behövde tas dessförinnan. Eftersom målgruppen enbart använder sig av iPads i skolan, medförde det att spelet implementerades för operativsystemet iOS. Det innebar att både ett programmeringsspråk och en plattform för iOS behövde finnas. Under de inledande efterforskningarna undersöktes flertalet alternativ, och de det slutligen vägde mellan var utvecklingsverktygen Corona, Xamarin och Xcode. Corona [15] och Xamarin [16] är båda utvecklingsmiljöer som stödjer utveckling för flera plattformar. Både Corona och Xamarin framstår sig vara väldigt breda plattformar, som erbjuder utveckling för både iOS och Android. Xamarin använder sig dessutom av så kallade nativa användargränssnitt, vilka möjliggör det för utvecklaren att enkelt skapa användarvänliga gränssnitt [16], vilket kan ses som en fördel över Corona.

Xcode, å andra sidan, gör att en applikation blir direkt knuten till enhetens operativsystem. Xcode är direkt riktad för utveckling av applikationer för Apple-enheter. Applikationer får en större chans att prestera bra när de konstrueras direkt i miljön de är ämnade att användas i. Försättningsvis är ramverken Multipeer Connectivity och SpriteKit, som används i detta projekt, anpassade för utveckling av applikationer i just Xcode. Det är de främsta orsakerna till att vi valde Xcode som plattform för utvecklingen av CirKva.

6.3 Evalueringsprocess

De evalueringsmetoder som valdes visades fungera väldigt bra. Vid testning av prototyperna hade det varit svårt att ha någon annan metod än "observation i laboratorium", då det inte var rätt målgrupp som utförde testet. Det innebar att evalueringen inte kunde utföras i den miljö spelet var tänkt för. Eftersom spelet även går ut på att en grupp personer ska sitta tillsammans, var vi tvungna att aktivt göra grupper av de som visade intresse för att vara med under testningen. Det svåraste under denna evalueringen var just att få ihop en tillräckligt stor grupp.

Vid testningen av den slutliga produkten kunde vi däremot testa på den riktiga målgruppen. Det innebar att vi vid denna evaluering valde metoden "observation

i fält”. Det fungerade väldigt bra och vi tror att denna metod var det bästa sättet att få en insikt i hur spelet mottogs och användes i den tänkta miljön. Vi tror inte att det hade varit lika bra att använda metoden “observation i laboratorium” för denna målgruppen, då det hade kunnat bli för mycket nya intryck. Om testgruppen både hade kommit till en ny lokal, träffat nya människor och testat ett nytt spel är chansen stor att deras uppmärksamhet hade riktats åt annat håll än just spelet. Således, genom att använda metoden “observation i fält” vid evalueringen, fick vi se hur målgruppen agerade och utförde de olika uppgifterna i sin naturliga miljö.

Vi är även nöjda med valet av intervjutyp som resulterade i en semistrukturerad variant. Vi tror att det var det bästa sättet att dels få svar på det vi undrade, och dels få respons på sådant vi inte funderat på själva. Om det hade varit en strukturerad intervju hade mycket av informationen testgrupperna delade med sig av aldrig kommit fram. I en ostrukturerad intervju anser vi att svaren hade blivit alldeles för varierande och det hade varit svårt att jämföra vad de olika personerna tyckte. Det hade helt enkelt varit svårare att få en tydlig blick av vad som var bra och vad som behövde mer arbete.

6.4 Användning av pappersprototyp

Pappersprototypen visade sig vara ett värdefullt hjälpmedel vid utvärdering av hur användarvänligt olika delar av spelkonceptet var. Den hade dock sina begränsningar. Vid testning av pappersprototyp hade spelarna möjligheten att bryta mot spelets regler innan spelledaren alltid hann reagera. Att spelartestarna hann göra saker i spelet som inte är tillåtet utan omedelbar feedback från spelet fick spelreglerna att framgå som oklara.

En av fördelarna med att använda sig av pappersprototyper är att de går mycket fortare att ta fram än en motsvarandemjukvara. De pappersprototyper som användes i utvecklingen av det här projektet var handritade för att det var mindre tidskrävande än att designa dem i ett program för bildbehandling. Det faktum att de var handritade har dock sina brister. Den grafiska utformningen blev som följd av detta begränsad jämfört med slutprodukten.

Eftersom pappersprototypen är begränsad både grafiskt och prestandamässigt så kan spelupplevelsen inte testas på ett rättvist sätt. Därför lades störst fokus på att analysera hur speltestarna förstod spelkonceptet, samt deras helhetsintryck, vilket ofta gav värdefull information som gjorde att spelkonceptet kunde förbättras.

6.5 Testgrupper vid evalueringar

Eftersom spelet hade en tydlig målgrupp, vore det mest optimala att utföra samtliga evalueringar med just den målgruppen. Men vid evalueringen av prototyperna och den slutliga produkten användes två olika testgrupper. Detta beror på att pappersprototyper är enkla att tillverka vilket möjliggjorde flera tester under kort tid där

olika delar av spelkonceptet förbättrades utifrån återkoppling. Det hade inte varit praktiskt att genomföra så många tester, så tätt inpå varandra, om dessa skulle utförts hos målgruppen. Dessutom fokuserade testerna, i ett sådant tidigt skede av projektet, på användarvänlighet i allmänhet snarare än på specifika implementeringar av riktlinjer mot målgruppen. Vi upplever därför att denna kompromiss ledde till en mer agil utveckling av spelkonceptet som i slutändan gynnat projektet.

6.6 Förväntningar och målsättningar

Det var en självklarhet för samtliga i projektgruppen att skapa ett så bra spel som möjligt, trots den tidsbegränsning som fanns i projektet. För att säkerställa att vi skulle nå det önskvärda resultatet, var det av stor vikt att sätta upp diverse delmål. Det fanns två mål som var grundläggande för projektet: att implementera ett spel som sträcker sig över flertalet kommunicerande enheter, och anpassa spelet för barn med kognitiva funktionsnedsättningar. Båda målen är relativt omfattande, och behövde därför brytas ner till mindre delmål.

Det första målet behövde delas upp i ytterligare delmål, då det inte var möjligt att få flera enheter att kommunicera med varandra det första vi gjorde. Därför sattes det upp ett mål att först och främst få två enheter att kommunicera med varandra och sända mindre objekt mellan varandra. Därefter skulle vi successivt få fler enheter att kommunicera med varandra tills vi nådde målet med fyra kommunicerande enheter.

För att det senare målet skulle kunna uppnås behövde vi först och främst ta reda på vilka riktlinjer som finns vid utveckling av applikationer för barn med kognitiv funktionsnedsättning. Dessa riktlinjer behövde tas i beaktande vid framtagningen av speldesign och koncept, för att spelet skulle passa projektets målgrupp. Med de olika riktlinjerna i åtanke kunde ett spel som sträcker sig över flera enheter, anpassat för barn med kognitiva funktionsnedsättning, implementeras.

Genom att vi satte upp sådana tydliga delmål underlättades utvecklingen av spelet. Dessutom blev det enklare att nå projektets två grundläggande mål. I det stora hela anser samtliga i projektgruppen att de viktigaste målen för spelet har uppnåtts inom den tidsbegränsning som fanns. Vi lyckades implementera spelet på så sätt att det kan spelas av fyra enheter samtidigt. Dessutom var spelet och dess koncept relativt väl anpassat för målgruppen. Det kan främst styrkas utifrån de evalueringar som har gjorts med barn med kognitiva funktionsnedsättningar.

Trots det fanns det främst ett delmål som inte riktigt uppnåddes under projektets gång, vilket var att en implementering av tillräckligt många nivåer med olika svårighetsgrader. Detta var något som var önskvärt från oss som utvecklare av spelet, då vi ville ge barnen möjlighet att själva få välja svårighetsgraden på nivåerna och på så sätt utvecklas i egen takt.

6.7 Målgruppsanpassning

Vad gäller hur väl anpassat spelet är för den tilltänkta målgruppen är det viktigt att titta på två olika saker, dels hur väl användargränssnittet är anpassat för målgruppen, dels hur väl spelkonceptet i sig är anpassat för målgruppen.

6.7.1 Anpassning av gränssnittet till målgruppen

Då det under de initiala efterforskningarna framkom ett antal konkreta riktlinjer för hur användargränssnittet bör se ut, finns det goda möjligheter att utvärdera gränssnittet.

Valet av den minimalistiska designen har underlättat för att följa riktlinjerna då stort fokus i riktlinjerna legat på att få fram ett enkelt gränssnitt utan onödiga och för detaljrika element som distraherar från uppgiften. Utifrån tabellen i avsnitt 2.1 uppfyller spelet 10 av 13 av de önskade riktlinjerna, och endast ett utav de 10 icke önskvärda riktlinjerna.

De önskvärda riktlinjerna som inte blev uppfyllda är:

- Indikera var användaren är.
- Visuella indikatorer vid lång laddningstid.
- Hjälpknapp.

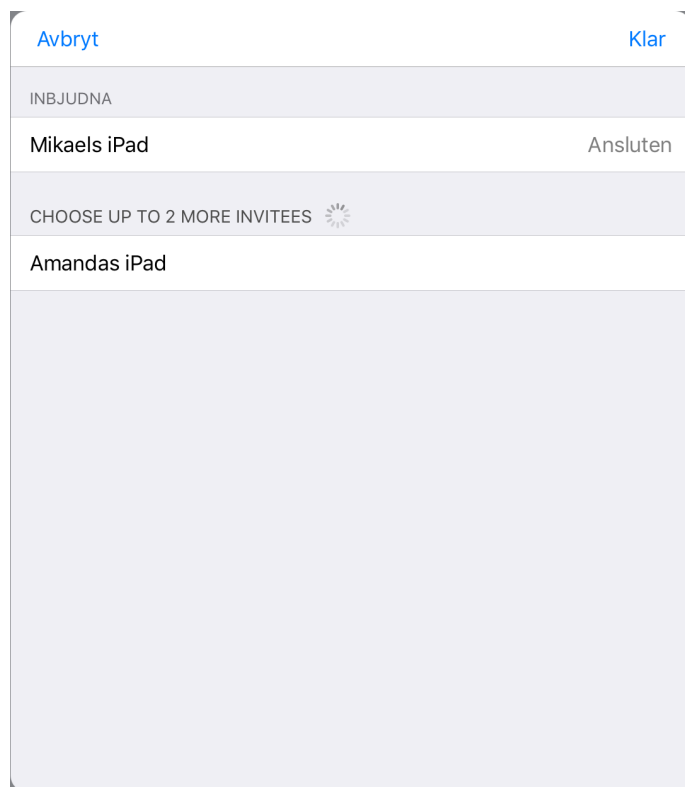
Visuella indikatorer vid lång laddningstid har inte blivit aktuellt mer än möjligtvis vid uppstart av programmet då byten mellan olika vyer har gått mycket snabbt och smidigt. En indikator på hur lång tid det tar för spelet att ladda i början har diskuterats, och ses som en möjlighet i framtiden. För tillfället är den största tidsåtgången från det att användaren klickar på spelikonen tills det att spelmenyn visas i sin helhet. Här emellan visas en vit skärm.

När det kommer till att indikera var användaren är i spelet saknas detta. Projektgruppen hoppas att det framgår tydligt var någonstans i spelet användaren är genom den tydliga vyn och rubriken i menyn för att välja nivå. I spelvyn ansågs det inte nödvändigt att explicit visa att spelet var igång.

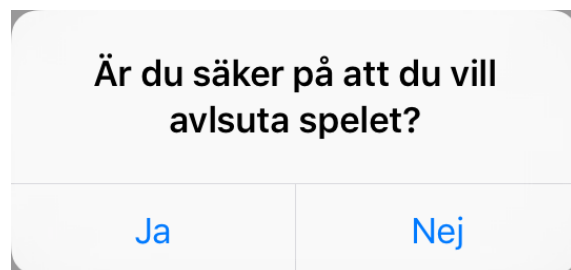
Under testtillfället med målgruppen uppstod viss förvirring under sammankopplingstillfället. Här var det en hel del väntetid som projektgruppen inte uppmärksammat tillräckligt. Under sammankopplingen måste de inbjudna spelarna vänta på att alla andra spelare har accepterat inbjudan och den spelare som bjuder in har startat en spelomgång. Detta indikeras inte direkt på enheterna och har på så sätt orsakat stor förvirring. Det indikeras inte heller tydligt nog att alla enheter inte har kopplats samman. Det var vanligt för användarna att klicka sig vidare till att välja bana innan alla enheter hade anslutits. Under sammankopplingen hade det således varit bra att ha något som indikerade att systemet arbetade och att användaren måste vänta innan de kan gå vidare till nästa steg.

Det finns ingen hjälpknapp i spelet, däremot finns ett läge för att träna själv, detta för att användaren ska kunna bli familjär med hur spelet fungerar. I varje vy i spelet finns det även en meny- eller tillbaka-knapp, som ger användaren möjlighet att återgå till menyn. Trots att projektgruppen ansåg att spelet inte behövde en hjälpknapp på grund av träna själv-läget uttrycktes det önskemål från målgruppen att kunna få instruktioner och tips under en spelomgång. Det kan således vara av intresse att se över valet att utesluta en hjälpknapp då det verkar som att behovet för en finns.

Bland de icke önskvärda elementen så innehåller spelet faktiskt ett par dialogrutor, vilka enligt Pavlov kan anses vara distraherande [14]. En av dialogrutorna dyker upp under sammankopplingsstadiet och en när spelaren avbryter ett spel. Det finns dock förmildrande omständigheter kring de dialogrutor som används. Dels är de standard för iOS och användarna antas vara familjära med enhetens gränssnitt. Det kan således vara en fördel att använda ett standardfönster istället för en som är utformad speciellt för spelet. Dels är den dialogruta som är standard i iOS 9 mycket enkel till sitt utseende, och följer i sig mångt och mycket de riktlinjer som följts under spelets utveckling.



Figur 6.1: Dialogruta för sammankoppling. Författarens egen bild.



Figur 6.2: Dialogruta för avslut av spel. Författarens egen bild.

Vad gäller dialogrutan för att avsluta ett spel har den ytterligare ett existensberettigande. Då det går att komma åt tillbakaknappen av misstag finns det en risk att mycket arbete från användarens sida kan gå förlorat. Ett centralt begrepp som kommer upp flera gånger i Tidwells Designing Interfaces är just att programmet inte ska förstöra för användaren och att användaren ska kunna röra sig i programmet utan att behöva oroa sig över att något oförutsett händer [41]. Utifrån de fördelar som dialogrutorna faktiskt medför ansågs de vara berättigade trots att de bör undvikas. Vid användartesterna visade det sig att dialogrutorna i sig inte medförde några problem för målgruppen. Det var som redan nämnts andra omständigheter kring sammankopplingen som orsakade problem.

Överlag anser vi att användargränssnittet följer de utsatta riktlinjerna, och att det fungerar relativt väl för målgruppen. En del detaljer, speciellt kring sammankopplingsmomentet, behöver ses över i vidare utveckling.

6.7.2 Anpassning av spelets upplägg för målgruppen

Under utvecklingen av spelet togs stor hänsyn till de riktlinjer som kom fram i kapitel 2.1, speciellt då de som kom från intervjun med Peter Börjesson. Utifrån den inledande efterforskningen hade följande riktlinjer extra stor vikt vid utformning av vad spelet skulle gå ut på.

- Belöningsystem, speciellt sådana där det finns flera nivåer, kan bli en låsning då fokus kan hamna på att samla alla belöningar.
- Stressfulla moment kan upplevas som svåra och obehagliga.
- Barn i särskolan har i mångt och mycket samma intressen som vilka barn som helst.
- Personer med autismspektrumtillstånd kan ha stora problem med att misslyckas.

Spelkonceptet utvecklades hela tiden mot att minimera inslag av ovanstående och i den slutliga produkten finner man inga av de konkreta saker som målgruppen har svårt för. Den har ett helt binärt belöningsystem, där något antingen är klart eller inte. Utöver detta är spelet utformat på ett sådant sätt att ingenting som händer i spelet påverkar karaktären eller spelaren negativt. Det finns inget i spelet som förstör för spelaren eller påverkar karaktärerna till det sämre, mer än möjligtvis ett felval

av väg hos spelaren. Dock är det alltid möjligt att ta sig tillbaka till utgångsläget genom att förflytta karaktären tillbaka samma väg som den kom till en eventuell återvändsgränd. På så sätt uppfyller spelet att inte innehålla misslyckanden.

Huruvida ett moment var stressigt diskuterades stundtals, speciellt kring karaktärernas förflyttningssätt. Flera av de förflyttningssätten som diskuterades tog kontrollen av karaktären ut ur spelarens händer på ett sätt som ansågs kunna bli stressande. För att vara på den säkra sidan valdes ett styrsätt där spelaren drar karaktären längs skärmen, vilket lägger spelets takt hos spelaren. Det finns ingenting i spelet som tvingar spelaren till att utföra ett moment under tidspress.

Under testning visade sig förflyttningmomentet i sig inte vara stressande, dock kunde hastigheten som andra spelare förflyttade karaktärer egga på en viss individ att flytta den karaktär den hade kontroll över snabbare.

Något som kan uppfattas som stressande för en del individer i målgruppen är kravet på samarbete för att kunna spela spelet till fullo. Då vissa individer kan ha mer problem än andra med kommunikation kan det så klart vara både stressande och obehagligt att behöva samarbeta med en grupp. Att kommunikationsmomentet kan utgöra en stressfaktor bekräftades under evalueringstillfället. När testpersonerna inte kunde föra fram vad de tänkte blev de otåliga och frustrerade. För barn som har problem med att kommunicera kan en lärare eller annan vuxen som känner barnet väl vara till stor hjälp, då denne kan ingripa efter barnets behov.

Själva syftet med projektet är trots allt att främja just samarbete, och det blir således en krock mellan stress och projektets syfte. Utifrån projektets målsättning frångicks den stressproblematik som finns kring kommunikation för en del individer. Förhoppningsvis ger spelet barnen möjlighet att öva på att föra fram sina tankar och åsikter, så att de i framtiden är bättre utrustade för att hantera den stress som kan uppstå vid kommunikation.

Utöver det ovan nämnda var målsättningen att det skulle vara ett roligt spel, oberoende av om spelaren har en kognitiv funktionsnedsättning eller ej. Barnen som testade slutprodukten sa nästan alla att de tyckte att spelet var roligt. Utifrån de tester som gjordes på dels oberoende testare, dels personer med anknytning till projektet Touch AT! framgick att även de tyckte att det var ett roligt, men stundtals klurigt spel.

Utifrån argumenten ovan anser projektgruppen att spelets upplägg har en potential att vara mycket väl anpassat till målgruppen, med reservation för att vissa individer kan uppfatta det som stressande till följd av samarbetskravet i spelet. Spelet ger dock en ypperlig möjlighet för barnen att öva på att hantera stressen som uppstår i samband med samarbetet.

6.8 Framtida utveckling

Om CirKva någon gång i framtiden ska publiceras på den kommersiella marknaden, så att särskolor i Sverige kan köpa in den, finns ett antal förbättringar som hade varit nödvändiga för att förhöja spelupplevelsen. Först och främst måste antalet spelbara nivåer utökas markant för att spelet över huvud taget skulle vara värt att användas frekvent. Det skulle även vara önskvärt att svårighetsgraden ökar i takt med antalet avklarade nivåer. Detta för att få till en mer stimulerande upplevelse.

Ett annat mycket högt prioriterat område som måste utvecklas är sammankopplingmomentet, där större indikation om vad som händer och mer reglering av när inbjudningar skickas krävs. Det behöver bli tydligt för användarna när systemet processerar inbjudningar och att inbjudna användare ska vänta tills en nivå blivit vald.

I dagsläget är spelet helt ljudlöst. Spelmusik hade förmodligen varit opassande i skolmiljö och, vid uppspelning, skapat förvirring då flertalet mobila enheter är inkluderade i samma spel. Dämpade ljudeffekter skulle däremot ge spelarna indikationer på att saker händer på just deras enheter och skulle till exempel kunna användas när en karaktär ställer sig på en knapp.

Vidare finns möjligheter för ytterligare finputsning av designen, som idag är väldigt platt. Tredimensionella figurer, exempelvis, hade gett en mer logisk bild av vad som sker i spelet med tanke på att vyn är ovanifrån. Då en karaktär rör sig hade exempelvis en sfär skapat känslan av att den rullar, vilket den tvådimensionella cirkel som används idag inte kan. Fortsättningsvis är spelets dialogrutor utformade i iOS standarddesign, vilket skär sig mot rådande gränssnitt i resten av programmet. Om fortsatt utveckling skulle ske hade säkerligen dessa fönster anpassats till att bli enhetlig med övrig design.

Utan att sabotera den önskvärda avskalade designen alltför mycket skulle en slags handling kunna göra spelet mer intressant. Det hade kunnat tillföra en grad av motivation till att vilja fortsätta spelandet och hade kanske frambringat ett tydligare mål. Detta hade i och för sig tvingat fram längre väntetider i spelet, som vissa människor kan uppleva som tröttsamma.

Fortsättningsvis har det diskuterats en hel del huruvida vi ska indikera för användare av CirKva att samarbete krävs och att de olika enheterna måste vara synliga för samtliga spelare för att de ska kunna klara av de olika nivåerna. En lösning som diskuterades var att tvinga spelarna att lägga de olika enheterna mot varandra för att på så sätt bilda en spelplan tillsammans, och för att säkerställa att spelarna inte sitter med varsin enhet på eget håll. För att lösa det var tanken att, för varje nivå, visa en bild för spelarna hur varje enhet ska placeras. Den här lösningen togs bort när det bestämdes att spelplanen skulle indikera hur enheterna skulle ligga med hjälp av portaler. Den ursprungliga lösningen hade i efterhand varit tydligare, vilket är något som kan utforskas i framtiden.

7

Slutsats

Vid projektets start sattes det upp som mål att utveckla ett spel anpassat till barn med kognitiva funktionsnedsättningar för att ge barnen möjlighet att öva på sin samarbetsförmåga och sina sociala förmågor. Spelet skulle ha en spelplan som sträckte sig över flera surfplattor.

CirKva är ett logikspel vars spelplan sträcker sig över flera iPads vilka kommunicerar med varandra. Därmed har ambitionen att utforma ett spel för flera surfplattor uppnåtts. Spelets utformning kräver att flera personer spelar tillsammans och uppmuntrar till kommunikation och samarbete mellan spelarna. Spelet kan därför bidra till att öka spelarnas sociala och emotionella förmåga, då samarbete och kommunikation ger spelarna tillfälle att öva på dessa. En ökad social förmåga kan anses bidra till en ökad social hållbarhet.

CirKva uppfyller även i stort de fastslagna riktlinjerna från 2.1, och utifrån de användartester som gjordes hos en särskoleklass visade det sig att eleverna inte hade några större problem med spelets gränssnitt. CirKva kan därför anses vara på god väg att vara väl anpassat för barn med kognitiva funktionshinder, även om viss vidare utveckling och testning behövs för att komma till rätta med de brister som kvarstår. Viss vidare utveckling är även att föredra för att förfina spelets funktionalitet och utseende. Även ett större antal nivåer skulle vara fördelaktigt.

De riktlinjer som fastslagits i kapitel 2.1 har tjänat projektets syfte väl, men är inte alltid helt representativa för hela målgruppen. Mer konkret forskning på vad ett användargränssnitt för barn med kognitiva funktionsnedsättningar kräver behövs.

Referenser

- [1] J. Ryberg. (2013, apr.) Ipad höjer motivationen i skolan. [Online]. Tillgänglig: <http://computersweden.idg.se/2.2683/1.502148/ipad-hojer-motivationen-i-skolan>.
- [2] U. Jämterud. (2016, feb.) Surfplattan blir delatagare i leken. [Online]. Tillgänglig: <http://www.skolverket.se/skolutveckling/forskning/amnen-omraden/it-i-skolan/relationer-larande/surfplattan-blir-deltagare-i-leken-1.205047>
- [3] P. Andersson. (2015, aug.) Mattebageriet 2 – lär dig räkna i köket med pappa som assistent. [Online]. Tillgänglig: <http://www.skolappar.nu/mattebageriet-2/>
- [4] O. Lidegran. (2014, jun.) Kalle kunskap tvåan – Öva matematik, språk, logiskt tänkande och kreativitet. [Online]. Tillgänglig: <http://www.skolappar.nu/kalle-kunskap-tvaan/>
- [5] Touch AT! (2016) Om projektet. [Online]. Tillgänglig: <http://www.touch-at.se/>
- [6] Skolverket. (2016, maj) Grundsärskolan är till för ditt barn. [Online]. Tillgänglig: <http://www.skolverket.se/publikationer?id=3202>
- [7] P. Börjesson *et al.*, “Designing technology for and with developmentally diverse children: A systematic literature review,” i *Proceedings of the 14th International Conference on Interaction Design and Children*, ser. IDC ’15. New York, NY: ACM, 2015, ss. 79–88. [Online]. Tillgänglig: <http://doi.acm.org/10.1145/2771839.2771848>
- [8] A. P. Hill *et al.*, *Translational Approaches to Autism Spectrum Disorder*. Cham, Schweiz: Springer International Publishing, 2015, kap. Epidemiology of Autism Spectrum Disorders, ss. 13–38. [Online]. Tillgänglig: http://dx.doi.org/10.1007/978-3-319-16321-5_2
- [9] M. Oskoui *et al.*, “An update on the prevalence of cerebral palsy: a systematic review and meta-analysis,” *Developmental Medicine & Child Neurology*, vol. 55, nr. 6, ss. 509–519, 2013. [Online]. Tillgänglig: <http://dx.doi.org/10.1111/dmde.12111>

[//dx.doi.org/10.1111/dmcn.12080](http://dx.doi.org/10.1111/dmcn.12080)

- [10] M. E. Weijerman och J. P. de Winter, “Clinical practice: The care of children with down syndrome,” *European Journal of Pediatrics*, vol. 169, nr. 12, ss. 1445–1452, 2010. [Online]. Tillgänglig: <http://dx.doi.org/10.1007/s00431-010-1253-0>
- [11] M. Davis *et al.*, “Guidelines for researchers and practitioners designing software and software trials for children with autism,” *Journal of Assistive Technologies*, vol. 4, nr. 1, ss. 38–48, 2010. [Online]. Tillgänglig: <http://dx.doi.org/10.5042/jat.2010.0043>
- [12] J. Mintz *et al.*, “Key factors mediating the use of a mobile technology tool designed to develop social and life skills in children with autistic spectrum disorders,” *Computers & Education*, vol. 58, nr. 1, ss. 53 – 62, 2012. [Online]. Tillgänglig: <http://www.sciencedirect.com/science/article/pii/S0360131511001710>
- [13] A. M. Figueroa och R. Juárez-Ramírez, “Towards a user model for the design of adaptive interfaces for autistic users,” i *Computer Software and Applications Conference Workshops (COMPSACW), 2014 IEEE 38th International*, jul. 2014, ss. 264–269. [Online]. Tillgänglig: <http://dx.doi.org/10.1109/COMPSACW.2014.47>
- [14] N. Pavlov, “User interface for people with autism spectrum disorders,” *Journal of Software Engineering and Applications*, vol. 7, nr. 2, ss. 128 – 134, 2014. [Online]. Tillgänglig: <http://dx.doi.org/10.4236/jsea.2014.72014>
- [15] Corona Labs Inc. (2016) Why choose corona sdk? [Online]. Tillgänglig: <https://coronalabs.com/why-choose-corona-sdk/>
- [16] Microsoft. (2016) Xamarin platform. [Online]. Tillgänglig: <https://www.xamarin.com/platform>
- [17] K. Putt. (2014, maj) Monument valley. [Online]. Tillgänglig: https://www.flickr.com/photos/k_putt/14190284512/in/photostream/
- [18] Apple Inc. (2016) Xcode ide. [Online]. Tillgänglig: <https://developer.apple.com/xcode/features/>
- [19] ——. (2016) About swift. [Online]. Tillgänglig: <https://swift.org/about/#swiftorg-and-open-source>
- [20] ——. (2015, okt.) Model-view-controller. [Online]. Tillgänglig: <https://developer.apple.com/library/ios/documentation/General/Conceptual/DevPedia-CocoaCore/MVC.html>

- [21] A. Syromiatnikov och D. Weyns, “A journey through the land of model-view-design patterns,” i *Software Architecture (WICSA), 2014 IEEE/IFIP*. IEEE, apr. 2014, ss. 21–30.
- [22] L. T. Kim. (2016) Mvp pattern. [Online]. Tillgänglig: <http://thefinestartist.com/android/mvp-pattern>
- [23] Google. (2014) Model view presenter gui design pattern. [Online]. Tillgänglig: https://commons.wikimedia.org/wiki/File:Model_View_Presenter_GUI_Design_Pattern.png
- [24] R. Frey. (2010) Mvc-process. [Online]. Tillgänglig: <https://upload.wikimedia.org/wikipedia/commons/a/a0/MVC-Process.svg>
- [25] Apple Inc. (2013, sep.) About multipeer connectivity. [Online]. Tillgänglig: <https://developer.apple.com/library/ios/documentation/MultiPeerConnectivity/Reference/MultiPeerConnectivityFramework/>
- [26] R. Daque. (2016) wifi. [Online]. Tillgänglig: <https://pixabay.com/en/wifi-wifi-signal-internet-network-1290667/>
- [27] IO-Images. (2016) Smartphone. [Online]. Tillgänglig: <https://pixabay.com/en/smartphone-mobile-phone-phone-1132675/>
- [28] WOODPUNCHER. (2014) Bluetooth. [Online]. Tillgänglig: <https://pixabay.com/en/bluetooth-connection-mobile-phone-468288/>
- [29] Apple Inc. (2016, mar.) About spritekit. [Online]. Tillgänglig: https://developer.apple.com/library/ios/documentation/GraphicsAnimation/Conceptual/SpriteKit_PG/Introduction/Introduction.html#//apple_ref/doc/uid/TP40013043
- [30] ——. (2016, mar.) Sknode. [Online]. Tillgänglig: https://developer.apple.com/library/ios/documentation/SpriteKit/Reference/SKNode_Ref/
- [31] ——. (2015, sep.) Skscene. [Online]. Tillgänglig: https://developer.apple.com/library/ios/documentation/SpriteKit/Reference/SKScene_Ref/
- [32] ——. (2014, sep.) Skview. [Online]. Tillgänglig: <https://developer.apple.com/library/ios/documentation/SpriteKit/Reference/SKView/>
- [33] A. Randolph. (1999) Ricochet robots. [Online]. Tillgänglig: <http://riograndegames.com/games.html?id=163>
- [34] Nintendo. (1996) Pokémon. [Online]. Tillgänglig: <http://www.pokemon.com/us/>

- [35] P. Staníček. (2014) <http://paletton.com/>. [Online]. Tillgänglig: <http://paletton.com>
- [36] SharkD. (2010, maj) Hsv color solid cylinder alpha lowgamma. [Online]. Tillgänglig: https://commons.wikimedia.org/wiki/File:HSV_color_solid_cylinder_alpha_lowgamma.png?uselang=sv
- [37] Y. Rogers *et al.*, *Interaction Design - Beyond human-computer interaction*, 4e uppl. John Wiley & Sons Ltd, 2011, kap. 7, ss. 229–258.
- [38] Apple Inc. (2016) Enumerations. [Online]. Tillgänglig: https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/Enumerations.html
- [39] D. C. Schmidt, “Wrapper facade—a structural pattern for encapsulating functions within classes,” 1999.
- [40] A. P. Association, *Diagnostic and statistical manual of mental disorders*, 5e uppl. Arlington, VA: American Psychiatric Publishing, 2013.
- [41] J. Tidwell, *Designing interfaces: patterns for effective interaction design*, 1a uppl. Sebastopol, CA: O’Reilly, 2006.
- [42] Apple Inc. (2015, sep.) Mcsession. [Online]. Tillgänglig: https://developer.apple.com/library/ios/documentation/MultiPeerConnectivity/Reference/MCSessionClassRef/index.html#//apple_ref/occ/cl/MCSession
- [43] ——. (2013, sep.) Mcpeerid. [Online]. Tillgänglig: https://developer.apple.com/library/ios/documentation/MultiPeerConnectivity/Reference/MCPeerID_class/index.html#//apple_ref/occ/cl/MCPeerID
- [44] ——. (2013, okt.) Mcadvertiserassistant. [Online]. Tillgänglig: https://developer.apple.com/library/ios/documentation/MultiPeerConnectivity/Reference/MCAdvertiserAssistant_class/index.html#//apple_ref/occ/cl/MCAdvertiserAssistant
- [45] ——. (2015, okt.) Uiviewcontroller. [Online]. Tillgänglig: https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIViewController_Class/index.html#//apple_ref/occ/cl/UIViewController
- [46] ——. (2013, sep.) McbrowserviewController. [Online]. Tillgänglig: https://developer.apple.com/library/ios/documentation/MultiPeerConnectivity/Reference/MCBrowserViewController_class/index.html#//apple_ref/occ/cl/MCBrowserViewController
- [47] ——. (2013, sep.) Mcadvertiserassistantdelegate. [Online]. Till-

- gänglig: https://developer.apple.com/library/ios/documentation/MultiPeerConnectivity/Reference/MCAdvertiserAssistantDelegate_class/index.html#//apple_ref/occ/intf/MCAdvertiserAssistantDelegate
- [48] —. (2013, sep.) McbrowserviewControllerdelegate. [Online]. Tillgänglig: https://developer.apple.com/library/ios/documentation/MultiPeerConnectivity/Reference/MCBrowserViewControllerDelegate/index.html#//apple_ref/occ/intf/MCBrowserViewControllerDelegate
- [49] —. (2013, sep.) Mcsessiondelegate. [Online]. Tillgänglig: <https://developer.apple.com/library/ios/documentation/MultiPeerConnectivity/Reference/MCSessionDelegateRef/>
- [50] —. (2013, sep.) Mcnearbyservicebrowserdelegate. [Online]. Tillgänglig: https://developer.apple.com/library/ios/documentation/MultiPeerConnectivity/Reference/MCNearbyServiceBrowserDelegateRef/index.html#//apple_ref/occ/intf/MCNearbyServiceBrowserDelegate
- [51] —. (2013, sep.) Mcnearbyserviceadvertiserdelegate. [Online]. Tillgänglig: https://developer.apple.com/library/ios/documentation/MultiPeerConnectivity/Reference/MCNearbyServiceAdvertiserDelegateProtocolRef/index.html#//apple_ref/occ/intf/MCNearbyServiceAdvertiserDelegate

A

Intervju efter testning av pappersprototyp

Frågor som ställdes efter testningen av pappersprototypen.

Generella frågor om spelarna:

- Hur gamla är ni?
- Vilken spelvana har ni?
- Har ni erfarenhet av samarbetsspel?

Frågor om sammankopplingstestet:

- Var det tydligt hur ni skulle göra?
- Hade ni föredragit att enheter i närheten blir inbjudna automatiskt, istället för att ni ska behöva skicka en inbjudan manuellt?

Frågor om själva spelet:

- Tyckte ni att spelet var roligt?
- Fann ni några svårigheter under spelets gång?
- Vad i utförandet av uppgiften fungerade bra? Om inget, varför?
- Vad tycker ni om spelets upplägg?
- Vad tycker ni om spelets svårighetsgrad?
- Fanns det något i spelet som ni saknade?
- Hur tycker ni att samarbetet i gruppen fungerade under spelets gång?

B

Intervju efter testning av slutlig produkt

Frågorna som ställdes efter testningen av den slutliga produkten.

Grupp 1

- Var det något speciellt som ni tyckte var svårt?
- Hur kändes det när plattorna lades ihop istället för att ha dem utspridda? Hur tyckte du att det var när vi byggde ihop spelplanen? Blev det lättare att se en helhetsbild av spelplanen eller hur tyckte ni att det var?
- Tyckte ni att det var ett roligt spel?
- Hur tyckte ni att det kändes med samarbetet?
- Hur kändes det när någon annan kom och styrde på den plattan som ni hade?
- Man hade ju kunnat säga hur man ska göra med. Hur tycker ni att man ska göra? Ska man sträcka sig och visa eller ska man försöka förklara hur man ska göra?
- Hur tyckte ni att spelet ser ut?
- Var det några färger ni hade svårt att se skillnad på?
- Har ni några tankar på hur man kan göra spelet bättre? Hur man skulle kunna göra det roligare eller lättare att spela?
- Hade det känts ok att en kompis använde eran egna iPad om ni spelade på era iPads?
- Har ni något annat att tillägga om spelet?
- Hur var det att kunna skicka grejer mellan iPadsen?
- Skulle ni vilja spela det igen?
- Hur tyckte ni att det var att kunna testa momenten först innan ni provade de svårare grejerna?
- Hur var spela självläget?
- Hur kändes det när någon annan gjorde något innan man själv hann tänka?
- Hade det varit bättre att kunna titta på banan innan någon kunde flytta runt på grejerna?
- Har ni tankar på hur man kan göra det lättare att förstå hur saker hänger ihop?

Grupp 2

- Tyckte ni att det var ett roligt spel?
- Var det något som var svårt?
- Hur gick det att samarbeta?
- Är det samarbete när man sträcker sig?
- Har ni några idéer om hur det kan bli ett bättre spel?
- Tycker ni att spelet ser bra ut?
- Hade ni svårt att se skillnad på några utav färgerna?

Grupp 3

- Tyckte ni att det var ett roligt spel?
- Vad var roligast?
- Var det något som var svårt?
- Vad var det som var svårt?
- Hur tyckte ni att det gick att spela med era kompisar?
- Hur kändes det när ni sträckte er över till eran kompis platta?
- Hur kändes det när eran kompis sträckte sig över till eran platta?
- Finns det något ni tyckte hade gjort spelet roligare?
- Hade ni några problem med hur spelet såg ut?
- Var det fint eller fult?
- Hade ni svårt att se skillnad på färgerna?

C

Multipeer Connectivity Framework

C.1 Klasser & Objekt

MCSession:

Detta är en klass innehållandes MCSession-objekt som handhar själva sessionen där enheterna utbyter data samt kommunicerar med varandra. Det är alltid en enhet, A, som skapar själva sessionen, vilket då gör det möjligt för denna enhet att bjuda in andra enheter till applikationens session. Om enhet A inte skapar sessionen, så har den möjlighet att delta i en annan enhets session, förutsatt att enheten som har skapat sessionen bjuder in enhet A [42].

MCPeerID:

Varje enhet måste ha ett specifikt ID som representerar den applikation som körs på enheten i fråga. Detta främst för att kunna identifiera samtliga enheter under en session, men även för att kunna skilja dem åt. Det objekt som sköter detta är kallas för peerID [43].

MCAdvertiserAssistant:

Detta är en klass som innehåller advertiser assistant-objekt som handhar inbjudningar till sessioner. När en enhet bjuder in en annan enhet till sin applikation kommer en inbjudan att skickas till den enheten. Advertiser assistant objektet gör det möjligt för enheten att se inbjudan. Dessutom handhar objektet enhetens svar på inbjudan, som antingen kan accepteras eller nekas [44].

UIViewController:

UIViewController-klassen gör det möjligt för användaren att hantera de olika vyerna i en iOS-applikation [45]. UIViewController består även av en underklass, MCBrowserViewController, som visar de närliggande enheterna som befinner sig på samma nätverk som enheten i fråga [46].

C.2 Protokoll

De protokoll som Multipeer Connectivity Framework består av beskrivs nedan.

MCAdvertiserAssistantDelegate

Detta protokoll beskriver de metoder som MCAdvertiserAssistant objekt kan implementera [47]. Mer om vad själva objektet gör kan ses i avsnitt C.1.

MCBrowserViewControllerDelegate

Detta protokoll beskriver de metoder som MCBrowserViewController objekt kan implementera [48]. Mer om vad själva objektet gör kan ses i avsnitt C.1.

MCSessionDelegate

Detta protokoll beskriver de metoder som ett MCSession objekt kan implementera för att handa sessioner [49]. Mer om vad själva objektet gör kan ses i avsnitt C.1.

MCNearbyServiceBrowserDelegate

Detta protokoll beskriver de metoder som ett MCBrowserServiceBrowser objekt kan implementera [50].

MCNearbyServiceAdvertiserDelegate

Detta protokoll beskriver de metoder som ett advertiser objekt kan implementera [51].

D

Kravspecifikation

Denna bilaga presenterar samtliga funktionella samt icke-funktionella krav som finns på systemet. Då projektet hade en begränsad tid, valdes en del krav ut som prioritering enligt följande schema:

HÖG: Användningsfallet måste implementeras innan projektets slut för att applikationen skall kunna fungera som tänkt.

MEDEL: Användningsfallet ska implementeras i mån av tid.

LÅG: Spelet kan gott och väl klara sig utan dessa funktioner. Men för att öka nivån på spelet, kommer det att implementeras om tiden räcker till.

D.1 Funktionella krav

Nedan presenteras samtliga punkter som systemet ska kunna klara av.

Sammankopplingsprocess:

- | | |
|---|------------|
| 1. Systemet ska kunna koppla samman flera enheter. | HÖG |
| 2. Systemet ska kunna känna av antalet enheter som är sammankopplade. | HÖG |
| 3. Systemet ska känna av när en eller flera enheter kopplas bort. | HÖG |
| 4. Systemet ska anpassa spelinnehållet till antalet enheter. | HÖG |

Spelets status:

- | | |
|---|------------|
| 1. Systemet ska känna av när spelet startas. | HÖG |
| 2. Systemet ska känna av när en nivå är avklarad. | HÖG |
| 3. Systemet ska känna av när spelet avslutas. | HÖG |

Spelets egenskaper:

- | | |
|---|------------|
| 1. Systemet ska ge karaktärer olika färger. | HÖG |
| 2. Systemet ska ge karaktärer samma egenskaper. | HÖG |

- | | |
|---|--------------|
| 3. Systemet ska ha tydliga färger. | HÖG |
| 4. Systemet ska stoppa en karaktär när den går in i ett fast föremål. | HÖG |
| 5. Systemet ska ge spelarna möjligheten att spela med single-player. | HÖG |
| 6. Systemet ska, för varje nivå, visa spelarna hur surplattorna skall läggas. | HÖG |
| 7. Systemet ska anpassa antalet karaktärer till antalet surfplattor. | HÖG |
| 8. Systemet ska erbjuda möjligheten att välja mellan olika nivåer. | MEDEL |
| 9. Systemet ska erbjuda olika svårighetsgrader. | LÅG |
| 10. Systemet ska erbjuda ljud. | LÅG |

Nedan presenteras samtliga punkter som spelarna ska kunna göra med själva spelet.

- | | |
|--|--------------|
| 1. Spelarna ska kunna starta ett spel. | HÖG |
| 2. Spelarna ska kunna bjuda in varandra till spelet. | HÖG |
| 3. Spelarna ska kunna avsluta spelet när de än önskar. | HÖG |
| 4. Spelare ska kunna flytta på karaktärer. | HÖG |
| 5. Spelarna ska flytta samtliga karaktärer till respektive mål, för att kunna klara en nivå. | HÖG |
| 6. Spelarna ska kunna inverka på spelplanen via diverse knappar på spelplanen. | HÖG |
| 7. Spelarna ska, för varje nivå, kunna ändra spelbanans form. | MEDEL |
| 8. Spelarna ska kunna välja kunna välja svårighetsgraden på nivån. | MEDEL |
| 9. Spelarna ska kunna plocka upp föremål från spelplanen. | LÅG |
| 10. Relativ ljudvolym ska kunna ställas in. | LÅG |

D.2 Icke-funktionella krav

1. Spelet ska inte avslutas oväntat.
2. Spelet ska vara underhållande och främja samarbete.
3. Spelet ska vara relativt lätt att förstå samt använda.
4. Gränssnittet ska vara lättförståeligt.
5. Det ska vara möjligt att lägga till fler plattor.
6. Det ska vara möjligt för flera spelare att spela spelet samtidigt.
7. Enheter ska kunna kommunicera med varandra.
8. Spelet ska vara medvetet om hur många enheter som är sammankopplade.
9. Systemet bör vara uppdaterat konstant.

E

Användningsfall

I denna bilaga presenteras de användningsfall som har tagits fram för spelet. Först presenteras de prioriterade användningsfallen, vilka är fullständiga. Därefter presenteras samtliga användningsfall som har tagits fram.

E.1 Prioriterade användningsfall

Nedan presenteras de användningsfall som har prioriterats i detta spel. Samtliga användningsfall har prioriterats som höga.

Nummer: 1

Namn: Bjuda in spelare.

Aktör: Spelaren som startar spelet (sessionen).

Mål: Att bjuda in fler enheter till spelet.

Beskrivning: Spelaren bjuder in andra spelare till spelet genom att låta systemet söka efter enheter i närheten som befinner sig på samma nätverk, och sedan välja de enheterna som ska vara med i en given spelomgång.

Förvillkor: Sant

Eftervillkor: Spelarnas enheter är sammankopplade och kan kommunicera med varandra.

Huvudflöde av händelser:

1. Spelaren startar spelet.
2. Spelaren trycker på knappen "Skapa nytt spel".
3. **Anta:** Samtliga enheter är ansluta på samma nätverk eller sammankopplade via samma Bluetooth.
4. Systemet letar efter tillgängliga enheter.
5. **Anta:** Systemet finner enheter som är online.
6. Spelaren väljer de enheter (spelare) som ska vara med i spelomgången.
7. Spelarna som bjuds in mottar en inbjudan (användningsfall 2 och 3).

Alternativt flöde:

3-4a. **Anta:** Systemet finner inga enheter på samma nätverk.

1. Spelarna ansluter sig till samma nätverk.

5-6a. **Anta:** Systemet finner inga enheter online.

1. Spelaren trycker på Avbryt-knappen.
2. Gå till steg 2 igen.

Nummer: 2

Namn: Acceptera inbjudan

Aktör: Spelaren som mottar inbjudan.

Mål: Att acceptera inbjudan till en spelomgång.

Beskrivning: Spelaren mottar en inbjudan till en spelomgång som accepteras. Det medför att spelet påbörjas med samtliga inbjudna spelare.

Huvudflöde av händelser:

1. Den spelare som har startat spelet skickar en inbjudan.
2. Systemet skickar en notifikation till mottagaren av inbjudan.
3. **Anta:** Mottagaren väljer att acceptera inbjudan.
4. Spelaren som skickade inbjudan mottar en notifikation om accepterandet av inbjudan.
5. Systemet påbörjar sessionen.
6. Systemet startar spelomgången.

Alternativt flöde:

3-4a. **Anta:** Mottagaren råkar neka inbjudan.

1. Gå till steg 1 igen.

Nummer: 3

Namn: Neka inbjudan.

Aktör: Spelaren som mottar inbjudan.

Mål: Att neka inbjudan till en spelomgång.

Beskrivning: Spelaren mottar en inbjudan till en spelomgång som denne väljer att neka.

Huvudflöde av händelser:

1. Den spelare som har startat spelet skickar en inbjudan.
2. Systemet skickar en notifikation till mottagaren av inbjudan.
3. **Anta:** Mottagaren väljer att neka inbjudan.
4. Spelaren som skickade inbjudan mottar en notifikation om nekandet av inbju-

dan.

5. Systemet avslutar sessionen.

Alternativt flöde:

3-4a. **Anta:** Spelaren råkar acceptera inbjudan.

1. Spelaren avslutar spelet.
2. Systemet avslutar sessionen.

Nummer: 5

Namn: Bilda ny form på spelbanan.

Aktör: Spelarna.

Mål: Att flytta om enheterna för att bilda en ny form på spelbanan.

Beskrivning: Innan varje nivå kommer en bild presenteras på enheterna, gällande hur enheterna ska sättas ihop för nivån i fråga. Spelarna lägger enheterna i den givna ordningen, och sedan startas spelet.

Förvillkor: Sant

Eftervillkor: Spelarna har lagt enheterna i den givna ordningen.

Huvudflöde av händelser:

1. Spelarna har startat spelet och valt en nivå.
2. Systemet visar hur enheterna ska läggas ihop.
3. **Anta:** Spelarna lägger enheterna på rätt sätt.
4. Systemet startar spelet på den givna nivån.

Alternativt flöde:

3-4a. **Anta:** Spelarna lägger enheterna på fel sätt.

1. Systemet hänvisar spelarna att lägga enheterna på det angivna viset.

Nummer: 6

Namn: Inverka på spelplanens design.

Aktör: Spelarna.

Mål: Att ändra spelplanens utseende genom att stå på en knapp.

Beskrivning: Spelaren ställer sig på en av de existerande knapparna. Det leder till att en vägg i samma färg/ett hinder försvinner.

Förvillkor: Sant

Eftervillkor: Designen på banan är modifierad.

Huvudflöde av händelser:

1. **Anta:** Spelaren flyttar sin karaktär till en färgad knapp.

2. Systemet tar bort det hinder tillhörandes knappen i fråga.
3. Systemet uppdaterar innehållet på spelplanen.

Alternativt flöde:

1-3a. **Anta:** Spelaren flyttar sin karaktär bort från den färgade knappen.

1. Systemet lägger tillbaka det hinder tillhörandes knappen i fråga.
2. Systemet uppdaterar innehållet på spelplanen.

Nummer: 7

Namn: Flytta karaktär

Aktör: Spelarna.

Mål: Att flytta sina karaktärer på enheterna.

Beskrivning: Spelaren flyttar runt sin karaktär genom att svepa åt den önskade riktningen.

Huvudflöde av händelser:

1. Spelaren flyttar sin karaktär genom att svepa åt det håll som karaktären skall röra sig åt.
2. **Anta:** Inga hinder finns i vägen.
3. Systemet uppdaterar karaktärens koordinater.
4. Systemet flyttar karaktären till den önskvärda platsen.

2-4a. **Anta:** Karaktären stöter på ett hinder.

1. Systemet stoppar karaktären.
2. Gå till steg 1 igen.

E.2 Samtliga användningsfall

Nedan presenteras samtliga användningsfall, de uppgifter som användaren kan utföra på gränssnittet, som har tagits fram för detta spel. Avsnittet innehåller både de prioriterade användningsfall, men även de som inte har haft lika hög prioritet i projektet. Orsaken till den lägre prioriteringen beror på att de inte är nödvändiga för att spelet skall kunna fungera på ett bra vis.

Samtliga användningsfall presenteras enligt den korta versionen. För den kompletta versionen av de prioriterade användningsfallen, se avsnitt E.1.

Sammankoppling

Nummer: 1

Namn: Bjuda in spelare.

Aktör: Spelaren som startar spelet (sessionen).

Mål: Att bjuda in fler enheter till spelet.

Beskrivning: Spelaren bjuder in andra spelare till spelet genom att låta systemet söka efter enheter i närheten som befinner sig på samma nätverk, och sedan välja de enheterna som ska vara med i en given spelomgång.

Nummer: 2

Namn: Acceptera inbjudan

Aktör: Spelaren som mottar inbjudan.

Mål: Att acceptera inbjudan till en spelomgång.

Beskrivning: Spelaren mottar en inbjudan till en spelomgång som accepteras. Det medför att spelet påbörjas med samtliga inbjudna spelare.

Nummer: 3

Namn: Neka inbjudan.

Aktör: Spelaren som mottar inbjudan.

Mål: Att neka inbjudan till en spelomgång.

Beskrivning: Spelaren mottar en inbjudan till en spelomgång som denne väljer att neka.

Nummer: 4

Namn: Gå ur spelet under en spelomgång.

Aktör: Spelarna.

Mål: Att ta bort sin enhet från de resterande ihopkopplade enheterna.

Beskrivning: En spelare avslutar spelet på sin enhet, vilket gör att de andra får en notis. Resterande spelare skickas automatiskt till huvudmenyn.

Inverkan på spelplanen

Nummer: 5

Namn: Bilda ny form på spelbanan.

Aktör: Spelarna.

Mål: Att flytta om enheterna för att bilda en ny form på spelbanan.

Beskrivning: Innan varje nivå kommer en bild presenteras på enheterna, gällande hur enheterna ska sättas ihop för nivån i fråga. Spelarna lägger enheterna i den givna ordningen, och sedan startas spelet.

Nummer: 6

Namn: Inverka på spelplanens design.

Aktör: Spelarna.

Mål: Att ändra spelplanens utseende genom att stå på en knapp.

Beskrivning: Spelaren ställer sig på en av de existerande knapparna. Det leder till att en vägg i samma färg/ett hinder försvinner.

Karaktärerna

Nummer: 7

Namn: Flytta karaktär

Aktör: Spelarna.

Mål: Att flytta sina karaktärer på enheterna.

Beskrivning: Spelaren flyttar runt sin karaktär genom att svepa åt den önskade riktningen.

Nummer: 8

Namn: Samla föremål från spelplanen.

Aktör: Spelarna.

Mål: Att karaktärer skall kunna plocka upp föremål längs banan.

Beskrivning: Genom att kollidera en karaktär med ett föremål, plockar den karaktären upp det föremålet från banan och har det till sitt förfogande.

Nummer: 9

Namn: Använda föremål.

Aktör: Spelarna.

Mål: Att karaktärer skall kunna använda sina föremål för att låsa upp nya delar av banan.

Beskrivning: Genom att en karaktär äger ett visst föremål och kolliderar med ett hinder som kräver det föremålet, kommer det hindret att avaktiveras. Om en karaktär exempelvis har plockat upp en nyckel, kommer denne kunna öppna dörrar genom att kollidera med dem.

Spelarna

Nummer: 10

Namn: Se bedrifter.

Aktör: Spelarna.

Mål: Spelarna skall kunna se om en nivå är avklarad eller inte.

Beskrivning: Spelaren ska i vyn där de väljer nivå kunna se vilka nivåer som är avklarade sedan tidigare.

Nummer: 11

Namn: Välj en nivå.

Aktör: Spelaren som startat spelet. (Värd)

Mål: Att starta en av dom redan upplåsta nivåerna.

Beskrivning: Spelaren ser de olika nivåerna denne kan välja och trycker på en av dom. Det medför att nivån startar.