CLONE WARS

Spilrapport

Computerspilprojekt

Antal anslag (inklusiv mellemrum): 42.903

Antal sider: 17.9

Afleveringsfrist: 20-12-2019

Undervisere: Lasse Juel Larsen & Sara Mosberg Iversen

Alexander Svanholm (Eksamens nr.: 474918)

BA | 5. semester | Tilvalg | SDU

Alsva19

Frederik D. Christensen (Eksamens nr.: 474919)

BA | 5. semester | Tilvalg | SDU

Frchr19

Nadia C. Carlsen (Eksamens nr.: 166753927)

BA | 1. semester | OpTek | SDU

Nacar19

Nickolai Petersen (Eksamens nr.: 459368)

BA | 1. semester | OpTek | SDU

Nipet18

FORORD

Spillet 'Clone Wars' er blevet udviklet i spilmotoren Unity og administreret gennem et Git Repository i programmet GitKraken, som har en vis form for kollaboration funktion tilgængelig, således at alle gruppens medlemmer har kunne programmere og designe på hver sin computer og dele dette med de andre i gruppen gennem et depot i skyen. Spillet, som udgør den konstruerende del af designprocessen, sammenlagt med denne rapport udgør eksamensprojektet i faget 'Computerspilprojekt'. Rapporten beskriver hele forløbet af spillets tilblivelse, samarbejdet i gruppen samt en teoretisk evaluering af spillet. Hele projektet er udarbejdet af Alexander Svanholm, Frederik D. Christensen, Nadia C. Carlsen og Nickolai Petersen.

Table of Contents

INDLEDNING 1
DESIGNPROCESSEN 2
KONCEPT OG PRE-PRODUKTIONSFASEN 2
PRODUKTIONSFASEN 5
GRUPPENS SPILLEREGLER OG MEDLEMMERNES OPGAVER 13
Alexanders Projektandel og hans stykke kode 14
Frederiks Projektandel og hans stykke kode 15
Nadias Projektandel og hendes stykke kode 16
Nickolais Projektandel og hans stykke kode 16
VURDERING AF SPILLET FRA ET SPILTEORETISK SYNSPUNKT 18
I. Core Mechanics 18
II. Level design 18
III. Game feel 20
Hvilke elementer kunne tillægges spillet? 21
Konklusion 22

INDLEDNING

I projektets forløb har det endelige produkt taget mange former. Dette afbildes i blandt andet konceptualiseringen af spillet, prototyping, spiltest og endeligt selve spillet. Med grundig bearbejdelse af produktet og konstant perspektivering, har designprocessen resulteret i et 2D platform multiplayer shooter spil, hvor underholdningskvalitet og spilbarhed har været i stor fokus fremfor samfundsrelevante emner.

Formålet har været at lave et sjovt, interaktivt multiplayer spil med replayability features. Spillet er blevet produceret med udgangspunkt i Tracy Fullertons 7-trins procesmodel, hvor første del er brainstorming efterfulgt af en papirprototype, som præsenteres og testes. Efterfølgende laves der en software prototype, som der i denne opgave omtales den digitale prototype. Designprocessen dokumenteres løbende under produktionsfasen og der playtestes undervejs for at integrere brugere og for at gøre spillet bedre.

Designprocessen har ikke været ligetil, og det var først halvvejs gennem første spilidé, at gruppen indså, at et samfundsrelevant fokus i spillet ikke var det, som de skulle bane efter. Et spil behøver ikke altid at have realistiske undertoner eller have en større mening. Det skal være underholdende og fungere som en pause fra virkeligheden. Denne tanke overgik til eksamensproduktet 'Clone Wars', som giver spilleren mulighed for at påtage sig rollen som en genial, men skør, videnskabsmand, der er blevet sat i et surrealistisk dilemma, om at dræbe sin egen klon eller dø selv. Spillet efterlader også spilleren med spørgsmål, for hvem er egentlig klonen?

Der vil i det følgende redegøres for gruppens udførte designproces. Her bliver der belyst og tydeliggjort, hvordan valgene gennem processen har været præget af iterativt arbejde, og hvordan refleksion og de forskellige opståede situationer i hhv. konceptfasen, pre-produktionsfasen og produktionsfasen har formet spillet 'Clone Wars'. Derefter vil der kommes ind på gruppens samarbejde, spilleregler og hvordan opgaverne var allokeret gennem designprocessen. Til sidst vil spillet vurderes fra et spilteoretisk perspektiv.

DESIGNPROCESSEN

KONCEPT OG PRE-PRODUKTIONSFASEN

NB! Årsagen til at koncept samt pre-produktionsfasen er slået sammen er, at vi undervejs i processen gik fra pre-produktion til koncept, da den originale spilidé blev skiftet ud med en ny.

Den første brainstorm session til projektet startede den 17/8., hvor der blev lavet mindmaps og tankestrømme om hvilken type spil, som gruppen kunne lave og som kunne være interessant at spille fra et spillercentrisk perspektiv. Der blev konceptualiseret femten forskellige spilidéer, der hver især indgik i en iterativ, demokratisk proces. Tre dage efter den 20/8. holdt gruppen et møde over Discord, hvor gruppen fortsatte udvælgelsesprocessen. Under denne elimineringsrunde reflekterede gruppen over, hvad der rent faktisk kunne lade sig gøre indenfor tidsrammen. Den nye liste blev indsnævret og gruppen blev enige om at skrive fordele og ulemper for hver af de ni resterende kandidater. Processen resulterede i et overordnet koncept: en 2D endless runner, hvor der blev draget inspiration fra allerede populære spil som *Subway Surfers* (Kiloo & SYBO Games, 2012), *Alto's Odyssey* (Team Alto, 2018) og *Jetpack Joyride* (Halfbrick Studios, 2012), der alle blev spillet igennem for at konkretisere konceptet yderligere.

Illustration 1: Moodboard til idé 1

Temaet 'Diabetes' blev valgt som tema for gruppens spil, og dette ledte til titlen 'Let's Diabetes', der både skulle være tillagt stor underholdningsværdi for spilleren men samtidig også give en samfundsrelevant indsigt i forskellige problemstillinger vedrørende sygdommen. For at blive enige

om det stilistiske udtryk, blev der udarbejdet et moodboard (se illustration 1) og diskuteret spørgsmål omkring det æstetiske udtryk af spillets komposition, perspektiv og hvilken følelse spilleren skulle have af dette. Da der var opnået enighed om det visuelle udtryk og koncept, blev der efterfølgende udformet en papirprototype af spillet.

Ifølge Tracy Fullerton (2008, s. 19-20) kan det være en god idé at eksperimentere med papirprototyper som en interaktiv repræsentation af et koncept, før man kaster sig ud i programmering. Årsagen til dette er, at det kan være dyrt i form af både tid og penge at få programmeringen igangsat, hvis der ikke engang er bred enighed om spillets game mechanics. Det kan også fremhjælpe kreativ tænkning og styrke innovation. At udforme og teste en papirprototype var en udfordrende proces, da der var tale om et endeløst spil, der derfor kan være svært at simulere på papir. Vi endte dog med at klistre flere papirstykker sammen, sætte det rundt om et objekt, i vores tilfælde en computerskærm, og lade spilleren styre en karakter hvor de løfter og sænker hånden for at bevæge karakteren op og ned, mens papirstykkerne roteres rundt, så banen langsomt afsløres for spilleren (se illustration 2).

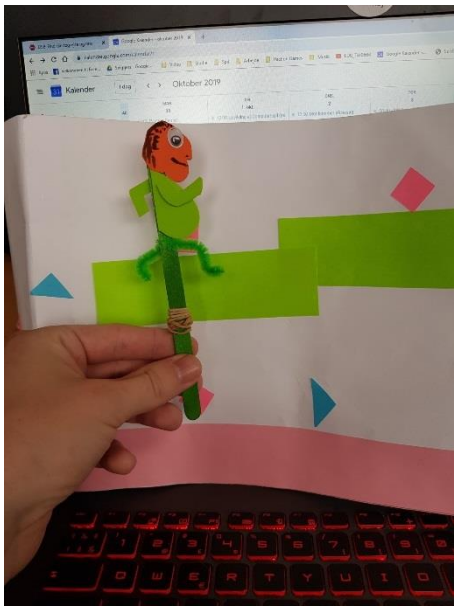

Illustration 2: Papirprototype til idé 1

Den 23/10. blev papirprototypen testet af en gruppe gymnasieelever, der udviste stor interesse for spilidéen og kom med forslag til forbedringer, der dannede grundlag for videre overvejelser. Med elevernes feedback i baghovedet nåede gruppen, i dialog med adjunkt og vejleder Lasse Juel

Larsen, frem til den konklusion, at spillet var for ensformigt, og at en indførelse af multiplayer, hvilket var et element, som gruppen ønskede at implementere i spillet, kunne skabe dynamik og store muligheder for optimering samt balancering af spillet. Dette resulterede i, at den forhenværende spilidé blev kasseret og designprocessen begyndte forfra med det endelige produkt: et lokalt 2D multiplayer platform-shooter spil ved navn 'Clone Wars', hvor en gal videnskabsmand har klonet sig selv, og nu kæmper de to identiske videnskabsmænd til døden i en hektisk duel med laserstråler for at være ene mand om identiteten.

Illustration 3: Moodboard til 'Clone Wars'

Det var vigtigt for gruppen, at den nye idé var opnåelig inden for den angivne tidsramme, og derfor gik de hurtigt til værks med at få udtrykt et koncept i form af moodboard (se illustration 3). I den efterfølgende weekend blev der afholdt en lang design sprint, hvor gruppen udarbejdede en digital, interaktiv prototype, både for at visualisere spilkonceptet, men også for at teste, om spillet ville kunne fungere i praksis. Da indsigterne fra første papirprototype allerede havde medført en afklaring omkring den nye spilidé, valgte gruppen at undlade endnu en udarbejdelse af en papirprototype. Et meget vigtigt element i spillet var en tydelig core mechanic, som er et begreb introduceret af Keith Burgun (2015, s. 37). Denne core mechanic er en sammensætning af core action: hvad spilleren gør, og core purpose: hvorfor spilleren gør det. Kigges der på spil fra et spilcentrisk perspektiv, præsenteret af Fullerton (2008, s. 2), er elementer som core mechanic og

mål meget vigtige for spillerens oplevelse. Derfor var spillets core mechanic et af gruppens fokuspunkter under første design sprint.

PRODUKTIONSFASEN

Den første prioritering med hensyn til produktionsfasen var at få skabt en spilbar digital prototype, således at der hurtigt kunne foretages første omgang af spiltests med henblik på game feel jf. (Swink, 2009). Game feel beskrives som en samlet betegnelse for den fulde oplevelse af et spils interaktivitet, samt forståelsen af spillet som en forlængelse af sanserne (Swink, 2009, s.10). Swink benytter sig i denne forbindelse også af begrebet polish til at beskrive de dele i spil, hvor interaktionen mellem spiller og spil bliver kommunikeret i form af såkaldte clues, der kan hjælpe spillere med at tolke effekten af deres handlinger (Swink, 2009, s. 5). Et begreb der også anvendes meget inden for game design er juice der, ifølge Bo Kampmann Walther & Lasse Juel Larsen, kan beskrives som værende sammenfaldende i dets betydning. "Swink's conception of polish is equivalent to what other game designers call ´juice´ [...] a composite of communicative effects that warrant maximum game output as response to minimum player input" (Walther & Larsen, 2019, s. 6). Begreberne polish og juice benyttes derfor på samme vilkår og betingelser i denne rapport. Da det første udkast til den digitale prototype som udgangspunkt ikke indeholdte polish, var der derfor stor relevans i at få feedback på særligt dette område. Fokuspunkterne for testen af den digitale prototype var ligeledes at få evalueret styringen i spillet og dermed også relationen mellem spillernes input i interaktion med spillets objekter samt spillets output, der nærmere kan bestemmes som feedback i form af lyd, kamerabevægelse og lignende, der forekommer når en spiller interagerer med et spilobjekt.

Ifølge Fullerton er iteration en vigtig del af en spilcentrisk proces (Fullerton, 2008, s. 14). Derfor skal vigtigheden af brugertests understreges, hvorfor gruppen målrettet har testet så ofte som muligt. Ved at teste ofte kunne der sikres et friskt syn på den digitale prototype gennem dens udvikling, så den hele tiden kunne rettes til undervejs.

Illustration 4: Første udkast til den digitale prototype udviklet i løbet af en dag

I forbindelse med den første omgang spiltest blev der indsamlet feedback, som gav anledning for flere opgaver til uddelegering i gruppen. For at have overblik over arbejdsfordelingen og hvad gruppens deltagere hver især arbejdede på, blev der oprettet et Google Sheet, hvor opgaverne blev skrevet ind og hvert medlem kunne skrive sig på det, som vedkommende var i gang med. MoSCoW-modellen blev taget i brug med det formål at strukturere samt effektivisere arbejdsbyrden, idet at projektets arbejdsopgaver opdeles efter relevans i kategorierne must-have, should-have, could-have og won't-have. Denne model blev grundstenen i den organisatoriske og konstruerende del, men i stedet for kategorierne should-have og could-have, blev det fusioneret til én samlet betegnelse ved navn nice-to-have. Der var dog, for hver af de to kategorier, forskel på opgavernes prioriteringsmæssige relation til hinanden. Derfor blev hver enkelt opgave påført en specifik farve, som skulle indikere prioriteringsniveauet og således også, hvor hurtigt de skulle fuldføres. Undervejs blev nogle opgaver også overført fra nice-to-have til must-have, samtidig med at der blev tilføjet nye opgaver på for dem begge. Størstedelen af must-have opgaverne har været baseret på mechanics, hvor nice-to-have overvejende har været det æstetiske i spillet. Det var blandt andet opgaver som at implementere screen shake eller lyde som feedback, når man interagerer med objekter - ting der ved udeblivelse ikke ødelægger spillet, men i den grad tilføjer ekstra polish og raffinerer det. I projektets forløb overgik gruppen til hjemmesiden Monday, som er et webbaseret struktureringsværktøj til organisering af arbejdsopgaver. Dog var dette en gratis 30-dages prøveperiode, så gruppen eksporterede data fra Monday til et Excel-sheet, så gruppen stadigvæk kunne have et overblik over arbejdsopgaverne. Der blev oprettet et board i struktureringsværktøjet Trello (se illustration 5). En stor fordel ved at benytte Trello frem for et manuelt system som

Google Sheets er, at Trello afspejler det dynamiske aspekt af processen, så hvert gruppemedlem kan følge med i de løbende ændringer og opdatere de andre.

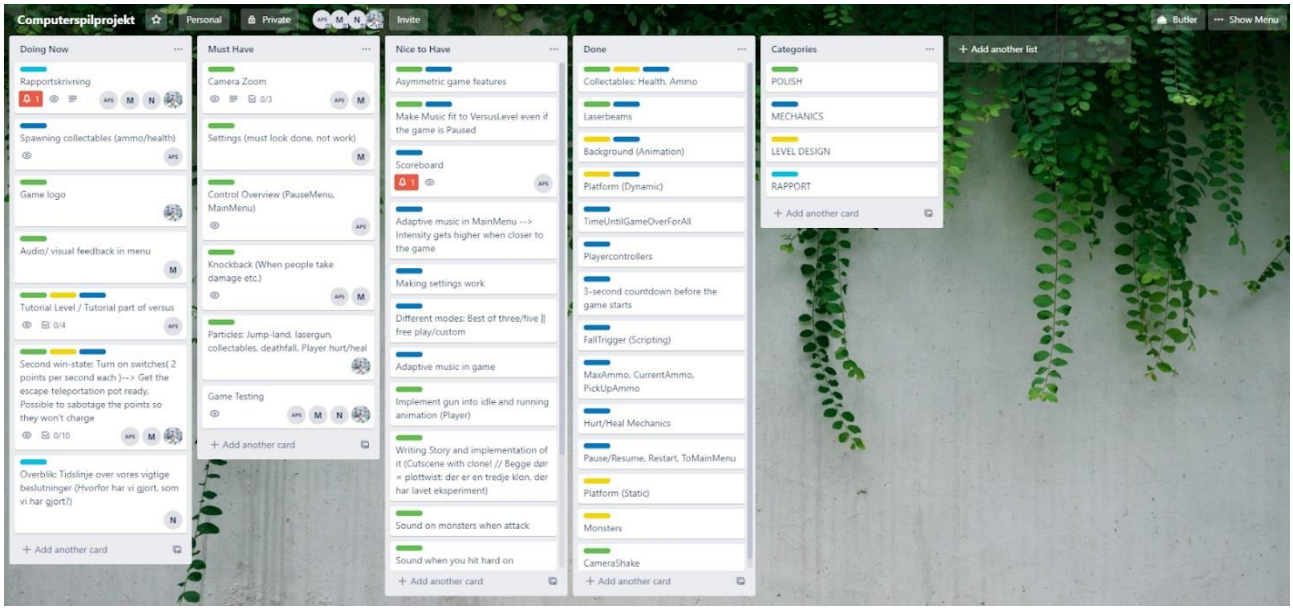

Illustration 5: Screenshot af organisering af opgaver via Trello

Enkelte assets blev hentet fra eksterne kilder såsom Assets Store, men langt de fleste assets såsom grafik og animationer er blevet lavet selv, hvor de fleste lyde er hentet men redigeret efterfølgende. Der er på sidste side refereret til alle eksterne assets.

En stor del af produktionsprocessen var playtesting, feedback og implementering af den givne feedback. I det tidlige forløb var dette for at undersøge om mekanikken i spillet var intuitiv og om spillerne kunne navigere rundt i spillet uden dybere forklaring. Efterfølgende samledes der feedback på, hvad der fungerede bedst designmæssigt. I første udgave af den digitale prototype af spillet var der eksempelvis bevægende platforme i hver side af banen, som gjorde at man kunne køre hurtigt op eller ned. Denne funktion blev dog hurtigt udnyttet som en FOO-strategi (Firstmove Order Optimal), et begreb præsenteret af Extra Credits (2012), da spillerne havde uendelige skud og brugte dette til at lave rapid-fire på sin modstander, en hurtig og nem måde at vinde spillet på. Der kunne også opstå en skakmat situation, som David Sirlin (n.d.) nævner i sin artikelserie 'Balancing Multiplayer Games, Part 2', hvor den spiller som var for langsom til at træde op på platformen, endte med ikke at kunne rykke sig, eller i hvert fald havde meget dårlig chance for at få ram på den anden. Altså kan resultatet af spillerunden allerede gennemskues, så snart den første spiller har trådt

op på platformen. Det førte til at platformene blev fjernet, også for at spillerne ikke valgte en statisk tilgang til spillet.

Gruppens playtesting gik også meget på at modarbejde ulighed i spillet. For eksempel kom ulighed til udtryk i tidlige versioner af 'Clone Wars', hvor spiller 1 var vendt mod spiller 2, mens spiller 2 havde ryggen til spiller 1. Dermed havde spiller 1 en fordel ved, at de kunne skyde modstanderen med det samme. Selvom ulighed også kan være godt i spil i form af for eksempel asymmetri, som nævnt af David Sirlin (n.d.) i sin artikel 'Balancing Multiplayer Games, Part 1: Definitions', hvor det kan føre til et mere dynamisk spil ved at give to spillere forskellige valg i starten. Men eftersom det hverken var et valg, eller har nogen fordel at starte med ryggen til sin modstander, var det vigtigere at spillet havde lighed i starten, så det blev implementeret at begge spillere stod rettet mod hinanden. Senere hen blev der tilføjet forhindringer, så man ikke blot står og skyder langt fra hinanden. For at indføre et mere strategisk brug af ammunitionen blev mængden af lasere pr. magasin nedsat til fem. I senere testing sessioner mente deltagerne, at fem skud var for mange til at starte med, og at tre ville passe bedre. De mente også, at det var fedt, at det var tilfældigt, hvorvidt det var health eller ammo, der spawnede ved siden af dem, da de kunne lægge strategier baseret på netop dette.

Til at starte med havde en af gruppens medlemmer designet DNA-strenge, som skulle repræsentere de objekter, som spilleren kunne indsamle for at få ekstra liv. Spillets health-UI havde kolber med grønne kemikalier. Gennem en playtesting session fik gruppen at vide, at de to konceptuelt ikke var forbundet til hinanden, og at det kunne være svært at opfange betydningen af DNA-strenge. Til trods for at DNA-strengene passede ind i spillets opsatte game world, så besluttede gruppen sig for at skifte dem ud med et objekt, som gav mere mening konceptuelt: hjerter. Hjerterne blev både implementeret i spillernes UI samt de health-pickups, som der tilfældigt spawner rundt omkring på platformene. Denne playtesting session fik også gruppen til at indse, at UI i toppen var for småt. Nogle af spillerne lod slet ikke mærke til, at man kunne holde øje med antal liv og den resterende tid for runden. Derfor besluttede gruppen at forstørre disse.

Adskillige spillere havde pointeret, at der muligvis var en mangel på replayability. Dette fik gruppen til at reflektere og brainstorme på, hvordan man kunne få løst dette. Gruppen endte med to nye elementer, som skulle spille centrale roller i spillet: second win-state og et score system.

Gruppen brainstormede på, hvordan dette second win-state kunne udformes, da medlemmerne ønskede, at det skulle være forbundet til historien, som blev fortalt i spillets begyndelse. Gruppen endte med et second win-state, hvor 'quantum-chargers' aktiverer en teleporter, når der kun er 10 sekunder tilbage. Dette gør det muligt for spillerne at lægge yderligere strategier, da de nu kan vælge mellem at forholde sig passivt og vente på, at teleporteren spawnede, eller at indtage en aktiv rolle og prøve at eliminere den anden spiller inden. Hvis Player 1 (P1) kun havde 1 liv tilbage og dens modstander P2 havde 5, så ville P1 formodentlig søge dækning eller gå på jagt efter liv. Dette kan være et irritationsmoment for P2, da denne naturligvis vil vinde spillet og ikke ønske, at P1 får en ekstra chance. Dette kan få P2 til at lægge en strategi for det videre forløb, som efter alt at dømme er centreret i det mål om at sørge for, at P1 ikke vinder runden. Hvis ingen af spillerne når at skyde hinanden eller slippe ud gennem teleporteren, så eksploderer laboratoriet, og begge spillere dør. Dette leder videre til det andet element, som potentielt kunne løse manglen på replayability: score system. For hver runde spillerne gennemfører giver et point til den spiller, som endte med at eliminere den anden. Hvis ingen af spillerne slipper ud eller eliminerer hinanden, så dør de begge, og der gives ingen point. Gruppen testede dette på en række testpersoner, og de var alle fan af de nye elementer. De mente dog, at 'quantum-chargers' blev en distraktion for målene i spillet, da de stod på banen uden noget formål end pynt og simulering af en bagvedliggende proces af klargøring af teleporteren. Gruppen valgte hermed at fjerne disse.

I forbindelse med det tekniske programmeringsarbejde har gruppen erfaret, at en lille ændring i koden kan resultere i en dominoeffekt der medfører, at uventede dele af spillet pludseligt ikke virker optimalt. Så når en brugertest ender med, at folk ´snyder´ i spillet ved f.eks. at dobbelthoppe i luften, fordi de udforsker spillets rammer, fremstår vigtigheden af brugertests særligt tydelig. Noget der frustrerede spillere allermest, var hvis de løb ind i sine egne skud og tog skade, fordi det ikke stemmer overens med den generelle forståelse af, hvordan projektiler virker. Fornemmelsen for game feel er i dette tilfælde ikke helt tydelig, da spillerne ikke er klare over årsagen for deres død, før de blev gjort opmærksom på det. Blandt andre bugs i spillet var der blandt andet også, at spillere kunne skyde, selvom de var døde. At bugs opstår er naturligt i computerspil, da en masse foruddefinerede elementer og scripts skal kunne arbejde sammen. Da gruppens projekt endte ud i tusindvis af elementer, der skulle arbejde sammen på kryds og tværs, heriblandt 2.550 linjer kode og tusindvis af spil objekter, så ville det være ualmindeligt og bizart, hvis fejl ikke hændte. Alle disse er dog blevet løst.

Efter en længere proces af programmerings- og designarbejde samt tests af de implementerede elementer vurderede gruppen, at polish og juice var det næste på dagsordenen. Der blev brainstormet på lyde, particle effects, flottere grafik og UI samt raffinering af animationer. Gruppen diskuterede mulighederne og udformede en liste over disse, som skulle optræde i diverse scenarier. Nedenstående tabel er en oversigt over de mange forskellige scenarier i spillet, og hvordan juice elementer såsom animation, visuelle effekter, lydeffekter, cinematiske effekter og ændringer i UI kommer til udtryk:

EVENT	**VISUELLE EFFEKTER**	**LYDEFFEKTER**	**CINEMATISKE EFFEKTER**	**User-Interface**
Spiller bliver ramt	Blodpartikler udspringer fra den skadede spillers krop	Der er en lydeffekt hver gang, spilleren bliver skadet Hvis ramt af monster, så afspilles en moster-attack lydeffekt ligeledes	Kameraet ryster (Screen shake)	Et hjerte forsvinder fra UI
Når spillere bevæger sig	Animation for løb sættes i gang		Kameraet er afhængig af spillernes positioner og zoomer ind og ud baseret på beregninger i et script.	
Når spiller samler health og ammo op	Pickup forsvinder ved kollision	Forskellige lydeffekter afhængig af om det er ammo eller health		Spillerens UI for health eller ammo opdateres

Spiller skyder	En laserstråle bliver initialiseret	Lyd af laserstråle		Ændring i UI ammo. Mister 1 per skud.
Når tid rammer 0:10	Teleporteren begynder at animere	Ved 0:15 afspilles en lydeffekt for teleporteren der er i gang med at blive aktiveret		
Når tid går ud	Eksplosion og score screen loades bagefter	Lydeffekt for eksplosion afspilles	Screen shake	UI forsvinder
Hvis en spiller vinder, og den anden taber	Teleporteren forsvinder, hvis en spiller slipper ud via denne Hvis en spiller bliver dræbt, fjernes spilleren. Uanset hvad loades score screen	Hvis man slipper ud via portalen, så afspilles en lyd	Screen shake	Hvis en spiller går ind i teleporteren, så fjernes UI, da runden slutter Hvis en spiller bliver dræbt, så fjernes UI Et point bliver tillagt den spiller, som vinder runden

Illustration 6: Screenshot af gameplay i dets færdige state

UI blev ombygget og moderniseret (se illustration 6). Dette gjorde det både nemmere for spillerne at se den aktuelle score, hvor meget tid der var tilbage og antal resterende liv samt skud. Spillerne blev også animeret forfra, hvor laser-pistolen blev fastgjort til hvert sprite, så videnskabsmændene altid løb rundt med pistolen, og at den ikke kun var visuel, når man skød.

Selve gameplay var naturligvis ikke det eneste, som spillet i sin helhed bestod af. En æstetisk tilfredsstillende hovedmenu, pause menu, scoremenu samt 'How to play' sektioner blev implementeret. Hovedmenuen blev udformet med formålet at skabe en laboratorie-setting allerede fra spillets begyndelse, så spilleren blev, så at sige, kastet ind i spillets rammer fra start til slut. Den panorerende effekt af kameraet var en cool feature, som ikke kunne modstås og har været anledning til positiv respons fra deltagere.

Det var essentielt, at en 'How to play' sektion blev integreret både fra hovedmenuen af og ved spillets begyndelse samt ved pause menu, så man altid kunne tilgå controls og objectives. Pause menu blev lavet for at tilgå denne 'How to play' sektion og for at give spillerne muligheden for at sætte spillet på pause, hvis dette ønskedes uden at forlade spillet med den risiko for at score count blev nulstillet. Det skulle ligeledes være muligt at gå tilbage til main menu fra spillet af, og dette var endnu et argument for, at pause menu blev lavet. Adskillige score screens blev også lavet, så spillerne efter en runde kunne opleve den reward, der ligger i at vinde. Hvilken score screen der

vises er afhængigt af udfaldet på runden. Desuden gav score screen også spillerne et overblik over den samlede score og gav dem mulighed for at holde en pause, da den næste runde først går i gang, når man trykker 'Restart'.

En cutscene blev også lavet, hvor voice-over samt tekst forklarer situationen mellem de to videnskabsmænd. Denne cutscene blev lavet for at integrere storytelling elementer, da dette muligvis kunne få spilleren til allerede at leve sig ind i spil situationen, før spillet overhovedet gik i gang.

GRUPPENS SPILLEREGLER OG MEDLEMMERNES OPGAVER

I gruppen blev der fra første møde foretaget en forventningsafstemning, hvor der blev drøftet hvad det individuelle gruppemedlems forventning til projektarbejdet og gruppens spilleregler var. Gruppen bestod af to 1. semesterstuderende og to 5. semesterstuderende, så derfor var både kompetencer og erfaring divergerende. Da to gruppemedlemmer i samme periode stod med et bachelorprojekt over hovedet, og da den tredje mand var familiefar og boede på Sjælland, blev der lavet en aftale om hvor meget tid, man hver især forventer at kunne afsætte til projektet.

Aftale om varighed samt hyppighed af møder blev inspireret af SCRUM, en agil projektorganiserede metode. Ifølge Clinton Keith benytter metodologien sig af såkaldte *sprints*, som er "time-boxed iterations of development [...] that create working versions of a game, which allow the customers to see how it is evolving, better understand where the game is going, and have meaningful conversations with the development team" (Keith, 2007, s. 22). De såkaldte *Daily Scrums* konverterede gruppen til *Weekly Scrums*. Gruppen mødtes én gang om ugen fysisk i minimum 2 timer. Derudover holdt gruppen ugentlige møder over Discord. Formålet med disse møder var at holde alle ajour med designprocessen samt status quo og at kunne dele problemstillinger, succesoplevelser og andet. Derudover har gruppen kommunikeret dagligt over Messenger og ved at skrive kommentarer til de forskellige arbejdsopgaver i Monday og Trello. Gruppen har tillige engageret sig socialt med hinanden for at skabe kemi.

Under udviklingen af den digitale prototype blev GitKraken anvendt til at opbevare projektet i skyen via versionsstyringsværktøjet Git gennem GitHub. For gruppen var det essentielt, at spillet

altid kunne spilles, og at det rent teknisk fungerede, før der blev brugt en masse tid på polishing. Dermed blev polish og juice gemt til sidst i processen. Testing er essentielt i en designproces, da det typisk er disse som afgør kvaliteten af spillet i sidste ende. Processen har været kendetegnet som iterativ, da vi især i produktionsfasen har lavet ændringer, testet, behandlet data for disse tests for dermed at ændre eller implementere nye elementer i spillet.

Gruppen blev enige om, at det var vigtigt, at alle holdt sig ajour med processens forløb, og at hvert gruppemedlem bidrog til hver del af processen som programmering, grafik, lyd, etc.. Baseret på erfaring og kompetencer udpegede gruppen Lead-roller. Disse ville have ansvaret for området, men opgaverne indenfor hvert felt blev fordelt mellem alle. I løbet af semestret begyndte Nadia at være i syv sind med valg af uddannelse og senere besluttede hun at afbryde sit studie. Dermed bestod gruppen praktisk talt af 3 medlemmer. Opgaverne og ansvarsområderne blev fordelt som følgende:

Alexanders Projektandel og hans stykke kode

Jeg indtog rollen som Lead Programmer og har primært stået for programmering og implementering af spilobjekter i Unity. Jeg har dermed også stået for at holde overblik over, hvad der skulle programmeres for at kunne udforme spillet, som vi i sidste ende ønskede, at det skulle se ud. Jeg gav de andre gruppemedlemmer det selvsamme overblik ved at oprette arbejdsopgaver i Trello. Spillet, dets funktionalitet og polish af elementer har været mit hovedfokus. Jeg har dog også lavet intro-cutscene og selv optaget lyde til denne.

```
void OnTriggerStay2D(Collider2D other)
{
    if (other.tag == "Ammo")
    {
        if (currentAmmo != maxAmmo)
        {
            currentAmmo = maxAmmo;
            Debug.Log("Ammo!");
            ammoSound.Play();
            Destroy(other.gameObject);
            spawnAmmo = true;
            P1Ammo();
        }
    }

    if (other.tag == "Health")
    {
        findGameManager = FindObjectOfType<GameManager>();
        if (findGameManager.P1Life != 5)
        {
            findGameManager.HealP1();
            Debug.Log("Health!");
            Destroy(other.gameObject);
            //Instantiate(HealthEffect, transform.position, transform.rotation);
            spawnHealth = true;
        }
    }
}
```

Illustration 7: Logikken bag ammo samt health pickups

Jeg har valgt at kommentere på følgende stykke kode (se illustration 7), da det godt og grundigt afspejler min fascination af kodesprog generelt. Kodestykket er inddelt i to forskellige conditional statements (if-sætninger). Disse er essentielle for spillet, da de trækker i en masse kode fra andre scripts for til sidst at kunne visualisere logikken bag ammo samt health-pickups. I den anden if-sætning vedrørende health-pickup eksempelvis kan vi se, at vi igennem et andet script finder spillerens resterende antal liv. Hvis spilleren har under 5 liv, så skal det health pick-up, som spilleren berører, forsvinde og give spilleren et ekstra liv. Jeg synes, at fremgangsmåden på hvordan man kæder logik sammen i programmeringssprog er fascinerende, og disse linjer af kode er et meget godt eksempel på netop dette.

Frederiks Projektandel og hans stykke kode

Jeg indtog rollen som Lead Sound Designer og har i den forbindelse stået for det meste af sound design. Dette indebærer blandt andet, at jeg har stået for at holde overblik over hvilke lyde, som vi skulle have implementeret i spillet. Et af mine hovedfokuspunkter har været spillets funktionalitet, så jeg har blandt andet stået for at programmere forskellige events samt fixet diverse bugs, når de på uforklarligvis opstod. Jeg har dog også lavet en masse design-relateret arbejde, blandt andet main menu, pause menu og win-states. Jeg har valgt at kommentere på følgende stykke kode (se illustration 8), da det illustrerer meget godt, hvordan eventbaseret kode fungerer. Når spiller 1 har opnået en sejr på den ene eller den anden måde skal der, såfremt pointet ikke er blevet tildelt en spiller endnu, gives et point til spiller 1. Dette opnås ved at lægge et point til spillerens nuværende score for derefter at opdatere det visuelle display til en score screen, der er afhængig af hvem der vinder. Så i dette tilfælde vil der selvfølgelig stå, at spiller 1 har vundet.

```
public void p1Win()
{

    if (PointGiven == false)
    {
        P1ScoreValue++;
        P1Score.text = ""+P1ScoreValue;
        BigP1Score.text = P1Score.text;
        PointGiven = true;
        whoWon.text = "Player 1 eliminated Player 2 and managed to escape the laboratory before it exploded!";
        P1WonVisual.SetActive(true);

    }
```

Illustration 8: Logikken bag pointsystemet

Nadias Projektandel og hendes stykke kode

Jeg er førsteårsstuderende på Lærings- og oplevelsesteknologi, og har derfor ingen tidligere kompetencer haft indenfor programmering eller spiludvikling. Derfor var det i starten programmeringen, der var en vigtig del for mig at forstå og lære mere om, for at kunne hjælpe ordentlig til i projektet. Dog tog jeg en beslutning i midten af 1. semester om, at studiet ikke var det jeg ville og derfor overgav jeg min programmeringsdel til Alexander, Frederik og Nickolai. Derfra har det derfor været vigtigere for mig at forstå fremgangsmåden i et kollaborationsprojekt, samt få øvelse i at skrive en faglig relevant rapport, da dette er universelle byggesten på en universitetsuddannelse, som jeg kan bruge til mit fremtidige studie. Dog fik jeg programmeret blandt andet et script til platforme, der skulle bevæge sig fra position A til position B. Det var en stor ting, da det var første gang jeg skrev et stykke kode helt på egen hånd - med hjælp fra tutorials - og en kæmpe succesoplevelse, da det endelig virkede. Samtidig var det fedt at opleve at man med kode kunne give en ellers livløs ting bevægelse. Samtidig viste det også, at en kode tiltænkt til ét formål, kan vise sig at være brugbart til andre formål. På grund af designmæssige grunde, endte spillet ikke med at inkludere bevægende platforme, men scriptet kunne stadig bruges til at få collectibles til at bevæge sig op og ned for at drage spillerens opmærksomhed hen på det givne objekt.

```
1 reference
private void Move()
{
    childTransform.localPosition = Vector3.MoveTowards(childTransform.localPosition, nextPos, speed * Time.deltaTime);

    if (Vector3.Distance(childTransform.localPosition, nextPos) <= 0.1)
    {
        ChangeDestination();
    }
}

1 reference
private void ChangeDestination()
{
    nextPos = nextPos != posA ? posA : posB;
}
```

Illustration 9: Screenshot af kode til bevægende platforme

Nickolais Projektandel og hans stykke kode

Jeg var Lead Graphics Designer og mit primære opgaveområde var at designe de forskellige spilelementer, animere og implementere. Jeg er førsteårsstuderende på Lærings- og oplevelsesteknologi, og med det, havde jeg meget, der skulle læres. I takt med undervisningen skred fremad og teorien satte sine faste spor, begyndte jeg at danne mig et billede af mine kompetencer og

hvad jeg kunne bidrage med til gruppen. Jeg har for det meste tegnet og animeret, men jeg har også programmeret. Jeg fik også gennem processen dannet mig et overblik over kodes struktur og funktionalitet ved at kigge på de andres scripts.

Monstrene i spillet blev mit personlige gennembrud, for hvad der startede som et lille sideprojekt fik hurtigt en central del i spillet. Efter en forelæsning i spilprogrammering lærte vi om såkaldte 'raycasts', hvilke gjorde det oplagt at benytte i mit sideprojekt. Af indlysende grunde skulle monsteret kunne bevæge sig og gerne så det gav mening for spiloplevelsen.

```
0 references
void Update()
{
    transform.Translate(Vector2.right * speed * Time.deltaTime);

    RaycastHit2D groundInfo = Physics2D.Raycast(groundDetection.position, Vector2.down, distance);

    if (portalDetection != null)
    { monsterInfo = Physics2D.Raycast(portalDetection.position, Vector2.down, distance);
    }

    if (groundInfo.collider == false)
    {
        if(movingRight == true)
        {
            transform.eulerAngles = new Vector3(0, -180, 0);
            movingRight = false;
        }
        else
        {
            transform.eulerAngles = new Vector3(0, 0, 0);
            movingRight = true;
        }
    }
    if (monsterInfo.collider == true)
    {
        if (monsterInfo.collider.tag == "Portal")
        {
            if (movingRight == true)
            {
                transform.eulerAngles = new Vector3(0, -180, 0);
                movingRight = false;
            }
            else
            {
                transform.eulerAngles = new Vector3(0, 0, 0);
                movingRight = true;
            }
        }
```

Illustration 10: Screenshot af MonsterScript

Første del af koden begynder med at få objektet til at bevæge sig, hvilket fungerede efter hensigten. Problemstillingen opstod, da jeg gerne ville have den til at opdage, at en platform ophørte og skulle stoppe. Svaret fandt jeg i brugen af raycasts. Inde i spillet oprettede jeg et tomt objekt foran monstret, og denne skulle fungere som en detektor og få programmet til at reagere når platformen ophørte.

Først skulle monstret bevæge sig med en konstant hastighed fra venstre mod højre. Derefter skulle en "Ground Detection" fortælle, hvornår en given platform ophører. Hvis den ikke er ophørt, så skal den fortsætte ad samme vej. Hvis den ophører, så skal den dreje 180 grader og fortsætte fra højre til venstre. Dette program fortsætter.

En mindre udfordring opstod, da teleporteren blev implementeret, da dette program ikke tog højde for denne. Derfor blev der i bunden af scriptet tilføjet en funktion der fortæller, at når collideren støder på et objekt med tagnavnet "teleporter" skal programmet opfatte det som værende enden af platformen og dermed og vende om. Et lille, men ganske betydningsfuldt script for mig, da dette var mit første møde med den fulde oplevelse af at selv skabe noget helt fra bunden og se det blive til et lille animeret selvkørende program.

VURDERING AF SPILLET FRA ET SPILTEORETISK SYNSPUNKT

I. Core Mechanics

Spillet er et 2D, lokalt multiplayer platform-shooter spil. To spillere bevæger sig fra platform til platform, armeret med laser-pistoler for enten at nedkæmpe hinanden og/eller holde fjenden i skak og undslippe gennem en teleporter, inden tiden er gået.

Keith Burgun definerer core mechanics ud fra to perspektiver: "Core actions: the things that you actually do in the system […] Core purpose: the reason that you do the core actions" (Burgun, 2015, s. 37). I 'Clone Wars' er core action at bevæge sig og skyde, imens core purpose er 'undgå at blive skudt og forårsag skade på din opponent'. Spillets core mechanic er hermed, at man som spiller efter alt at dømme vil forsøge at være bedre stillet end sin opponent ved at undgå at tage skade ved at undvige i form af bevægelse og skyde for at skade sin opponent. Med core mechanics opnås det absolutte mål for (næsten) alle spil - at vinde. I dette spil vinder man enten ved at eliminere den anden spiller, og dette hænder, når denne spiller har mindre end 1 liv tilbage, eller ved at træde ind i teleporteren, når den spawner.

II. Level design

Med brug af Smith, Cha & Whitehead (2008, s. 76) model for analyse af 2D Platformer levels, så er 'Clone Wars' opbygget af "platforms". Disses strukturer er skabt af en bred bund for neden og for hvert opadgående trin, tilspidses det, der derved danner en pyramide. Fire af platformene indeholder

derudover en mulighed for at opsamle ammunition og liv, i.e. “collectable items”. Disse platforme er sat i hver af de fire retninger og ud til kanten. Dette er med øje for at modarbejde FOO-strategier, der vil tvinge den enkelte spillere til at skifte mellem defensivt og offensivt spil. Spilleren er fra starten udstyret med tre skud og skal i den forbindelse kort efter spillets start gøre sig selv sårbar ved at bevæge sig rundt på banens platforme for at indsamle skud. Den anden spiller kan dog vælge at stille sig i samme situation.

Spillet kan vindes på to måder:
I midten spawner en teleporter, der først aktiveres, når der er 10 sekunder tilbage af runden. Teleporteren bevogtes af monstre og her skal spilleren igen tage et strategisk valg, inden tiden render ud. Enten skal spilleren aktivt gå efter at skyde modstanderen eller gøre sig sårbar og bevæge sig mod midten, undgå monstrene og flygte via teleporteren. Det kan dog være svært, da den anden spiller har det samme mål: at vinde. Når den ene spiller er gået igennem teleporteren, så deaktiveres den, og den anden spiller dør hermed pga. den efterfulgte eksplosion.

Spillet er symmetrisk og afbalanceret. Banen er statisk og symmetrisk på y-aksen. Begge spillere starter med samme vilkår: begge har tre skud og starter på neutrale platforme lige langt fra ammunition, liv og monstre. “A multiplayer game is balanced if a reasonably large number of options available to the player are viable--especially, but not limited to, during high-level play by expert players.” (Sirlin, n.d.). Dog kan det tilføre spillet et strategisk element, hvis pickups i siderne er tilfældige. Hvis et hjerte spawner i venstre side i starten af spillet, og et ammo pickup spawner i højre side, så har P2 en fordel, da denne har op til 5 ekstra skud til rådighed, hvor P1 kun har 3. Dog sætter P2 sig i en sårbar situation, hvis den aktivt forsøger at dræbe P1, da monstre kan beskadige ham på vejen.

En god måde at motivere spillerne til at forsætte spillet er, at interaktion udløser belønninger, som John Hopson nævner i sin artikel ‘Behavioral Game Design’ (2001). Da spillet er en meget kort, men intens oplevelse med kun halvandet minuts spilletid, så kan der ikke gives langvarige belønninger, men i stedet belønnes spilleren for eksempel ved, at modstanderen mister et hjerte, når pågældende bliver skudt. Der bliver samtidig spawnet både ny ammunition og ekstra liv, som også er en belønning til spillerne, såfremt de tager en chance og mister liv på det eller er meget aktive og bruge alt deres ammunition. Dog bliver disse belønninger generet tilfældigt, så der er en vis

usikkerhed i, at spillerne ikke altid kan få lige dét, som de har brug for. Grundet en variabel ventetid for et spawn kan usikkerheden sætte spilleren i et dilemma, om hvorvidt de skal blive eller fortsætte (Hopson, 2001). Samtidig er der en score bar, der tæller hvor mange runder spillerne har vundet og dette giver anledning til at spille igen.

III. Game feel

I kapitel 11 af Steve Swinks bog (2009, s. 179-183) redegør han for, hvordan regler i et spil arbejder sammen (rules metrics). Her inddeler han dem i tre kategorier: high-, mid- og low-level rules. High-level rules kan siges at være usynlige. Spildesigneren fortæller ikke direkte med skilte eller tekst hvad spillerens mål med en bestemt aktion er, men kommunikerer det i stedet gennem spilobjekter og polish. "For example, collecting 100 coins in Super Mario 64 gives you a star, which in turn unlocks various star doors in the castle." (Swink, 2009, s. 180). I 'Clone Wars' er der en tydelig high-level rule, som går på, at når der skydes på modstanderen, mister den ramte et hjerte/liv. Dette motiverer spillerne til at fortsætte, indtil modstanderen har nul hjerter som udløser win-state for spilleren, der stadig har hjerter tilbage. Dog er der endnu en high level-rule, som først udløses, når der er gået 80 sekunder af spillet og ingen af spillerne er døde. En teleporter aktiveres, der giver spillerne mulighed for et second-win state, hvor den ene af de to spillere kan tage teleporteren væk, før timeren rammer nul og laboratoriet eksploderer. Dette er forklaret til spillerne gennem historien og 'How to play' sektionerne.

Mid-level rules eller medium-level rules forklares af Swink som at være regler, der midlertidigt ændrer det overordnede mål som high-level rules har sat. Det fremgår i spillet i form af collectibles som ekstra liv og ammunition til laser-pistolen. Disse objekter roterer og svæver for at drage spillerens opmærksomhed og informere om, at der er noget interaktivt. Hvis spilleren har brugt sine tre skud og dermed er løbet tør, bliver deres primære mål at finde ny ammunition, da de ellers ikke kan gøre skade på modstanderen og således ikke opfylder den pålydende high-level rule. Samme er gældende for ekstra liv. Hvis spilleren kun har ét liv tilbage, vil de muligvis have en større tendens til at opsøge flere liv, således at det bliver sværere for modstanderen at eliminere en. Alt dette bindes sammen og udgør en del af det større billede: game feel.

Game feel er ifølge Swink defineret som "Real-time control of virtual objects in a simulated space, with interactions emphasized by polish." (2009 s. 32). Med det beskriver han, hvad et spil er, eller

hvad følelsen af et spil skal være. Men for at skabe en 'god' game feel skal en spildesigner få de tre elementer: real-time control, spatial simulation og polish til at samarbejde som en treenighed (se illustration 11). Hvis der for eksempel kun eksisterer real-time control og polish, bliver resultatet en udelukkende æstetisk oplevelse, hvor spillerens handlinger ikke har nogen betydning.

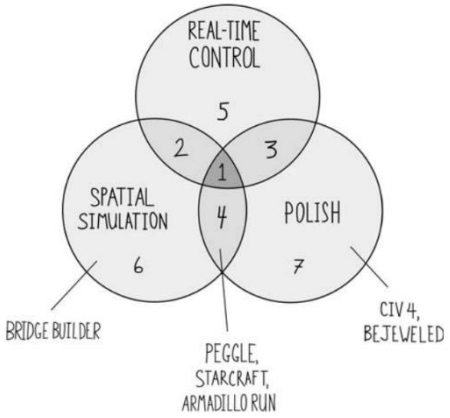

Illustration 11: Swinks treenighed (2009 s. 8)

For at opnå game feel, skal 'Clone Wars' altså bestå af alle tre dele. Real-time control og spatial simulation var det første der kom til grund i den digitale prototype, da der ellers ikke havde været noget at vise. Prototypen fik også hurtigt polish-effekter, så man kunne se, at sin modstander eller en selv tog skade. Foruden dette havde der ikke været nogen grund til at lave playtesting, da testpersoner blot ville løbe rundt uden mål eller motivation. Med endnu mere polish, som lyde på skud, ammunition og ekstra liv der bevægede sig og et kamera der ryster, når en af spillerne bliver ramt, formår spillet at vække følelser hos spilleren - irritation, glæde, motivation og meget andet, samtidig med at det føles mere inviterende og levende.

Hvilke elementer kunne tillægges spillet?

Hvis der havde været mere tid til udvikling af spillet, så kunne man indarbejde en række elementer, som muligvis vil styrke den affordance om en god spiloplevelse, som spillet besidder. 1) Man kunne implementere muligheden for visuel brugertilpasning, hvor spilleren selv kan tilpasse sin karakter visuelt og give den kosmetiske våbenændringer, hårfarve, etc.. Gruppen forsøgte at implementere knockback, når spilleren blev skudt. Det endte dog med ikke at være et tilfredsstillende resultat. 2) Når man bliver ramt, så kommer der feedback i form af lyd og ændring af UI. I stedet for knockback designede gruppen en blodeffekt, som bliver initialiseret, når denne

spiller bliver ramt. Intentionen bag dette var at give en følelse af, at man blev skudt. Et funktionelt og polished knockback ville dog have været foretrukket. 3) Adaptivt musik kunne implementeres, så intensiteten i musikken blev højere, jo tættere spillerne var på at opnå målet. 4) Man kunne komponere en masse forskellige lyde, som kan afspilles, når spilleren rammer en platform. Hvilken lyd der skal afspilles kunne ved hjælp af et script være afhængigt af, hvor 'hårdt' spilleren rammer platformen. 5) Man kunne implementere en række modes. Eksempelvis kunne man have modes som 'Best of three/five/custom'. Dette kunne tilføre spillet et stresselement. Man kunne også have et mode med noget asymmetri over sig. En af karaktererne kunne være et monster og have en fuldstændig divergerende rolle og evne modsat videnskabsmanden. 6) Man kunne også implementere network features og/eller AI, så folk fra to forskellige netværk kan spille mod hinanden eller spille mod en computerstyret karakter.

Konklusion

Ved hjælp af Tracy Fullertons 7-trins procesmodel, har gruppen formået at producere eksamensproduktet i form af 'Clone Wars'. Spildesign processen har dog ikke været en lige vej fra position A til B. Der opstår konstant nye problemstillinger og nye idéer formes undervejs i processen. Et produkt som fra et spilcentrisk perspektiv virker fornuftigt kan have sine udfordringer, når man ser det fra et spillercentrisk perspektiv. Derfor har det været en vigtig del i skabelsen af 'Clone Wars' at have en struktur med playtesting, feedback, implementering, iteration. Selvom resultatet har været tilfredsstillende, efterlades der stadig mange gode elementer uimplementeret, som det fremgår af ovenstående afsnit. Men på grund af store uforudsete ændringer i, blandt andet, spillets genre, core mechanics og historie, har tid været en udfordring. Med en solid spilteori indenfor core mechanics, game feel og polish i bagagen, har gruppen formået at udvikle et surrealistisk 2D, multiplayer spil der føles levende, og med en historie, som spillerne kan indleve sig i. Der bydes op til, at spilleren gentager spillet flere gange ved, at der er to win-states, og at det er et hurtigt, underholdende versus spil, hvor spillerne gennem rewards motiveres til at finde ud af, hvem der er bedst til at svinge sin laser-pistol i et hasarderet laboratorie.

LITTERATURLISTE

- Burgun, K. (2015). *Clockwork Game Design* (1st ed.) New York: Routledge. https://doi.org/10.4324/9781315756516
- Extra Credits. (2012, December 20). *Balancing for Skill - The Link from Optimal Power to Strategy - Extra Credits* [Video file]. Retrieved from https://www.youtube.com/watch?v=EitZRLt2G3w
- Fullerton, T. (2008). *Game Design Workshop: A Playcentric Approach to Creating Innovative Games* (2nd ed.) Elsevier Morgan Kaufmann
- Halfbrick Studios. (2012). Jetpack Joyride [Mobile video game]. Brisbane, Australia: Halfbrick Studios.
- Hopson, J. (2001, April 27). *Behavioral Game Design*. Retrieved from https://www.gamasutra.com/view/feature/131494/behavioral_game_design.php
- Keith, C. (2007, February). Scrum Rising. *Game Developer*, 14(2), 21-26. Retrieved from http://www.clintonkeith.com/resources/ScrumRising.pdf
- Kiloo & SYBO Games. (2012). Subway Surfers [Mobile video game]. Aarhus, Denmark: Kiloo Games.
- Sirlin, D. (n.d.). *Balancing Multiplayer Games*, Part 1-2. Retrieved from http://www.sirlin.net/articles/balancing-multiplayer-games-part-1-definitions http://www.sirlin.net/articles/balancing-multiplayer-games-part-2-viable-options
- Smith, G., Cha, M. & Whitehead, J. (2008). A framework for analysis of 2D platformer levels. *Proceedings of the 2008 ACM SIGGRAPH symposium on video games*, 75-80. New York, NY, USA: ACM
- Swink, S. (2009). *Game Feel: A Game Designer's Guide to Virtual Sensation* (1st ed.) Elsevier Morgan Kaufmann
- Team Alto. (2018). Alto's Odyssey [Mobile video game]. Toronto, Canada: Snowman.
- Walther, B. K., & Larsen, L. J. (2019). Bicycle kicks and camp sites: towards a phenomenological theory of game feel with special attention towards 'rhythm'. *Convergence: The International Journal of Research into New Media Technologies*, 1-21. https://doi.org/10.1177%2F1354856519885033

LISTE OVER EKSTERNE ASSETS BRUGT I SPIL

Visual

https://www.gamedeveloperstudio.com/graphics/viewgraphic.php?item=1q592f4l6a4d0m609p

https://assetstore.unity.com/packages/2d/textures-materials/crazy-lab-escape-gui-game-74052

https://assetstore.unity.com/packages/2d/environments/robot-shooting-game-sprite-free-93902

https://pixelation.org/index.php?topic=18875.0

https://waneella.tumblr.com/post/158869673302/more-backgrounds-for-pocket-rumble

Sound

https://assetstore.unity.com/packages/audio/sound-fx/8-bit-style-sound-effects-68228

https://assetstore.unity.com/packages/audio/sound-fx/sound-fx-retro-pack-121743

https://freesound.org/people/PacmanGamer/sounds/165660/?fbclid=IwAR2BmCb3pkjlOuOTPrIyO4vn2nvWTys2EMaAzVjZfkWsN3yaNp3bQQsq8Oc

https://freesound.org/people/Leszek_Szary/sounds/172207/

https://freesound.org/people/Alexmattucci/sounds/403023/

https://freesound.org/people/Cyberkineticfilms/sounds/135435/

https://opengameart.org/content/monster-sound-effects-2

https://assetstore.unity.com/packages/audio/music/rock/chiptune-rock-pack-1-57965

https://assetstore.unity.com/packages/audio/music/electronic/8-bit-chiptrance-vol-1-free-85231

Fonts

https://www.1001freefonts.com/barcade.font?fbclid=IwAR18uND1PxULK2tnmwQzpt79UySsPM4EuEQ5RE3dFlti5_oCZQrCQLfgdKw