

The Spaceship

Fleranvändarspel för barn med användning av
fysiska reglage
Multiplayer game for kids utilizing physical controls

Andreas Ekberg
Gustav Johansson
Kenton Larsson
Lovisa Svensson
Lucas Hägg
Ramez Rizek
Rebecka Sahlin.

Examinator: Daniel Jönsson

Sammanfattning

Projektet handlar om utvecklingen av ett spel med fysiska reglage som riktar sig till barn. Målet är att främja social interaktion och erbjuda stimulerande pusselutmaningar. Genom användarcentrerad design och forskning inom spelutveckling skapades en engagerande och användarvänlig spelupplevelse.

Tekniska specifikationer för systemet presenterades, inklusive de fysiska reglagen och programvaran. Spelet utvecklades med dessa aspekter i åtanke: liktfördelat engagemang mellan två spelare, att svårighetsgraden är anpassad för barn och främjande av social interaktion. Systemarkitekturen och säkerhetsaspekterna beaktades också. Utvecklingsmiljön för spelet inkluderade Unity som spelmotor samt Inkscape och FL Studio för grafik- och ljudskapande. Med vägledning av framtagna designprinciper samt genom återkoppling från användartester identifierades potentiella problem och spelet kunde förbättras.

Genom en iterativ process och kontinuerlig förbättring lyckades projektet framgångsrikt leverera ett spel som uppfyller de uppsatta målen och förväntningarna, vilket syftade till att tilltala och engagera barn i en underhållande och utmanande spelupplevelse.

Innehåll

Sammanfattning	i
Figurer	v
1 Inledning	1
1.1 Bakgrund	1
1.2 Syfte och målsättning	1
1.3 Frågeställning	1
1.4 Avgränsningar	2
1.5 Spelbeskrivning	2
2 Relaterat arbete	3
2.1 Att utveckla spel för barn	3
2.2 Tema till spelet	4
2.3 Funktioner i liknande spel	4
3 Metod - Design	6
3.1 Principer	6
3.2 Tema	7
3.3 Pussel	8
3.4 Grafik	10
3.5 Ljud	10
4 Metod - Teknisk implementation	11
4.1 Systemarkitektur	11
4.2 Fysiska reglage	12
4.3 Spelmotor	13
4.4 Scener	14
4.5 Startskärm	14
4.6 Light Puzzle	15
4.7 Crane Puzzle	15

4.8 Orbit Puzzle	15
4.9 Användartester	16
5 Resultat 18	
5.1 Fysiska reglage	18
5.2 Slutresultat på spelet	18
5.3 Användartester	20
5.3.1 Användartest 1	20
5.3.2 Användartest 2	22
5.3.3 Användartest 3	23
6 Analys och diskussion 24	
6.1 Design	24
6.2 Teknisk implementation	25
6.3 Användartester	26
6.4 Potentiell vidareutveckling	26
6.5 Etisk och samhällelig reflektion	27
6.6 Källkritik	27
7 Slutsatser 28	
7.1 Frågeställning 1	28
7.2 Frågeställning 2	28
Litteraturförteckning 29	
A Utvecklingmetodik 31	
A.1 Utvecklingsmetodik	31
A.2 Tidsplan	31
A.3 Organisation	32
A.4 Mötesprinciper och rutiner	33
A.5 Versionshantering	34
A.6 Dokumentationsprinciper	34
A.7 Kravhantering och spårning	34
A.8 Story/task-hantering	35
A.9 Användartester	36
A.10 Testningsprinciper och rutiner	36
A.11 Modelleringsrutiner	36
B Reflektion över systemutvecklingsprocessen 37	
B.1 Lucas Hägg	38

B.2 Gustav Johansson	39
B.3 Andreas Ekberg	40
B.4 Kenton Larsson	40
B.5 Ramez Rizek	41
B.6 Rebecka Sahlin	41
B.7 Lovisa Svensson	42

Figurer

3.1	Moodboard som användes under utvecklingen.	7
3.2	Insperation från spelet <i>Among us</i> till vänster och spelet <i>It takes two</i> till höger.	8
3.3	Wireframes för Planet- och Light Puzzle, samt Crane Puzzle från vänster till höger uppifrån.	9
3.4	Prototyp av Crane Puzzle i Figma	9
4.1	Container-diagram för ett spel med fysiska reglage	11
4.2	Funktionaliteten och layout av två reglage uppsättningar	13
4.3	Representation av spelets scener och stadier i ordning, från start till slut.	14
4.4	Kodimplementation för att föra en satelliten i en omloppsbana samt ändra på omloppsbanan med hjälp av musens position.	16
5.1	De fysiska reglagen och kretskortet.	18
5.2	Startskärmen för båda spelarna.	19
5.3	Första pusslet Light Puzzle. Figuren representerar vyn för spelare 1 respektive spelare 2.	19
5.4	Andra pusslet <i>Crane Puzzle</i> . Figuren representerar vyn för spelare 1 respektive spelare 2.	19
5.5	Tredje pusslet Orbit Puzzle. Figuren representerar vyn för spelarna.	20
5.6	Statistik på svar om hur kul användare tyckte pussel 1 var. 1 är mycket tråkigt och 5 är mycket kul.	21
5.7	Testares svar på hur kul de tyckte pussel 2 var, där 1 är mycket tråkigt och 5 är mycket kul.	21
5.8	Testare som känner barn svarade på hur svårt de tyckte spelet var, där 1 är mycket enkelt och 10 är mycket svårt.	23
5.9	Testare som inte känner barn svarade på hur svårt de tyckte spelet var, där 1 är mycket enkelt och 10 är mycket svårt.	23
A.1	Gantt-schema för projektets utvecklingsprocess.	31
A.2	Produkt backlog som användes under projektets gång.	35

Kapitel 1

Inledning

Under detta projekt har ett 2D-spel som använder fysiska reglage utvecklats. Spelets huvudsakliga innehåll centrerar kring att två användare med hjälp av samarbete skall ta sig genom en serie utmaningar i form av pussel. I detta kapitel följer en beskrivning av projektets syfte, mål och bakgrund. Vidare presenteras relevanta frågeställningar samt avgränsningar rörande projektet.

1.1 Bakgrund

Spelidén fick först liv av Karljohan Lundin Palmerius, universitetslektor och docent i visualisering och medieteknik på Linköpings universitet. Tanken var att ge besökarna ett nytt spel till visualiseringscentret i Norrköping, med mål på att väcka problemlösning och samarbete mellan besökarna. Karljohan har varit kund och handledare till detta projekt.

1.2 Syfte och målsättning

Denna rapport har i syfte att ge en tydlig bild över ett genomförande av ett mjukvaruprojekt inom kursen TNM094 Medietekniskt kandidatprojekt. Projektets mål och krav var att skapa och undersöka hur ett fleranvändarspel som placeras i en publik miljö ger upphov till ökat samarbete och engagemang mellan mäniskor. Spelet ska styras med fysiska reglage och en spelare ska inte kunna klara spelet själv. Denna rapport syftar till att ge läsare en helhetssyn av utvecklingsprocessen, ge inblick i vilka problem som uppstått, samt hur dessa problem har bemötts.

Den publika miljön som projektet hade som målmiljö bestämdes till att bli en rymdutställning på Visualseringscenter C i Norrköping, där målgruppen bland besökarna är barn. Ett syfte med projektet var således att undersöka hur ett fleranvändarspel utvecklas för att skapa engagemang hos barn från sju och uppåt.

1.3 Frågeställning

Följande frågeställningar är utformade för att besvara frågor som beskriver utmaningar rörande projektets målsättning. Dessa frågor behandlas under utvecklingen och besvaras av innehållet i denna rapport.

- Vilka designprinciper ska spelet följa för att bibehålla barnens intresse och engagemang i en publik miljö?

- Hur kan spelet utvecklas så att båda användarna upplever att de har utfört lika många uppgifter efter spelets slut, samt att alla reglage har samma användningsmängd?

1.4 Avgränsningar

Spelet har utvecklats specifikt för användning av två personer. Spelet använder sig av två skärmar samt två uppsättningar av reglage som varje spelare använder och kontrollerar. Slutprodukten, det vill säga ett spel med reglage och skärmar, antas vara placerat i en publik miljö med periodvis hög ruljangs av människor. Detta kräver en hållbar ställning så att produkten tål hantering. Det kommer inte göras.

Utöver detta så kommer spelet endast vara tillgängligt för datorer med *Windows* som operativsystem, då det skulle ta för lång tid att utveckla ett spel som fungerar på exempelvis *Linux* eller andra operativsystem.

1.5 Spelbeskrivning

Spelet utspelar sig i och runt ett rymdskepp som kraschlandat på en okänd planet. Två astronauter måste ta sig genom skeppet för att hämta en nödsignal och skjuta iväg den för att kunna bli räddade. I skeppet träffar de på ett antal utmaningar i form av pussel, som de tillsammans måste lösa. Några av problemen spelarna stöter på är att laga rymdräkter, placera satelliter i omloppsbanor och lösa kodlås. Med hjälp av ett antal fysiska reglage, som beskrivs i *avsnitt 4.2*, har vardera spelare möjlighet att kontrollera olika objekt. Varje pussel innehåller minst en utmaning som kräver att spelarna samarbetar via sina reglage eller verbal kommunikation. Vardera pussel är utformade på så sätt att de båda spelarna har en unik vy av spelaren eller exklusiv kontroll över viktiga kontroller. Nödvändig information som undanhålls för en spelare kan på så sätt ges till den andra spelaren.

Kapitel 2

Relaterat arbete

Detta kapitel fokuserar på det arbete som gjorts i förberedelse inför projektet. Här presenteras en undersökning av målgruppen, design- och pusselprinciper.

2.1 Att utveckla spel för barn

Målet med förundersökningen var att besvara ett antal frågor, framförallt gällande projektets designaspekter, som påverkar målgruppen. Några av de viktigaste av dessa frågor var:

- Hur gör vi spelet lättförståeligt för målgruppen?
- Vad ska vi tänka på när vi skapar gränssnitt?
- Hur håller vi barnens intresse i längre perioder?
- Vilka liknande spel finns och vad gör dessa bra i förhållande till målgruppen?
- Vilka färger bör användas?
- Hur kan vi göra design som är inkluderande av barn med olika funktionshinder, exempelvis färgblindhet eller epilepsi?

Enligt författaren Banu Manav i *Color-emotion associations and color preferences: A case study for residences* kan barn lätt bli distraherade av irrelevanta objekt och därför är det viktigt att utforma spelet utefter principer som både underlättar för och är tilltalande för barn. Bland annat har färger en stor påverkan på hur tilltalande något är för barnen. Färger som gör det lättbegripligt och attraktivt för barn är ljusa och färgglada färger [1].

En annan viktig aspekt gällande användning av färger är personer med färgblindhet, som har det svårt att urskilja vissa färger. För att undvika detta kan färger, enligt företaget *Adore Color*, anpassas genom att generera en färgpalett som är godkänd för färgblinda [2].

Ytterligare aspekt är att enligt Felix Gyllenstig Serrao i bloggen *Spelläraren - Spel, föräldrarskap och pedagogik* så ger spel med tydliga mål, regler och hinder en trygg och förutsägbar miljö för barn. De spelen ger en viss kontrollerad kontext där inga oväntade överraskningar inträffar [3].

Anpassad UX-design är en viktig aspekt när det kommer till att skapa applikationer för barn, Mariia Kasym beskriver i *UX Design for Children: How to Create a Product Children Will Love* att motoriken hos barn är sämre i jämförelse med vuxna. Barns motorik påverkar därför spelets svårighets

och komplexitet. Vidare tar artikeln upp hur den mentala förmågan hos barn skiljer sig ifrån vuxna, barn kan lätt tappa fokus, bli uttråkade snabbt och har även generellt mindre tålmod. Barn behöver mycket interaktion för att något ska kunna behålla deras uppmärksamhet i längre perioder. Som lösning på detta beskrivs hur konstant feedback och belöningar kan användas för att öka interaktionen, och därmed bibehålla barnens uppmärksamhet. Andra exempel på element som kan göra upplevelsen mer interaktiv inkluderar ljudeffekter och animationer [4]. Vidare är det, enligt författarna av *Effective Feedback Procedures in Games for Health*, bra att ha en tydlig distinkt skillnad mellan interaktiva och icke-interaktiva objekt. Det är även lättare för barn när välbekanta symboler och objekt används [5].

2.2 Tema till spelet

Ett spel som gav inspiration var *Among Us*. Utöver att spelet har temat rymden och utspelar sig i ett rymdskepp är det mycket populärt bland barn[6]. Miljön är futuristisk och främmande med högteknologisk utrustning med många korridorer och rum. Bakgrundsfärgerna är dämpade med en del svart, grått och vitt. Interaktiva objekt är detaljerade och använder sig av färgglada färger [7]. Atmosfären kan framstå stressig eftersom spelet går ut på att sabotera för de andra spelarna.

Två andra spel gruppen hämtade inspiration ifrån var de 3-dimensionella förstapersons äventyrsspelet *We Were Here* och *It Takes Two*. Det är två kooperativa spel där spelarna måste samarbeta och kommunicera för att avancera i spelets handling. I *We Were Here* befinner sig karaktärerna i en värld med många olika miljöer exempelvis grottor, fängelsehålor och slott. Beroende på var karaktärerna befinner sig så förmedlar spelet olika känslor. Spelet går exempelvis från en dystopisk känsla med dämpade gråa färger till livliga känslor med mycket färgglada färger. Karaktärerna i spelet *It Takes Two* är två stycken små dockor som befinner sig på en bakgård. Med rolig design har detta spel lyckats att få sina objekt se levande ut. Färgerna i spelet är färglada med mycket kontrast [8] [9].

En viktig del som berör temat är, enligt författarna av *The Structural Characteristics of Video Games: A Psycho-Structural Analysis*, att spelbakgrunden inte ska vara för detaljrik. Detta beror på att det kan få bakgrunden att framstå som rörigt vilket kan leda till att spelare blir distraherade och missar viktiga moment i spelet. Författarna menar på att huvudobjekten bör sticka ut ur bakgrunden, och inte blandas in med de övriga objekten i en scen [10].

En del inspiration har också hämtats från *Pinterest*, vilket är en social medieplattform inriktad på visuell inspiration och delning av bilder. Det är ett onlinesamhälle där användare kan skapa och organisera virtuella anslagstavlor. Detta gör Pinterest till en plattform där man kan söka och samla information om specifika ämnen och projekt. Inspirationen hämtades från en blandning av verkliga fotografier och ritningar av bland annat rymden och spelets objekt [11].

2.3 Funktioner i liknande spel

Among Us har flera olika typer av funktioner, några av dessa funktioner går under namnet *tasks*. Användarna kan interagera med dessa tasks för att klara av pussel. Exempelvis finns ett pussel där spelaren ska matcha kablar utifrån färg. När alla kablar är ihopkopplade är tasken och därmed pusslet klart. Det finns ett annat pussel där spelaren ska dra ett passerkort i en korthållare. Kortet måste dras i en bestämd hastighet för att klara utmaningen. På detta sätt testas spelarens motorik på ett intressant sätt vilket gav gruppen inspiration till detta projekt.

Spelet *We Were Here* innehåller pussel där spelarna måste interagera med varandra. Då spelarna har två olika vyer främjar dessa pussel kommunikation. I exempelvis ett pussel finns det ett schackbräde med osynliga fallhål som spelare 1 behöver ta sig över. Spelare 2 har lösningen och vet vilken väg

som spelare 1 ska gå och behöver kommunicera detta med spelare 1. Detta främjar problemlösningsförmåga och kommunikation mellan spelarna.

It Takes Two har ett pussel där spelarna ska ta sig över ett hål. Spelare 1 har en hammare och spelare 2 har spikar. Spelare 2 kan kasta sina spikar på väggen och med hammaren kan spelare 1 då svinga sig över till andra sidan. Detta låter spelarna öva på sin reaktions- och samarbetsförmåga.

Arkadspelet Pac-Man använder fysiska reglage som kontroller. Detta är ett spel med få enkla funktioner. Spelaren styr positionen av Pac-Man med hjälp av en *joystick*. Pac-Man har ett enkelt koncept som är lätt för en ny spelare att lära sig. Det är också ett spel som brukar befina sig i publika miljöer. [12]

Kapitel 3

Metod - Design

Under följande avsnitt visar hur information som insamlats under förundersökningarna, beskrivna i kapitel 2, utgjort grunden för gruppen när de har fattat beslut och gjort val under skapandet av olika aspekter av projektet.

3.1 Principer

När spel utvecklas är det viktigt att ta hänsyn till målgruppen. För detta projekt är målgruppen, som tidigare nämnts, barn. Att förstå målgruppens intressen och behov gör att spelets innehåll kan anpassas, vilket underlättar utvecklingen och ger möjlighet till att skapa spel som tilltalar och engagerar. Annat som påverkar spelupplevelsen är målgruppens spelvanor och spelförmåga. Genom att använda design som är bekant och som tar spelarens förmågor i åtanke går det att forma en spelupplevelse som minimerar känslor av frustration och som sällan är överväldigande för spelaren. För att fastställa dessa attribut hos målgruppen så gjordes en förundersökning i avsnitt 2.1 under planeringsstadiet av projektet, där resultatet sedan kunde användas för att försäkra en genomgående design som beaktar målgruppen.

Förundersökningen användes som grund för att skapa en rad olika designprinciper som kunde ge vägledning under flera delar av projektet. Gällande färger kunde undersökningen slå fast att spelet bör använda ljusa och färgglada färger för att göra spelet mer attraktivt och lättbegripligt för barn. Användning av specifika färgkombinationer av denna typ kan även användas för att göra inkluderande design, såsom för färgblinda och epilepsi. Genom att ha tydliga och enkla mål, trygga ramar och förutsägar design så kan barnen få en positiv spelupplevelse, där de kan lära sig och underhållas på ett sätt som är lämpligt för deras utvecklingsnivå och ålder. En annan aspekt som uppenbarades i undersökningen var att barn har sämre motorik än vuxna. Detta är något som var viktigt att tänka på vid design av dels reglagens spatiala placering och antalet reglage, men även vid design av spelmoment. I dessa fall behövde antalet reglage som används parallellt begränsas för att inte överbelasta barn. Vidare behöver spelet ge frekvent återkoppling, bland annat genom att använda mycket animationer och ljudeffekter. Med återkoppling får spelaren bekräftelse på sina prestationer, vilket ger motivation som får spelaren att engagera sig mer i spelet. Slutligen är det bra att ha distinkt skillnad mellan interaktiva och icke-interaktiva objekt, samt att dessa objekt såväl som symboler är välbekanta.

Följande designprinciper användes genom utvecklingsprocessen:

- Använd ljusa och färgglada färger.
- Tydliga och enkla mål.

- Trygga ramar, inga överraskande moment.
- Designa utifrån barns motorik.
- Ge feedback så ofta som möjligt.
- Använd mycket animationer och ljudeffekter.
- Inga effekter som framkallar epilepsi hos individer.
- Distinkt skillnad på interaktiva och icke-interaktiva objekt.
- Använd bekanta objekt och symboler.

3.2 Tema

Utifrån önskemål från kund gavs två olika förslag på det som skulle bli spelets centrala tema, rymden eller ekologitema. Alternativen baserades på de teman som förekommer på Visualiseringscenter C i Norrköping. Gruppen kunde tidigt komma överens om att rymd tema skulle passa bra för projektet eftersom det gav fler möjligheter i vilken spelupplevelse man kan skapa samt att medlemmarna kände att rymden är mer intressant.

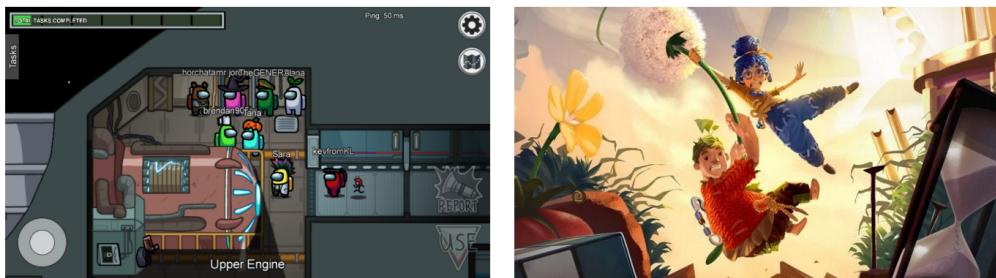
Förundersökningen i avsnitt 2.2 gav en grund till att skapa följande tema och moodboards, se Figur 3.1. Among Us, se Figur 3.2, är spelet som gav projektets riktlinjer kring design. Då temat på projektet var rymden, kände gruppen att Among Us objektdesign och perspektiv var ett mål att gå efter.

Bakgrunden är en betydande del av varje scen som tillför mycket till upplevelsen. För att välja passande färger och mönster på bakgrunder var det viktigt att tänka på hur de skulle påverka objekt i spelet och användarnas uppfattning. Bakgrundens färger bör vara neutrala för att framhäva och tydligt urskilja objekt från sin omgivning. Om bakgrunden skulle vara för detaljrik så riskerar detta att göra kompositionen av scenen rörig, vilket kan leda till att användare blir distraherade och missar viktiga delar av spelet. Dessa aspekter togs till hänsyn när de implementerades i spelet. Resultatet var en genomtänkt och välbalanserad design som gjorde det enkelt för spelarna att spela och lösa pusslet.

It Takes Two, se Figur 3.2, är ett bra exempel på hur objekt kan designas för att tilltala spelets målgrupp. Då deras objekt innehåller starka färger och ger en känsla av liv tilltalar det barn på ett effektivt sätt. Detta inspirerade designen av projektets objekt genom att använda färgstarka färgkombinationer och en animerad stil.



Figur 3.1: Moodboard som användes under utvecklingen.



Figur 3.2: Inspiration från spelet *Among us* till vänster och spelet *It takes two* till höger.

3.3 Pussel

Pussel är den centrala delen som utgör spelets innehåll. Med tema bestämt samt givna krav från kund så kunde gruppen börja överväga de pussel som skulle skapas. De främsta kraven som formade hur ett pussel kunde skapas var att spelet är till för två spelare som har separata skärmar, samt att samarbete mellan spelarna skulle uppmuntras och vara nödvändigt för att klara av spelet. Här sattes även en målsättning på att båda spelarna skulle få lika mycket att göra i varje pussel, samt att användningen skulle fördelas jämnt mellan de olika reglagen.

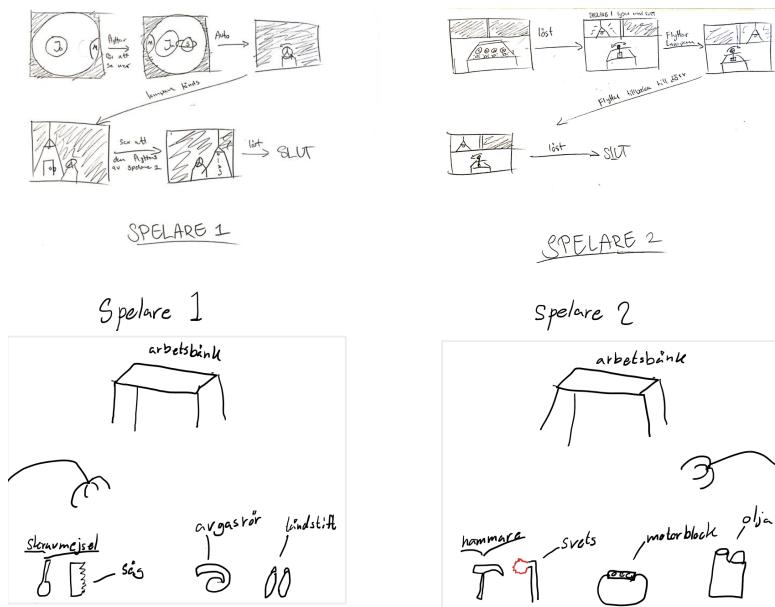
Vidare gjordes tydliga avgränsningar för vad som skulle kunna inkluderas i ett pussel. Något som direkt kunde kopplas till innehållet var att helt utesluta explicit innehåll, såsom våld och droger. En annan viktig aspekt var att slutprodukten antas vara placerad i en publik miljö med periodvis hög ruljangs av mäniskor, och därmed potentiella användare. Detta innebär att den totala speltiden för att klara av spelet behövde begränsas. Som påverkade det totala antalet pussel i spelet, samt varje pussels enskilda komplexitet och svårighetsgrad. Den totala speltiden som skulle eftersträvas bestämdes till ca 10 minuter som uppskattades till 20 minuter för barn. Vilket skulle ge upphov till maximalt tre pussel av passande svårighetsgrad.

Med dessa grundpelare att utgå ifrån fick varje enskild gruppmedlem i uppdrag att komma på minst två mindre pusselidéer eller delmoment till pussel. Vid ett senare tillfälle tog gruppen beslut angående vilka idéer som skulle vidareutvecklas i en *brainstorming session*. Inför brainstorming sessionen delades gruppen in i två mindre grupper, där vardera grupp fick i uppdrag att, med hjälp av brainstorming, integrera idéer till minst ett fullständigt pussel. För att anses som ett fullständigt pussel skulle dessa innehålla:

- Tydlig början och slut
- Samtliga utmaningar med tillhörande lösningar
- Beskrivning av vilka reglage som används och när.

Under detta skede hade även projektets designprinciper fastställts utifrån förstudien i 2.1, vilket gav ytterligare vägledning som varje pusselidé skulle rätta sig efter. En första version av varje pussel illustrerades i form av en *wireframe* på papper, vilket gav en översiktlig bild av pusslets struktur, se Figur 3.3.

Med wireframes för flera pussel kunde gruppen mötas för att diskutera vilka av dessa som var mest lovande utifrån de tillkommande kraven som tagits upp ovan. Gruppen beslutade att gå vidare med totalt tre pussel; *Light Puzzle* där en lampa förflyttas för att lysa upp delar av ett rum, *Crane Puzzle* där spelarna ska matcha olika objekt med hjälp av en kran, och slutgiltigen *Planet Puzzle* där ordning på planeter skulle kopplas i en kombination av knapptryckningar. För att kunna testa dessa spel på en



Figur 3.3: Wireframes för Planet- och Light Puzzle, samt Crane Puzzle från vänster till höger uppifrån.

djupare nivå i form av användartest så skapades prototyper av spelet i *Figma*, som är en webbapplikation för gränssnittsdesign[13]. Resultatet blev två separata prototyper, som visas i Figur 3.4 en för Crane Puzzle och en för både Planet Puzzle och Light Puzzle. Med dessa prototyper kunde ett första användartest utföras enligt avsnitt 4.9.



Figur 3.4: Prototyp av Crane Puzzle i Figma

Med återkoppling från det första användartestet, se avsnitt 5.3.1, så gjordes en sammanställning av vilka aspekter av vardera pussel som var i behov av förändring, samt vad som saknades eller var överflödigt. Ett tydligt resultat från detta var att Planet Puzzle var för otydligt, då det saknade återkoppling för en spelare. Ett annat problem med detta pussel var att en av spelarna upplevde att den inte fick göra tillräckligt. I brist på idéer som skulle kunna förbättra pusslet gjorde att gruppen tog beslut om att helt slopa pusslet. Utifrån sammanställningen kunde dock övriga pussel uppdateras med ny funktionalitet och förbättrad design inför den kommande implementationen i spelmotorn, som beskrivs i avsnitt 4.3.

Under tiden då implementation av resterande pussel påbörjades började två gruppmedlemmar konceptualisera en idé på ett nytt pussel för att ersätta Planet Puzzle, det som skulle komma att bli *Orbit Puzzle*. Detta pussel földe i stora drag samma process som tidigare pussel, vilket började med en definition av pusslets beståndsdelar och reglage, med tillhörande wireframe. Eftersom det första användartestet redan gjorts, samt att gruppen hunnit bekanta sig med spelmotorn, så togs beslutet att direkt implementera pusslet utan att skapa en prototyp i Figma. Med en hastig implementation så kunde även detta pussel vara klart inför nästkommande användartest.

Implementationen i spelmotorn resulterade i en *Minimal Viable Product (MVP)* av spelet, d.v.s. en version där varje pussel kunde spelas från start till slut med användning av reglagen. Detta gav möjlighet till ett andra användartest, som även här resulterade i en sammanställning av förbättringar som kunde utforskas. Som framgår i resultatet av det andra användartestet, se avsnitt 5.3.2, så uppstod en viss förvirring hos samtliga testare under ett specifikt moment i pusslet. Problemet grundade sig dels i att spelarna fick för lite återkoppling på att de gjort framsteg i pusslet, men även något simplistiskt innehåll. Den stora förändringen som gjordes utifrån denna återkoppling var att utöka Orbit Puzzle med ytterligare ett moment, samt att lägga till återkoppling i form av ljud.

3.4 Grafik

Allt grafiskt i spelets scener skapades i det vektorbaserade ritprogrammet *Inkscape* [14], som är en kostnadsfri plattform. I ritprogrammet ritades allt ifrån objekten till bakgrunden, och exporterades till SVG-format som därefter implementerades i spelmotorn. Objekten kunde behöva ändras i form av skala så därför användes det skalbara vektorgrafika formatet SVG som standard. Innan ritningar exporterades behövde de omvandlas till objekt eftersom de bestod av *paths*, då det inte var möjligt att importera SVG-bilder i spelmotorn när de innehöll *paths*.

Verktyget som var mest användbart för projektet var Inkscapes pennverktyg som skapade *Bézierkurvor* och raka linjer samt formverktyget som skapade bland annat cirklar och kvadrater.

Funktionen *Layers* användes för att separera element och hantera dessa individuellt. Lagerfunktionen gav friheten att ändra grafiken utan att påverka andra delar i ritningen, vilket gjorde det lätt och flexibelt att skapa fler versioner, samt ändra specifika delar samtidigt genom att gruppera objekt i samma lager.

3.5 Ljud

Ljud är en faktor som ger mer liv och känsla till ett spel. Till detta projekt skapades bakgrundsmusik med hjälp av en *Digital Audio Workstation* (DAW) vid namn *FL Studio* [15]. I FL Studio skapades en projektfil för varje låt. I en projektfil kan instrument, och mönster till dessa, skapas och placeras ut i en tidslinje som representerar låten. Sedan kan den exporteras till en ljudfil. Designen av musiken baserades på pusslets tema, stämning, utmaning och när i spelet scenen äger rum. Totalt skapades fem stycken låtar som tillhör respektive scen i spelet. Första låten som besökarna hör tillhör startskärmen. Denna designades med mystik och rymd i åtanke, för att direkt försöka väcka nyfikenhet. Den inkluderar också ljud som påminner om knapptryck på en instrumentpanel, vilket är tänkt att matcha det visuella i scenen, se avsnitt 5. Den sista låten i spelet designades med spänning i åtanke, för att ge spelarna en känsla av en sista utmaning. Till kranpusslet, som äger rum i mitten av spelet, designades en helt annan typ av låt. Rymddräktens delar, kranen och de verktyg som måste plockas upp i pusslet ser lite lustiga ut. Därför designades låten till att vara rolig och lekfull. På liknande sätt designades resterande musik.

Förutom bakgrundsmusik brukar spel också inkludera ljudeffekter. Till detta spel användes två ljudeffekter för att ge återkoppling till spelarna. Ett kort pling för att representera framsteg och ett kort surr för att representera misslyckande. Dessa applicerades till Orbit Puzzle och hämtades från samarbetsarkivet freesound [16].

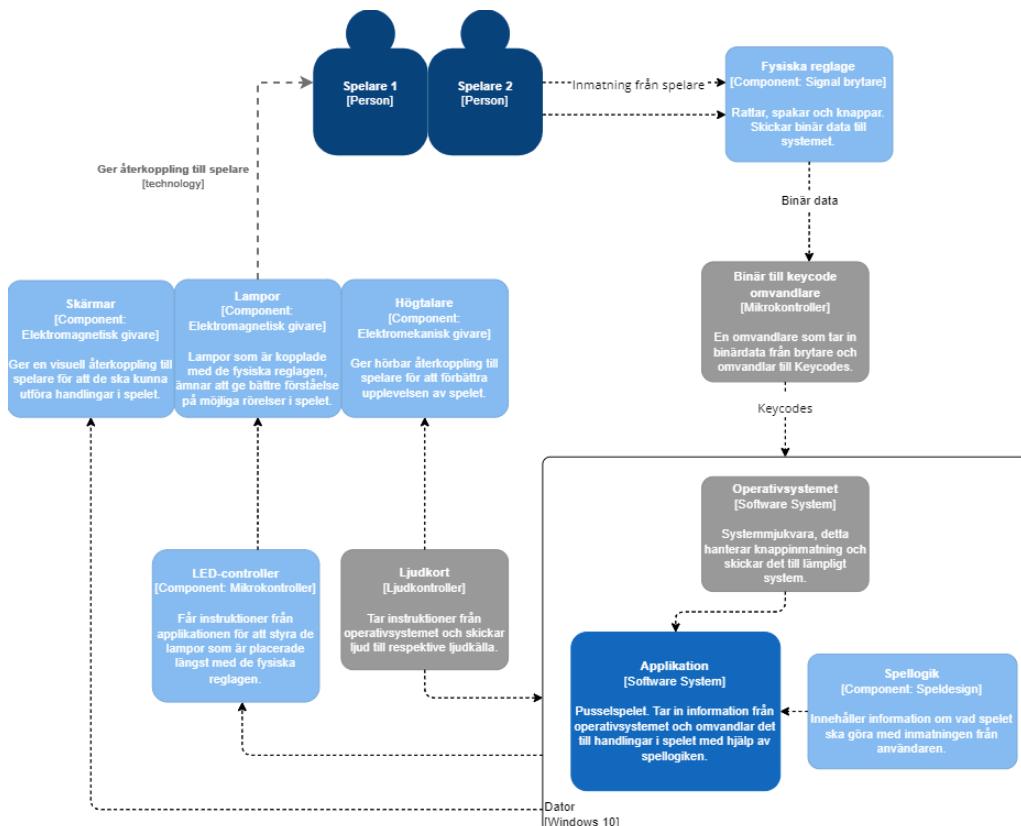
Kapitel 4

Metod - Teknisk implementation

I detta avsnitt beskrivs två viktiga delar av spelets implementering. Först presenteras sammansättningen av fysiska reglage för att styra olika funktioner i spelet. Därefter ges en detaljerad beskrivning av hur spelmotorn *Unity* användes för att skapa spelets scenstruktur, implementera komponenter och programmera spel-funktioner.

4.1 Systemarkitektur

Figur 4.1 visar ett *Container-diagram* som presenterar de olika högnivåkomponenterna eller *containers* av systemet och deras beroende mellan varandra [17]. Syftet med diagrammet är att ge en tydlig översikt över systemets struktur, vilket underlättar för identifiering av potentiella problem samt förenklar förståelsen för hur de olika komponenterna samverkar för intressenter.



Figur 4.1: Container-diagram för ett spel med fysiska reglage

Spelet utvecklades för operativsystemet *Windows 10*, vilket spelade en viktig roll i hanteringen av informationsflödet mellan olika delar av systemet. För att använda fysiska reglage kopplades ett multi-kontrollerkort till datorn. Med kortet tillkom ett förinstallerat program som innehöll ett gränssnitt för bland annat kartläggning av inmatningar från reglage, och program som omvandlade signalen från reglagen till tangentbordsinmatningar.

När en användare interagerade med reglagen skickades signalen från reglagen till datorn genom det etablerade gränssnittet. Själva spelet, alltså applikationen, hade ansvaret att svara med lämplig återkoppling i form av uppdatering av bilder och ljud till användaren.

Applikationen kommunicerade även med spellogiken, som tog beslut baserat på spelarens inmatning och bestämde typen av återkoppling till spelaren. Bildskärmar behövdes för att ge visuell återkoppling till spelaren. Dessutom användes belysta knappar och högtalare för att förbättra spelupplevelsen. Knapparnas belysning styrdes av ytterligare en inbyggd kontroller i kontrollerkortet, och ljudet sköttes av datorns interna hårdvara som kopplades mot högtalare.

Säkerhet är en viktig aspekt inom systemarkitektur. Eftersom spelet ställs i en publik miljö så utvecklades spelet utan beroendet till en databas och därmed en nätverksuppkoppling som reflekterats i Figur 4.1. Detta skär ner på möjliga sårbarheter som systemet kan introducera på en dator, då den inte öppnar upp en väg för attack. Eftersom spelet inte sparar någon information om användarna så kan inte möjlig skadlig mjukvara som redan finns på en dator utnyttja spelet.

4.2 Fysiska reglage

De fysiska reglage som spelet behöver bestämdes av gruppen efter att pusseldesignen var avklarad. De bestämdes utifrån den budget som universitetet presenterat samt det som skulle behövas för att avklara varje pussel. Nedan är en lista över alla reglage:

- Joystick x2
- Färgbelysta knappar x8
- Rattar x2

Komponenterna som användes till att skapa uppsättningarna av fysiska reglage är inköpta från leverantören *Ultimarc* som specialiseras sig på att bygga system för arkadspel[18]. De levererade multi-kontrollerkortet, kablar, knapparna, joysticks och rattarna som presenteras i avsnitt 5.1. Utöver detta införskaffades följande nödvändiga delar:

- Skokartong x2
- En fyra meter nätverkskabel
- Silver- och eltejp

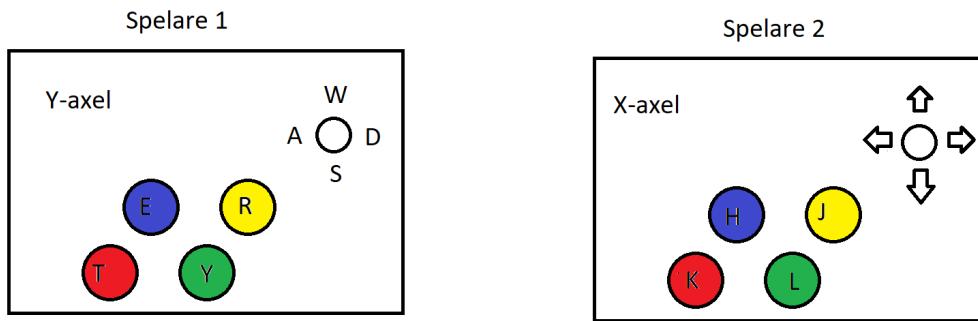
Reglagen är fysiska strömbrytare som kan användas för att styra olika enheter eller funktioner. När en person trycker på brytaren aktiveras en kontakt som i sin tur skickar en binär signal, det vill säga en signal som antingen är på eller av.

En dator kan emellertid inte direkt tolka denna binära information och omvandla den till en önskad funktion eller handling i operativsystemet. Ett kontrollkort som är specialanpassat användes för detta syfte.

Interface for PC to Arcade Controls (IPAC) är ett kontrollkort som är skapat för att koppla upp brytare till en dator. Utöver portarna för de fysiska reglagen så innehåller kretskortet en annan kontroller för att styra lampor *LEDs*. En IPAC valdes i detta projekt för dess funktionalitet till skillnad från andra lösningar som utforskades, såsom *Arduino*[19].

För att säkerställa en rättvis spelupplevelse skulle varje spelare tilldelas en egen uppsättning reglage, vilket möjliggör att båda kan delta jämligt i spelet. Eftersom alla reglagen behövde kopplas till kretskortet placerades detta i en av kartongerna. Reglagen som placerades i den andra kartongen fick en förlängningskabel med hjälp av nätverkskabeln. Det fanns totalt 30 kablar som behövde förlängas och en nätverkskabel innehåller åtta kablar. Nätverkskabeln klipptes därför i fyra delar, på så sätt fanns det tillräckligt med skarvsladdar och reglagen kunde placeras minst en meter ifrån varandra. För att göra skarvningarna stabilare löddes alla kablar och insulerades med eltejp.

Programvaran *WinIPAC* användes för att ställa in vilken inmatning respektive reglage skulle skicka, som presenteras i Figur 4.2. Knapparna skulle representera bokstäverna *E,R,T,Y,H,J,K,L*, spakarna programmerades till *W,A,S,D* och pilarna på tangentbordet, sist justerades rattarna baserat på musens läge i X- respektive Y-led.



Figur 4.2: Funktionaliteten och layout av två reglage uppsättningar

Eftersom knapparna var belysta kopplades även dess lampor till IPAC-kretskortet. Dessa styrdes via det medföljande *Software Development Kit (SDK)*. Det är ett bibliotek med funktioner som användes till att styra lamporna i varje knapp. Knapparnas färger kalibrerades som röd, blå, grön och gul för att representera objekt i spelet, knapparna har statiska färger för att följa designprinciper som nämns i avsnitt 2.1. Biblioteket är skrivet i programmeringsspråket C++, vilket skapade komplikationer med att integrera belysningen i Unity. Istället skapades en extern applikation som justerade knapparnas belysning till de bestämda färgerna som visas i Figur 4.2.

4.3 Spelmotor

Användandet av en spelmotor underlättar utvecklingen av spelet då det tillhandahåller en mängd olika verktyg som förenklar implementation av de delar som bygger upp spelet. Spelvärlden, objekt, spellogik och regler är exempel på dessa delar. Det är också i spelmotorn som texturer och ljud hanteras efter att de är framtagna. Valet av spelmotor blev Unity. Detta bestämdes tidigt eftersom några medlemmar har tidigare erfarenhet med spelmotorn, samt att den är välkänd för utveckling av 2D-spel.

Spelet använder sig av Unitys 2D-läge som bas. Det innebär att en scen i spelmotorn har X-axeln riktad horisontellt, Y-axeln vertikalt, och Z-axeln “in i skärmen”. En scen innehåller alltid en kamera som sänder den visuella informationen till en skärm. För att spelet ska kunna visas på två skärmar

behövde varje scen innehålla två kameror, en som är riktad till *Display 1* och den andra till *Display 2*. I scener kan spelobjekt läggas till, även kallat *GameObjects*. På spelobjekt kan komponenter läggas till, vilket är funktioner som appliceras på spelobjekt. Funktioner skrevs i så kallade C#-script som använder sig av klasser för att hantera objektinformation. Klasserna innehåller som grund en *Start()*-funktion som körs en gång när scenen startar, och en *Update()*-funktion som körs en gång varje *frame*, vilka kom till stor användning under projektet. Det är dock inget krav för C#-script att använda dessa funktioner.

Unity innehåller även färdiga komponenter som kan appliceras på spelobjekt. För att hantera objekts texturer innehöll varje spelobjekt komponenten *Sprite Renderer*. För att infoga ljud till spelet användes den färdiga komponenten *Audio Source* som spelade upp musik i en scen.

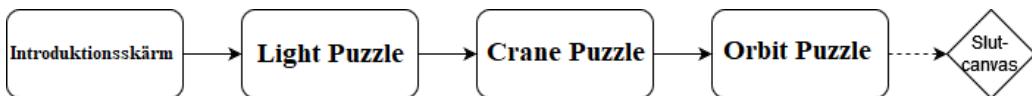
Övergångar mellan scener hanterades i scriptet *LevelLoader*, som laddar scener då nivåernas olika krav är uppfyllda. För en mjukare övergång mellan scener användes funktioner som applicerade fördjörningar och *fade outs* på scenen.

Inmatningar från fysiska reglage hämtades automatiskt genom datorn eftersom reglagen är kopplade till tangentbordsinmatningar. För att hantera inmatningar till spelet behövde funktioner hämta information när spelaren tryckte på en viss tangent som var kopplad till reglagen, till exempel med den inbyggda funktionen *Input.GetKeyDown* som kör en funktion när en viss tangent är nedtryckt. Eftersom ratten hanterade information från musen hanterade exempelvis funktionen *Input.GetAxis("Mouse X")* inmatningar när spelaren rörde på musen horisontellt, som var kopplad till ena spelarens ratt.

Spelet innehöll även ett script med en funktion kallad *IdleTimeout*, som laddade spelets startskärm ifall tre minuter hade passerat utan några inmatningar till spelet. Denna implementerades för att säkerställa att spelet återvände till ett startläge ifall spelare i den publika miljön lämnade utan att spela färdigt spelet.

4.4 Scener

Spelet är indelat i fyra scener som representerar olika pussel. Den första scenen är en startskärm som alltid är påslagen när spelet inte spelas. Efter att spelarna startat spelet tas de vidare till Light Puzzle, följt av Crane Puzzle och slutligen Orbit Puzzle. Vid fullföljande av pusslet visas en slutskärm för spelet. Figur 4.3 visar spelets scener i ordning.



Figur 4.3: Representation av spelets scener och stadier i ordning, från start till slut.

4.5 Startskärm

Den tekniska implementationen av startskärmen består av ett script som övervakar vilka knapptryck som spelarna gör via reglagen. En if-sats kollar ifall båda spelarna tryckt på samma reglageknapp och börjar då förstora planeter (*GameObjects*) i scenen. Denna funktion är tänkt att ge spelarna en snabb introduktion till samarbetsaspekten. Samtidigt startar ett partikelsystem som representerar stjärnor. Det ser då ut som att karaktärerna i spelet håller på att kraschlanda på en planet. Efter förstoringen är klar kallas *LevelLoader* för att starta det första pusslet.

4.6 Light Puzzle

I spelets första pussel måste spelarna flytta på en lampa höger och vänster för att avslöja två färgkombinationer som spelarna måste trycka in för att avsluta nivån. Spelarna befinner sig inte i samma rum inuti spelet, och ser därav olika saker på sina skärmar. Båda spelarna ser varsin tavla med fyra cirklar som representerar färgkombinationerna de måste inmata.

Den ena spelaren ser ett släckt rum med en lampa som lyser upp innehållet i rummet. Lampans ljus innehåller en *Sprite Mask* som tillåter valda spelobjekt med *Sprite Renderer*-komponenten, som hanterar hur ett spelobjekt visas, att synas då masken positioneras över den. När den andra spelaren flyttar på lampan visas rummets objekt, bland annat en tavla som visar två färgkombinationer med fyra färger som vardera spelare ska trycka in med de färgade knapparna.

Den andra spelaren befinner sig i ett tånt rum där det finns ett valv och ett fönster där den tidigare nämnda lampan syns. Andra spelaren kan, som tidigare nämnt, flytta lampan som finns i ena spelarens rum. Detta gör spelaren med sin ratt, och när spelaren vrider den vrider även valvet i rummet och lampan i fönstret förflyttas höger eller vänster. Detta är implementerat på så sätt att när spelet hämtar inmatningar från ratten ökar eller sänks lampans position i x-led, och en rotation appliceras på valvobjektet. Spelaren kan även använda sina färgade knappar till att mata in en färgkombination.

Båda spelarna har varsin färgkombination att mata in, där ena spelaren ser sina färginmatningar på en rad och den andra spelaren på den andra raden. Varje knapptryck från spelaren samlas i en lista och när den listan har fyra knapptryck jämförs det med den korrekta kombinationen. Om den är rätt går spelarna vidare, annars töms listan och processen börjar om.

4.7 Crane Puzzle

Kranpusslet består av tre huvuddelar, att kunna flytta på kranen med joysticken, att kranen kan plocka upp objekt och att två objekt kan paras ihop. När en användare rör på joysticken förs kranen mellan två positioner med en bestämd längd. Om användaren vill plocka upp ett objekt från golvet åker kranen ner och längst ner på sin färd skjuts en *raycast* ut. Denna raycast är en stråle som skickar information om något som den träffar. Vid en träff skickas positionen av ett objekt som kopplas samman med positionen av kranen vilket leder till att de translateras tillsammans. När kranen är över arbetsbänken kopplas positionerna bort och objektet hamnar på bänken och sedan placeras i en lista. Varje objekt som kan plockas upp av kranen har en tagg som berättar vilket objekt som den kan paras ihop med. Om det finns två objekt i listan med samma tagg räknas det som en korrekt kombination och annars flyttas objekten tillbaka till sin ursprungliga position.

4.8 Orbit Puzzle

I detta pussel styr båda spelarna en satellits omloppsbana runt en planet med hjälp av reglagerattarna. Den ena spelaren styr radien och den andra styr hastigheten i omloppsbanan. Det finns också en siluett som har en egen omloppsbana runt planeten. När spelarna lyckats matcha siluettens omloppsbana dyker det upp fyra färgknappar framför satelliten i omloppsbanan. Båda spelarna ska trycka på motsvarande reglageknapp när satelliten når en färgknapp och totalt finns det fyra stycken. Därefter är pusslet avklarat.

Den tekniska implementeringen består av två olika spelstadier. I det första spelstadiet ska spelarna styra satelliten. Som nämnt tidigare kopplas rattarna till datormusens position. Den ena ratten flyttar på musens x-position och den andra dess y-position. Dessa värden normaliseras, baserat på skärmens

storlek, och användes sedan för att skala vinkel respektive längd på en hypotenus, alltså avståndet, som beräknas med hjälp av de trigonometriska funktionerna *cosinus* och *sinus*. Se Figur 4.4.

```
radiusSatelite = Input.mousePosition.x/Screen.width;
velocitySatelite = (Input.mousePosition.y - Screen.height/2) / (Screen.height/2);

//Moving satelite
angleSatelite += velocitySatelite * Time.deltaTime;

Vector2 offset = new Vector2(Mathf.Sin(angleSatelite), Mathf.Cos(angleSatelite))*radiusSatelite;
Satelite.transform.position = offset;
```

Figur 4.4: Kodimplementasjon för att föra en satelliten i en omloppsbanan samt ändra på omloppsbanan med hjälp av musens position.

I det andra spelstadiet har spelarna lyckats matcha satellitens omloppsbanan med siluettens omloppsbanan, vilket kallas genom att räkna avståndet mellan satelliten och siluettens positioner. Har avståndet varit tillräckligt kort inom ett lämpligt tidsintervall aktiveras andra spelstadiet. Vid detta moment så inaktiveras rattarna för att spelarna har uppnått det första delmålet.

På samma sätt som satelliten matchar siluettens omloppsana, kan satellitens position i nästa stadio jämföras med positionen av de färgade knapparna som tillkommer i denna fas. Varje färg är kopplad till två knappar, en för vardera spelare, som då måste tryckas när satelliten är i rätt position. Då spelarna lyckas trycka på rätt knapp vid rätt tillfälle får de ljudåterkoppling i form av ett pling. Trycker de på fel knapp, eller misslyckas med att trycka i tid, får de ljudåterkoppling enligt 3.5. Efter att båda spelarna lyckats med utmaningen fyra gånger så är pusslet, och därmed spelet, avklarat.

4.9 Användartester

Användartester är en viktig del i systemutveckling. Till följd av den agila utvecklingsmetodiken krävs användartester eftersom utvecklingsförloppet följer en iterativ metod. En undersökning skulle utföras i slutet av varje test där deltagarna fick svara på frågor i form av ett formulär. Formuläret syftar till att besvara specifika frågor gällande användarupplevelse och dylikt. Med data från användartester kunde systemet testas igen för att mäta framgången av den nya iterationen. Dessa mätningar gav ett resultat som ämnar att besvara frågeställningarna i avsnitt 1.3.

Det utfördes totalt tre användartester under utvecklingsprocessen: Den första med enbart Figma-prototyper, det andra vilket använde en MVP som implementerats i Unity med de mest nödvändiga funktionerna och objekten, samt det sista på spelmässan *Lincon*. Användartesterna genomfördes med personer i åldersgruppen 20 - 30 år. I ett försök att öka relevansen av användartesterna försökte gruppen inrikta sig på personer som hade nära koppling till målgruppen, och därav en djupare förståelse av deras vanor och beteenden. Därmed svarade testare på frågor gällande hurvida de kände personer i den sökta målgruppen samt om de trodde att barn skulle klara av spelet.

Användartesterna började med en introduktion där en av utvecklarna presenterade omgivningen som spelet utspelar sig i. Utvecklaren presenterade även relevant information angående testerna, exempelvis att de utvecklare som befann sig i rummet inte var tillgängliga för att hjälpa testarna, förutom då utvecklarna kunde konstatera att en bugg uppstått. Utvecklarna uteslöt också informationen att det var menat för testarna att kommunicera och samarbeta, för att efterlikna den publika miljö som slutprodukten skulle komma att befina sig i. Resterande utvecklare i rummet satt även och antecknade sina iakttagelser av testarna under testets gång. Efter testerna var klara diskuterade utvecklarna deras iakttagelser och svaren i formuläret för att sammanställa resultatet.

Det fanns ett par skillnader mellan första och andra användartesterna. Den största skillnaden mellan

användartesterna var att spelet, i första användartestet, bestod av en prototyp skapad i Figma. Prototypen var simplistisk och utvecklare behövde kontinuerligt uppdatera spelarnas vy under testets gång för att simulera visuell återkoppling. Under det andra användartestet var spelet implementerat i Unity och var ett fungerande spel från början till slut. En annan skillnad mellan testerna var formulärens innehåll. Det första användartestet var mer inriktat på att ta reda på vad testare tyckte om spelet i helhet. Därför innehöll första formuläret mindre frågor som var inriktade på vad de tyckte om pussel och reglage. Inte lika mycket design. Exempelvis: *Hur uppenbart var det att förstå vad man skulle göra för att lösa pusslet?* Andra användartestet innehöll ett större antal frågor samt ett bredare fokus. Detta innebär att andra användartestet inte bara innehöll frågor om pusslena utan formuläret innehöll frågor om exempelvis design, placering av reglage och allmänna frågor.

Ordningen på pussel under första och andra användartestet skiljde sig något. Första pusslet Planet Puzzle på det första användartestet innehöll en del som gruppen tog beslut om att ta bort inför nästa användartest. Andra och tredje användartestet innehöll även Orbit Puzzle som inte var med på första, alltså en o-testad spelidé, samt startskärmen. Följande pusselordning användes under det första användartestet: Planet Puzzle, Light Puzzle och Crane Puzzle. Pusselordningen för det andra och tredje användartestet var istället: Startskärm, Light Puzzle och Crane Puzzle och Orbit Puzzle.

Det sluttgiltiga användartestet, alltså det tredje, utfördes på spelmässan LinCon, där många besökare var spelintresserade och dessutom tillhörde många spelets målgrupp. I detta stadie var spelet i slutfasen där bland annat ljudåterkoppling, nya animationer samt start- och slutskärm hade implementerats. Detta användartest var unikt till skillnad från de tidigare eftersom spelet var utställt i en publik miljö. Detta test använde inget formulär, men på spelmässan hölls en diskussion med alla spelarna i syfte att samla information om spelets upplevelse. Dessutom iakttog utvecklarna varje spelomgång.

Kapitel 5

Resultat

I detta kapitel kommer den skapade slutprodukten att presenteras. Slutprodukten utgörs av två lådor med fysiska reglage som är kopplade till IPAC, samt spelet som i sig består av fyra separata scener. För att koppla samman reglagen med datorn används en USB-kabel och för att försörja de belysta knapparna med ström så användes en nätagadapter.

5.1 Fysiska reglage

Figur 5.1 visar placeringen av de fysiska reglagen. Det används ett IPAC kretskort för båda stationerna. Kretskortet placerades i en av lådorna som sedan kopplades med förlängningskablar till den andra lådan med reglage.



Figur 5.1: De fysiska reglagen och kretskortet.

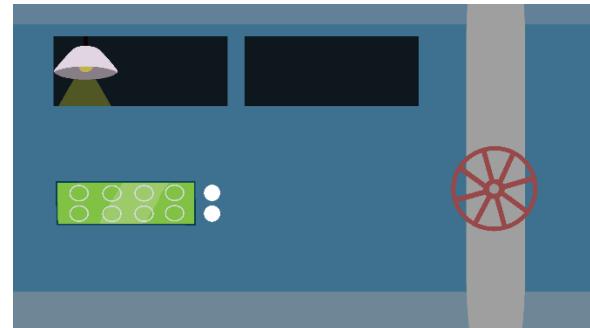
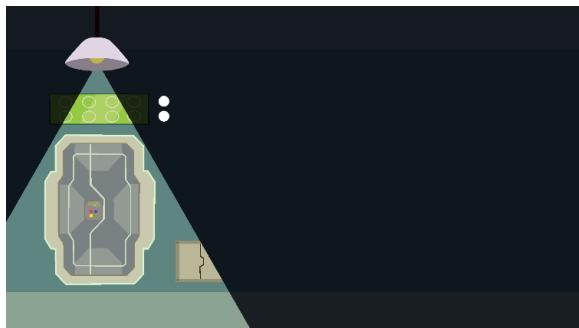
5.2 Slutresultat på spelet

Startskärmen, i Figur 5.2, innehåller två cirklar. När spelare 1 trycker på en av färgknapparna lyser den vänstra cirkeln upp i respektive färg. När spelare 2 trycker på en av färgknapparna lyser höger cirkel upp i respektive färg. När dessa cirklar har samma färg påbörjas första pusslet.



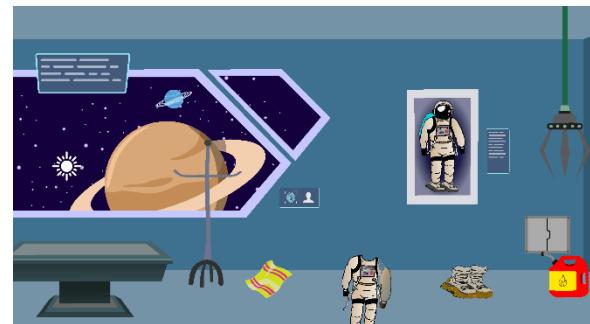
Figur 5.2: Startskärmen för båda spelarna.

Light Puzzle, i Figur 5.3, är ett pussel där spelare 2 ska förflytta positionen på lampan. Denna lampa kommer då lysa upp fler objekt för spelare 1. När lampan är placerad längst till höger visas två färgkoder. Dessa färgkoder representerar de kombinationer som spelarna ska trycka på färgknapparna. När de klarat detta går spelet vidare till nästa pussel.



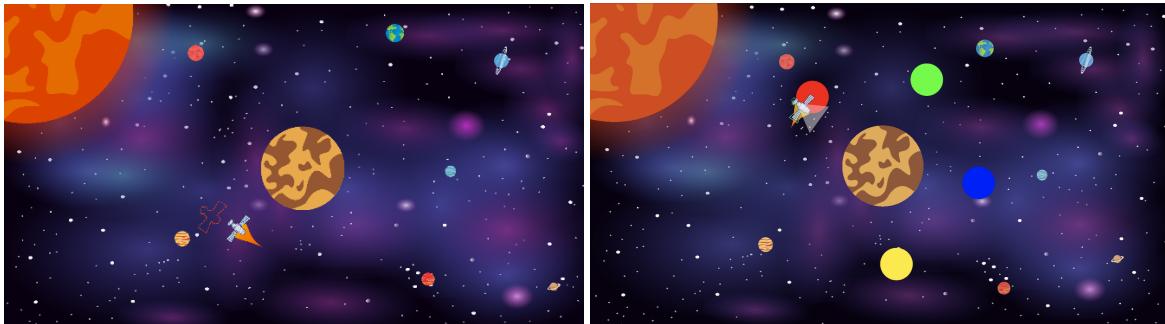
Figur 5.3: Första pusslet Light Puzzle. Figuren representerar vyn för spelare 1 respektive spelare 2.

Figur 5.4 visar resultatet av det andra pusslet. I detta pussel kan båda spelare röra vardera kran höger och vänster. Kranarna kan även plocka upp de objekt som är placerade under kranen, för att sedan lägga dom på bordet. När två objekt paras ihop placeras objekten på deras respektive startposition om det var fel kombination eller så placeras de på klädställningen utifall att spelarna har parat ihop rätt objekt. När alla objekt är på klädställningen så är pusslet klart.



Figur 5.4: Andra pusslet *Crane Puzzle*. Figuren representerar vyn för spelare 1 respektive spelare 2.

Orbit Puzzle, i Figur 5.5, är ett pussel där spelarna ska samarbeta för att styra en satellit till den rätta omloppsbanan. Spelare 1 styr vilken avstånd satelliten ska ha från planeten medan spelare 2 styr hastigheten på satelliten. När satelliten är i den korrekta omloppsbanan börjar satelliten ta bilder av planeten. När satelliten är i rätt omloppsbanan ska båda spelare trycka in en färgkombinationen för att klara av pusslet.



Figur 5.5: Tredje pusslet Orbit Puzzle. Figuren representerar vyn för spelarna.

Följande ordning blev den slutgiltiga pusselordningen: Startskärm, Light Puzzle, Crane Puzzle och Orbit Puzzle.

5.3 Användartester

Nedan ges resultatet från projektets samtliga användartester.

5.3.1 Användartest 1

Formuläret i det första användartestet bestod av åtta frågor per pussel. Dessa frågor upprepades på vardera pussel. Första pusslet gav i helhet en negativ reaktion medan andra pusslet gav i helhet en positiv reaktion. Nedan presenteras en sammanställning av svar från formuläret samt utvecklarnas iakttagelser för vardera pussel.

Planet Puzzle & Light Puzzle

- Hur uppenbart var det att förstå vad man skulle göra för att lösa pusslet?

Majoriteten av svaren inkluderade att det inte var självklart att man skulle samarbeta i början av pusslet. Det stod även att det skulle varit tydligare vad syftet av första pusslet var.

- Hur mycket behövde du kommunicera med din medspelare? Kändes det likvärdigt?

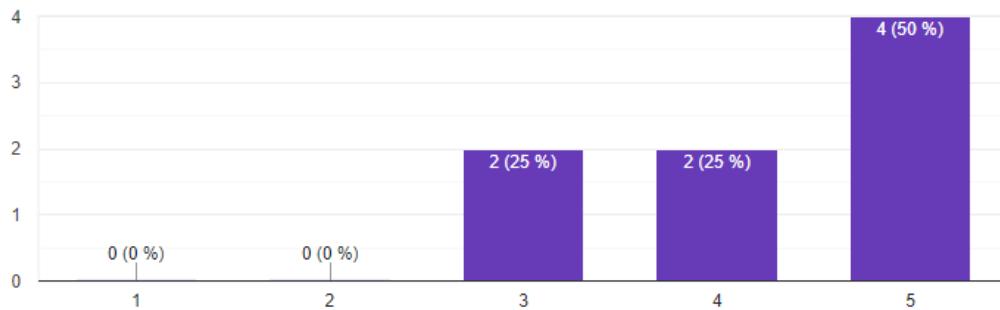
Det vanligaste svaret var att de behövde kommunicera mycket vilket var positivt. Många sa även att det kommunikationen kändes likvärdig. Ett par testare löste det helt utan att kommunicera, då de till en början trodde att det var en tävling, samt att den ena personen visste svaret på pusslet då det var ett faktabaserat svar.

- Saknades det någon vägledning?

Denna fråga fick en hel del blandade svar. Några tyckte att det var bra att det fanns mindre vägledning då det var kul att försöka lista ut vad de skulle göra. Några testare kände sig vilsna och skulle tyckt det var bra om några tips fanns om en spelare tappar spelets röda tråd. Tre testare

föreslog en startskärm med instruktioner. Nästan alla testare ansåg att det saknades vägledning, vare sig detta var positivt eller negativt.

Figur 5.6 visar resultatet av hur roligt testarna upplevde att första pusslet var. Testarna kunde välja mellan alternativen 1 till 5. Nummer 5 indikerar på att testaren tyckte spelet var mycket roligt och nummer 1 indikerar på att testaren tyckte att spelet var mycket tråkigt.



Figur 5.6: Statistik på svar om hur kul användare tyckte pussel 1 var. 1 är mycket tråkigt och 5 är mycket kul.

Crane Puzzle

- Hur uppenbart var det att förstå vad man skulle göra för att lösa pusslet?

Hälften av alla testare tyckte att det var uppenbart. Andra hälften tyckte att det blev uppenbart efter de hade testat sig fram. Därmed var det uppenbart efter spelarna hade kommunicerat och pratat om vad de såg på deras skärmar. Ett par testare förstod inte helt att vissa objekt hängde ihop men det var enkelt att testa sig fram enligt testarna.

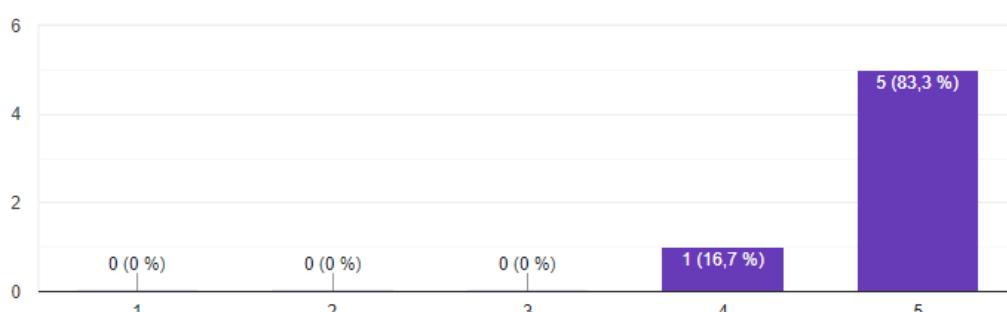
- Hur mycket behövde du kommunicera med din medspelare? Kändes det likvärdigt?

Alla testare skrev antingen mycket eller likvärdigt.

- Saknades det någon vägledning?

Alla skrev nej men vissa svar inkluderade även följande: *"men kanske ett par hints om man fastnar"*.

Figur 5.7 visar resultatet av hur roligt testarna tyckte att andra pusslet var. Detta gav ett resultat som indikerar på att testarna tyckte att andra pusslet var roligare än första då 83% av testarna valde nummer 5 som svar.



Figur 5.7: Testares svar på hur kul de tyckte pussel 2 var, där 1 är mycket tråkigt och 5 är mycket kul.

Utvecklarnas iakttagelser

Första pusslet var ett svårt pussel. Spelare 1 fick ingen återkoppling på sin skärm föreens pusslet var klart då hen inte behövde göra något för att gå vidare. Spelare 1 behövde bara kommunicera svaret med den andra spelaren. Det tog även väldigt lång tid för spelare 1 att fatta att hen hade lösningen som skulle kommuniceras till den andra spelaren. Spelare 2 missförstod, återkommande, uppgiften och fattade inte vad de skulle göra för att klara uppgiften. Det var även återkommande att spelare 2 inte förstod vilka reglage som var menade för användning.

Pussel 2 var ett enklare, och i allmänhet mer uppskattat pussel från båda spelarnas perspektiv. Något återkommande var att början var svår, men efter en kort stund lyckades testarna kommunicera fram lösningen. Efter det var pusslet lätt att lösa för samtliga testare.

5.3.2 Användartest 2

Andra användartestet hade två formulär, ett för testarna som känner barn och ett som inte känner barn. Varje formulär har samma innehåll med 22 frågor. I helhet gav testerna positiva åsikter om pusslena förutom Orbit Puzzle där testerna möte på svårigheter under pusslet. Nedan presenteras en sammanställning av de mest relevanta svar från formuläret samt utvecklarnas iakttagelser.

- Något som var otydligt med designen? Visste du exakt vad det var för objekt? Om inte, vilket/vilka? Varför var de otydliga?

Majoriteten av designen till objekten tyckte testarna var tydlig. De tyckte att designen var välanpassad för barn, enkel och inte för mycket distraktioner. En spelare tyckte att spelet liknade ett annat barnspel de hade spelat när de var yngre. Ett objekt som var otydligt var fönstret i Light Puzzle där de önskade tydligare ramar på fönstret. Handduken var även lite svår att förstå att den passade med hjälmen.

- Löste ni pusslet jämlikt mellan er?

I helhet svarade testarna ja, vissa tyckte dock att medspelaren fick kommunicera mer, men de ansåg att de kunde tagit mer initiativ själv och kommunicerat mer. De upplevde att de fick göra mindre på deras skärmar.

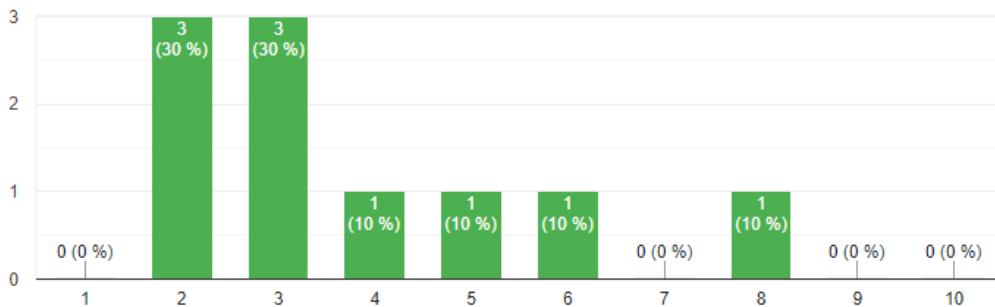
- Fördelades uppgifterna jämnt mellan er? Om nej, varför?

I helhet tyckte testarna att uppgifterna var jämt fördelade men resultatet visade att vissa pussel var lite ojämna i uppgifter. Nästan alla tyckte att det var bra fördelat. Men två personer tyckte att de hade lite mindre att göra än sin medspelare i Light Puzzle.

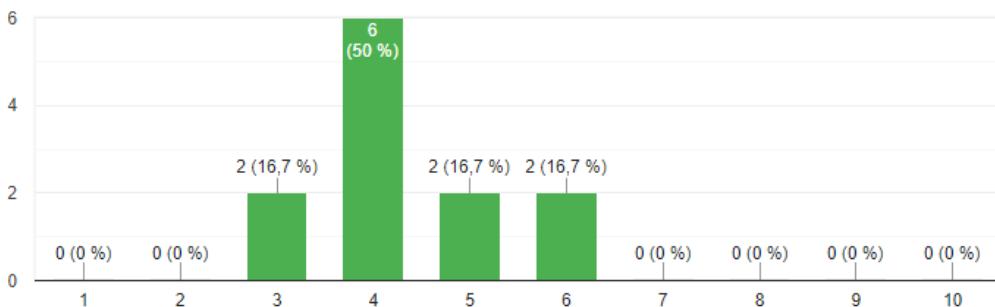
- Använde du alla reglage lika frekvent, eller skulle det ha varit önskvärt med ytterligare funktioner för något av dem?

Det var bra användning av alla reglagen. Testarna önskade att man fick använda reglagen ännu mer, främst på Orbit Puzzle. Dessutom önskade de att knapparna skulle användas mer.

I Figur 5.8 och 5.9 visas resultatet för hur svårt testarna tyckte att spelet var. Testarna kunde välja mellan alternativen 1 till 10. Nummer 10 indikerar på att testaren tyckte spelet var mycket svårt och nummer 1 indikerar på att testaren tyckte att spelet var mycket enkelt. Figur 5.8 och 5.9 skiljer sig med svar från testerna där testare kände respektive inte kände barn. Figur 5.8 visar ett medelvärde på 3.8 medan Figur 5.9 visar ett medelvärde på 4.3.



Figur 5.8: Testare som känner barn svarade på hur svårt de tyckte spelet var, där 1 är mycket enkelt och 10 är mycket svårt.



Figur 5.9: Testare som inte känner barn svarade på hur svårt de tyckte spelet var, där 1 är mycket enkelt och 10 är mycket svårt.

Utvecklarnas iakttagelser

Orbit Puzzle uppskattades av testarna men den hade en bugg. Datorns muspekare, som styrs av ratten, fastnade återkommande på en viss position vilket gjorde det svårt för spelarna att klara spelet. Även Light Puzzle var ett uppskattat pussel som i regel gick snabbt att lösa. Pusslet Crane Puzzle var för vissa grupper ett enkelt pussel men för andra grupper var det svårt. Dock var det bara svårt till en början men när testarna listade ut vad de skulle göra, gick det snabbt att lösa.

5.3.3 Användartest 3

Alla besökare på LinCon tyckte spelet var kul. Vissa hade det svårare än andra. Besökare med spelvana klarade spelet utan vägledning. En kommentar från ett barn, sju år gammal, sa: *"Why did you even make this game!"* i frustration över att ha fastnat i Orbit Puzzle. Vissa barn inom målgruppen fick hjälp av sina föräldrar att klara spelet. De besökarna som komмуunicerade med varandra under spelets gång hade det lättare att klara av spelet än de som inte gjorde det. De flesta besökarna hade svårt att klara av Orbit Puzzle. En observation som gjordes var att majoriteten av spelare vid startskärmen prövade att interagera med alla reglage utan att komмуunicera, och trots detta kunde ta sig vidare från startskärmen.

Kapitel 6

Analys och diskussion

Under detta kapitel analyserar och diskuteras arbetet som utförts under arbetetsgången. Hur problem hanterats som uppstått under tiden beskrivs i detta kapitel där eventuella förändringar som behöver göras nämns.

6.1 Design

Under projektet har gruppen lagt stort fokus på att skapa en design med målgruppen i åtanke och samtliga medlemmar känner att detta har uppnåtts till övervägande del. Mycket är tack vare designprinciperna som tagits fram under förstudien, se avsnitt 3.1, samt användartester som gav möjlighet till en iterativ utvecklingsprocess. Trots detta finns det flera delmoment som varit bristande ur denna synpunkt. Till att börja med så uppstod problem med Planet Puzzle, som under det första användartestet visade sig sakna flera av de designprinciper som varje pussel skulle förhålla sig till. Framförallt gällde detta en avsaknad av tydliga mål och en brist på återkoppling. Dessa problem är något som, enligt designprocessen för pussel (se avsnitt 3.3), bör ha identifierats före skapandet av en prototyp. En anledning till att detta var att förstudien utfördes parallellt med att pusselidéer började utformas, och därmed fanns inte designprinciper som vägledning i det tidiga stadiet av pusseldesignen. Ett annat problem med detta pussel var att en spelare fick för lite att göra, vilket helt gick emot ett av spelets målsättningar, d.v.s. att båda spelarna skulle uppleva att de utför lika många uppgifter. Detta överseende var inte relaterat till designprinciperna utan berodde istället på att Planet Puzzle ansågs vara en del i ett större pussel tillsammans med Light Puzzle. Med detta resonerades att även om en spelare fick göra mindre i en del av pusslet så gjorde det inget så länge nästa del erbjöd mer till denna spelare. Här visade sig därmed användartester vara en kritisk komponent i utvecklingen för att kunna försäkra att spelet skulle kunna uppnå sitt syfte.

Ett annat problem som upptäcktes i samband med användartest av Planet Puzzle var gällande hur tydligt spelet förmedlar att spelarna är tillåtna att samarbeta. Som nämns tidigare i avsnitt 4.9 så gavs inga uttryckliga instruktioner gällande spelets regler, istället gavs endast en beskrivning av den tänka miljön som spelarna skulle befina sig i. Detta var för att efterlikna ett scenario och en miljö och likt den slutprodukten har designats för. Att Planet Puzzle under denna fas utgjorde starten på spelet, i kombination med att pusslet hade otydliga mål, kunde synliggöra att spelarna till en början hade svårt att förstå att de skulle samarbeta. För att lösa detta valde gruppen att inkludera en enklare startskärm, se Figur 5.2. Som går att se i resultatet av det sista användartestet, se avsnitt 5.3.3, så lyckades startskärmen med att lösa detta problem fullt ut. Problemet grundar sig i att funktionen som kontrollerar att spelare klickar samma knapp är för simplistisk. Detta ger upphov till att spelarna slumpmässigt lyckas trycka på samma knapp utan att de är medvetna om innebörden av detta, vilket minimerar syftet med en startskärm. För att undvika detta skulle en något mer invecklad funktion som

inte ger möjlighet till att slumpmässig interaktion med reglagen inte är en effektiv strategi.

6.2 Teknisk implementation

Genom att välja IPAC-kortet kunde gruppen dra nytta av flera fördelar för att skapa kontroller till spelet och uppnå det övergripande målet med projektet. För det första erbjuder IPAC-kortet möjligheten att ansluta rattar, till skillnad från de övriga reglagen, som använde fysiska strömbrytare, skickar rattarna information i form av en analog signal som kan ha olika värden. Detta var viktigt eftersom vissa pussel i spelet krävde nyanserade rörelser och inte enbart enkelriktad styrning.

En annan fördel med IPAC-kortet är dess flexibilitet när det gäller kartläggning av knappar till datorns olika inmatningar. Det betyder att varje knapp kan kopplas till en unik tangentbords- eller musinmatning. Detta gav gruppen mer frihet att skapa en varierad och anpassningsbar spelupplevelse. Eftersom reglagen även införskaffades från samma leverantör så blev det enklare att koppla ihop allting då kortet skapades för detta syfte.

En ytterligare fördel med IPAC-kortet är dess dedikerade LED-kontroller. Detta gör det möjligt att styra och anpassa belysningen av knapparna genom programvaran. Man kan till exempel justera färgerna på de belysta knapparna för att passa spelets design eller till och med lägga till ytterligare belysningselement runt reglagen för att skapa en mer attraktiv visuell upplevelse.

I jämförelse med Arduino-kortet erbjuder IPAC-kortet en mer specialiserad och färdig lösning för att koppla upp och styra fysiska reglage till datorn. Att använda en Arduino skulle kunna ge gruppen större flexibilitet och kontroll över reglagens funktionalitet. Genom att programmera Arduino-kortet i C# och integrera det direkt i Unity-miljön skulle det vara möjligt att skapa anpassade och avancerade funktioner som är specifika för spelet. Detta skulle kunna öppna upp för en mer skräddarsydd spelupplevelse och ge möjlighet att förbättra spelmekanismer och interaktioner. Även om Arduino skulle möjliggöra utvecklingen av mer avancerade funktioner och direkt integration med Unity, så skulle det också kräva mer tid och arbete för att implementera och anpassa reglagens funktionalitet till spelet.

En begränsning med IPAC och Arduino är det maximala antalet inmatningar som systemen kan hantera[18][19]. Den billigaste modellen som går att köpa tillåter upp till 32 knappar utan möjligheten till kontroll av belysning. För de flesta spel som kräver inmatning från flera spelare är detta mer än tillräckligt. Men om projektet kräver fler än 32 knappar kan det vara nödvändigt att uppgradera till en mer avancerad version av IPAC eller leta efter andra alternativ. En grundläggande Arduino har möjligheten att koppla 14 knappar vilket inte var tillräckligt för syftet av detta projekt, en mer avancerad version hamnade i samma prisklass som en grundläggande IPAC så det blev det självklara valet.

När det kommer till metodens replikerbarhet och reliabilitet av projektet så finns det punkter att ta upp. Det finns möjlighet att replikera både studiens vetenskapliga aspekter och själva spelet genom att läsa rapporten. Rapporten säger inte allt, såsom detaljerad kodimplementation, men metoden beskrivs på ett tillräckligt djup för en grupp studenter ska kunna återskapa ett liknande spel. I avsnitt 2 presenteras information om studier och andra spel som inspirerat gruppen. Designprinciperna som presenterades i avsnitt 3.1 togs fram för att hjälpa gruppen besvara undersökningsfrågorna som ställdes på detta projekt. Eftersom designprinciperna kom från forskningsartiklar som har undersökt effektiviteten av dessa metoder så har det förstärkt reliabiliteten av projektet.

6.3 Användartester

Testare som deltog i det första och andra användartestet fick en introduktion från utvecklarna, tidigare nämnt i avsnitt 4.9, som presenterade omgivningen där slutprodukten var tänkt att vara monterad. Under testens gång observerades testarnas interaktion med spelet och därefter fick testarna svara på ett formulär. För att säkerställa att korrekt och användbar information insamlas, kunde gruppen använda flera metoder för att samla in data. Till exempel kunde gruppen använt skärminspelningar, berättat utförligare information rörande spelet eller till och med intervjuat testdeltagarna efter att de spelat klart. Detta skulle ge en heltäckande bild av användarupplevelsen, och hjälpa att påvisa eventuella buggar stötts på under testet. En förändring som skulle kunnat göras är att ge testarna mer information, förslagsvis kunde de fått veta att det är ett samarbetspel samt att deras vyer är olika. Just denna information hade dock skadat resultatet av användartesterna, då viss relevant återkoppling hade försummats. Detta kan ses i exempelvis resultatet av det första användartestet (5.3.1), som påvisade att samarbetskravet behövde förtydligas i spelet, vilket gjorde att en startskärm inkluderades. Hade testare fått veta innan testet att de skulle samarbeta så hade detta problem inte kunnat identifierats. Ett bekymmer med de två första användartesterna var att testarna inte utfördes med användare från den önskade målgruppen, utan istället på studenter i åldern 20 - 30 år. För att förbättra resultatet behövde testerna utföras av den önskade målgruppen barn, vilket försämrade valideten på användartesterna. Exempelvis skulle flera tester med den faktiska målgruppen kunnat ge insikt i hur samtliga pussel uppfyller flertalet av designprinciperna, såsom hur bekanta objekt och symboler används eller hur väl interaktiva objekt urskiljer sig från bakgrunden. Lösningen med att försöka hitta användartestare med koppling till målgruppen var till viss grad en effektiv lösning, men användning av den faktiska målgruppen skulle vara optimalt.

6.4 Potentiell vidareutveckling

Projektet pågick under en relativt kort period sett till projektets skala, vilket gjorde att vissa moment inte gick att planera in, därav hade många moment och delar av projektet kunnat vidareutvecklas. Det hade varit möjligt att vidareutveckla objekten med mer detaljerade inslag samt att göra bakgrunden mer verklighetstrogen. Dessutom hade ljuskällor kunnat läggas till, vilket hade skapat ett realistiskt djup i scenen genom skuggor och ljusriktning. Detta hade fungerat bra i scenen som visas i Figur 5.3, då den hårda kanten hade ersatts av en ljusstråle. En annan aspekt som hade gjort spelet mer levande är animering av flera objekt. Flera delar av spelet är statiska, tillägg av fler rörliga delar samt en mer detaljerad rörelse hade resulterat i mer dynamiska scener.

Ytterligare förbättringar som kunnat gjorts är förflyttning av spelarna mellan scener i spelet. För att förmedla en berättelse skulle en introduktionsvideo och animerade övergångar implementerats. Introduktionsvideon skulle visat spelaren slutmålet och bidra med en förståelse för varför spelet utspelar sig i rymden och ge allmän kontext till spelets pussel. Övergångarna hade hjälpt till att leda spelare från pussel till pussel, då ett tydligt problem från användartesterna var att spelarna inte förstod var de befann sig i förhållande till varandra.

Även om det i nuvarande versionen av spelet används en hel del återkoppling i form av både ljud och visuella aspekter, så går detta att förbättra och vidareutveckla, framförallt med utökad användning av ljud och ljus. Ljus kan användas för att ge använderen återkoppling i form av interaktiva ljus på de fysiska reglagen, för förmedla klickbarhet under något av spelets moment. Ett annat användningsområde för ljus är den lockande aspekten, där ljusanimationer i den publika miljön kan användas för att väcka ett intresse hos förbipasserande. En separat funktion gruppen önskat att implementera är små högtalare i behållaren för reglagen. Även detta skulle kunnat fungera som en lockande aspekt i den publika miljön, där högtalarna på reglagen skulle ge ifrån sig ljud. Dessa högtalare skulle dock

främst innebära att interaktion med rattar eller knappar kan ge ljudåterkoppling. Vidare är de belysta knapparna just nu statiska under hela spelförloppet, och här skulle även ljusanimationer kunna läggas till för att förtynliga återkoppling i samband med interaktionen med dessa.

Ljudeffekter används till viss grad under flera av spelets pussel, men för att förbättra spelupplevelsen skulle dock flera ljudeffekter kunna läggas till. Den existerande återkopplingen, både den ljudbaserade och den visuella, är i sina grundläggande stadier och kan förbättras.

6.5 Etisk och samhällelig reflektion

Eftersom spelets målgrupp var barn fanns ett extra stort ansvar när det kommer till etik och samhälleliga aspekter, eftersom det är lätt att påverka barn[20]. Förundersökningen utgör en stor del av det arbete som föreligger att slutprodukten är passande, och inte har innehåll som kan anses som stötande, eller på något sätt problematisk för målgruppen. Även om analysen främsta syfte är att göra systemet attraktivt för användare inom målgruppen så tas även hänsyn till de restriktioner som bör finnas vid användning av explicit innehåll.

En annan viktig aspekt som har behandlats är hur systemet designats för att inte utesluta användare som lider av någon form av funktionsnedsättning. Termen är bred och innehåller flera typer av funktionsvariationer som påverkar barn på olika sätt [21]. På grund av projektets begränsade resurser så var det svårt att implementera lösningar som tar hänsyn till samtliga funktionsvariationer, även om det var en strävan. En bättre lösning var att fokusera på de funktionsvariationer som har enklare lösningar och som faktiskt var genomförbara inom projektets ramar. De funktionsvariationer som påverkade spelets design var epilepsi och färgblindhet. Av dessa låg färgblindhet i fokus då detta kräver mest åtanke vid design av spelet i allmänhet. Som nämnts i avsnitt 2.1, används en ljus och färgglad kombination av färger. Det viktiga var att inte använda färgkombinationer som hindrar färgblindas möjlighet att urskilja aspekter i spelet. Åtgärder för att inkludera barn med epilepsi var enkelt och innefattade att inte använda blinkande ljuseffekter i spelet eller på de fysiska reglagen som var belysta.

6.6 Källkritik

Det är viktigt att kunna bedöma kvaliteten på olika källor. Opålitliga nyheter och information sprids lätt och därför är det viktigt att kunna urskilja vad som är sant och falskt. De främsta källorna för information har varit skrivna av skaparna av produkten såsom I-PAC och Unity egna hemsidor [22][18].

Tidigt i processen genomfördes en målgruppsanalys och en undersökning av designprinciper för att skapa riktlinjer angående vad som bör inkluderas och vad som bör undvikas i spelet. De mest trovärdiga artiklarna kom ifrån kvalificerade nyhetstidningar samt företag som specificerar sig inom UX.

Kapitel 7

Slutsatser

Målet med detta projekt var att planlägga och utveckla ett system som ämnade att öka engagemang och social interaktion i en publik miljö. Eftersom spelet har utvecklats för barn som spelare så implementerades fysiska reglage som inmatningsenhet för att göra spelet mer interaktivt. Flera användartester utfördes för att utvecklingslaget ska få feedback och att projektet håller sig till kundens vision.

7.1 Frågeställning 1

- Vilka designprinciper ska spelet följa för att bibehålla barnens intresse och engagemang i en publik miljö?

Under systemutvecklingen nyttjades de designprinciper som nämns i avsnitt 3.1. Resultatet av användartesterna visade att vissa principer hade mer tyngd än andra.

För att hålla barnens engagemang i spelet är det viktigt att ge frekvent återkoppling. Detta kunde uppnås genom användning av visuell återkoppling och ljudeffekter som ger spelaren bekräftelse på framsteg. Förutom detta är det viktigt att inkludera tydliga mål för att hålla barnens intresse. På detta sätt kan spelet bibehålla barnens engagemang under längre tid.

7.2 Frågeställning 2

- Hur kan spelet utvecklas så att båda användarna upplever att de har utfört lika många uppgifter efter spelets slut, samt att alla reglage har samma användningsmängd?

En iterativ systemutveckling och noggrann planering av spelets moment är avgörande för att säkerställa att användarna utför likvärdiga uppgifter. För att ytterligare förbättra spelet och uppnå balans i användningen av reglage kan användartester som utformats efter projektet målsättningar genomföras. Genom att användare, i synnerhet de inom målgruppen, får tillgång till spelet eller delar av det och sedan samla in deras återkoppling kan man utvärdera om användarna upplever att de har utfört lika många uppgifter. Dessa tester kan också påvisa om användarna upplever att användningen av reglagen är jämnt fördelad mellan de olika reglagen. Genom att kombinera noggrann planering och användartester kan man sträva efter att skapa en optimal spelupplevelse där användarna känner sig jämlika och engagerade.

Litteraturförteckning

- [1] Banu Manav, *Color-emotion associations and color preferences: A case study for residences*, Color Res. Appl. 2007
- [2] Färger på färghjul som är okej för färgblindhet, *färgblind*, u.å., hämtad: 2023-05-09, Tillgänglig: <https://color.adobe.com/sv/create/color-accessibility>
- [3] Felix Gyllenstig Serrao, *Diagnoser och dataspel en given kombination*, hämtad: 2023-05-24, Tillgänglig: <https://spellararen.wordpress.com/>
- [4] Mariia Kasym, *UX Design for Children: How to Create a Product Children Will Love*, Eleken, u.å., hämtad: 2023-05-09, Tillgänglig: <https://www.eleken.co/blog-posts/ux-design-for-children-how-to-create-a-product-children-will-love>
- [5] Kelly Bower, Paul Krebs, Claudine J. Lamoth, and Elizabeth J. Lyons, *Effective Feedback Procedures in Games for Health*. *Games for Health Journal*, Mary Ann Liebert, Inc. 2013
- [6] The Game Awards, *The Game Awards*, u.å., hämtad: 2023-05-09, Tillgänglig: <https://thegameawards.com/>
- [7] Among Us, *Among Us*, Steam. u.å, hämtad: 2023-05-25, Tillgänglig: https://store.steampowered.com/app/945360/Among_Us/
- [8] We Were Here, *We Were Here*, Steam, u.å, hämtad: 2023-05-25, Tillgänglig: https://store.steampowered.com/app/582500/We_Were_Here/
- [9] It Takes Two, *It Takes Two*, Steam, u.å, hämtad: 2023-05-25. Tillgänglig: https://store.steampowered.com/app/1426210/It_Takes_Two/
- [10] Richard T.A. Wood, Mark D. Griffiths, Darren Chappell, and Mark N.O. Davies, *The Structural Characteristics of Video Games: A Psycho-Structural Analysis*, CyberPsychology and Behavior. 2004
- [11] Pinterest, *Pinterest*, u.å, hämtad: 2023-05-25. Tillgänglig: <https://www.pinterest.se/>
- [12] Pac-Man, *Pac-Man*, u.å, hämtad: 2023-05-25, Tillgänglig: https://www.spelbutiken.se/produkt/arrcade-1-up-legacy-pacmania-bandai-namco/23F9NQ/?gclid=Cj0KCQjwjrjBhD0ARIAsAMLvnF-cc-eKkfAzlfJFheADgs9oLy0mzQHTKcp2vSc4u2AcquhhY7dL7-waAsHpEALw_wcB&gclsrc=aw.ds
- [13] Figma, *Figma: the collaborative interface design tool*, u.å, hämtad: 2023-05-25, Tillgänglig: <https://www.figma.com/>

- [14] InkScape, *InkScape*, hämtad: 2023-05-09,
Tillgänglig: <https://inkscape.org/>
- [15] ImageLine, *FL-Studio*, hämtad: 2023-05-25,
Tillgänglig: <https://www.image-line.com/fl-studio/>
- [16] Freesound, *Freesound*, hämtad: 2023-05-25,
Tillgänglig: <https://freesound.org/>
- [17] Simon Brown, *The C4 model for visualising software architecture*, Creative Commons, hämtad: 2023-05-05,
Tillgänglig: <https://c4model.com/#ContainerDiagram>
- [18] Ultimarc, *I-PAC Ultimate I/O*, hämtad: 2023-05-10,
Tillgänglig: <https://www.ultimarc.com/control-interfaces/i-pacs/i-pac-ultimate-i-o/>
- [19] Arduino, *Arduino Hardware*, 2022-04-11, hämtad: 2023-05-25
Tillgänglig: <https://www.arduino.cc/en/hardware#boards-1>
- [20] Rädda Barnen, *Barn och Självkänsla - Minska Sociala Mediers PåverkanBARN*, Rädda Barnen.
u.å, hämtad: 2023-05-10,
Tillgänglig: <https://www.raddabarnen.se/rad-och-kunskap/foralder/barns-sjalvkansla-minska-sociala-mediers-paverkan/>
- [21] 1177, *Samhällsstöd till Barn med Funktionsnedsättning*, 1177.se, 2020. Hämtad: 2023-05-08
Tillgänglig:
<https://www.1177.se/Stockholm/barn–gravid/vard-och-stod-for-barn/funktionsnedsattning-hos-barn/samhallsstod-till-barn-med-funktionsnedsattning/>
- [22] Unity, *Unity - Manual*, hämtad: 2023-05-10,
Tillgänglig: <https://docs.unity3d.com/Manual/index.html>
- [23] Ken Schwaber and Jeff Sutherland, *Scrumguiden*, 2020-11, hämtad: 2023-05-09
Tillgänglig: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Swedish.pdf>

Bilaga A

Utvecklingmetodik

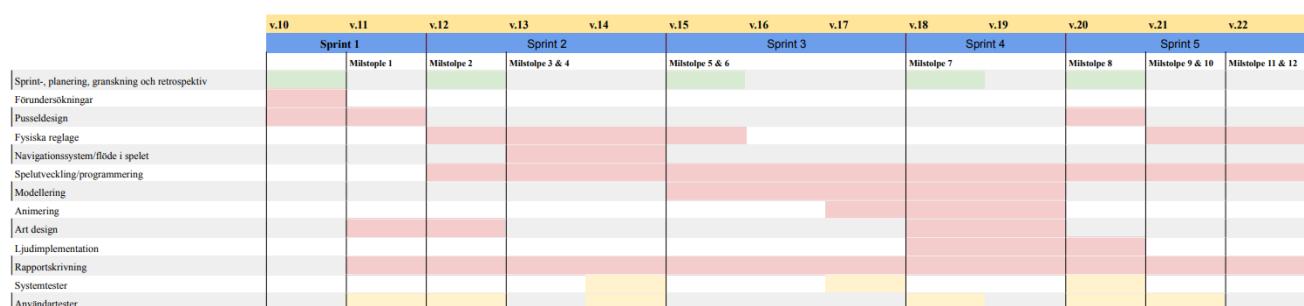
Scrum är en agil utvecklingsmetod som används för att hantera och leverera produkter på ett effektivt sätt. Genom att följa Scrum-metodens riktlinjer och principer kan utvecklingslaget skapa en slutprodukt som uppfyller kundens önskemål på ett effektivt och kvalitativt sätt.

A.1 Utvecklingsmetodik

Scrum är en agil utvecklingsmetodik som har använts i detta projekt. Att arbeta agilt innebär att medlemmarna ska ha regelbunden och transparent kontakt med varandra. Det innebär också att arbeta flexibelt och ta beslut under projektets gång. En agil utvecklingsmetodik passade projektet bra eftersom medlemmarna är studenter med flexibla scheman. Det finns också få ursprungskrav från kunden vilket gjorde arbetslaget möjligt att anpassa till slutprodukten. Scrum kombinerar fyra formella tillfällen för granskning och anpassning, vilket tillsammans utgör sprintaktiviteten. Dessa aktiviteter är sprintplanering, dagligt scrummöte, sprintgranskning och sprintretrospektiv. Sprintar är hjärtat i Scrum, där idéer omsätts till värde. Ett scrumteam är typiskt 10 personer eller färre och definierar endast tre specifika ansvar. Produktägare, Scrum master och utvecklare. [23] Mängden medlemmar och olika ansvar i Scrum passar storleken på projektet.

A.2 Tidsplan

Projektet pågick under 14 veckor och 6 sprintar hann fullföljas. Tidsplanen representeras i ett Gantt-schema, se Figur A.1, som visar hur sprintarna fördelar över tid samt vad projektteamet fokuserade på. En del av momenten arbetas parallellt med varandra då de fördelar mellan utvecklare. Schemat ändrades utefter projektets utveckling.



Figur A.1: Gantt-schema för projektets utvecklingsprocess.

Att planera upp varje sprint kräver olika milstolpar som vägleder och säkerställer att inget förbises. Den innehåller viktiga funktioner och är rangordnade enligt följande:

Milstolpe 1: Fastställa projektets designprinciper.

Designprinciperna innehåller regler och beskrivningar som ska följas.

Milstolpe 2: Första pappersprototyp på två pussel.

Grundidén för ett pussel måste behandla mekanik, regler, restriktioner, samtliga steg av pusslet samt målet. Här ska även användandet av reglage beskrivas, det vill säga hur användare tillåts interagera med olika delar av pusslet och omgivningen. Är dessa krav uppfyllda så ska en simpel prototyp skapas med papper, kartong, eller digitala verktyg som exempelvis *Microsoft Paint*. Denna prototyp ska ge en grov överblick av pusslets uppbyggnad och ge möjlighet att testa pusslets grundläggande mekanik och flöde.

Milstolpe 3: Slutföra pusseldesignen baserat på användartester av pappersprototyper.

Pusslet redo för att börja implementeras i *Unity*.

Milstolpe 4: Moodboard av spelets tema.**Milstolpe 5: Reglage kopplas och testas så att de fungerar på dator.****Milstolpe 6: En skiss på layout av reglage är skapad och godkänd.****Milstolpe 7: Minimum Viable Product, MVP**

MVP, ett spel som ska spelas av två personer på varsin skärm, man ska kunna styra spelet med reglage, det har en start till pussel till slut och tillbaka till start. Två pussel. Ha en spellängd på 10 minuter.

Milstolpe 8: Texturer och modeller är skapade och redo att importeras till spelet.**Milstolpe 9: Spelet har en väldefinierad layout för reglage som styr spelet.**

En mer välplanerad och snyggare layout för reglagen och högtalarna.

Milstolpe 10: Ljudeffekter skapas och implementeras i spelet och de kan spelas ut på specifikt placerade högtalare.**Milstolpe 11: Slutprodukt, samtliga krav uppfyllda, tester genomförda och avklarade.****Milstolpe 12: Dokumentation om montering och underhållning av systemet.**

A.3 Organisation

Genom att ha haft tydliga ansvarsområden och roller i projektet kunde alla grupper arbeta effektivt och samarbeta för att uppnå projektets mål. Detta ledde till en smidigare och mer effektiv utvecklingsprocess och slutresultatet blev en produkt som var väl genomtänkt och uppfyllde kundens behov och förväntningar.

Några organisatoriska roller som följer av en agil process är:

- **Produktägare**

Produktägaren ansvarar över sprintbackloggen där hen ser till att allt framträder tydligt och genombränt. Endast produktägaren får godkänna ändringar och beslut, dessutom har hen en transparent kontakt med kunden.

- **Scrum-master**

Scrum-master ser till att projektet drivs framåt, har en tydlig bild av delmålet och hjälper till medutvecklingen. Scrum-master ser dessutom till att alla utvecklare håller sig till utvecklingsmetodiken och Scrums principer. Scrum-mastern kan eventuellt bytas ut mot någon i projektteamet som är mer erfaren inom en sprint. Den individ med mest erfarenhet bör vara Scrum-master och vid behov hjälpa till med uppgifterna.

- **Dokumentansvarig**

Alla utvecklare kommer att skriva på den slutgiltiga rapporten men det är en person som har ansvar över att den är skriven bra och i tid.

Utöver dessa roller behövdes ytterligare arbetsområden som gällde för detta projekt

- **Spelutvecklare**

Ansvaret är att implementera en miljö i spelmotorn. Alla spelutvecklare är engagerade i implementeringen av spelmotorn då det kommer vara många olika delar som ska utföras korrekt.

- **Grafikdesigner**

Ansvaret är att alla objekt och bakgrunder får en design baserat på temat och målgruppen. Allt ska vara sammanhängande och färgerna ska vara färgblindsgodkänt.

- **Leveledesigner**

Ansvaret är att de pussel som spelet innehåller utvecklas. De är ansvariga för att skapa ett användargränssnitt och för att planera hur pusslen ska lösas. Denna grupp består av två mindre grupper bestående av två personer. Vardera grupp har ansvar över ett pussel var.

- **Ljuddesigner**

Huvudansvaret för att skapa passande ljud och grafik för spelets alla nivåer tillhör tre personer. Alla i gruppen kommer dock medverka och bidra till den slutgiltiga produkten inom detta område.

- **Reglageansvarig**

Ansvaret är att få hårdvaran I-PAC att fungera med spelmotorn.

A.4 Mötesprinciper och rutiner

Alla möten hade en bestämd start- och sluttid. Mötens hölls med konsistens och jämna mellanrum för att ge en bra struktur. Alla medlemmar i arbetslaget deltog på mötena. Varje möte hade ett specifikt syfte och agenda som alla medlemmar arbetade tillsammans med att försöka uppnå. Mötens var transparenta, vilket innebar att alla uppmuntrades till att dela sina tankar, idéer och bekymmer.

- Sprintplaneringsmöte:

Detta möte påbörjade varje ny sprint genom att fastställa arbetet som skulle utföras i sprinten. Den resulterade planen skapades gemensamt av hela arbetslaget. Mötet varade ungefär två timmar.

- Dagligt Scrummöte:

Arbetslaget skulle granska framstegen mot sprintmålet och kunde anpassa sprintbackloggen samt arbetsplaneringen vid behov. Mötet skulle hållas på kommunikationsverktyget *Discord* klockan 10:15 varje vardag och varade i ungefär 15 minuter.

- Sprintgranskning:

Ett möte hölls efter varje sprint där arbetslaget granskade utfallet och tog beslut om kommande anpassningar. Arbetslaget presenterade också arbetsresultatet för kunden och diskuterade framstegen mot produktmålet. Mötet varade max 2 timmar långt.

- Sprintretrospektiv:

Under dessa möten granskade arbetslaget senaste sprinten med avseende på medarbetare, samverkan, processer, verktyg och definitionen av klar. Mötet varade i max 2 timmar.

A.5 Versionshantering

Git användes som versionshanteringsststem, med *GitHub* som verktyg. GitHub är en plattform för versionshantering som fungerar väl för samarbete mellan flera utvecklare i ett mjukvaruprojekt.

I GitHub arbetade projektgruppen med mindre grenar som var baserade på projektets huvudgren. Huvudgrenen innehöll all färdig kod som var redo att publiceras. I de mindre grenarna arbetade projektets utvecklare med spelets funktionaliteter. När en utvecklare påbörjade en uppgift skapades en ny gren för specifikt den funktionaliteten och döptes efter utvecklarens förnamn och vilken funktionalitet som skulle utvecklas, till exempel “anders-raketstyrning”.

När en utvecklare ansåg sin funktionalitet vara färdigutvecklad var tanken att denne skulle göra en *pull request* till huvudgrenen så att en av gruppens övriga utvecklare testade och utvärderade koden så att den kunde sammanfogas med huvudgrenen, men i praktiken blev det inte så. Eftersom rutiner, för när en utvärdering skulle utföras, saknades blev funktionerna inte testade kontinuerligt och alla funktioner sammanfogades till huvudgrenen i projektets slutskede.

A.6 Dokumentationsprinciper

Under sprintplaneringens möten fördes ett mötesprotokoll upprättas som sammanfattade de olika synpunkterna som presenterades under planeringen. Vissa synpunkter som förekom var vilka user-stories som skulle implementeras, vilka prioriteringar dessa hade, målet med sprinten, skifte av prioriteringar utifall det hade blivit någon spillover från föregående sprint samt tid och utvecklar allokering under sprinten.

I slutet av varje sprint hölls ett granskningsmöte där bland annat en demo på de funktioner som implementerats i föregående sprint visades upp och återkopplingen från de inbjudna intressenterna dokumenterades.

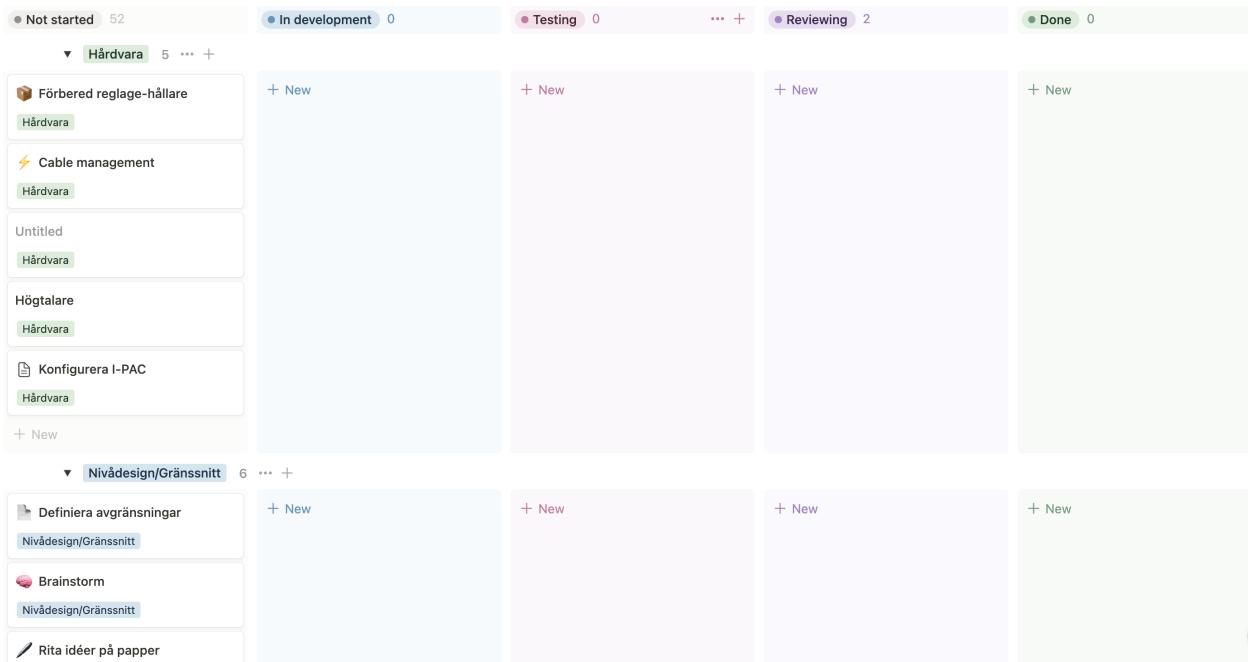
A.7 Kravhantering och spårning

Alla kravspecifikationer kom från antingen kundmöten eller generella bestämmelser inom gruppen med anknytning till projektets målbild. Dessa krav dokumenterades till en början i mötesdokument. Under projektets gång arbetade gruppen med två sammankopplade produktbackloggar, en allomfattande med samtliga arbetsuppgifter som existerade inom projektet, samt en detaljerad med pågående

uppgifter i aktuell sprint.

För att fastställa en majoritet av de moment som skulle komma att existera under utvecklingen började gruppen med en *brainstorming* session, där resultatet blev ett antal *epics*. En epic bestod av ett flertal relaterade arbetsuppgifter som var för stora för att inkluderas i en sprint, och dessa behövde därför brytas ner i mindre hanterbara *tasks*, innan de kunde ingå i en produkt backlog för att sedan implementeras. Det var alla dessa tasks som utgjorde den första *backloggen*. Dessa innehöll en utförlig specifikation av tillhörande krav och test som krävdes för att en task skulle anses vara färdig. Denna backlogg var alltså en behållare för samtliga tasks som skulle utföras under projektets gång, och de kunde placeras i ordning under ett av tre olika kategorier.

Vid sprintplaneringen bestämde gruppen ett antal tasks som skulle genomföras under kommande sprint, samtliga av dessa hamnade i den andra backloggen. Denna backlog hade i syfte att underlätta utvecklingsprocessen, då samtliga aktiva arbetsuppgifter under en sprint synliggjordes. Likt den första backloggen på placerades även här tasks i en av flera kategorier som beskrev status på denna, se Figur A.2. Då alla krav initialt ingår i epics, och vartefter kommer att brytas ner till taskt, då är dessa produkt backloggar det främsta verktyget för kravhantering och spårning.



Figur A.2: Produkt backlogg som användes under projektets gång.

A.8 Story/task-hantering

Som nämnts ovan har projektet initialt delats upp i ett antal epics som gav en överblick på högre nivå av samtliga funktioner och mål inom ett sammanhängande område. Varje epic tilldelades till minst en ansvarig person, som hade i uppgift att bryta ner den i mindre beståndsdelar *tasks*, som vid senare tillfälle skulle ingå i en sprint. Ansvarig hade även i uppdrag att alla tasks får en utförlig beskrivning, komplett med krav och tester. Varje task skattades utifrån två aspekter, vad den tillförde till projektet i form av värde, samt hur mycket ansträngning som krävdes för att utföra den. Med denna skattning kunde samtliga uppgifter rangordnas och få en prioritet inom utvecklingsprocessen, samt utifrån detta placeras i en produkt backlog.

A.9 Användartester

Användartesterna skapades under projektets gång för att utvärdera produkten. Ett problem med dessa användartester var att det med stor sannolikhet skulle bli problematiskt att få tillgång till personer inom målgruppen för att utföra testerna. För att undvika att förlora värdet i användartesterna inriktade sig gruppen på personer som hade nära koppling till målgruppen, och därav en djupare förståelse av deras vanor och beteenden. En undersökning skulle utföras i slutet av varje test där deltagarna fick svara på frågor i form av ett formulär. Formuläret syftar till att besvara specifika frågor gällande användarupplevelse och dylikt. Med data från användartester kunde systemet testas igen för att mäta lyckandet av den nya iterationen. Dessa mätningar gav oss resultatet som ämnar att besvara frågeställningarna i avsnitt 1.3.

A.10 Testningsprinciper och rutiner

När en ny task blivit implementerad skulle en kravlista skrivas av utvecklarna som förklarade vad som ska testas för att funktionen skulle bli godkänd, denna lista testas av övriga. Det skedde också en granskning av minst två andra utvecklare som säkerstälde att koden följer den angivna kodstandarden. Användartesterna utfördes av en ansvarig och bestod av två delar. Första delen var ett internt användartest där alla som inte hade arbetat med funktionen utförde ett test för att åtgärda problem som missats under utvecklingen. Andra delen var ett användartest med utomstående parter där de fick utföra givna instruktioner som antecknas för att sedan ge förbättringsområden och åsikter.

A.11 Modelleringsrutiner

Eftersom gruppen valde att använda en agil utvecklingsmetodik som diskuterades i avsnitt A.1 sätts vissa riktlinjer på hur systemet skulle modelleras. En av anledningarna med att använda en agil projektledningsmetod var att utveckla produkter på ett flexibelt sätt. Eftersom projektet ständigt genomgick förändringar börjar man med att skapa en så kallad rough design up front, med detta menar man att från början skapades en design av systemet på en abstrakt nivå. Avsikten med detta var att skapa en visuell representation av systemarkitekturen som beskrev slutprodukten utan att gå in för mycket på detaljer. Unified Modeling Language användes för att skapa ett diagram som representerade produkten, vilket diskuteras i avsnitt ???. Denna modelleringsmetod lämpade sig bäst för detta projekt då den används främst för programvarukonstruktion vilket var målet med detta projekt, till skillnad från andra metoder såsom Entity Relation Diagram som används för databasdesign som detta projekt saknar. Detta tillvägagångssätt tillät gruppen att påbörja systemutvecklingen tidigt medan modelleringsaspekten fördjupas i samband med projektets gång. Vilket garanterade möjligheten till anpassning av produkten då den inledande designen inte var satt i sten och hjälper även alla intressenter att fastställa de tekniska svårigheter som kunde uppkomma under projektets gång.

Bilaga B

Reflektion över systemutvecklingsprocessen

Under veckan hade vi två möten för att organisera vårt arbete. Det första mötet användes för att tilldela individuella uppgifter till varje medlem i teamet. Dessa uppgifter diskuterades sedan närmare under det andra mötet, där vårt mål var att nå gemensamma beslut. Ett exempel på en sådan uppgift var att utveckla idéer för pussel i vårt spelprojekt. För att underlätta processen skapade vi en bank med pusselidéer, vilket tog mycket tid. Syftet var att ha en resurs för brainstorming och idéutbyte. Men i efterhand anser vi att det kan ha inneburit onödig dubbeltarbete. Det verkar som om vi kunde ha kombinerat dessa två aktiviteter och genomfört dem samtidigt för att spara tid och undvika upprepning.

Trots detta hade vår metod ändå fördelar. Genom att ha separata möten för att tilldela uppgifter och sedan diskutera dem, kunde vi hålla hela teamet på samma sida när det gällde vilken typ av spel vi skulle utveckla. Det bidrog till att skapa en gemensam förståelse och riktning för projektet.

En annan fördel med denna mer omfattande planeringsprocess var att vi fick möjlighet att noggrant överväga och diskutera våra idéer innan vi implementerade dem. Genom att ha en tydlig och väl genortnäkt plan kunde vi potentiellt undvika hinder eller problem längre fram i spelets utveckling.

Gruppen planerade att produkten skulle levereras tre veckor försent vilket resulterade i att gruppen fick mindre tid att utveckla spelet. Detta ledde till att bakgrundsljud och ljudeffekter inte blev klara. Dessutom kunde inte problemen från användartesterna åtgärdas i tid. Projektet följde och var i fas med tidsplanen, men tiden fanns inte för att bli helt *klar*. Några milstolpar, exempelvis milstolpe 8, skulle vara klar nästa vecka enligt tidsplanen, se avsnitt A.2. De slutgiltiga objekten och texturerna blev klar betydligt längre bak i tiden då gruppen ansåg att spelet hade varit dött med bara *placeholders*. När vi kollar tillbaka på planeringen ser vi därmed att gruppen har planerat milstolpar på fel sätt. Det vi inte tänkte på då var att flera punkter kan arbetas på samtidigt effektivt. Därför blev milstolpar klar snabbare än vad tidsplanen sa. Dock innehåller tidsplanen parallellt arbete men förmodligen inte lika mycket som var möjligt.

Tidsplanen var för övrigt bra att ha under projektet för att hålla utvecklingen och planeringen på rätt spår. Desto längre projektet gick, desto mindre aktuell blev tidsplanen.

När det gäller programmet Notion uppskattade gruppen det. Denna programvara har varit till stor hjälp för att organisera uppgifter, dokumentera och planera sprints. Problemet med Notion var dock att gratisplanen inte hade tillräckligt med utrymme, så innehållet minimerades och att en ny Notion-tavla skapades. En tavla som användes flitigt var den som innehöll dokumentation och information om projektet.

Till en början planerade vi att använda ett system som höll reda på när tasks startade och när de blev klara, men vi insåg snabbt att detta inte bidrog till mycket och slutade snabbt att göra detta. Under planeringsmöten skapades många tasks som under sprinten kändes onödiga då de var klara väldigt

fort eller delvis ingick i en annan task. Vi märkte också att rutinerna kring tasks fungerade bättre i början då varje task var mindre och hade en tydlig definition av klar. Efter några sprints skapade vi fler tasks än vad vi trodde vi skulle hinna för att inte få slut på tasks under sprinten men konsekvensen av detta var att produktbackloggen blev överfylld med tasks som inte hade en definition av klar och därmed förlorade sin vikt. En bättre kommunikation under dagliga möten hade kunnat hjälpt till att tydligt förstå vilka tasks som inte skulle bli klara eller var onödiga.

Inför projektet planerades det för fyra typer av möten, varav samtliga av dessa kom till användning, och i varierande grad lyckades uppfylla sina syften. Under projektet så tillfördes två mötesansvariga med uppgift att strukturera möten och se till att relevant information togs upp på respektive möte. Tanken med detta var att till viss del ersätta den organisoriska uppgiften av en Scrum master, för att undvika att en person skulle tvingas till att exklusivt arbeta med den uppgiften. Ett mindre problem under projektet var allmänt slarv med närvoro på möten, framförallt i början och i slutet av projektet. Detta kunde i flera fall bero på brist i kommunikation, och blev aldrig problematiskt då rimliga anledningar till frånvaro nästan alltid kunnat ges. Väl under möten så fungerade dessa som tänkt. Dagligt Scrum var en typ av möte som stundvis kunde känna överflödigt, men i det stora hela var ett bra sätt för gruppen att kunna dela med sig, och dagligt möte satte även press på medlemmar att prestera. Granskning och retrospektiv hölls i direkt anslutning. Här tog alla medlemmar upp tankar om våra existerande processer och testade att ändra på flera rutiner som uppfattades som problematiska. Dessa möten hjälpte också att förbereda oss inför nästa sprint, genom att ge en god överblick av projektets status. Ett moment som uteblev under flertalet sprintgranskningar var presentation av resultat för kunden. Anledningen till att kundmöten ofta övergavs var många gånger en brist på betydande resultat som var värdiga att dela med sig av. Detta kombinerades med att gruppen kände att dessa möten inte gav tillräckligt mycket för att föra projektet i rätt riktning, eftersom kundens krav var få. Kunden sa även att han agerade mer som en handledare, vilket innebar att han var tillgänglig för frågor, men hade en begränsad roll när det kom till att ge återkoppling. Därför hölls få kundmöten och efter ett visst skede i projektet så slutades dessa möten bokas helt.

Vi sammanförde viktiga versioner och använde generellt sett grenar på rätt sätt. Även om vi inte alltid hade en konsekvent namnstandard för grenarna, så pushades inga ändringar till huvudgrenen (master) om de inte hade genomgått noggrann testning av den utvecklare som arbetade i grenen. I projektplanen hade vi planerat att utveckla varje funktion separat och sedan pusha dem till huvudgrenen. Tyvärr fungerade inte detta fullständigt eftersom det var utmanande att utveckla funktioner samtidigt som vi garanterade att de var kompatibla med den specifika grenen de tillhörde. Slutligen hade vi en gren för varje pussel i spelet. Funktionerna utvecklades delvis i separata grenar och delvis i pusselgrenarna.

Gruppen hade avsikt att genomföra kvalitetssäkring under utvecklingsprocessen. Enligt planen skulle två andra utvecklare testa varje funktion när en utvecklare ansåg att den var klar. Tyvärr märkte vi tidigt att denna process inte följdes, vilket ledde till att vi beslutade att ta upp problemet vid det kommande retrospektivmötet för sprinten. Det var oklart när kvalitetssäkringen skulle utföras, och vi kom överens om att det var utvecklarens ansvar att planera dessa tester. Trots detta beslut så genomfördes ingen kvalitetssäkring, vilket vi tror berodde på tidsbrist och en bristande uppfattning om dess betydelse. Tack vare GitHub's enkla versionshanteringssystem har det varit enkelt för oss att återgå till tidigare versioner om det har behövts. Det har också känts mer praktiskt att först sammanföra grenar och sedan leta efter buggar.

B.1 Lucas Hägg

Jag har varit utvecklare och mötesansvarig tillsammans med Andreas. Som utvecklare har jag programmerat spellogik i Unity och skapat ljudprototyper i FL Studio. Som mötesansvarig har jag och Andreas hjälpts åt att hålla i de olika momenten som ingår i Scrum. Jag har också bidragit med idéer

och planering under hela projektets gång.

I praktiken så ska det finnas en Scrum master och en produktägare i Scrum [23]. I stället bestämde sig gruppen för att ha två mötesansvariga. Alltså saknade gruppen en del ansvar när det kommer till utövandet av Scrum. Det fanns ingen ansvarig för produktbackloggen, och ingen som såg till att Scrum processen följdes och att alla förstod den under projektets gång. Som utvecklare ska jag och de andra utvecklarna tillsammans slutföra tasks i produktbackloggen till sprintens slut. Jag ska också vara delaktig i de andra Scrum processerna som dagliga möten, sprintplaneringar, granskningar och retrospektiv. Det mesta har fungerat bra med några små svårigheter när det kommer till ansvarsområden och utövandet av Scrum. Som nämnt i projektplanen så har varje task/post i produktbackloggen en ansvarig utvecklare. Varje ansvarig har som mål att se till att sina tasks blir slutförda till sprintgranskningen men i verkligheten kändes det som att många blev kvarglömda och förflyttade till nästa sprint. Det kan bero på flera olika saker. Jag tror att de största anledningarna var slarv eller oklar task. I utövandet av Scrum ska varje task i produktbackloggen ha en tydlig definition av klar, vilket visat sig vara svårt att definiera till ett projekt som har få krav som vårt. Det är också möjligt att dessa tasks har varit för breda i sin definition, vilket gjort det svårt att veta för de ansvariga när de är klara. I andra fall har det varit slarv från gruppmedlemmar som inte gjort klart sina uppgifter. Vi nämnde också i projektplanen att tasks skulle få prioritet. De fick dock inte prioritet utifrån hur svår tasken var att utföra, utan snarare hur snabbt den behövde bli klar. Tasks med låg prioritet var ofta de som släpades efter till nästa sprint, så det är möjligt att detta system kom med en nackdel. Vi har haft dagliga möten som är till för att dela med sig hur det går och vad alla ska arbeta på den dagen. Alla har inte varit delaktiga på dessa möten och många har struntat i att säga något under mötena. Hade gruppen haft en Scrum master och produktägare är det möjligt att resultatet av vårt utövande varit annorlunda. Till exempel hade kanske vår produktbacklogg haft färre eftersläpande tasks, då någon hade kunnat ansvara över att alla tasks var kompletta. Med en Scrum master hade det varit lättare att vägleda alla till rätt spår under våra dagliga möten.

B.2 Gustav Johansson

Min roll i detta projekt var utvecklare. Jag har även haft mer ansvar i designen av spelet där jag har producerat objekt som visas i spelet. Jag hade ansvar att se till att objekten skapades till ett av pusslena. Jag har även deltagit i alla användartester som har utförts. Där hade jag ansvar att samla ihop personer som ville göra våra användartester och att skapa ett schema med tider då personerna skulle komma.

Då projektets planering har utgått från utvecklingsmetoden Scrum så delade vi upp gruppen som utvecklare, Scrum master och produktägare. Gruppen beslöt sig för att ha kvar rollen utvecklare men istället för Scrum master och produktägare gav vi två utvecklare lite mer ansvar exempelvis att skapa dagsordningen för stand-up möten och sprintplanering etcetera. Dock, alla deltagare i projektet har haft ansvarsområden. Dessa ansvarsområden bestäms genom våra sprintplaneringar då alla deltagare deltar i en *backlog grooming session* vilket innebär att gruppen diskuterar de inlägg som finns i backloggen och sedan placeras prioritet och vem som har ansvar. De utvecklare som i slutändan får ansvar över ett inlägg är inte nödvändigtvis den personen som kommer implementera denna task, utan ansvarshavaren ska istället se till att den blir gjord. Detta resulterade i att gruppens ansvarsfördelning blev effektiv och att tvärgrupper inte blev nödvändiga. Då utvecklare fick nya ansvarsområden varannan vecka har utvecklarna nästan alltid något att göra. Om en utvecklare inte fick något att göra så kunde personen nämna detta i ett av alla dagliga möten. Detta fungerade delvis. När personen, som behövde hjälp, bad om stöd var det enkelt att hjälpa personen med tasken eller att förklara för personen vad den kan göra. [23]

Vissa tasks blev kvar från en sprint till en annan. Detta beror på dålig kommunikation med gruppen och dålig beskrivning av tasken. Vissa tasks täckte stora områden, vilket egentligen borde ha delats upp i mindre tasks. Detta gjorde det svårt för ansvarig att bestämma vad den och andra ska göra för att klara tasken. Detta hade gått att lösa med bra kommunikation genom att deltagarna hade kunnat ändrat tasken eller hjälpt till.

Tidsplanen skapades under planeringsfasen. Denna tidsplan var bra att ha då projektgruppen kunde kolla på denna tidsplan inför eller under varje sprintplanering (se avsnitt A.2). Dock var inte tidsplanen och milstolparna rimliga under alla steg av projektet. Effekten av detta var att implementeringen av spelet startades försent. Det resulterade i att utvecklarna fick skapa hela spelet under bara några veckor medan planeringen av projektet tog mer än dubbelt denna tid. Detta gjorde det stressigt i sista delen av projektet och inte mycket tid att åtgärda uppstående fel.

Efter sprintgranskning skulle vi pratat med produktägare men då vi knappt hade en produktägare så fick vi ingen direkt feedback från ägaren. Detta var ett problem då vi ville använda oss av utvecklingsmetodiken Scrum för att efterlikna hur ett projekt skulle kunna se ut i arbetslivet. En produktägare är en stor del av Scrum metoden då personen ska komma med återkommande feedback och nya tasks vilket inte blev fallet. [23]

I helhet är jag nöjd med projektets resultat. Även fast det ibland gick sakta framåt hjälpte gruppen varandra så gott vi kunde och skapade ett bra spel tillsammans.

B.3 Andreas Ekberg

Rollen jag hade var främst utvecklandet av spelet. Jag hade också en del i framtagning av idéer och göra de idéer vi tog fram till en prototyp i Figma. Jag utvecklade i stor del kranpusslet och de olika funktionerna som är kopplade till den delen av spelet. Under användartesterna fanns jag tillgänglig för att kunna svara på frågor om funktionaliteten samt hoppa in vid eventuella buggar då jag hade en bra övergripande förståelse för strukturen av hela spelet.

Projektet i teorin var uppdelat efter Scrum där alla hade ansvarområden. Dessa ansvarområden delades ut på planeringsmöten och gav alla chansen att arbeta på olika saker under olika sprints. I praktiken blev det mer att samma personer sitter med det som utdelades under sprints i början för de var insatta i ämnet och gjorde det svårt för andra att sätta sig in i det. Det ledde i sin del till att uppdelningen av arbete inte blev jämt utdelat. Inte nödvändigtvis för att vi inte försökte utan att mer att tiden det skulle ta att lära upp någon som inte suttit i Unity kunde läggas på att implementera i Unity. Många av rutinerna vi skapat var också väldigt optimistiska som sedan inte kunde efterföljas. I projektets helhet har det gått bra och vi har en slutprodukt värd att visa men det finns många moment som kunde förbättras eller förändras då många av problemen har sin grund i att vi inte hade erfarenheter inom ämnet. Hade vi gjort om projektet med de kunskaper vi har nu hade vi kunnat omstrukturera en del för att ha mer tid i slutet av projektet.

B.4 Kenton Larsson

Under projektet har min roll varierat något. Till en början ägnade jag tid till att göra förstudier rörande projektets målgrupp. Då projektet lämnade planeringsfasen arbetade jag i stor utsträckning med att få fram idéer till pussel, för att sedan bearbeta dessa och skapa prototyper i Figma. Den specifika prototypen jag varit med och skapat är den till spelets Crane Puzzle. Jag var även med och konceptualisade Orbit Puzzle, samt var en av utvecklarna som implementerade detta i Unity. Under projektets användartester var jag delaktig i att dels dokumentera hur testare interagerade med spelet

och varandra, men även i att uppdatera Figma-prototypen i realtid.

I sin helhet känner jag att projektet har fungerat väl och planeringen har definitivt varit en bidragande faktor till detta, trots att den inte följs in i minsta detalj. Att planeringen inte kunnat följas helt var dock väntat, och är till viss del en produkt av den agila utvecklingsmetoden som används. Ett tillvägagångssätt som såg viss förändring under projektet var utvecklingsprocessen för pussel. Det första pusslet som skapades var kran-pusslet, vilket hade en tydlig och förbestämd process från konceptualisering, till prototyp i Figma, och slutligen till implementation i Unity. Orbit Puzzle följde inte samma process utan gick direkt från koncept till Unity. Anledningar till detta var flera; brist på tid, önskan att göra användartest i Unity, samt en ökad säkerhet i utvecklarnas förmåga att implementera funktioner i Unity.

B.5 Ramez Rizek

Min huvudsakliga roll i detta projekt var reglageansvarig. Min roll var att testa och färdigställa reglagen så att de kan användas i spelet. I denna roll fick jag blandannat skapa och dekorera behållare som skulle rymma alla reglage, programera belysningen på de belysta knapparna via den medföljande *Software development kit*, återanvända en internetkabel som en förlängningskabel då reglageuppsättningarna ska hållas minst två meter ifrån varandra och montera alla komponenterna och såg till att reglagen fungerade som de skulle. Utöver min huvudsakliga roll deltog jag i förarbete stadiet där vi bildade idéer på vilka pussel som skulle skapas och hjälpte med att skapa prototyper av spelet för användartester.

Eftersom vi valde Scrum som utvecklingsmetodik så borde mina uppgifter varit uppdelade i tasks som hela utvecklingslaget kunde arbeta på. Alla komponenter av reglagen behöver i sin tur också vara tillgängliga för utvecklingslaget.

Eftersom vi saknade en allmän plats att förvara alla komponenter så försvarades valet av att alla ska kunna arbeta på de. Det slutade med att jag förvarade allting eftersom jag huvudsakligen jobbade med reglagen och det var mitt ansvar att leverera ett färdigt system från start till slut utan att dela upp arbetet i tasks.

B.6 Rebecka Sahlin

Min roll har huvudsakligen varit utvecklare under projektet. I starten av projektet gjorde jag en del förundersökningar inom målgruppen, varefter vi även skapade ett antal färgpaletter och moodboards. En del tid ägnades även till att kommer på idéer för spelets pussel. Jag hade senare ansvar över att skapa objekt och bakgrund till spelet. Jag bidrog till användartesterna genom att skriva frågor till formulären och hjälpa till att göra anteckningar under det första användartestet.

Inför förundersökningen samlade gruppen frågor och tankar kring olika områden. En förstudie genomfördes sedan för att undersöka olika informationskällor och bedöma deras kvalitet och pålitlighet. Vidare sammanfattades nödvändig information för att svara på gruppens tankar. Eftersom gruppen inte hade erfarenheten inom alla områden, så var det viktigt att göra förundersökningar. Till en början var det svårt att avgöra vad som behövde utredas, men allt eftersom projektet fortskred dök det upp fler frågetecken om var vidare utredning behövde påbörjas. Genom att tidigt bestämma vad vi behövde tänka på kunde vi undvika att slösa tid på förstudier senare.

När jag skulle göra färgpaletter tog jag inspiration från pinterest och andra spel och filmer inom rätt tema. Jag använde färgblindverktyget för att få fram godkända färgblindhetsfärger. Det här verktyget gjorde mitt arbete mycket enklare och hjälpte mig att unvika fel färger.

Det tog mig längre tid än väntat att komma på pusselider. Det var svårt att fatta ett gemensamt beslut och ha alla nödvändiga element för att både göra spelet roligt, lärorikt och samarbetskravande. Det slutade med att vissa delar eller hela pusslet fick ändras efter behov. Användartesterna var till stor användning då det blev mer uppenbart vad som behövde ändras.

B.7 Lovisa Svensson

Jag har arbetat mycket med själva spelmotorn och sett till att övriga i gruppen har fått en stabil grund till att arbeta med den, eftersom jag hade en del förkunskaper innan projektet. Jag jobbade en del med studier kring hur vi använder versionshantering på bästa sätt vilket resulterade i att jag satte upp ett GitHub-repository med ett Unity-projekt i 2D.

Det har varit lite blandade uppgifter när det kommer till spelets funktionalitet. Jag har utvecklat en funktion som ställer om spelet om det gått en viss tid utan att någon interagerar med reglagen, vilket är bra för ett spel i en publik miljö där människor kanske lämnar området mitt i spelet. Jag har även utvecklat spelets startskärm och funderat kring vilka sätt som verkar enklast för att starta spelet, vilket är en funktion som kräver att två spelare håller in knappar samtidigt för att spelet ska starta. Utöver det har jag gjort ett par andra funktioner som mest är relaterade till styrning och för att visa objekt som en lampa lyser på.

Vi har en slutprodukt som jag anser möter förväntningarna men jag är lite bekymrad över min egen roll i projektet. Jag har gjort mitt bästa för att hänga med i gruppen men det har funnits många oklarheter som gjort det svårt att arbeta. Exempelvis när många utvecklare arbetar med många separata funktioner som i slutändan ska sättas ihop är det svårt att veta exakt hur man själv bör utforma sin funktion för att stämma överens med resten. Jag tror att alla tasks i Kabanantavlan och att vi aktivt arbetade med tavlan under sprintmötena ändå bidrog till en god helhet över planeringen men de små funktionerna som hör till spelet var svåra att planera inför och föll lite mellan stolarna. Det är också svårt att säga vad som skulle förbättras. Den här gruppen har inga egentliga erfarenheter inom varken spelutveckling eller Scrummetoden och därför har vi fått lära oss allteftersom vi arbetar med projektet. Hade vi gjort detta igen hade vi antagligen kommit fram till en plan som var mer heltäckande och skulle fungerat bättre i praktiken.