

Projektrapport Grupp D, TNM094

Kortspelsaktivierad strid i VR

Andreas Wahlgren
Anton Lindskog
Arvid Magnusson
Martin Hag
Oskar Tengvall
Viktor Sjögren
William Malteskog

Examinator: Daniel Jönsson

Sammanfattning

Denna rapport täcker ett utförande av ett Medietekniskt kandidatarbete inom skapande av ett *VR-spel*. Projektets syfte var att med *VR-teknik* utveckla ett kortspel som kan köras på ett lokalt nätverk, spelare mot spelare. Arbetet undergick med översyn av en kund som hade vissa krav om strukturen i spelet.

Projektarbetet har använt sig utav utvecklingsmetodiken Scrum, med viss modifiering för att anpassa det för gruppen. Varje medlem har haft ett ansvarsområde att ansvara för så att arbetet utförs, under tiden har de kunnat arbetat inom andra områden under projektets gång. Samarbete vid behov skedde via kommunikation på distans.

Under arbetets gång skedde utveckling inom många områden såsom modeller, texturer, kod och musik. Allt detta resulterade i ett fartfyllt spel som utmanar användarnas kognitiva förmåga och skicklighet att prestera under tidspress.

På grund av att *VR-utrustningen* flyttades till ett annat hus begränsades åtkomsten till utrustningen under ett antal veckor, vilket bidrog till förseningar i tidschemat. På grund av den pågående pandemin och karantänen begränsades sammankomster och möten, som i sin tur begränsade samarbetet, *VR-arbetet* och projektets utveckling. Den grundläggande speldelen behölls och anses tillräckligt fullbordat trots detta, även om betydligt mer funktionalitet från början var eftersträvat.

För att skapa en storhetskänsla utan att kompromissa med prestandan så jobbades det mycket med placering och distans till objekt i scenen. För att uppnå ett kognitivt belastande men samtidigt taktiskt spel så används ett turbaserat upplägg med kort tid för varje tur. För att skapa ett gränssnitt som inte förlitar sig på text så har grafiken designats med åtanke att den ska vara väldigt intuitiv för användaren.

Innehåll

Sammanfattning	i
Figurer	iv
Tabeller	v
1 Inledning	1
1.1 Bakgrund	1
1.2 Syfte	1
1.3 Frågeställning	1
1.4 Avgränsningar	2
2 Bakgrund	3
3 Relaterat arbete	5
3.1 Yu-Gi-Oh!	5
3.2 Pokémon	5
3.3 VR-utrustning	6
4 Redogörelse för arbetet	7
4.1 Implementation	7
4.1.1 Prototyp i GameMaker: Studio	7
4.1.2 2D-Grafik i Procreate	8
4.1.3 3D-Grafik i Maya och Blender	8
4.1.4 3D-Animation i Blender	9
4.1.5 Ljud och Musik	9
4.1.6 Programmering och komponenter i Unreal Engine	10
4.1.7 Systemarkitektur	10
4.1.8 Serverimplementation i Unreal Engine	11
4.2 Utvecklingsprocess	12
4.2.1 Versionshantering	12
4.2.2 Kvalitetssäkring	12

4.2.3	Dokumentation	12
4.2.4	Kravhantering	12
4.2.5	Organisation	13
4.2.6	Tidsplan	13
4.2.7	Kundmöten	14
5	Resultat	15
5.1	Spelet	15
5.2	Spelmekanik	16
5.3	Gränssnitt till användaren	16
5.4	Scenuppbryggnad	17
6	Analys och diskussion	18
6.1	Analys av metod	18
6.1.1	Organisation	18
6.1.2	Tekniska lösningar	19
6.1.3	Källkritik	20
6.2	Analys av resultat	20
6.3	Arbetet i ett vidare sammanhang	20
7	Slutsatser	21
7.1	Besvarande av frågeställningar	21
7.2	Vidareutveckling	22
Litteraturförteckning		24
A	Ansvarsområden	26
B	Användartester	27

Figurer

3.1	Elementschema med anvisning för svagheter och styrkor för elementen.	6
4.1	Prototypens spelmiljö i GameMaker: Studios exekvering	7
4.2	Golem modell med armatur i Blender	8
4.3	Exempel på animacionsmiljö i Blender	9
4.4	Exempel på blueprint-struktur	10
4.5	Grundläggande systemarkitektur	11
4.6	Gantt-schema	13
5.1	Spelet i action från spelarens vy.	15
5.2	Modellen som indikerar tid som gått under spelets rundtur.	16
5.3	Exempel på ett spelkort i spelet. Kortet ändrar vädereffekten till åska.	16
5.4	Exempel med highlight runt korten i spelet.	17
6.1	Ett exempel på hur Trello-tavlorna har sett ut under projektet.	19
B.1	Användartest prototyp del 1	27
B.2	Användartest prototyp del 2	28
B.3	Användartest prototyp del 3	29

Tabeller

Kapitel 1

Inledning

I denna rapport redogörs det arbete som gjorts inom ett projekt som pågått under en termin i kursen TNM094, Medietekniskt kandidatprojekt, som genomfördes av sju studenter som studerar till civilingenjör inom Medieteknik vid Linköpings Universitet (LiU) på Campus Norrköping. Projektet gick ut på att skapa ett kognitivt belastande spel för användaren där användaren utmanades med snabbhet och skicklighet vid hantering av kort i en immersiv miljö. Projektet har getts ut som uppdrag av en kund som är forskare på universitetet.

1.1 Bakgrund

Teknik fortsätter att utvecklas exponentiellt under det senaste decenniet. Ett stort framsteg är speciellt virtuell verklighet, kortfattat kallat *VR*. Denna teknik utmanar de redan befintliga sätten som media konsumeras, oftast stillasittande framför en skärm med någon typ av kontroll för att interagera. Det *VR*-tekniken istället tillåter är att användaren kan virtuellt befinna sig i en simulerad värld, där rörelser från användaren utifrån verkligheten projiceras in i den simulerade världen. På det sättet får användaren ett mer immersivt och naturligare sätt att interagera med teknik och information. Grundtanke med projektet var att utnyttja denna teknik för att sammanfoga en klassisk spelform som kortspel med en relativt ny teknikform.

1.2 Syfte

Projektet är tänkt att ge en slutprodukt som blandar en klassisk spelform i form av kort med *VR*. Spelet ska köras med två spelare som spelar mot varandra, där bågge spelarna bär ett *Head-mounted-display* (förkortat *HMD*) och styr sina rörelser med *VR-kontroller*, så kallade *Stavar*. Mekaniken i spelet ska leda till att användaren hela tiden har flertal val och på det sättet utmana spelarens kognitiva förmåga och göra varje gång som spelet spelas till en unik upplevelse.

1.3 Frågeställning

Frågeställningar som projektet bygger på:

- Hur kan en storhetskänsla skapas i spelet utan att kompromissa för prestanda med den hårdvara som finns tillgängligt?

- Hur kan vi uppnå ett spel som är både kognitivt belastande och underhållande men som samtidigt erhåller en struktur i turfördelningen och kräver att spelaren måste göra taktiska val?
- Hur kan ett gränssnitt skapas som inte förlitar sig på text utan representeras mest med form och färg som snabbt ger användaren en klar bild om vad objekten gör?

1.4 Avgränsningar

Spelet kan enbart testas med en viss typ av hårdvara och operativsystem på grund av den arbetsmiljö projektet arbetas under. Därav testas spelet enbart för dessa som finns tillgängligt och kan därför inte garanteras till att fungera för några andra system. Det som spelet testas för är *Windows 10* med *HTC Vives VR-redskap*.

Ingen publik server kommer att skapas för att köra spelet på. Spelet kommer istället köras på en *Listen-server* som skapas när två spelare startar spelet.

Spelet riktas mot personer i åldern 15-25 år, och förväntas förstå spelets regler och VR-utrustningen innan de börjar spela.

Kapitel 2

Bakgrund

Här presenteras kundens krav och mål kring projektet. Vilka önskemål som projektgruppen hade samt även den slutgiltiga idéen som valdes till projektmål.

Under första kundmötet fick projektgruppen reda på vilka visioner som kunden hade med projektet och vad han önskade att arbetets främsta fokus skulle ligga på. Projektgruppen fick stor frihet att tolka dessa krav själva och komma fram till en spelidé som innehöllade dem.

Ett av kundens mål med projektet var att det skulle vara immersivt. Vad som ansågs som immersivt var brett beskrivet, men en viktig del av det var att spelet skulle känna trovärdigt. Användaren ska känna att spelet beter sig på ett förväntat sätt utifrån hur verkliga världen fungerar. Objekten i spelvärlden ska vara logiskt gjorda och ha tydliga kopplingar mellan föremålen och deras funktionalitet.

Ett annat krav var att det skulle innehålla någon sorts kortliknande funktionalitet. Kunden berättade om sin vision att spelet skulle vara skicklighetsbaserat och att användaren skulle ha ett stort antal kort till sitt förfogande. Det stora antalet kort leder till att användaren behöver något sorts system att sortera dem. Det handlar om att ge spelaren en kognitiv överbelastning där hur sorteringen struktureras är avgörande om spelaren kommer hinna fatta rätt beslut och hitta mest lämpat kort under sin tur. Kunden underströk vikten att fokuset skall vara på spelets mekaniker och miljön i den immersiva världen skall inte ta upp för mycket av användarens fokus.

Förutom dessa två krav var det inte mycket mer kunden önskade kring projektet. Om spelet var multiplayer skulle det gå att spelas från enskilda datorer, grafiken skulle inte ha *low poly* stil utan vara lite mer naturtrogen samt målgruppen var för lite äldre män eftersom spelet är kognitivt krävande.

Utifrån dessa krav kom projektgruppen på massor av idéer till hur spelet skulle kunna se ut. Projektgruppen var enade att de ville försöka få till den immersiva känslan genom att använda sig utav VR med HMD. Fördelarna med HMD var att gruppen redan hade tillgång till HTC Vive utrustning som även är en lättillgänglig produkt. Eftersom HTC Vive kan använda sig utav det enkla systemet *SteamVR*, som har användarvänlig dokumentation för ihopkoppling av VR-utrustning, ansågs det vara ett bra beslut [17]. Till sist enades medlemmarna till ett koncept som låt genomförbart, roligt och överensstämde med kundens krav.

Spelet som beslutades att utveckla spelas av två spelare. Spelarna äger varsin golem där målet är att få ned motståndarens golems livmätare till noll. Detta görs med hjälp av kort som spelaren har på hand där korten har olika effekter i världen. Den första typen av kort som finns är kommandokort. Det finns två attackkort som skadar motståndarens golem samt två blockeringskort som reducerar skadan från inkommande attacker. Det finns även ett helarkort som ger sin golem mer liv. En annan korttyp som finns är elementkort. Dessa läggs på sin golem för att ge deras attack extra skada. Elementen som finns är vatten, natur, el och eld och dem har vardera ett element de är starkt emot och ett som de är svagt emot. Med båda golems element beräknas skadan på attacker ut beroende på vilka element som

ägs samt om motståndaren har det motsatta elementet. Sedan finns det väderkort som ändrar på vädret i spelet. Vädret gör så att elementet på båda golems som är svagt mot vädereffektens element tas bort. Sist finns det ett kort som ger spelaren extra kort på hand. Detta finns med eftersom det går att kombinera kort. Två svaga attackkort kombineras till ett starkt attackkort. Två svaga blockkort kombineras till ett start blockkort. Elementkorten kombineras och blir till vädereffekter. Kombinationen sker av spelaren på hand om spelaren önskar det.

Om en spelares golem får alla fyra element blir den *spectral*. Detta betyder att elementen försvinner i motbyte för att nästa kommandokort blir extra kraftfullt. Om man lägger ett attackkort slår den med mer skada, ett blockeringskort förvandlas till att hela sin golem istället.

Spelarna använder sig av HMD för att se spelet och interagerar i spelet med handkontroller. Miljön som spelet utspelar sig i är inspirerat från antika grekland där spelarna står på varsin piedestal i en arena med båda golems i mitten. Spelarna turas om att spela varsitt kort där de har en viss tid på sig att göra ett val.

Kapitel 3

Relaterat arbete

Kunden hade vissa krav och rekommendationer kring utvecklingen och spelidén för att uppnå en underhållande och bekväm upplevelse i en VR-miljö, vilket beaktades och implementerades i form av frågeställningar till detta projekt. En av sakerna som kunden tryckte på var att tänka på vilket avstånd som man placerar objekt i fokus på, då ett objekt som ligger för nära eller för långt bort kan bli suddiga och pixliga. Enligt dokumentation på Unreal Engine bör ett objekt, för att lättast kunna beaktas, vara placerad mellan 0.75-3.5 meter[1] samt att ögonens höjd bör vara runt 160 cm över marken. Man bör heller inte ändra för mycket på standardvärdena för *FOV* (field of view), *motion blur* eller *DOF* (depth of field) i ett VR-projekts inställningar i Unreal Engine . För VR-headsetet HTC Vive finns det även ett minimumkrav på prestanda som motsvarar en svarstid och bildfrekvensen som bör ligga på 17-33 millisekunder respektive 90 Hz /90 fps (frames per second) för en naturlig upplevelse [2]. En högre svarstid skulle leda till att det blir en otakt mellan rörelse från spelaren och feedback på displayen vilket både kan upplevas förvirrande och störande och en lägre bildfrekvens skulle kunna leda till symptom som åksjuka eller liknande.

Gruppen strävade efter att skapa ett relativt unikt VR-spel men då det finns en så stor mängd VR-spel och kortspel så gick det inte att undvika att ta en viss inspiration från andra spel och projekt.

3.1 Yu-Gi-Oh!

Då gruppen tilldelades uppgiften att skapa ett spel med kortspelsaktivverad strid var det första som dök upp, för majoriteten i gruppen, kortspelet *Yu-Gi-Oh!* [3]. I *Yu-Gi-Oh!* duellerar två spelare i form av kortspelsaktivverad strid med varsin kortlek som innehåller olika typer av monster, formler och fallor med varierade attribut. Dessa attribut kan vara t.ex. hur mycket skada de ger, hur mycket liv de har eller om de har någon speciell effekt som kan påverka spelet. Spelarna har en hand med kort att välja mellan och turas om att välja vilket monster de vill lägga ut i mål att besegra sin motståndare. Det som utifrån detta spel på sätt och vis applicerats i projektet är idén med korthanden, tur-baseringen samt duell mellan spelare i andrahand genom monster.

3.2 Pokémon

Ett annat klassiskt spel som tagits en del inspiration av är kortspelet *Pokémon* [4]. *Pokémon* är, precis som *Yu-Gi-Oh!* och projektets resultat, ett spel med kortspelsaktivverad strid mellan två spelare med varsitt monster, eller i detta fall *pokémons*, med olika attribut som slåss mot varandra. Det som är unikt med detta kortspelet är att varje kort har även ett element som eld, vatten, el, gift och många

fler, som har sårbarheter och styrkor mot andra element. Detta är något gruppen valde att ta med i spelet i en simplare form med endast fyra olika element som var och en har en styrka och en svaghet i en cirkulär ordningsföljd som kan ses på Figur 3.1.



Figur 3.1: Elementschema med anvisning för svagheter och styrkor för elementen.

3.3 VR-utrustning

Kunden gav gruppen till en början flera alternativ till vilken utrustning/medel som kunde användas för att utveckla spelet till. Ett exempel som var presenterat var ett *touchbord* som skulle kunnat agera som ett naturligt sätt att spela ett kortspel på. Detta av skälet att kortspel oftast spelas runt ett bord. Gruppen hade dock stark motivation till att utveckla spelet till just HMD, tillhörande handkontroller och den VR-utrustningen som fanns tillgänglig var HTC Vive. HTC Vive är inte den enda HMD som finns tillgänglig för VR men gruppen ansåg att HTC Vive uppnådde gruppens krav. Andra populära alternativ, såsom *Oculus Rift*, hade inte passat projektet bättre oavsett [16].

Kapitel 4

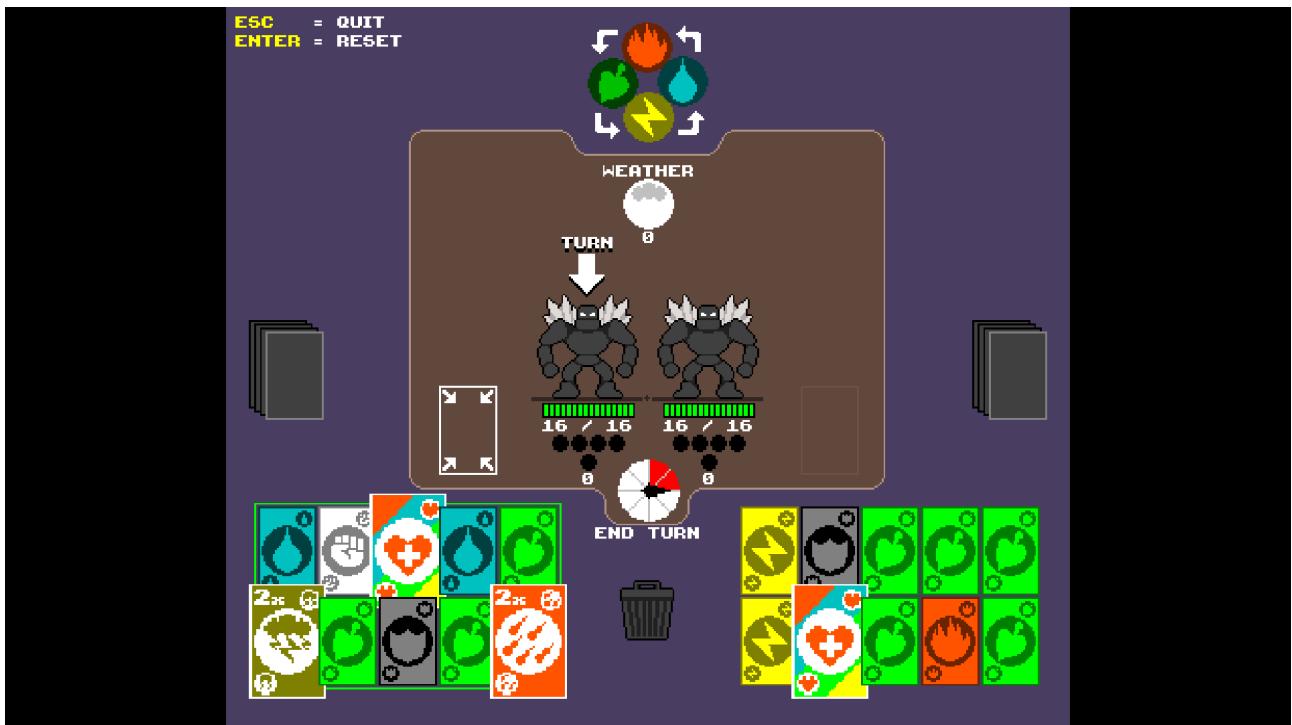
Redogörelse för arbetet

Detta kapitel redogör arbetet uppdelat i implementation och utvecklingsprocess.

4.1 Implementation

För att fullborda projektgruppens vision av projektet krävdes flera olika verktyg där spelresurser (som bild-, ljud- och animationsfiler) kunde skapas och sedan exporteras för användning i spelet. Hur dessa element skapades och vilka verktyg som används beskrivs i detalj i detta kapitel.

4.1.1 Prototyp i GameMaker: Studio



Figur 4.1: Prototypens spelmiljö i GameMaker: Studios exekvering

Innan arbete på själva projektspellet började skapades en mycket simpel föregångare med hjälp av *GameMaker: Studio* [7]. Det är en spelmotor och utvecklingsmiljö som innehåller flera verktyg, till

exempel en integrerad bildredigerare, och är speciellt effektiv för att skapa tvådimensionella spel med pixel-grafik. Detta utnyttjades för att snabbt utveckla en spelprototyp där alla viktiga spelregler implementerades med kod, medan grafiskt skapande inte krävde mycket fokus.

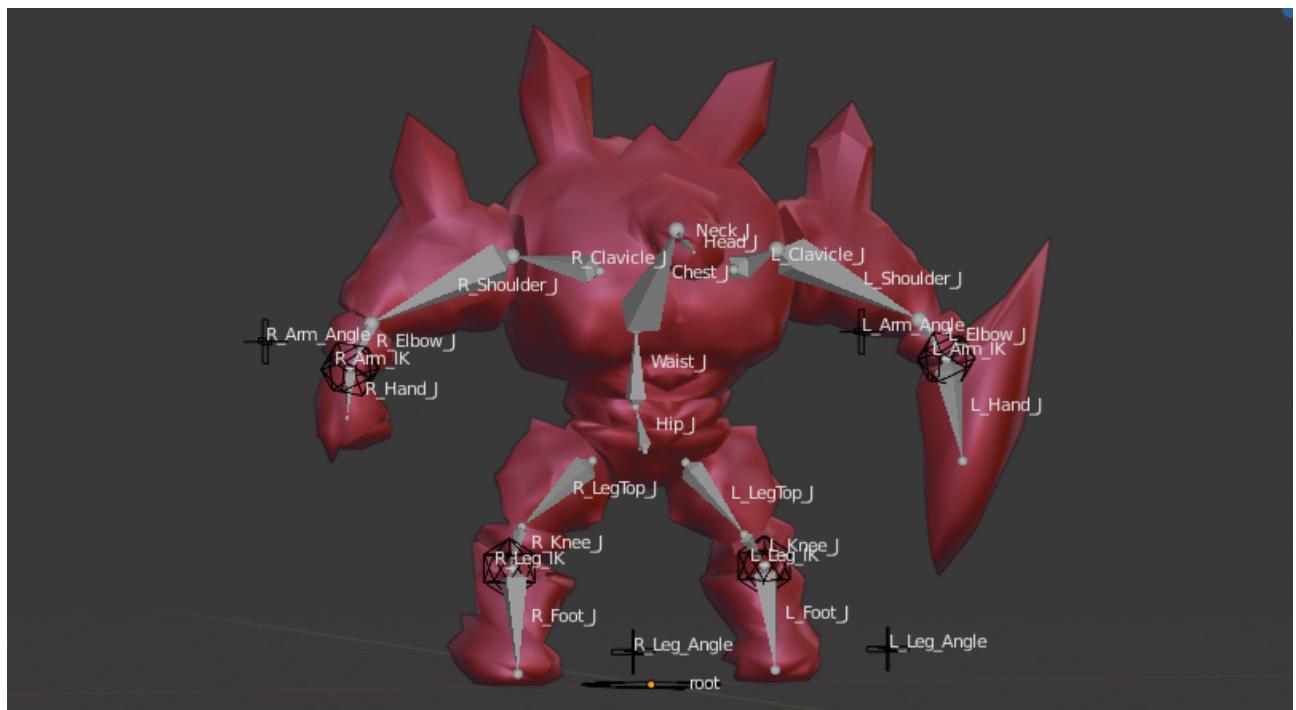
Istället för att två spelare använde varsina VR-utrustningar på separata datorer kopplade via en server kunde två spelare dela samma skärm och turas om att använda en datormus. Med denna kunde spelare dra, placera och klicka på grafiska representationer av sina spelkort på sin sida av skärmen omkring i en nära statisk spelmiljö för att kontrollera varsin golem-pjäs enligt figur 4.1.

Gruppen kunde snabbt speltesta projektidén med en färdig prototyp för att få underlag på vad som skulle göras i det faktiska spelet. Framför allt skulle detta hjälpa med återkoppling på spelregler som implementerats och generellt hur roligt spelet som koncept var för testare.

4.1.2 2D-Grafik i Procreate

Spelkortens framsidor behövde bilder som skiljde dem åt och illustrera det element och attribut de ska representera. Dessa skapades med hjälp av applikationen *Procreate* [8], en iPad-applikation med bildredigerings-verktyg för att sedan exportera bilder i JPG-format. Dessa kunde sedan appliceras som material på 3D-modeller av korten.

4.1.3 3D-Grafik i Maya och Blender



Figur 4.2: Golem modell med armatur i Blender

Spelets två spelare behövde tredimensionella representationer av spelkort att spela med. En kortmodell skulpterades i *Blender* [9] för att representera dessa, med redigerbart material på deras fram och baksida.

Även vissa miljö-objekt modellerades i Blender, som piedestaler för spelarna att stå på och ett antikt colosseum (arena) som spelet utspelar sig i.

Spelarna behövde dessutom en varsin pjäs i form av en golem (stenjätte), och dessa skulle representeras av kopior av en och samma 3D-modell. Denna modellerades i *Maya* [10]. Med en så kallad ”Mesh” (det synliga av modellen) fäst på en osynlig armatur kunde modellen röra/böja sina kroppsdelar. Den exporterades i fbx-format för att sedan animeras i *Blender* enligt figur 4.2. (se nästkommande rubrik)

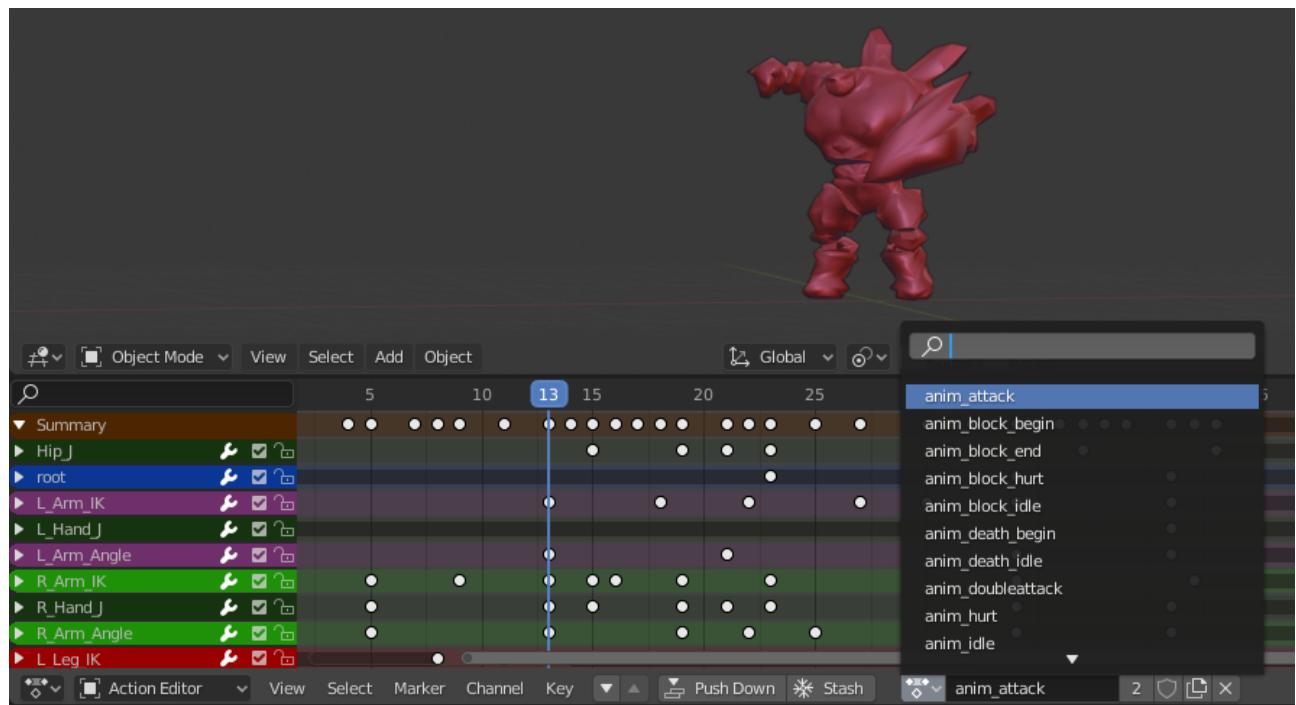
Notera att dessa moment kunde ha utförts i enbart en av de benämnda 3D-mjukvarorna (*Maya* eller *Blender*), eftersom de innehåller liknande utvecklings-möjligheter för sina respektive inbyggda verktyg. De uppdelades här helt enkelt baserat på gruppmedlemmars erfarenheter i dem.

4.1.4 3D-Animation i Blender

Golem-modellen behövde animationer för att meddela för spelare vad de gör i spelet (Till exempel behövdes en attack-animation som spelas upp när en spelare aktiverar ett attack-kort). Dessa animationer skapades i *Blender*.

Med så kallade *keyframes* kunde modellen poseras genom att man rör/böjer på kroppsdelar med hjälp av dess armatur. Flera keyframes uppställda på en tidslinje ger modellen en animerad övergång mellan dem enligt figur 4.3.

Dessa exporterades återigen i fbx-format, men denna gång ingick animationens information i dem för att kunna implementeras i grafikmotorn *Unreal Engine*.



Figur 4.3: Exempel på animationsmiljö i *Blender*

4.1.5 Ljud och Musik

Ljudeffekter fick låg prioritet och därför användes ljudeffekter tillgängliga från internet. Specifikt hämtades ljudfiler ifrån *freesound.org*, en vinstfri webbsida med offentliga ljudbibliotek. Dessa ljud effekter hade syftet att indikera och förtydliga för spelare vad som händer i spelmiljön. Till exempel att en attack-ljudeffekt spelas när sin golem slår eller blir slagten av den andra.

Musik komponerades i *FL Studio* [11], en arbetsstation för digitalt ljud med instrument som antingen ingick i FL Studios inbyggda ljudbibliotek eller ljudfonter tillgängliga ifrån offentliga ljudbibliotek på internet. Separata musik-ljudfiler skapades, en som skulle spelas i spelmenyn och flera under spelets strid, som alternerar mellan varandra beroende på vad som händer i striden. Till exempel spelas musik under tiden en spelare kan lägga ett kort, medan den växlas till en annan då kortet aktiveras och en golem agerar.

4.1.6 Programmering och komponenter i Unreal Engine

Projektet programmerades med *blueprints*, enligt figur 4.4, internt i utvecklingsmiljön/spelmotorn Unreal Engine. Blueprints är en visuell programmeringsmiljö där blockprogrammering utnyttjas för att kalla på funktioner och använda variabler. Fördelar med detta är att koden inte behövde kompileras vilket ledde till en snabbare iterativ process. Det gick även att kombinera blueprints med skriven C++ kod som i så fall skulle ha skrivits i *Visual Studio* [12], men sådan kod undveks helt.



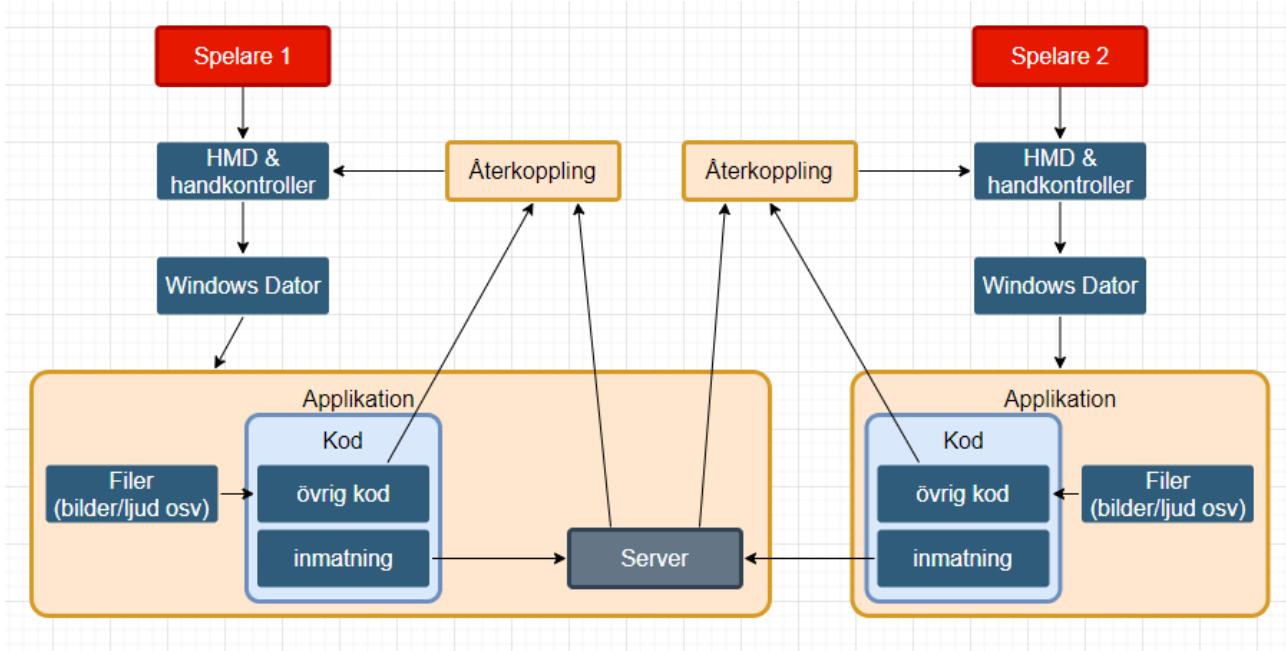
Figur 4.4: Exempel på blueprint-struktur

Assets (komponenter som filer) till spelet som grafik, animationer och musik importerades från olika verktyg beskrivna i överstående rubriker i detta kapitel. Kravet var att dessa program kunde exportera i filformaten som var bestämda för att Unreal Engine skulle kunna läsa in dem korrekt. De bestämda filformats-standarderna var följande:

- Ljud- och musikfiler hade ".WAV-format, valdes för att .WAV behåller ljudkvalitet
- Bildfiler hade antingen ".PNG"eller ".JPG-format, valdes för att de är vanliga bildformat
- Animations- och 3D-modellfiler hade ".FBX-format, valdes för att det är ett universellt 3D-format för alla av de använda/relevanta verktygen (Maya, Blender och Unreal Engine)

4.1.7 Systemarkitektur

Spelare skulle påverka spelet med hjälp av HMD och handkontroller. Denna inmatning skulle kopplas till en Windows dator (det var inte krav på att operativsystemet skulle vara Windows, men var snarare en egenbestämd begränsning) och skulle sändas till en applikation skapad med *Unreal Engine* (en spelmotor med förefintlig VR kompatibilitet och användes som huvudsaklig del av systemet). VR-utrustningens position och orientering spårades som variabler för spelkoden att använda sig av. Detta möjliggjordes med SteamVR, som kopplade VR-utrustningen och spelmotorn. Applikationens kod som kördes under tiden producerade en återkoppling för HMD-utrustningen, medan inmatningskoden först sändes till en server innan återkopplingen för att se till att en spelares inmatning påverkat alla deltagares återkoppling, se figur 4.5.



Figur 4.5: Grundläggande systemarkitektur

Specifikt, två spelare skulle spela spelet på respektive hårdvara med länkad serveruppkoppling via ett lokalt nätverk (specifikt en *listen-server* [13] som kördes på den ena spelarens dator för den andra att ansluta sig till), eftersom det var dugligt och simplare att implementera än online-uppkoppling.

4.1.8 Serverimplementation i Unreal Engine

För att två spelare ska kunna spela i samma scen i *Unreal Engine* krävdes arbete som omfattade att använda mycket av Unreal Engines egna visuella skriptningsspråk, blueprint, som nämnts tidigare. I blueprint skapades först en prototyp för att få två spelare/datorer att koppla upp sig på samma scen utan någon VR-funktionalitet för att se till att den delen fungerade. När det fungerade att få två datorer i samma scen märktes det dock att bara ena spelaren såg den andra röra sig men inte tvärtom. Detta problem undveks till en början.

Därefter satte gruppen upp den befintliga server *Blueprint*-implementationen som nämnts ovan med en existerande *VR-preset* som finns i Unreal Engines standardbibliotek. Denna *preset* har grundfunktionalitet för att få VR att köras i Unreal Engine. Ett problem som tog lång tid att felsöka var att kontrollerna till VR inte fungerade för den spelare som anslöt sig efter att ena spelaren startat en server. När detta löstes påbörjades en lösning för problemet med att spelarna inte såg varandras händelser, detta var för att *replication* inte var korrekt implementerat [5]. Replication ser till att objekt inuti scenen kan behandlas på samma sätt, det vill säga om en spelare håller ett kort kan andra spelare se att det kortet är upplockat och nu hålls av den andra spelaren. För att få detta att fungera behövdes de flesta handlingar i spelet som att knyta ihop handen, ta upp kort, slänga kort och så vidare byggas på med mer blueprint kod. Principen för att få det att fungera var att lägga till en komponent som sköter så att händelsen sker på serverägarens sida och att information skickas ut till den andra spelaren på servern. Samt en komponent som ser till så att information som den uppkopplade spelaren gör skickas till serverägaren. På detta sätt sker allting som har dessa komponenter tillagda simultant och på det sätt som spelarna förväntar sig när de spelar spelet.

4.2 Utvecklingsprocess

4.2.1 Versionshantering

Text- och kodbaserade filer versionshanterade med Git [14]. Med Git kunde utvecklare arbeta på samma filer samtidigt och privat på olika så kallade *branches* för att sedan föra samman de textstrimmor som varje utvecklare ändrat eller lagt till i samma fil som befinner sig på ett gemensamt *repository* online på webbplatsen *GitHub*. Dock användes nästan enbart *master-branchen* av spelmekanikutvecklarna, medan en annan branch användes av Server och VR implementations-utvecklarna.

Google Drive användes som fildelnings-tjänst för komponenter som inte var rent textbaserade (till exempel bild och ljudfiler), eftersom Git inte kunde versionshantera sådana på lämpligt vis.

Gits användning skedde genom Unreal Engines *Source Control*, vilket var Unreal Engines inbygga funktionalitet som automatiskt kunde uppdatera Git branches via internet avlägset så länge den var given en anvisning till ett lokalt Git repository. Git repository är en plats på datorn med filer som i tur ska skickas till en master branch, där olika personers textstrimmor förs samman. Att använda denna funktion underlättade versionshantering ytterligare, och var kritisk för att utvecklare skulle kunna arbeta utan att slösa tid på att vänta på varandra.

4.2.2 Kvalitetssäkring

Kvalitetssäkring på själva speldén gjordes genom användartester, se Bilaga B, på prototypen som vid detta lag var färdigutvecklat, för att få återkoppling på om speldén verkade omtyckt eller välkonstruerad, som sedan skulle behållas i åtanke inför utveckling av det faktiska spelet.

För själva projektspelet skulle all kod föras ihop från alla git branches (med branch merge) då det verifierats att de olika kodfilerna inte skulle ge upphov till konflikter när de förs samman i slutet av varje sprint. På grund av problem (som begränsad åtkomst till VR salar och parallella kurser som åt upp mer tid från projektet än förväntat) som resulterade i längsammare utveckling av spelet behölls utvecklaces versioner relativt separata tills nära slutet av projektets gång. Turligt nog var filer fortfarande i stort kompatibla efter de tre sammanslagningar som gjordes totalt under 2 månader. Efter dessa hopsättningar gjorde användartester. Även ett användartest med kund hölls för ytterligare säkerställning att spelet verkade omtyckt eller välkonstruerad, likt prototypen.

4.2.3 Dokumentation

I Unreal Engines blueprints skrevs kommentarer som förklarade nästan all funktionalitet i dem för att se till att den som läste kommentaren i tur lättare kunde förstå funktionaliteten. Dessa hade ingen standard, eftersom de som arbetade med koden förväntades använda dem som ledtrådar och därefter själva förstå kommentarer baserat på hur koden såg ut. Ständig kontakt mellan projektmedlemmar kompenserade för oklarheter.

4.2.4 Kravhantering

På Trello uppdaterade uppgifters status ständigt i form av kommentarer på moment, checklistor innanför momentfönster som kunde bockas av och möjligheten att flytta moment bland kolumner som "backlog", "to do", "doing" och "done" beroende på dess status. På så vis kunde gruppmedlemmarna hålla koll på vad som gjordes, vem som gjorde det och vad som fanns kvar att göra utan några svårigheter. Utöver det hölls möten för att garantera att alla gruppmedlemmar var informerade om

allt nödvändigt som mötesprotokoll. Detta ansågs mer än tillräckligt utan att göra förklaringar för belastande.

4.2.5 Organisation

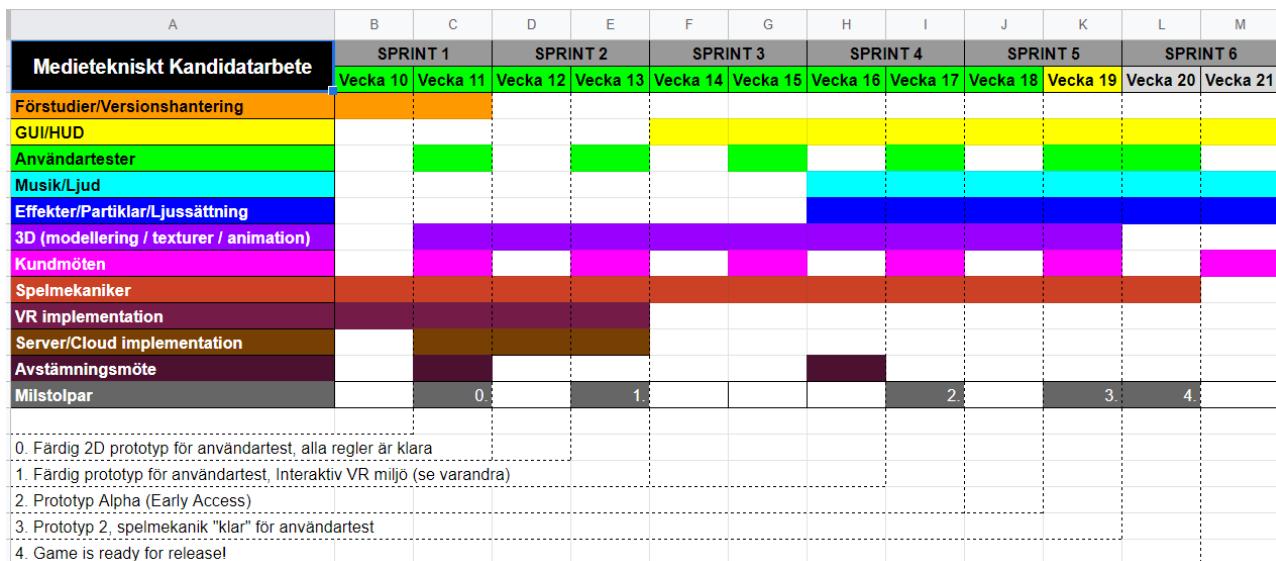
Med Scrum arbetade projektgruppen i regel som ett team snarare än med individuella uppgifter [18]. I detta projekt delades utvecklare dock upp i ansvarsområden någorlunda, med samarbete när det kom till just parprogrammering, eller härledning genom samtal med *screen-share* på IP-applikationen *Discord* [15].

Till exempel fick utvecklare hjälp av andra utvecklare när de hade fastnat på ett problem i Unreal Engine blueprints, som en annan kunde lista ut lösningen till tack vare att de kunde se varandras arbetsmiljöer via screen-share. Liknande härledning skedde med 3D animering och modellering, eftersom dessa utfördes på individuella hårdvaror, men med screen-share-baserat samarbete som kompromiss.

Däremot kunde flera personer jobba på separata animationer för samma modell samtidigt, eftersom dessa kunde föras ihop i en fil utan att påverka varandra. Liknande process gällde bilder (och alla sorts filer som inte praktiskt sätt kunde arbetas på samtidigt).

Under varje sprint utdelades arbetsområden för var utvecklare på så vis att alla fick pröva på flera områden. Konstant kontakt via möten och internet blev en kompenstation för utvecklare som hanterade uppgifter separerat från andra i projektgruppen.

4.2.6 Tidsplan



Figur 4.6: Gantt-schema

För att säkerställa att moment fullbordades i tid användes ett Gantt-schema, vilekt är ett flödeschema med moments tidsramar placerade över en tidsaxel. Schemat kunde anpassas efter ny information tagen från analys, testning eller diskussioner om sprints från schemalagda projektmötten. Därför var det mycket viktigt att ha till goda prototyper för återkoppling från användartesterna.

VR- och serverimplementation (att se till att VR utrustning fungerar, att en dator ger korrekt återkoppling i testapplikationer som använder sig av VR och att serverfunktionalitet kommer att kunna programmeras på önskat sätt senare) påbörjades tidigt eftersom det var det mest väsentliga av alla

moment och hela poängen med spelet var att det skulle kännas ”verkligt” (vilket inte åstadkoms väl på något annat vis än med VR med det beslutna upplägget.) Spelmekaniker kunde utvecklas parallellt med de mesta andra momenten genom att göra så att funktioner som krävde assets (modeller osv) först använde sig av simpla platshållare tills modeller och annat skapats och enkelt byttes ut. Därför arbetades spelmekanikerna på genom nästan hela tidsplanen.

Modeller och deras textur påbörjades tidigt eftersom de var de viktigaste av alla assets. GUI/HUD (grafiskt användargränssnitt) påbörjades senare för att det inte var lika väsentligt och därför lägre prioritet på. Samma sak gällde de flesta andra momenten, som ljud och partiklar. Användartester och avstämningssmöten skulle ske med jämna mellanrum för att få återkoppling med externa åsikter. Dock hölls dessa bara två gånger på grund av begränsad tillgänglighet från kunden. Nedan var det beslutna schemat som användes. (Se figur 4.6)

4.2.7 Kundmötens

Två avstämningssmöten hölls under projektet för att få återkoppling med om spelet uppfyllde de krav och önskemål kunden hade. Överlag så fick gruppen ganska fria tyglar kring vad som utvecklades.

Kapitel 5

Resultat

I detta kapitel presenteras resultatet av projektet.

5.1 Spelet

Målet var ett VR-spel som utmanar spelarnas kognitiva förmåga och prototypen var till stor hjälp för att få till ett resultat som var just det. Spelet utövas av två spelare på ett lokalt nätverk, där hårdvaran är datorer med HTC Vive kontroller och HMD. Spelarna börjar spelet genom att välja att vara host av en server eller att gå med den redan existerande servern. De två spelarna transporteras sedan till spelet som ses i figur 5.1 där vardera spelare placeras vid varsin spelyta och omsluts av den storslagna arenan. Framför sig har de två golems som ska genomgå den storartade striden.



Figur 5.1: Spelet i action från spelarens vy.



Figur 5.2: Modellen som indikerar tid som gått under spelets rundtur.

5.2 Spelmekanik

Spelet bygger på omgångar, att spelarna efter varandra har en begränsad tid på sig att utföra sina drag. I figur 5.1 ses användargränssnittet som spelarna ställs inför. Det finns elementkort som används dels för att kontra motståndarens element, men korten i sig kan även kombineras för att skapa en varande vädereffekt. Utöver de fyra olika elementkorten finns det attackkort och försvarskort, dessa kort används för att få sin golem att attackera eller försvara sig.

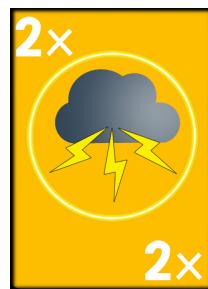
Spelarnas kognitiva förmåga sätts alltså på prov när de under tidspress ska hålla reda på vilka kort de har tillgång till, vilka element motståndarens och sin egna golem har, samt om en vädereffekt är i spel.

5.3 Gränssnitt till användaren

För att ge spelaren information om hur lång tid det är kvar och har gått under en rundtur valdes en klocka som modell eftersom den är förknippad med tid. En bild på klockan som används i spelet syns i figur 5.2. I klockan motsvarar ett helt varv hela spelarens runda.

För spelaren att veta vems tur det är placerades en platta som ändrar färg under spelplanens spelbräde, där grönt visar att det är ens tur och rött att det är motståndarens tur. Detta syns i figur 5.1.

Korten i spelet är färglagda med olika färger utifrån vad de gör. Korten som styr golems är gråa för att vara i samma färg som golems. På samma sätt valdes elementkortens färg att vara i samma färg som elementen, vilket gör det enkelt att hitta dem i korthanden. Kart som ger användaren möjlighet att plocka flera kort eller hela sin golem har regnbågsfärgad bakgrund för att visa att de är specialkort. Kart som är från sammanslagning av kort får även texten $2\times$ i hörnen för att indikera detta, ses i figur 5.3.



Figur 5.3: Exempel på ett spelkort i spelet. Kartet ändrar vädereffekten till åska.

Det användes också *highlights* runt objekten som man kan interagera med i spelvärlden så att det inte är oklart vad som går att interagera med och vad som inte går, vilket visas i figur 5.4.



Figur 5.4: Exempel med highlight runt korten i spelet.

5.4 Scenuppbyggnad

För att lösa problemet med att få en storhetskänsla i spelet utan att förlora överblicken på vad som händer bestämdes det att placera spelaren på en platå så de kan se ut över scenen. Detta medför att objekten kan göras stora och att spelaren fortfarande kan se vad som sker i spelet.

Arenan som omsluter spelarna i scenen användes som en visuell barriär för att blockera spelarnas sikt i scenen. Detta utnyttjades för att kunna minimera området som effekter behövs genereras i. Tack vare detta kunde projektet innehålla effekter som annars hade varit för krävande att rendera. Behovet av modeller för miljön minskades även då arenan tar upp den mesta av spelarnas sikt.

Kapitel 6

Analys och diskussion

I detta kapitel analyseras och diskuteras organiseringen, resultatet och de metoder som har använts i projektet. Det förs dessutom en diskussion kring etik och samhälleliga aspekter relaterade till arbetet.

6.1 Analys av metod

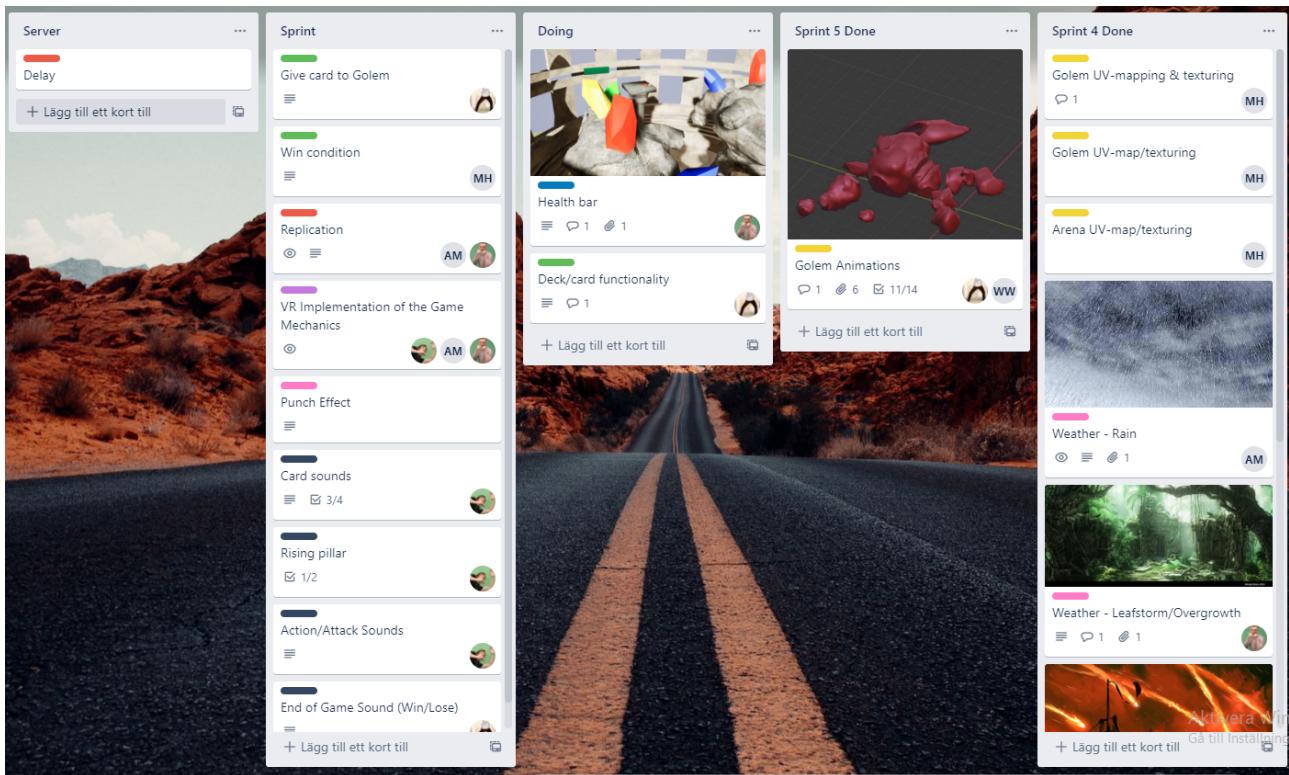
På grund av coronavirusets sammanträffande utbrott och påföljd karantän under projektet fick det arbetsupplägg som planerats i projektplanen ändras på kort varsel. Mycket av arbetet har fått utföras på distans istället, vilket har gjort att sprintplanering och arbetsfördelning fått anpassats efter det. Något som direkt är värt att nämna är att gruppen överlag har klarat att bedriva arbetet på distans mycket bra (åtminstone med tanke på motgångarna). Det som ligger till grund för det är en god kommunikation i gruppen och även användandet av programvaror som *Trello* och *Discord*.

6.1.1 Organisation

Projektgruppen har använt den agila utvecklingsmetoden *Scrum* och har till stor del följt riktlinjerna enligt metoden. En medveten förändring som dock gjordes var att tillåta längre morgonmöten än 15 minuter eftersom mycket av arbetet har utförts på distans. Dessa morgonmöten blev därför en viktig del av kommunikationen under arbetet för att kunna stämma av och diskutera läget i projektet. Inför varje sprint har det hållits ett mer omfattande möte där sprinten planeras, vilket har varit en viktig del av projektet. Mötena har dock lidit av en viss brist i närvaro hos projektmedlemmarna av olika anledningar, vilket försvårade planeringen. Om samtliga hade deltagit så skulle det ha varit lättare att planera lagom mängd arbete och delegera ut arbetsområden till utvecklarna.

Gruppen delade upp projektet i två-veckors sprints. Planeringen av sprints utgick till stor del från det Gantt-schema som skrevs i början av projektet och underlättade mycket i upplägget av varje sprint. Det är alltid svårt att planera hur lång tid olika delar av projektet kommer att ta, men den tidsplan projektgruppen skapade har överlag stämt bra överens med tiden som faktiskt lagts på respektive del. Den enda del som egentligen tog längre tid än planerat var att få spelet att fungera för två spelare. Mer om det under tekniska lösningar.

För att dokumentera planeringen av projektet och sprints har *Trello* använts. Många av medlemmarna i gruppen hade innan projektet inte använt *Trello*, men programmet gick snabbt att förstå och har fungerat mycket bra för samtliga. *Trello* har gjort det enkelt att delegera uppgifter till utvecklarna och för alla att se vem som jobbar med vad. Ett exempel på hur *Trello*-tavlorna sett ut under projektets gång kan ses i figur 6.1.



Figur 6.1: Ett exempel på hur Trello-tavlan har sett ut under projektet.

En sak som kunde ha gjorts bättre med Trello är att ha mer specifika och konkreta kort. I detta projekt har en del kort varit för breda och generella vilket har gjort det svårt att avgöra om kortet kan anses som klart eller om det krävs mer arbete.

Metoden som används vid utveckling av spel dikteras mycket av den hårdvara och de verktyg som finns tillgängliga just vid det tillfället. Dessutom utvecklas spelbranschen i snabb takt och det dyker ständigt upp nya tekniker och metoder att utföra saker på. På grund av detta kan det i framtiden bli svårare att återskapa detta spel genom att använda samma metod som användts detta projekt. Med det sagt så har ändå spelet en relativt god replikerbarhet då de tekniker och verktyg som utvecklas nästan alltid bygger vidare på tidigare verktyg. Det innebär att det alltid kommer att gå att skapa samma saker som i detta spel, fast antagligen på ett enklare och bättre sätt. På grund av detta har spelet en någorlunda låg reliabilitet. Med nya verktyg och tekniker så ändras metoden för att göra saker vilket gör att just denna metod relativt snabbt kan bli uppdaterad. Dessutom avgör utvecklarnas kunskap och färdigheter i stor utsträckning slutprodukten.

6.1.2 Tekniska lösningar

Ett tidskrävande problem som uppstod tidigt i projektet var att få två spelare att korrekt kunna se och spela mot varandra i VR. Denna process beskrivs i kapitel 4. Som en förstudie till projektet hade gruppen utforskat olika metoder för att få två spelare att kunna spela mot varandra. En metod [6] hittades och testades och ansågs tillräcklig för detta spel. Problemet uppstod då projektet drogs igång och denna metod skulle överföras till att vara kompatibel med VR och dessutom mer avancerad spelmekanik. Detta tog mycket tid att lösa och fick som konsekvens att resurser från andra områden fick läggas på detta. Här skulle det kunnat utförts en mer noggrann förstudie.

6.1.3 Källkritik

Den mest relevanta informationen för detta projekt har kommit från dokumentation, forum och kortare artiklar. Majoriteten av informationen har kommit från Unreal Engines egna dokumentation, genomgångar och exempel. Denna information är högst pålitlig och utvecklarna har i första hand utgått från denna källa under projektet. Då den dokumentationen inte räckt till har de även ett forum där det är fritt fram att ställa frågor. Här har det varit viktigare att kritiskt granska de svar som funnits. En metod har varit att endast kolla på de trådar som besvarats av en anställd på Epic Games (företaget som utvecklar Unreal Engine) eller en erkänd erfaren och pålitlig utvecklare, som kan identifieras på den rank de har på forumet.

6.2 Analys av resultat

Överlag så är resultatet präglat av att en hel del önskade funktioner fick skalias bort under utvecklingen då det inte fanns tillräckligt med tid. Ett sådant exempel var att ha publik med i arenan, vilket hade varit en häftig effekt men som i slutändan fick prioriteras bort.

Ett mål med projektet var att spelet skulle innehålla någon typ av spelgenomgång för de nya spelarna. Även detta fick prioriteras bort på grund av tiden men tack vare att grafik och spelmekanik designades att vara intuitiv och användarvänlig så går det relativt snabbt att förstå hur spelet fungerar. Det vittnar även användartesterna om som gjordes på prototypen för att testa spelmekaniken.

6.3 Arbetet i ett vidare sammanhang

Spel i VR kan framkalla illamående och huvudvärk hos vissa användare. Saker som ett väldigt överväldigande gränssnitt och mycket skakningar och rörelse i VR-headsetet ökar risken för det. Detta har alltid varit i åtanke under utvecklingen. Ett exempel på det är vädereffekterna som hade kunnat skapats med skakningar och blinkande ljus. Istället behölls dessa effekter relativt enkla för att minska den visuella påfrestningen för användaren.

Som tidigare nämnts är spelet tänkt att utmana spelarnas kognitiva belastning, stressstålighet och strategiska tänkande vilket är egenskaper som är användbara i spelsammanhang såväl som i vardags- och arbetslivet. Spel som detta som är baserade på strid eller våld i någon form kan låta avskräckande då somliga undviker att involvera sig med sådana. Det är något som förhoppningsvis ska ha åtgärdats genom att de visuella objekten, effekterna och gränssnittet alla är fiktiva och inte är direkt baserade på verklighet.

Interaktionen med korten kräver inga större rörelser eller förflyttningar vilket gör att även användare med rörelsehinder eller andra svårigheter kan spela detta spel utan större svårigheter. Man behöver heller inte spela spelet stående vilket gör att rullstolsbundna och andra användare kan använda produkten sittandes eller liknande.

Kapitel 7

Slutsatser

Projektets syfte var att skapa ett strategibaserat kortspel i VR där två spelare ska, med hjälp av varsin HMD och handkontroller, spela mot varandra i en virtuell miljö. Syftet med spelet var att förse användaren med flera valmöjligheter för att utmana användarnas kognitiva förmåga. Med hjälp av strategi och list som kan utnyttjas i spelet, tävlar användarna om att vinna över den andra. Genom kundmöten, användartester och noggrann spelbalansering kunde målen uppfyllas för de tidigare nämnda fallen.

7.1 Besvarande av frågeställningar

Hur kan en storhetskänsla skapas i spelet utan att kompromissa för prestanda med den hårdvara som finns tillgänglig?

Genom god placering och distans för objekt i den virtuella världen kan storhetskänsla uppnås. Tidigt i projektet bestämde projektgruppen att spelet ska ha lägre antal polygoner i modeller med syftet att kunna spara in på prestandan. Projektgruppen har kommit överens om att för att behålla en storhetskänsla för både golems och arenan är en god lösning att placera spelarna på upphöjda plattformar i den virtuella miljön istället för att skala ner eller göra drastiska ändringar på modellerna. Upphöjda kan spelarna få uppfattningen om hur stor arenan och båda golems är, utan att riskera att modeller hamnar för långt bort eller för nära. Det riskerar att bli en obehaglig upplevelse för användarna om saker placeras för nära eller för långt bort i den virtuella miljön. (Med en HMD kan det mänskliga ögat automatiskt försöka att fokusera och det vill man undvika på alla sätt man kan.) För att förhindra att prestandan och främst upplevelsen försämras i spelet har inställningar såsom *FOV*, *motionblur* och *depth of field* inte experimenterats med utan bibehållits till rekommenderade inställningar från ytter källor.

Hur kan vi uppnå ett spel som är både kognitivt belastande och underhållande men som samtidigt erhåller en struktur i turfördelningen och kräver att spelaren måste göra taktiska val?

Genom att förse användarna med valmöjligheter och tidspress ökar den kognitiva belastningen för hjärnan. Spelet gynnar spelarnas förmåga att kunna agera snabbt och välja sin strategi beroende på situation under spelets gång. Spelarna kan inte bara lägga ut kort utan även sortera kort, ta upp nya kort och kombinera kort med varandra för att få andra effekter, vilket ökar valmöjligheterna och även effekten på den kognitiva belastningen.

För att hålla spelet underhållande använder spelmotorn sig utav slumpfaktor när spelarna drar nya kort. Slumpfaktorn är balanserad för de olika korttyperna och bidrar med spännande och unika spelomgångar vid varje nytt spel. Som tidigare nämnt är spelet anpassat för att låta användarna komma med egna taktiker och strategier, bland annat defensiva eller offensiva. En spelare kan upptäcka ett överläge med sina kort på handen över den andra spelaren och kan välja att spela defensivt för att utnyttja

sin position och vinna genom att inte låta motståndaren göra något vinnande drag. En spelare kan också upptäcka innan motståndaren får övertag med kort och då spela offensivt för att vinna innan motståndaren hinner bygga upp ett försvar.

Spelet har utöver en engagerande spelmekanik också spännande miljö som ändrar tillstånd efter spelarnas handlingar. När spelarna lägger sina kort spelar dennes golem ut detta i form av animation eller färgändring. Spelarna kan även lägga kort som ger vädereffekter i arenan för att öka den immersiva känslan.

För att förtydliga för användarna vems tur det är i spelet använder spelet sig av ett bord framför spelarna som ändrar färg och även ett fickur som ligger på bordet som visar hur lång tid spelaren har för att göra draget. Bordet ändrar färg till grönt när det är spelarens tur för att tydligt indikera och ge feedback till spelaren att det är dennes tur. Fickuret tickar ner ett helt varv för att ge spelaren en uppfattning om hur mycket tid spelaren har att göra sitt drag på. Eftersom det är svårt att ge bra feedback i VR utan att det blir konstigt eller obehagligt för spelarna är det bra att lägga in fler än en indikation/feedback på vad spelaren måste göra eller vad spelaren är i för situation.

Hur kan ett gränssnitt skapas som inte förlitar sig på text utan representeras framförallt med form och färg som snabbt ger användaren en klar bild om vad objekten gör?

Spelet har många moment och mycket information som är viktigt för användaren att kunna uppfatta och handla utefter. Till exempel är det viktigt för användaren att veta när det är dennes tur att lägga ut ett kort som nämndes ovan. Det är speciellt viktigt i VR att ge användarna tydlig feedback och återkoppling på deras handlingar i spelet, eftersom det är svårt att läsa text och användarna måste få ha fri orientering för att inte ge en obekväm upplevelse. Genom att använda naturliga objekt från verkligheten som relateras med dess funktion kan användarna intuitivt hänga med i spelet. Spelet använder sig av kort med symboler för att spelarna ska slippa läsa texter på korten och istället få en uppfattning om vad kortet gör/handlar om. Färger och symboler som används är till exempel eld, rött, löv, grönt osv. För att användarna snabbt ska kunna förstå spelet hjälper dessa intuitiva val projektgruppen gjort, istället för att behöva lära dem något nytt. Dessa val av objekt och symboler har visats vara intuitiva och fungera bra enligt svaren från användartesterna. Användartesterna har också givit bra feedback som projektgruppen tidigare inte tänkt på och övervägt att använda sig av. Till exempel var efterfrågan att få sortera sina kort på handen stor, varpå gruppen övervägde att lägga ner arbete på den funktionen. Funktionen togs bort vid senare tillfälle för att det blev mer av en besvärlig effekt då det inte såg bra ut snarare än en bra funktion som spelaren kan använda sig utav.

7.2 Vidareutveckling

Vad som planeras att vidareutveckla i projektet är några förändringar för att förbättra den visuella upplevelsen. Gruppen har länge diskuterat spelarnas indikation på vems tur det är och kom på idén om att spelarens golem skulle kunna sträcka ut sin arm fram till spelaren (häftig upplevelse i VR-miljön) för att visa att spelaren ska lägga/aktivera ett kort i golems hand. För att förtydliga det ännu mer skulle handen kunna få en *outline* och lysa när spelaren närmar sig med ett kort mot golems hand för att indikera att om spelaren släpper kortet kommer golem att få det i handen och aktivera det. Denna lösning hittades inte med i utvecklingen men projektgruppen är enade om att detta skulle förstärka upplevelsen i både spelsammanhanget och miljön.

Fler indikationer och återkopplingar till spelarna som till exempel ljusindikationer, vibrationsfeedback och ljudindikationer skulle ge en bättre första upplevelse för nya användare specifikt. Innan man vet hur funktionerna i spelet fungerar kan det vara svårt att förstå hur man spelar även om de flesta funktionerna är anpassade till att vara intuitiva. Vibrationer, som stöds i handkontrollerna, skulle öka intuitionen om vad spelaren kan greppa med kontrollerna eller var spelaren kan lägga kort till exempel.

Vibrationer skulle även öka inlevelsen om en golem skulle få ett attack-kort, slå motståndarens golem för att sedan ge båda spelarna vibration i handkontrollerna för att visa på kraften i de stora stenjättarna i arenan.

Indikationerna för varje golem element (kristaller på ryggen) skulle kunna förbättras. Istället för att spelarna ska behöva titta bort mot båda golems kristaller på ryggen för att se vilka element som är aktiverade, skulle bordet framför spelarna kunna ha iconer som fylls i eller lyses upp för att indikera just detta. Detta skulle kunna förbättra spelupplevelsen genom att tillåta spelaren ha mer fokus på bordet och sina kort.

Ljudet i spelet skulle kunna förbättras och anpassas i miljön. För att öka storhetskänslan i den stora arenan skulle eko kunna läggas till för vissa av ljuden. Till exempel när golems släss och stenar flyger, eller när tiden går ut och det ger ifrån sig ett ljud för att sporra spelaren. För att miljön inte heller ska kännas tom och orealistisk behövs *ambient sound* som alltid finns i verkligheten. Fågelkvitter, vindsus eller andra naturliga ljud skulle kunna läggas in i bakgrunden för att ge miljön mer liv. Ljudindikationer på när spelaren handskas med kort skulle också ge bra feedback till spelaren så att det skulle ge en naturlig känsla att plocka med korten.

För att spelet ska kännas flytande, utan avbrott eller konstiga händelser, skulle förfining av animeringar, modeller och spelfunktioner behövas. Även fler buggtester för att upptäcka buggar som förstör spelupplevelsen skulle behövas, men som inte har hunnits med.

Litteraturförteckning

- [1] Epic Games, *Virtual Reality Best Practices*, Unreal Engine, hämtad: 2020-05-12
<https://docs.unrealengine.com/en-US/Platforms/VR/DevelopVR/ContentSetup/index.html>
- [2] Raaen K., Kjellmo I., *Measuring Latency in Virtual Reality Systems*, International Conference on Entertainment Computing, Springer, 2015, hämtad: 2020-02-27
https://link.springer.com/chapter/10.1007/978-3-319-24589-8_40
- [3] Konami, *What Is Yu-Gi-Oh!?*, Yu-Gi-Oh!, hämtad: 2020-05-12
https://www.yugioh-card.com/en/about/parents_what-is-ygo.html#tcg
- [4] Nintendo, *How to play the Pokémon trading card game*, Pokémon, hämtad: 2020-05-12
<https://tcg.pokemon.com/en-us/how-to-play/>
- [5] Epic Games, *Actor Replication*, Unreal Engine, hämtad: 2020-05-12
https://docs.unrealengine.com/en-US/Resources/ContentExamples/Networking/1_1/index.html
- [6] Matt, *How to use Multiplayer Sessions in your Unreal Engine 4 game*, Couch Learn, 2019-07-31, hämtad 2020-03-03, <https://couchlearn.com/multiplayer-sessions-in-your-unreal-engine-4-game/>
- [7] YoYo Games Ltd., *GameMaker: Studio*, GameMaker: Studio, 2018, hämtad 2020-05-27, <https://docs.yoyogames.com/>
- [8] Savage Interactive, *Procreate*, Procreate, 2019, hämtad 2020-05-27, <https://procreate.art/ipad>
- [9] Blender Foundation, *Blender 2.82a*, Blender, 2020, hämtad 2020-05-27, <https://www.blender.org/>
- [10] Autodesk, *Programvara för 3D-datoranimering, modellering, simulering och rendering*, Maya, 2020, hämtad 2020-05-27, <https://www.autodesk.se/products/maya/overview/>
- [11] Image-Line Software, *FL STUDIO 20*, FL Studio, 2020, hämtad 2020-05-27, <https://www.image-line.com/flstudio/>
- [12] Microsoft, *Visual Studio*, Visual Studio, 2020, hämtad 2020-05-27, <https://visualstudio.microsoft.com/>
- [13] Valve, *Hosting a Local (Listen) Server for Valve Games*, Steam, 2020, hämtad 2020-05-27, https://support.steampowered.com/kb_article.php?ref=5452-TASB-6078

- [14] Jonathan Silva, *What is a Git repository?*, Axosoft, 2018-04-10, hämtad 2020-05-27, <https://blog.axosoft.com/learning-git-repository/>
- [15] Discord Inc., *Ett nytt sätt att chatta med dina communityer och vänner.*, Discord, 2020, hämtad 2020-05-27, <https://discord.com/>
- [16] Singla, Ashutosh and Fremerey, Stephan and Robitza, Werner and Raake, Alexander, *Measuring and comparing QoE and simulator sickness of omnidirectional videos in different head mounted displays*, 2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX), IEEE, 2017, hämtad 2020-05-28, <https://www.liebertpub.com/doi/full/10.1089/g4h.2017.0114>
- [17] SteamVR — Everything you need to know, PC Gamer, 2015-05-28, hämtad: 2020-05-28
<https://www.pcgamer.com/steamvr-everything-you-need-to-know>
- [18] Ken Schwaber och Jeff Sutherland, *The Scrum Guide*, Scrum.Org and ScrumInc, 2017, hämtad: 2020-05-28
<https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Swedish.pdf>

Bilaga A

Ansvarsområden

Tabell A.1: Projektets medlemmar med ansvarsområden

Namn	Ansvarsområde	Summering av enskilt arbete
Anton Lindskog	Kundavstämning och kontakt	Ansvarade för kontakt mellan kunden och övriga kontakter under projektet. Var delvis dokumentansvarig och höll ordning på backloggen. Gjorde modellering och riggning av 3D-modeller samt anordnade SFX (ljudeffekter).
Andreas Wahlgren	Användartester	Ansvarade för spelprototyp i 2D och balansering av spelmekaniken. Ansvarade för användartester och dokumentering av feedback från användare. Sammanställde en manual för användarna om spelets funktioner. Arbetade med en del modeller och animationer.
Arvid Magnusson	Server och VR-implementation	Arbetade med att få till server och VR-implementeringen. Ansvarade för vädereffekterna i spelet och skapade 2d-grafiken för korten.
Martin Hag	Ljus och effekter	Ansvarade för versionshanteringen i projektet. UV-mappade och texturerade golem modellen och arenan. Kodade kortleksmekanik och golemmekanik i UE4. Integrerade animationerna med koden iUE4. Undersökte <i>volume based shaders</i> för vädereffekter. Arbetade även med koden för rundorna i spelet och skapade modellen, uv-mappade och la på texturer på klockan för att kunna visa tiden.
Oskar Tengvall	Spelmekanik och musik	Utvecklade spelprototyp som spelades under användartester och bidrog med manualsekrivning. Implementerade spelkortens spelmekanik, komponerade musik och animerade golems.
Viktor Sjögren	GUI/HUD och gränssnitt	Arbetade med serverimplementation, skapade GUI-element som livmätare. Skapade en resande pelare och viss kortinteraktion i VR med Blueprint-skriptning samt skapade ett partikelsystem som efterliknar en vädereffekt.
William Malteskog	3D-modellering	Arbetade med serverimplementation för att få två spelare att kunna spela mot varandra lokalt. Modellerade och texturerade meteorer samt implementation av vädereffekten eldstorm

Bilaga B

Användartester

Fanny & Peder

Fanny got the hang of it snabbt genom Manualen! Positivt!

Spelet:

Det var svårt att veta vems tur det var i början.
Glömmer dra nya kort
Fanny- "Peder har eld, då drar jag vatten"
Fanny- "Peder, nu slår jag dig"
Fanny- "Men hallå vi har glömt att dra kort"

Peder- "Nu ska jag slå dig, för jag är snäll"

Fanny- "Du spelar fel"

Peder- "Jag kan dra tre extra kort!" Fattade han direkt

Tar lång tid tills de fattade att man kunde kombinera kort på sin hand.

Det tog lång tid att förstå väderkortens funktioner.

Peder var uppmärksam på hur mycket skada han skulle göra i förtid.

Peder drog eld "sak" (firestorm) och märkte inte att han tog bort sitt gröna element.

Fanny- "Jag blockar mig gånger två"

Peder kom på att man kan lämna över turen genom att klicka på klockan!

Fanny- "nu måste jag skydda mig känner jag"

PEDER VANN.

----Eftersnack:

Weather - Superelement? Man fattar att det att det var starkare eftersom det stod x2

det svåraste var att fatta att man kunde dra nya kort.

Svårt att förstå hur man gör skada, och elementen är svåra. Vissa element är starka mot andra. Den övre bilden på elementen hjälpte Fanny men Peder tänkte inte på den. Känns som man kan nörra in sig på detta spel för att bli bättre, ändå lärde man sig det snabbt tycker Peder. Fanny gillar Golem, den är gullig.

Fett kul, bra jobbat! - sa de innan de gick.

Figur B.1: Användartest prototyp del 1

Thobias & Lina

Lina " vad var el bra mot"
Tobbe, " aha nu blev jag skadad" Fattade också direkt att kunna dra kort.
Tobbe tror han hänger med nu redan. Försökte kombinera attack med block.
Det gråa i block såg ut som den inte var available - bra feedback.

Frågade och sen förstod att man kunde kombinera kort på hand nu.

Tobbe " EYY gubben ändrar sig, coooool!"

Lina "Healade du dig nu, chicken?"
Tobbe "det står här uppe vad som är bra mot vad!"
Lina, svårt att veta vad.
Tobbe " kommer jag ha kvar elementet nästa runda?"
Lina slängde kort, men det hjälper inte!! Vad är slänga kort bra för?
Lina "Jag behöver det där plus tre kortet"
Lina förstår att Tobbe inte kan döda henne för hon ser hans hand.
Tobbe förstod att Lina inte skulle göra någon damage.

LINA VANN

Det känns som det var korta runder, till en början känns det dåligt. Få mer tid som nybörjare.
Bra att guiden fanns men den syntes inte först. Kul att få feedback på att kombon var
starkare. Det var svårt utan attackkort??
**-Kanske någon indikation att om man lyfter ett kort så ser man vad man kan
kombinera de med.**
Jag hade velat att den drog kort åt mig.

EN GÅNG TILL

Tobbe försökte sortera sina kort på hand.
-Indikation i förväg på vad väder släcker ut!!! Håller man i Firestorm vill man se typ kryss
över vatten på sig själv och motståndargolem!
Soptunnan är för stor för sitt syfte, för man får inte något för att slänga kort. (Vi förklarade att
ibland är möjligheten att slänga kort bra för att har man fullt på hand men inte vill lägga ut
något kort av någon taktisk anledning så är det bra att kunna slänga ett kort för att kunna dra
ett nytt!)

Om man har under ett visst antal kort så får man dra fler kort, skulle kunna vara en grej!

Figur B.2: Användartest prototyp del 2

Andreas, Anneli & Alexander

Svårt att se i början vems tur det var

Försökte kombinera i aktiveringsfältet men sedan började kombinera på hand.

Glöm inte att ta kort sa Andreas som tittade på!

Börjar tänka, eld, nej vatten är starkt!

Alex förstod snabbt väder, "nu dödar jag din grej"

Alexander vann!!!

Det kändes punishing att missa ett kort, så kanske bättre att man automatiskt får nytt kort.
(återkommande feedback)

Jag hängde inte riktigt med att Elementen hängde med en under Golems. Om jag gör ett väder "blir jag det". Andreas noterade dock axlarna fort för elementen. När jag attackerade så "consumades" mina element, och jag ba aha! Tänkte jag skulle få super sen.

Det var svårt att okej, jag ser att han har dessa element, okej är det denna den är stark mot? Men förstår att man måste kunna spelet utan och innan

förväntar mig att om ett kort ligger redo ska man kunna swappa det genom att lägga ett nytt. Det är lite frustrerande att se att hon har eld men jag har bara grönt.
Det bästa är väl ändå att bygga upp attackerna. Jag kör mer på attack än defense.
Ju mer attack kort man får av samma, desto färre borde man få på hand.

Figur B.3: Användartest prototyp del 3