

Miles Herman T. Flikke

Utforskning av foleybasert lyddesign for dataspill

Proessen av å lage og implementere et foleybasert lyddesign for dataspill med FMOD og Unity

Bacheloroppgave i Bachelor i Musikkteknologi

Veileder: Trond Engum

Mai 2024



Miles Herman T. Flikke

Utforskning av foleybasert lyddesign for dataspill

Proessen av å lage og implementere et foleybasert lyddesign for dataspill med FMOD og Unity

Bacheloroppgave i Bachelor i Musikkteknologi
Veileder: Trond Engum
Mai 2024

Norges teknisk-naturvitenskapelige universitet
Det humanistiske fakultet
Institutt for musikk



Kunnskap for en bedre verden

Sammendrag:

Denne oppgaven går ut på å utforske Foleylyd og bygge på tidligere erfaringer med lyddesign for å forbedre implementering av og kvalitet på lyddesign for spillet *Hairy*. *Hairy* er et 2D side-scrolling dataspill hvor man spiller som hårstrået *Hairy*, som faller av hodet på en kokk inn i diverse gryter med mat. Målet er å komme seg til enden av banen og samle poeng ved å spise diverse typer mat på veien, og unngå andre matbiter og hindre som fjerner helse og poeng. Oppgaven består av en praktisk del og denne rapporten. Den praktiske delen tar for seg arbeid med lyddesignet av *Hairy*, fra lydopptak til implementering i Unity med FMOD. I rapporten diskuterer jeg prosessen, tankegang rundt arbeidet, hvordan lyddesign endrer seg når man jobber med foley i motsetning til samplebibliotek, og refleksjoner rundt arbeidet. Jeg viser til diverse spill, tekster, og videoer som formet fremgangsmåten min i det praktiske arbeidet.

Abstract:

In this assignment, I will be exploring Foley sound and building on previous experiences with sound design to improve the implementation and quality of my sound design for the video game *Hairy*. *Hairy* is a 2D side scroller video game, in which you play as Hairy, a strand of hair who falls from the head of a chef into various dishes. The goal is to make it through the stages and earn points by eating different foods along the way, and avoiding obstacles and foods that decrease health and points. The assignment is made up of creative work and this report. The creative work involves the process of working on the sound design for *Hairy*, from audio recordings to implementation in Unity with FMOD. In the report I discuss the process, thought surrounding the creative work, how sound design changes when working with Foley instead of samples from sound libraries, and reflections. I cite and mention various games, texts, and videos which shaped my approach to the creative work.

Innholdsfortegnelse

1. Introduksjon og tema	3
2. Redegjørelse av begrep og teori.....	4
Lyddesign	4
Foley	4
Lytterutmattelse	5
Suspension of disbelief.....	5
Diegetisk og ikke-diegetisk lyd	6
Auditory icons og earcons	6
3. Praktisk arbeid.....	6
Før opptak.....	7
Foley og opptak av lyd	7
Tekniske problemstillinger	8
Opptak	9
Hairys spising.....	9
Prosessering av foleyopptak	10
Liv opp / Liv ned.....	10
Svimmel Hairy	11

Portal.....	11
Bevegelse	12
Kræslyder	13
Button press	15
Bakgrunnslyd	16
Etter opptak.....	16
Bygging av lydeffekter i FMOD / Tilfeldig lydutvalg	16
Volum	18
Lengde på lydeffektene.....	18
Implementering i Unity.....	19
FMOD i Unity	19
3D elementer.....	19
Koding.....	21
4. Diskusjon / Evaluering	23
5. Referanseliste	25
Kilder.....	25
Spill.....	26
6. Utstyr.....	27
7. Vedlegg	28

1. Introduksjon og tema

Videospill er et audiovisuelt medie som har vært i stadig utvikling siden tiden 1950-tallet. Med den raske teknologiske utviklingen etter andre verdenskrig, har måten vi lager og spiller spill på endret seg drastisk. Fra den tidlige eraen av dataspill som krevde arkademaskiner, dyre hjemmekonsoller, og masse penger for å produsere spill. Nå er vi i en tid hvor hvem som helst kan benytte seg av verktøy som Unity for å lage et eget spill.

Gjennom min tid på bachelor i musikkteknologi på NTNU har jeg fått en introduksjon til Unity, komposisjon, og lyddesign for spill. I høsten 2023 var jeg med på et lite team som utviklet spillet *Hairy*. Jeg tok for meg komposisjon, lyddesign, og litt FMOD implementering, men siden ingen av oss hadde laget et spill før, måtte vi legge en samlet innsats på å faktisk ha et funksjonelt ferdig produkt. Fokus på lyddesignets kvalitet ble sidestilt for dette. Vi brukte for det meste standard lyder fra lydbanker, og tidsmangel lot meg ikke fokusere på lyddesignet til den grad jeg ønsket.

I denne oppgaven ønsker jeg å besøke *Hairy* igjen for å utforske lyddesign og produksjon av lydeffekter for audiovisuelle medier. Jeg skal starte blankt, og kun bruke egne foleyopptak for å gi spillet et helhetlig lyddesign. Fokuset i lydarbeidet mitt ligger på kreativ bruk av objekter og lydprosessering for å skape lyder som passer spillets estetikk og hendelser, samt bruk av flere lyder over hverandre for å skape større og mer komplekse lyder. Naturligvis vil jeg og se litt på implementeringen av lyd gjennom FMOD inn i Unity, mulighetene det gir, og effektiv bruk av verktøyene. Til slutt vil jeg sammenligne de to lyddesignene og reflektere på hva jeg har lært om å jobbe som lyddesigner for et spill.

2. Redegjørelse av begrep og teori

Lyddesign

Nashville Film Institute, en filmskole i Nashville, USA, definerer lyddesign (sound design) som «the art of creating the audio for film, television, advertising, music, and other productions», og nevner “kreative opptak”, “miksing”, “sampling”, “lydredigering”, “effektrekker”, og “film scoring” som arbeidsoppgaver for lyddesigneren. For denne oppgaven skal jeg ta på meg alle disse arbeidsoppgavene unntatt musikken. I arbeidet mitt lagde jeg komposisjoner som jeg var fornøyd med. Denne gangen vil jeg heller fokusere på de andre oppgavene som jeg ikke har utforsket like mye. Grunnet dette vil denne oppgaven være uten musikk, samt i spillet som er vedlagt.

Foley

I boken *The Complete Guide to Game Audio: For Composers, Musicians, Sound Designers and Game Developers*, Aaron Marks definerer Foley i kontekst av filmverden som «the art form

which adds believable sound garnishment to on-screen character movements” (Marks, 2009, s.272).

Fra Oxford English Dictionary: “In film, television, etc.: sound effects created to mimic ambient noises, typically using a variety of objects and practical methods; (also) the process of creating such effects”. (Oxford, 2023).

I denne oppgaven vil foley defineres som en blanding av de to. Når jeg bruker begrepet Foley i denne oppgaven mener jeg lyd som jeg selv tar opp for å eventuelt videre bearbeide for å passe handlinger, atmosfæren, og estetikken i spillet. Grunnet forskjeller i dataspill og film som audiovisuelle medier, vil det være en løsere definisjon på hva som kvalifiserer som ‘realistiske’ lyder.

Lytterutmattelse

Jeg forsøkte å finne en kilde på begrepet «sound fatigue», som jeg vil oversette på norsk til «lytterutslitelse», men fikk bare definisjoner som angikk mer fysiologisk utmattelse.

Definisjonen jeg ønsker er nærmere det Bjørn Jacobsen beskriver i blogginnlegget «How to maintain immersion (+ reduce repetition & listening fatigue) in game audio:». Han skriver: «This [nuisance] also happens if a sound is used over and over again in the same game. We create listening fatigue through repetition, and we generate boredom from monotony. » (Jacobsen, 2018).

Når jeg bruker begrepet lytterutmattelse mener jeg da gjentatt bruk av lik lyd som fører til irritasjon og kjedsomhet hos spilleren.

Suspension of disbelief

‘Suspension of disbelief’ er et engelsk idiom som defineres i Merriam-Webster English som «to allow oneself to believe that something is true even though it seems impossible» (Merriam-Webster, 2016). Begrepet brukes ofte i diskusjon om audiovisuelle medier og til hvilken grad seeren kan leve seg inn i fantasien som presenteres. Dette er noe som må tas i betraktning i denne oppgaven, ettersom jeg bruker ‘ekte’ foleyopptak for et spill med mange surrealistiske elementer.

Diegetisk og ikke-diegetisk lyd

I Oxfords *A Dictionary of Media and Communication* defineres diegetisk lyd som hendelser innen ‘diegesis’, som de definerer som «The spacio-temporal world depicted in the film» (Chandler, 2011, s. 100). Ikke diegetisk lyd er da hendelser som skjer utenfor, og ikke er del av det karakterene i verden hører. Disse ikke-diegetiske lydene er spesifikt laget for å gi spilleren informasjon.

Auditory icons og earcons

I konferansebidraget til Patrick Ng og Keith Nesbitt “Informative Sound Design in Video Games”, diskuterer de bruken av ‘auditory icons’ og ‘earcons’. De viser til W. W. Gavers arbeid og beskriver auditory icons som «representation based on the naturally occurring sounds we encounter in everyday life». For earcons går de til M. S. El-Nasr og S. Yan, og forklarer det som «abstract, synthetic tones that can be used in structured combinations to create auditory messages». I lyddesign for spill brukes både auditory icons og earcons for å formidle informasjon til spilleren. Auditory icons kommer i form av lydeffekter som høres når spilleren interagerer med spillverden, f.eks. i forskjellen i lyd fra å trække i gress og betong. Earcons kan observeres i form alt fra av musikk, som i Mario Kart serien hvor tempoet tar seg opp når man er på siste runde rundt banen, eller en lys lyd når du blokkerer et angrep på akkurat riktig tidspunkt, men gjenspeiler ikke gjenstander i spillets verden.

3. Praktisk arbeid

I denne delen vil jeg diskutere det praktiske arbeidet og resultatet av det. Jeg vil gå inn i tanker og problemstillinger jeg hadde før og under arbeidet, hvordan det løste seg, og resultatet av det. Arbeidet og tanker bygger på teorien nevnt i kapittel 2, samt egne erfaringer og preferanser jeg har hatt gjennom å spille dataspill gjennom mitt liv. Spillet jeg jobber med, *Hairy*, er et 2D side scrolling dataspill. Dette er samme sjanger som *Super Mario Bros.* spillet fra 1985.

Før opptak

Foley og opptak av lyd

For å begynne å lage et lyddesign for et spill må man først vite hva slags lyder som trengs. Marks anbefaler lyddesignere å samle lyder som passer spillets sjanger og estetikk i en mappe (Marks, 2009, s.312) for å ha en palett man kan gå ut ifra. Siden jeg skal lage alle lydene som brukes i spillet fra bunnen av trengs ikke dette, men det er en god påminnelse om å ha god struktur på arbeidet sitt for å slippe rot, og om å ha en fokusert estetisk retning. I boken har Marks et intervju med David Chan, lyddesigner for og eier av Giant Sandbox Studios, som spesifiserer at det er viktig å lage lister over alt man vet vil trenge lyd (Marks, 2009, s-317). Det har jeg bestemt å gjøre, og har da delt det inn i to deler: diegetisk og ikke-diegetisk lyd. Lydene som skal konstrueres er som følger:

Diegetiske lyder / auditory icons

- Karakteren Hairy
 - o Svømming
 - o Spising
 - o Vokaliseringer
 - o Kræsj
- Portal
- Objekter
 - o Hindre
 - o Topp og bunn
- Atmosfære

Ikke-diegetiske lyder / earcons

- Menyknapper
- Endring i poengsum
 - o 'Skade' lyd og 'helse' lyd
- Svimmelhet / Inaktivitet

For å lage denne listen begynte jeg med å se gjennom hvilke spillhendelser som allerede var lydsatte i det tidligere lyddesignet, og hva slags samples som hadde blitt brukt for dette. Deretter spilte jeg gjennom spillet et par ganger til for å forsøke å finne flere hendelser som kunne lydsettes. Jeg innså at det ville være lurt å gi spilleren et tegn om at de nærmet seg mål, lydsette knappetrykk i hovedmenyen, og få Hairy til å lage lyd når han styres opp eller ned. I tillegg valgte jeg å lage distinkte lydeffekter for de diverse matbitene og objektene Hairy kan interagere med.

Hairy er et spill som ikke bygger tungt på realisme, men det er fortsatt viktig å forholde seg til «suspension of disbelief». Ved å kun bruke egne opptak kan det være en risiko for at lyddesignet blir for realistisk, som kan ta spilleren ut av verden. I stedet for å fokusere på realisme, vil jeg heller tenke på å lage et lyddesign som er passende for konteksten.

Hver lyd trenger ikke være realistisk og forsøke å imitere den virkelige verden, men å forsøke kan gi karakteren Hairy mer menneskelige kvaliteter som kan skape positiv kobling mellom karakter og spiller. En blanding av realistisk og surrealistisk må finnes for å samspille med spillets estetikk, og bygge Hairy som karakter.

Tekniske problemstillinger

Mange relevante problemstillinger rundt det tekniske aspektet ved lydopptakene presenterte seg med en gang jeg begynte prosessen. Hva slags mikrofon egner seg best for foleyopptak? Hvilken samplerate egner seg best? For eksempel, så innså jeg raskt at mange av opptakene jeg trengte å gjøre var best gjort hjemme. Jeg har ikke tilgang til et studio som spesifikt er tilpasset foleyopptak og det eventuelle rotet som medfører, og mange av tingene jeg hadde sett for meg å ta opp var vrirne å frakte rundt, som en 16 kilos kettlebell og middager, så jeg benyttet meg av soverommet mitt og utstyret der. Mikrofonen jeg har tilgjengelig her er en Røde Podmic, og jeg fikk senere låne en Zoom H5 Handy-Mic for en dag for å ta opp bakgrunnslyd.

Når det kom til samplerate valgte jeg å ta opp i 96000 Hz for å beholde en god lyd kvalitet når jeg f.eks. måtte strekke en lyd. Jeg har et opptak av at jeg slår i en sommerkotelett som måtte senkes to oktaver. Hadde den vært tatt opp i 44100 Hz hadde jeg mistet transienter og kvalitet i lyden. Dette unngikk jeg ved å ta opp i en høyere lyd kvalitet. Dette ble da nedsamlet til 44100 Hz når jeg eksporterte lydene fra Audacity og Ableton Live 11.

Opptak

Hairys spising

En av de første utfordringene som presenterte seg var hvordan jeg skulle få lyden av Hairys spising til å høres ut. I spillet vil Hairy spise mange forskjellige typer mat; kjøttboller, gulrøtter, tomater, sjokolade, og kirsebær. Jeg ønsket at lyden som lages av Hairy sin spising skal stemme med teksturen til maten, og kunne fremstå som realistisk, uten å bryte med virkelighetsbildet i spillet. Den beste måten å få lyden jeg ønsket tenkte jeg ville være ved å bruke opptak av at de matvarene blir spist, og gi dem lett prosessering til et punkt det er behagelig å lytte til. Jeg er sensitiv til tyggelyder og smatting, så jeg ville sørge for at det ikke ble noe som kunne distrahere spilleren.

Jeg ville først ta opptak av at jeg selv spiser alle disse tingene for å så bearbeide lydfilene til å passe deres behov i spillet. Under denne prosessen var ikke kirsebær, moreller og andre bær i sesong, så jeg prøvde å bruke kiwi for å få en våt fruktlyd som kunne minne om det. Når det nærmet seg frist for innlevering var bær i sesong og tilgjengelige i butikkene, så jeg endte opp med å bruke jordbær i stedet. Jeg tok og opptak av at jeg spiste gulrøtter, sjokolade, kjøttboller, og tomat. I noen tilfeller, som med sjokoladen, var lyden ikke dynamisk nok til å bryte gjennom lydbildet. Skulle jeg tatt nye opptak eller prosessert dem, hadde de vært distraherende og ugjenkjennelige fra hva lydene skulle representere i spillet. Det jeg valgte å gjøre var å ta lydopptak av at jeg bet i mat som knaser og lager høy lyd. Jeg benyttet meg av potetgull og saltstenger, og la de til lydene som trengte dynamikken. Jeg tok og opptak av at jeg lagde overdrevde gapelyder med munnen. Jeg hadde spiselydene fra spillet Minecraft som referansepunkt for disse.

Selv om spiselydene gir spilleren informasjon om hva de spiser, er det og passende å legge til informasjon som sier dem om det er en god eller dårlig ting. Jeg tok lydopptak av at jeg brakk meg, sa 'æsj', og andre negative reaksjoner for mat som skader, og positive reaksjoner for maten som gir poeng og helse. En menneskelig reaksjon vil gi spilleren et direkte referansepunkt fra virkeligheten til om det de gjør er en god eller dårlig ting.

Spiselydene oppstår ofte når man spiller, så et fokuspunkt var å sørge for at spilleren ikke vil oppleve lytterutmattelse og gå lei lyden. I det første lyddesignet til *Hairy* var det kun en lyd som

ble avspilt når man spiste noe, som ble slitsomt i lengden. Variasjoner og blanding av lyder vil være viktig. I FMOD kapittelet ser jeg på hvordan man kan lage variasjoner i lydeffektene med FMOD.

Prosessering av foleyopptak

Lydopptakene skjedde i forskjellig tempo, så i stedet for å vente til jeg var ferdig med alle opptakene, begynte jeg å prosessere dem fortløpende etter opptakene. Her vil jeg gå inn på tankegangen min rundt prosesseringen av lydfilene, hva jeg ønsket å gjøre med lydene, og hvordan jeg bearbeidet dem.

Liv opp / Liv ned

Jeg ønsket å ta i bruk auditory icons og earcons for å formidle informasjon til spilleren. Et av disse er informasjon om helsenivået til Hairy. Det er en bar som viser poengsum og hvor mye helse Hairy har øverst til venstre når man spiller, men den kan være vanskelig å følge med på for en konsentrert spiller. Da passer det å bruke en lyd til å informere spilleren om endringer i helse og poengsum, slik at de kan holde øynene sine på banen. Earcons er laget av motiver, som bygger på musikalske egenskaper som rytme, toner, klang, register og dynamikk (Ng & Nesbitt, 2013). Med dette i bakhodet valgte jeg å bruke en elektrisk gitar som verktøy. Siden lydeffektene jeg skulle lage informerer spilleren om at helsen deres går opp eller ned, ville jeg ha gjenkjennelig musikalsk bevegelse i form av noter som går opp og ned.

For lyden som spilles av når du går opp i liv og poeng spilte jeg da en harmonic gå gitarens lyseste streng, og for når Hairy mister liv holdt jeg inn 12e bånd på den mørke E-strengen og slapp den for å gå ned en oktav. For å få mer bevegelse i lydene tok jeg i bruk en frequency shifter, og klang for å få en lenger lyd.

Lyden aktiveres både når spilleren går opp og ned i lengde og får eller mister poeng, så jeg ville at den 'negative' lyden skulle føles smertefull ut. For å oppnå dette prøvde jeg diverse ting som vring og bit-crushing, men lyden endte opp med å skille seg for mye ut i lydbildet når jeg prøvespilte. Til slutt endte jeg på en tremolo for å få lyden hakkete, og frequency shift ned.

Svimmel Hairy

En annen earcon jeg ville lage var for når Hairy kræsjer i et hinder og blir immobilisert i noen sekunder. Jeg ønsket at denne lyden skulle gjøre spilleren oppmerksom på at de ikke kan bevege seg for en liten tidsperiode. I det originale lyddesignet var det brukt en klassisk tegneserielyd som f.eks. blir brukt når noen slår hodet og får stjerner eller fugler flyvende i sirkel rundt hodet. For å erstatte dette ville jeg ha en lyd som vekket følelsen av å være svimmel. Svimmelhet er ikke noe som har en direkte sonisk ekvivalent, så jeg visste at jeg måtte være kreativ. Når jeg tenker på å være svimmel, tenker jeg piping i ørene, en langstrakt lyd, bevegelse, og forvirring. Dette er ordene jeg hadde i tankene da jeg prøvde å lage en passende lyd.

Jeg ville forsøke å strekke ut en kort lyd, så jeg fant et lydklipp av at jeg slo på en gaffel med en kjele og la på masse delay og klang for å strekke den ut. Lydfilen hadde en brå attack, men lang hale, så jeg tenkte den egnet seg godt. Så la jeg til en pitch shifter for mer bevegelse i lyden, vring for mer støy i de høye frekvensene, og en bit crush for å gjøre lyden skarpere.

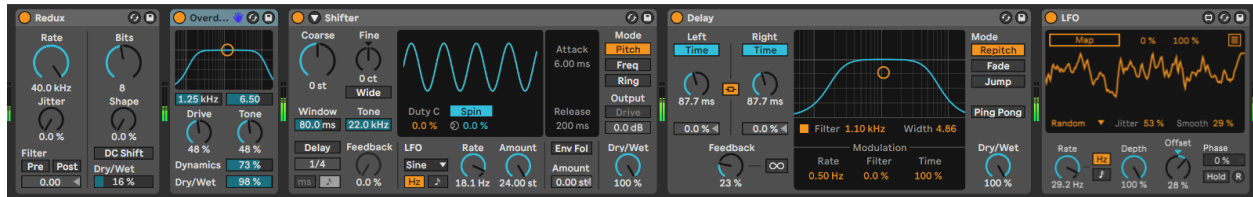
Det føltes fortsatt ut som at lydeffekten manglet noe, og jeg hadde allerede gjort det første lydklippet ugjenkjennelig, så jeg fant et lenger opptak av meg som lagde merkelige lyder med munnen i håp om å legge til mer dybde. Lydklippet hadde bass som jeg følte kunne bidra til den grumsete følelsen man kan få av å være svimmel. Jeg satt den samme prosesseringen på som for det første lydklippet, men prøvde forskjellige verdier på effektene.

Portal

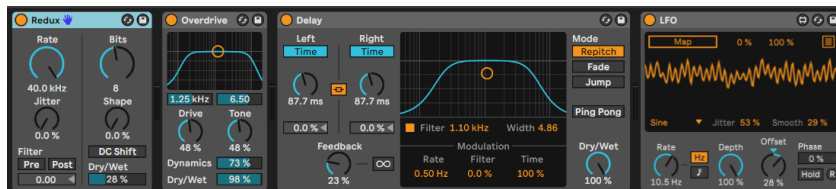
En av de nye tingene som jeg valgte å lydsette var portalen på slutten av hver bane. Formålet med dette er å la spilleren vite at de nærmer seg slutten av banen. Slike portaler finnes ikke i vår verden, så jeg følte det var lov å være kreativ og selv bestemme hvordan den skulle høres ut.

Datamaskinen min er ekstremt bråkete. Når jeg har Unity, Ableton Live, FMOD, og dusiner av faner i nettleseren min oppe høres den nærmest ut som en støvsuger. Jeg tenkte det kunne vært gøy å ta opp støyet og behandle det. Jeg brukte en EQ for å få mer diskant, etterfulgt av både pitch og frequency shifting med tilfeldig LFO, og til slutt fase. Jeg klarer ikke helt å finne ordene for å forklare sluttresultatet, som jeg synes gjør det passende som lyd for en uforklarlig portal i en gryte mat. Jeg syntes at den fortsatt manglet litt dybde, og ønsket å legge til noe mer. På siste dagen min med opptak hadde jeg en boks med Red Bull på pulten. Jeg skjøv den for å lage plass

til noe annet, og tenkte at lyden hadde vært bra å ta opp. Ved å dra boksen i sirkel over pulten fikk jeg en opp og ned bevegelse i lyden. Figurer 3.1 og 3.2 viser prosesseringen jeg brukte. Målet var å få så mye bevegelse inn som mulig. Enderesultat var fortsatt litt lyst, så jeg doblet sporet og pitchet lydfilen ned to oktaver. Trikset om pitch shifting kommer fra serien Waveform på YouTube av kanalen Marshall McGee. Marshall McGee er en profesjonell lyddesigner med mange års erfaring fra bransjen, og har laget videoer om lyddesign for spill siden 2016.



Figur 3.1: prosessering av kanne 1



Figur 3.2: prosessering av kanne 2

Bevegelse

Noe av det viktigste for meg å lydsette var at Hairy lager lyder når spilleren velger å bevege han opp og ned. Med det tidligere lyddesignet var det ingen lyd når man beveget på karakteren. Fra egen erfaring med spill er det kritisk å ha lyd på spillerbevegelser slik at man får tilbakemelding fra spillet og føler mer kontroll. Siden Hairy naturlig beveger seg fremover, mens spilleren styrer bevegelsene oppover og nedover, var det viktigste å få en klar følelse av vertikal bevegelse. Det første jeg da tenkte på var lyden av fugler som flager med vingene. Av gjenstander jeg hadde tilgjengelig på soverommet mitt, tenkte jeg at en pocketbok ville være det som kom nærmest den lyden. Jeg fant en og viften den opp og ned foran mikrofonen min, og klippet og sorterte lydfilen etter oppover og nedover bevegelsene. Denne lyden funkete fint, men var manglende. Det var vanskelig å se for seg hva det er Hairy svømmer gjennom; man følte ingen motstand. For en annen lydeffekt hadde jeg en bolle med gelé tilgjengelig, som jeg bestemte meg for å bruke igjen. Jeg tok på meg en gummihanske og tok opptak av at jeg grep og slapp geléen i bollen.

Gripelydene ble lagt med oppover bevegelsen av boken, og slippelydene med nedover bevegelsen.

Kræslyder

En av største mekanikkene i spillet er at Hairy kolliderer i gjenstander. Hver bane har egne hindere (tresleiver, metalløser, skjeer og bananer), samt en topp og bunn (grytelokk og bunn i kjelen). Topp og bunnen skader Hairy, mens hindrene sender Hairy flyvende tilbake og gjør han immobil i noen sekunder. Det første lyddesigner for spillet brukte kun en metallyd for topp og bunn, mens hindrene lagde en «boing» effekt, som av en metallfjær. Disse lydene ga en komisk effekt, men manglet dybde. Spilleren ville ikke hørt de og tenkt at de gjorde noe dårlig som de mister poeng og liv for. For å fikse dette tenkte jeg det var viktig å lage en tung lyd som kutter gjennom lydbildet. Disse filene skulle da brukes for alle lydeffekter hvor Hairy kræsjer med et hinder.

Topp og bunn

Når Hairy svømmer for høyt eller lavt kræsjer han med lokket eller bunnen av kjelen han svømmer i. For å få en realistisk lyd brukte jeg selv en metallkjele med lokk. Jeg ville ha en lyd som resonnerte, så jeg trengte noe hardt, men lett å slå med. Jeg endte opp med å bruke en gaffel, som hadde en lang klang. I FMOD la jeg til en liten delay for å skille det fra andre metallyder og gi spilleren inntrykk av at det er hult på den andre siden.

Impact

For å lage 'impact' lyden gikk jeg ut ifra en video jeg hadde tidligere sett på egen fritid, «Making a powerful punch sound from scratch || Waveform» av Marshall McGee på YouTube. Han vektlegger viktigheten av en god sub (<60 Hz) i lydeffekten for tyngde. I videoen anbefaler han å bruke seleri, hardt brød, og kjøtt for å lage lydene. Av de tre hadde jeg kun kjøtt i form av en sommerkotelett tilgjengelig, så jeg improviserte med egne hender og en trekloss. Det poengteres og i en video fra kanalen *Masahiro Sakurai On Creating Games* at det kan være lurt å ha en rask attack og lett klang på impact lyder (Sakurai, 2023).

Jeg dupliserte kjøttlyden to ganger og senket de med en og to oktaver, og la vregg på de to dupliserte. På lydsporet med den som var senket en oktav, EQet jeg for å tydeliggjøre midtfrekvensene (ca. 400 Hz til 2,000 Hz), og EQet for kun bass og sub på den som var senket 2

oktaver. Over disse tre lydsporene la jeg til opptak av at jeg slår på egen handflate, og slår en trekloss i musematten på pulten min.

Tresleiv

For å få lyden av at Hairy kolliderer med en tresleiv brukte jeg den samme treklossen. Jeg slapp den på pulten, slo den inntil, og la de lydklippene med den tidligere 'impact' lyden. For å få trelyden litt større satt jeg en klang på lydsporet i FMOD.

Skje

For å lage skjelyden brukte jeg gaffelen på et whiskeyglass med prosessering (figur 3.3). Dette var fordi jeg ønsket en lysere lyd i forhold til de andre metallydene, men fortsatt hadde litt tyngde. I FMOD senket jeg lydene noen semitoner for å en litt hulere klang.



Figur 3.3: Prosessering på lydspor i Ableton Live.

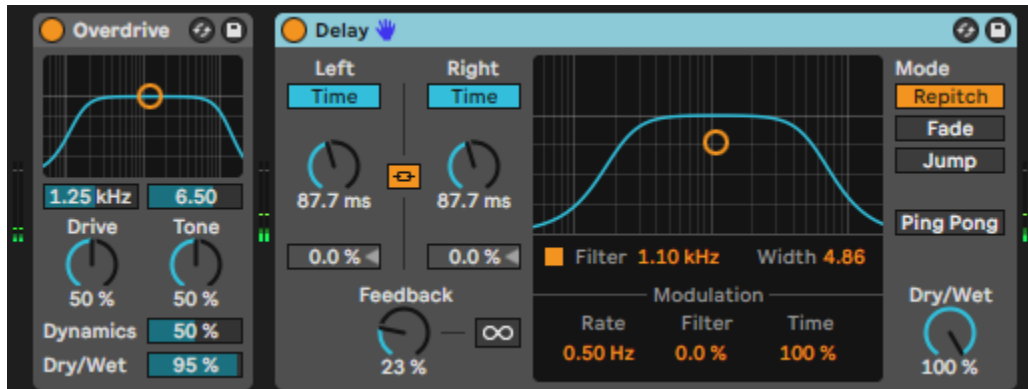
Banan

Bananen var det mest vriene hinderet å finne en lyd til. Bananer lager ikke en sterk og distinkt lyd når man slår dem. Jeg måtte tenke på hvilke kvaliteter (auditory icons) man tenker en banan har, og finne noe man kan gjenskape det med. Jeg tenkte på teksturen banan har når man spiser det, som er klissete og mykt. Dette er egenskaper jeg og kobler til gelé. Jeg kjøpte en liter med gelé som jeg helte inn i en bolle og slo til med hånden min i en gummihanske (figur 3.4).

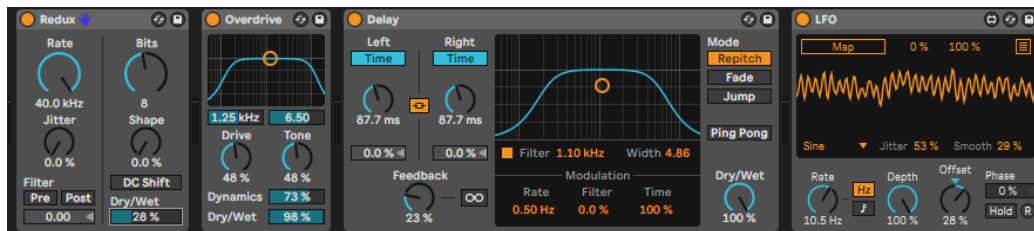


Figur 3.4: gelé og gummihandske.

For å gi lyden mer tyngde dupliserte jeg sporet og senket den ene med en oktav. På originalsporet brukte jeg vregng for å få en tydeligere lyd, og delay med høy moduleringsperiode for å gjøre lyden større og våtere (figur 3.5 og 3.6). Til slutt brukte jeg en gate for å fjerne støy.



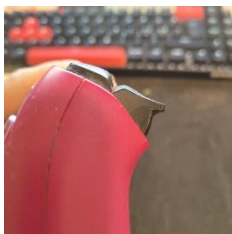
Figur 3.5: Prosessering 1



Figur 3.6: Prosessering 2

Button press

For trykkelyden i menyen tenkte jeg at det mest effektive ville være å finne noe å selv trykke, så jeg brukte en Playstation 5 DualSense kontroll. Jeg testet alle knappene og spakene for å se hvilke som lagde mest lyd, og benyttet meg av de bakerste analogknappene (figur 3.7).



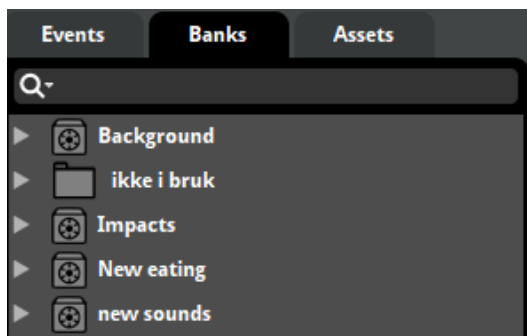
Figur 3.8: Analogknappene.

Bakgrunnslyd

Handlingene i spillet skjer i et restaurantkjøkken. Jeg tenkte at det ville være passende å ha et tilsvarende lydopptak å bruke for bakgrunnslyden i spillet. Jeg fikk låne en Zoom H5 Multi-Track Handy Recorder, og tok opptak at jeg lagde middag i eget kjøkken.

Etter opptak

Når opptakene ble ferdige, gikk jeg videre til å jobbe i FMOD. FMOD er et mellomvare program som lar deg behandle lyder for implementering i spill. Når man jobber i FMOD lager man 2D eller 3D lydtidslinjer som man laster opp til lydbanker. I disse tidslinjene kan man legge til effekter, justere volum, panorere, og mer. For hver ting i *Hairy* som skulle lydlegges måtte jeg lage en egen lydeffekt i FMOD. Jeg sorterte lydbankene ut ifra hvilken type lyd det var (figur 3.9).



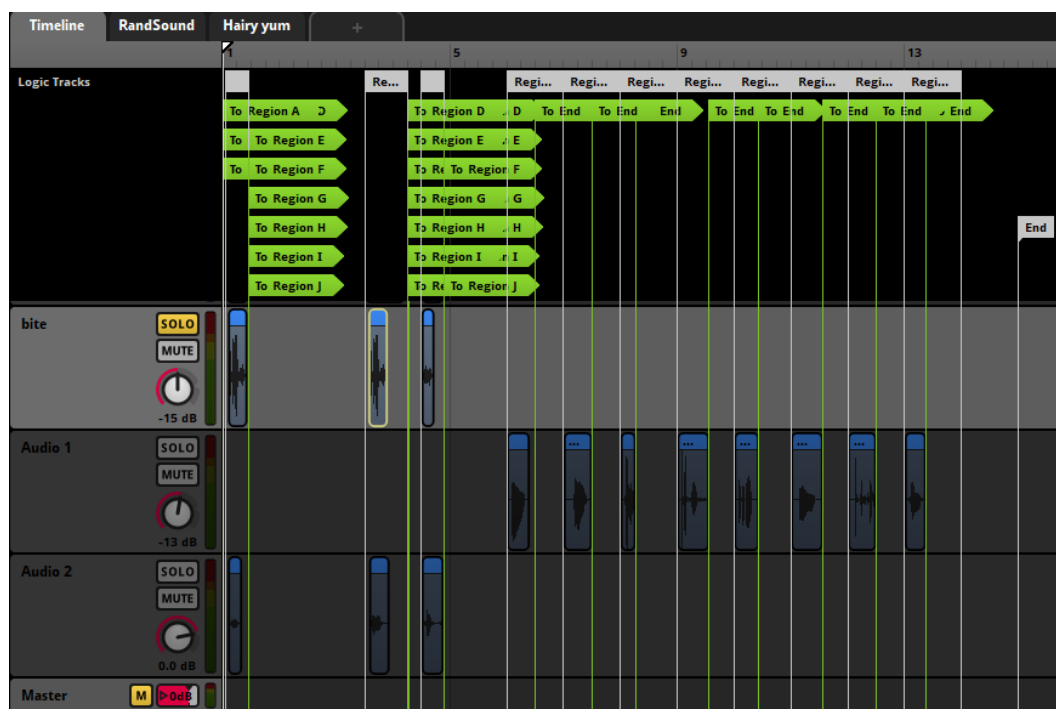
Figur 3.9: Lydbanker.

Bygging av lydeffekter i FMOD / Tilfeldig lydutvalg

Sist jeg jobbet i FMOD var prosessen av å bygge lydeffekter veldig enkel fordi jeg kun brukte én lydfil per hendelse. Denne gangen legger jeg sammen lydfiler for å bygge større og mer komplekse effekter. Jeg tenkte mye på lytterutmattelse når jeg jobbet med lyddesignet til *Hairy*. I det originale lyddesignet så spilte den samme lyden av hver gang *Hairy* spiste en kjøttballe eller en sjokoladebit, og likt var det med maten som skader han. Jeg ønsket å finne en måte å få mer variasjon i lydene samtidig som spilleren gjenkjenner lyden og hvilken handling den tilhører.

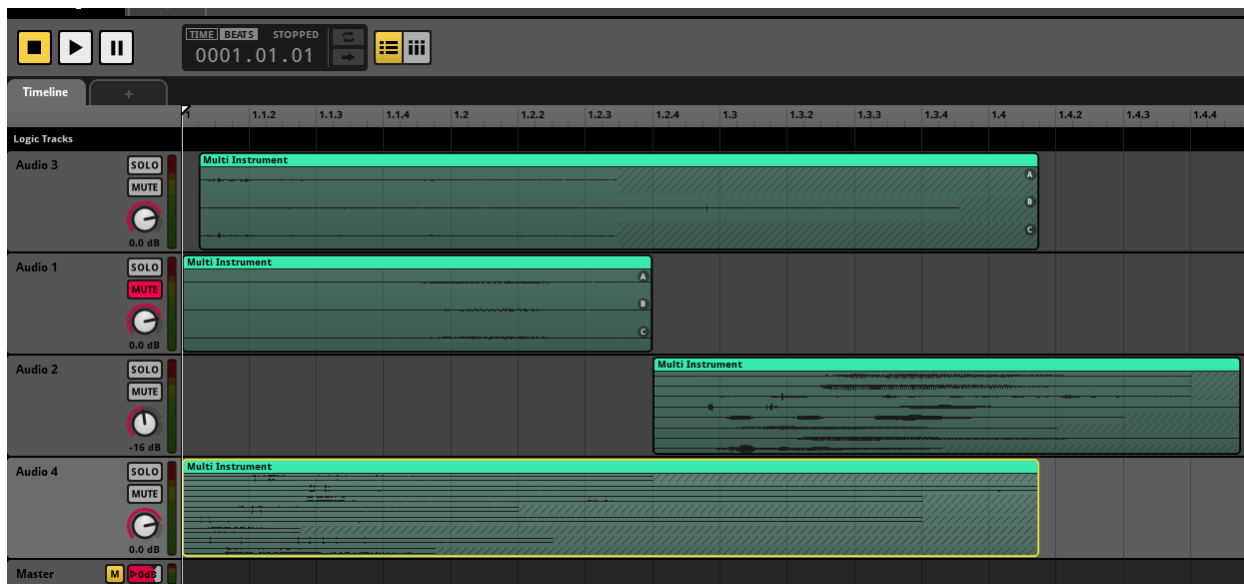
En løsning jeg så for meg var noe vi hadde vært innom før, overgangsregioner (figur 3.10). Disse lar deg hoppe fra en del av tidslinjen til en annen. Jeg prøvde å gjøre dette for å lage spiselydene, men innså at det var mange ulemper ved denne metoden. For det første så var det ekstremt

tungvint å sette opp. Tidslinjen ble veldig uoversiktlig og unødvendig rotete. Man kan justere sannsynligheten på de forskjellige overgangene til regionene, men om man jobber med en overgang til en av ti regioner, er det en 10% sannsynlighet for at overgangen skjer. Om sannsynligheten treffer to overganger så vil den som ligger øverst i 'Logic Tracks' få prioritet. Det er også en mulighet at den ikke går til noen av dem, og fortsetter langs lydsporet og spiller av flere lyder. Da måtte jeg sette opp enda en 100% sannsynlig overgang til slutten av tidslinjen bak de andre. Dette må gjøres for hver eneste region, og blir unødvendig tungt arbeid. I tillegg, om to lydfiler skulle spille av samtidig kunne man for eksempel ikke velge en tilfeldig tyggelyd av de fem man har. For kjøttbollene trengte jeg å ha to lydfiler gående samtidig, men jeg kunne ikke variere hvilke som spilte sammen.



Figur 3.10: Region oppsett

Løsningen for alle disse problemene var å bruke et 'Multi Instrument'. Multi Instruments lar deg laste opp flere lydfiler i et instrument. Instrumentet kan da tilfeldig velge et av lydklippene å spille av. Det praktiske med Multi Instrument er at det kan legges i et lydspor som et vanlig lydklipp. Du får i essens et lydklipp med flere lydklipp i seg. Dette er akkurat det jeg hadde behovde for å få tilfeldige lyder av samme natur til å spille samtidig. I Figur 3.11 ser vi hvordan den samme lydeffekten for kjøttbollene ser ut med bruk av Multi Instrument.



Figur 3.11: Multi Instrument oppsett

Volum

I videoen «Strike a Balance with Sound Effects [Audio]» av kanalen “Masahiro Sakurai on Creating Games” diskuterer Masahiro Sakurai hvordan man skal mikse lydbildet for å fortelle spilleren om viktige hendelser som skjer i spillet. Masahiro Sakurai er en dataspilldirektør og spilldesigner, mest kjent for hans arbeid med *Kirby* og *Super Smash Bros*. Seriene utgitt av Nintendo. Han sier at man, med tanke på lydnivå, skal «ta i betraktning deres viktighet for spilleren og spillkarakteren, så prioritere dem tilsvarende» (Masahiro Sakurai, 2023). Ut ifra dette forsøkte jeg å lage en prioriteringsliste over hvilke lydeffekter som skulle være høyest og lavest i lydbildet. Det viktigste tenkte jeg er endringer i spillerhelse, og det lavest prioriterte er bevegelse. Mellom disse har vi kræsje i hinder og tak eller guly, spising, og svimmelhet. Svimmelhet er et unikt tilfelle, fordi Hairy er ikke handlingsdyktig i perioden han er svimmel. Ingen andre lyder (med unntak av helse ned som starter samtidig) vil kunne spille av i denne perioden, så volumet på lydklippet trenger ikke justeres med tanke på annet en ‘helse ned’ lyden. De to andre unntakene er menyknappene og atmosfæren.

Lengde på lydeffektene

Jeg måtte og ta lengden på hver lydeffekt i betraktning med lyddesignet. *Hairy* er et spill hvor mye aktivitet kan oppstå på kort tid. Banene er fylt med mat som burde spises, mat som ikke burde spises, og hindringer å unngå. Varer lydeffektene for lenge vil de begynne å overlappes over

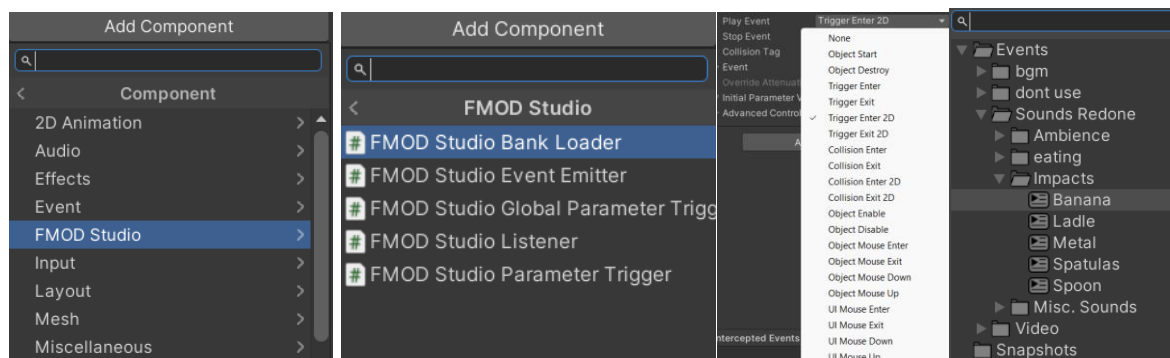
hverandre, og lydbildet vil bli rotete, som fører til at spilleren ikke får en klar idé av hva som skjer. Jeg satt en regel at spiselydene ikke burde vare lenger enn ett sekund, og helst mindre. Svimmelhet lydeffekten varer i 4 sekunder, men begynner å dempes halvveis inn, for å gå med antall sekunder man ikke kan gjøre noe etter at man treffer et hinder.

Implementering i Unity

Etter at alle lydeffektene er klare i FMOD må de implementeres til Unity. I 'Unity Asset Store' kan man laste ned et FMOD integreringsverktøy.

FMOD i Unity

Spillelementene som man kan interagere med er lagret i Unity som 'prefabs. Disse har alle komponenter til seg, som kollisjonsfelt, animasjoner, og annet. Gjennom FMOD integreringen kan man tilegne de lydegenskaper (figur 3.12 og 3.13). For spillelementer som hindre og mat må man legge en 'FMOD Studio Event Emitter'. Disse tillater deg å velge en hendelse med prefaben som utløser en valgt lyd fra FMOD lydbanken (figur 3.14. Måten lyden trigges og avsluttes på velger man selv (figur 3.15).

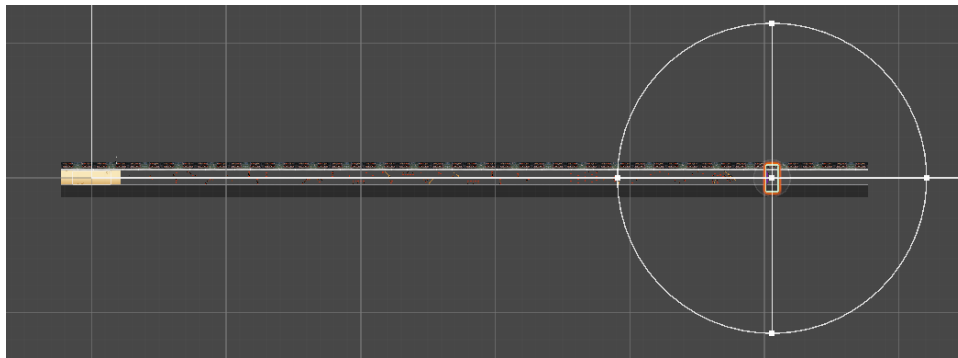


Figurer 3.12, 3.13, 3.14, og 3.15

3D elementer

Det var en prefab som måtte lydsettes på en annen måte enn resten. Grunnen til at jeg valgte å lydsette portalen som funker som målstrek på hver bane var for å gi spilleren en idé om at de

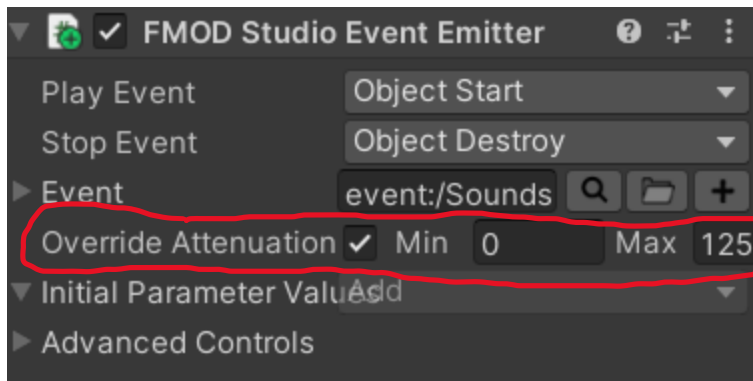
nærmer seg slutten av banen. Lyden skal først høres helt til høyre i stereobildet, og gradvis sentreres jo nærmere man kommer den. Et problem som oppsto var at lyden av portalen kunne høres over hele banen, og ikke kun innen sirkelen i figur 3.16. Løsningen til dette problemet var å legge lydklippene i en '3D Timeline' og bruke en 'FMOD Spatializer' på mastersporet, så huke av på 'User' under 'Envelopment', som vist i figur 3.17. Dette plasserer lydsporet tredimensjonelt i Unity. Man trenger til slutt en FMOD Listener på kameraelementet i Unity. For å justere hvor langt unna lydeffekten kan høres fra må man bruke 'Distance Override'. Dette kan man velge i FMOD eller Unity. Jeg valgte å bruke den i Unity for å enklere kunne se distansen i forhold til banens størrelse (figur 3.18).



Figur 3.16: Lyttefelt til portalen på Level 1.



Figur 3.17: FMOD Spatializer.



Figur 3.18: Override Attenuation i Unity.

Koding

Så langt har vi sett på implementering av lydeffekter på prefabs, men det er ikke alle lydeffekter som kan implementeres på denne måten. Bevegelsene til Hairy styres direkte i scripts. Scripts er kode som Unity leser for å utføre handlinger i spillet. Spilletets funksjonalitet ligger i disse. Unity bruker programmeringsspråket C#. Om jeg skal lydsette svømmingen til Hairy i spillet må jeg da kode det inn i ett av scriptene. Dette gjorde jeg i filen SnakeManager.cs (figur 3.19). Jeg må initiere en if statement som ser på vinkelen til Hairy som endres ved å trykke opp og ned pilene, så kalle inn FMOD med 'FMODUnity.RuntimeManager', be Unity spille av lyd med 'PlayOneShot', og til slutt legge inn filplasseringen til filen jeg ønsker å spille av.

```

void SnakeMovement()
{
    snakeBody[0].GetComponent<Rigidbody2D>().velocity = snakeBody[0].transform.right * speed * Time.deltaTime;
    CheckHeight();
    //TODO: Maybe use the unity input system later
    if (Input.GetAxis("Vertical") != 0)
    {
        // LIMITS TURN RADIUS BETWEEN 60 AND -60 DEGREES
        // https://discussions.unity.com/t/how-to-limit-the-rotation/203224/2
        Vector3 currentRotation = snakeBody[0].transform.eulerAngles;
        snakeBody[0].transform.Rotate(new Vector3(0, 0, -turnspeed*Time.deltaTime * -Input.GetAxis("Vertical")));

        // Account for multiples
        currentRotation.z = currentRotation.z % 360;

        // Then account for wrapping
        if (currentRotation.z > 180)
            currentRotation.z -= 360f;

        // Modify the rotation vector and reassign it
        currentRotation.z = Mathf.Clamp (currentRotation.z + -Input.GetAxis("Vertical")*Time.deltaTime*-turnspeed, MIN_ROTATION, MAX_ROTATION);

        snakeBody[0].transform.rotation = Quaternion.Euler (currentRotation);

        if (Input.GetAxis("Vertical")< 0)
        {
            FMODUnity.RuntimeManager.PlayOneShot("event:/Sounds_Redone/Misc. Sounds/Swim up");
        }
        if (Input.GetAxis("Vertical")> 0)
        {
            FMODUnity.RuntimeManager.PlayOneShot("event:/Sounds_Redone/Misc. Sounds/Swim Down");
        }
    }
}

```

Figur 3.19: Koden lagt til for å implementere lydeffektene fra FMOD.

Jeg måtte og inn i koden for å spille av lyd når Hairy kræsjer i toppen eller bunnen av banen (figur 3.20), og når helseverdien går opp eller ned (figur 3.21).

```
        //
    if (timer >= targetTime)
    {
        Debug.Log("Damage Topp");
        timer = 0.0f;
        mb.ShortenSnake();
        FMODUnity.RuntimeManager.PlayOneShot("event:/Sounds Redone/Impacts/Metal");
        HighScoreTracker hs = GameObject.Find("HighScoreTracker").GetComponent<HighScoreTracker>();
        hs.ScoreCollision();
    }

}
else if (height <= MIN_HEIGHT)
{
    MAX_ROTATION = 60f;
    MIN_ROTATION = 0f;
    snakeBody[0].transform.eulerAngles = new Vector3(
        snakeBody[0].transform.eulerAngles.x,
        snakeBody[0].transform.eulerAngles.y ,
        25.0f);
    if (timer >= targetTime)
    {
        Debug.Log("Damage Bunn");
        timer = 0.0f;
        mb.ShortenSnake();
        FMODUnity.RuntimeManager.PlayOneShot("event:/Sounds Redone/Impacts/Metal");
        HighScoreTracker hs = GameObject.Find("HighScoreTracker").GetComponent<HighScoreTracker>();
        hs.ScoreCollision();
    }
}
else
{
    MAX_ROTATION = 60f;
    MIN_ROTATION = -60f;
}
}
```

Figur 3.20: Kode fra SnakeManager.cs som ser om spillerkarakteren treffer maksverdi eller bunnverdi for høyde.

```

// Update is called once per frame
void Update()
{
    timer += Time.deltaTime;
    if (ph.healthCount > currentHealth)
    {
        Debug.Log("Nom!");
        currentHealth = ph.healthCount;
        FMODUnity.RuntimeManager.PlayOneShot("event:/Sounds Redone/Misc. Sounds/Length up");
        sl.value = currentHealth;
        fill.color = greenColor;
        timer = 0.0f;
    }
    else if (ph.healthCount < currentHealth)
    {
        Debug.Log("Æsj");
        currentHealth = ph.healthCount;
        FMODUnity.RuntimeManager.PlayOneShot("event:/Sounds Redone/Misc. Sounds/Length Down");
        sl.value = currentHealth;
        fill.color = redColor;
        timer = 0.0f;
    }
}

```

Figur 3.21: Fra filen InitializeHealthBar.cs

4. Diskusjon / Evaluering

Jeg gikk inn i denne oppgaven med mål om å få innsyn og erfaring i lyddesign for spill, foleyarbeid, behandling og design av lydeffekter, og implementering av lyd i Unity med FMOD. Utgangspunktet mitt var studierfaring med arbeid innen lyd for audiovisuelle medier og spilldesign, og å ha laget et tidligere lyddesign for *Hairy*. Det lyddesignet var basert på ferdiglagde lydfiler fra samplepakker og andre kilder. Jeg ønsket å forbedre på det gamle lyddesignet ved å lage egne foleyopptak, og behandle de med lydbehandlings programvare for å passe spilllets audiovisuelle estetikk og kontekst. Jeg hadde et ønske om å dykke dypere inn i FMOD og Unity for å utforske mulighetene for implementering som finnes mellom de to programvarene, men jeg innså etter hvert to ting. Den første var at spillet jeg jobber med ikke har realistiske muligheter for kompleks implementering. Det hadde vært enklere og mer givende å utforske dette med en større produksjon. Den andre tingen er at oppgaven hadde blitt for lang og omfattende for en bachelor, og jeg heller ville prioritere arbeid med lydeffekter og kreativ behandling av foleyeffekter. Arbeidet med FMOD og Unity var fortsatt veldig givende. Jeg forstår at det er essensielt å ha kunnskap i FMOD, Unity, og kodespråket C# for å kunne jobbe med lyddesign for spill på profesjonelt nivå. Tydelig kommunikasjon over alle ledd i en produksjon er viktig for effektivt arbeid. Det var flere ganger jeg ønsket at jeg hadde bedre

kunnskap i C# for å vite hvor i filene jeg måtte gjøre endringer. Noen spillelementer mangler, som f.eks. toppscoren i menyen på siste bane. Det er ikke meningen at helse opp og ned lydene skal spilles av i det man begynner på en bane, men jeg syntes at det passet greit som et varsel om at spillet er i gang.

Om jeg skal sammenligne det nye lyddesignet av *Hairy* med det gamle så synes jeg at det nyere har flere positive aspekter ved seg. Når jeg i ettertid har gått tilbake til det gamle lyddesignet er det nærmest ingen lyder som skiller seg ut som minneverdige ¹. Bruk av lydklipp fra biblioteker er veldig praktisk og kan være veldig nyttig, men er ikke lette å huske. Flere ganger når jeg spilte gjennom det lurte jeg på om jeg ikke hadde hørt noen av lydene i en annen kontekst. Lyder som er designet for spesifikt bruk skiller seg ut i mengden, og er over tid mer minneverdige. Egne lyder for de individuelle typene hindre synes jeg lagde en morsommere spillopplevelse, og variasjonen i lydeffektene som spilte i like scenarier gjorde at jeg ikke ble lei av å høre på dem. Unike lydeffekter for hver type mat ga meg mer innlevelse i spillverden. Jeg følte meg mer delaktig av å høre de små variasjonene i lyd som oppsto for hver kjøttbolle som ble spist, hver kollisjon, og hver bevegelse spillerkarakteren tok.

Gjennom arbeidet mitt med foleylyd har jeg blitt tvunget til å tenke på objekter, lyd, og deres relasjon på en annen måte. Auditory icons, informasjon samlet om objekter gjennom lyd, og earcons, strukturerte syntetiske lyder som sammen sender informasjon, har vist seg å være veldig nyttige verktøy som vil endre måten jeg tenker på og forholder meg til lyd.

Om jeg skal sammenligne lyddesign arbeid med foleyeffekter og forhåndslagde lydeffekter, så er det fordeler og ulemper med begge. Å kunne ta opp egne lydeffekter gir lyddesigneren en større grad med frihet til å tilpasse lyden til bildet, stor frihet i lydutvalg, og dypere innblikk i hvordan man kan tilpasse en lyd til settingen. Jeg anerkjenner at det til tider var vanskelig å få en lyd jeg var fornøyd med, og at det kunne vært enklere å bruke klipp fra lydbibliotek for å lage earcons og andre mer abstrakte lydeffekter som portalen, men det er en kunstnerisk glede i å forme alle lydene selv som jeg tenker lager et helhetlig bedre lydbilde. Lyddesigneren burde kunne improvisere for å lage en passende lyd til enhver situasjon, og å kunne ha frihet til å forme hver eneste lyd vil gi et resultat som er større enn summene sine.

¹ Originalversjon av *Hairy*: <https://jazzhjornet.itch.io/hairy-in-the-soup>

Den eneste gangen gjennom prosjektet jeg ønsket at jeg kunne bruke ferdiglagde lydeffekter var for bakgrunnslyden. Som student har jeg ikke mange midler og muligheter til å få tilgang til et profesjonelt kjøkken for å ta lydopptak mens det er folk i arbeid der. I teorien så kunne jeg ha laget et eget kjøkken ved å ta mange opptak av fotspor, kjøkkenutstyr, stemmer, bevegelser, maskiner, og lignende, men tiden, fokuset, og innsatsen det hadde krevd hadde i mine øyne ikke vært verdt det over å bruke et profesjonelt kjøkken. Opptakene mine som ble brukt for det endte opp med å bli mer støy enn ambiens, og jeg ser nytten av å kunne ha laste ned en lydfil tilpasset det du spesifikt trenger. Ideelt så vil jeg si at egne opptak er det beste å jobbe med for lyddesign, men det er et behov for lydbiblioteker som andre har laget. Ut ifra hvilken posisjon man er i er ikke alltid man har tilgang til utstyret som hadde gitt best opptak. En god lyddesigner burde kjenne hva som vil fungere best innen rammene man har.

5. Referanseliste

Kilder

- Blattner, M. M., Sumikawa, D. A., & Greenberg, R. M. (1989). Earcons and icons: their structure and common design principles. *Human Computer Interactive*, 4(1), 11-14. doi: 10.1207/s15327051hci0401_1
- El-Nasr, M. S., & Yan, S. (2006). Visual attention in 3D video games. *Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology, Hollywood, California*.
- Gaver, W. W. (1986). Auditory Icons: using sound in computer Interfaces. *Human Computer Interaction 2*: 167-177.
- Jacobsen, B. (2018, 16 mai). How to maintain immersion (+ reduce repetition & listening fatigue) in game audio:.. *The A Sound Effect Blog*. <https://www.asoundeffect.com/game-audio-immersion/>.
- Marks, A. (2009). *The Complete Guide to Game Audio For Composers, Musicians, Sound Designers, and Game Developers* (Second Edition). Focal Press.
- Marshall McGee. (2017, 23.7). *Making a powerful punch sound from scratch || Waveform*. <https://www.youtube.com/watch?v=vGM7dFFfmJ4>

- Marshall McGee. (2018, 11.11). *How To Easily Make Anime-Style Sound Effects*. https://youtu.be/8uJnIijUnX8?si=S6ud5maUpZRq_r0x.
- Masahiro Sakurai on Creating Games. (2022, 2.11). *Strike a Balance with Sound Effects [Audio]*. <https://www.youtube.com/watch?v=qKjD5qbLUeI>.
- Masahiro Sakurai on Creating Games. (2024, 2.4) *Balancing Ambient Sounds [Audio]*. <https://www.youtube.com/watch?v=jAgi6Uk1CU0>
- *[Audio]*. (<https://youtu.be/6Ns03q8noEA?si=oGZEL-POzrSAV9q3>).
- Masahiro Sakurai on Creating Games. (2023, 24.11). *Strong Attack, Light Reverb [Audio]*. <https://youtu.be/-nWjT49-cds?si=wLt9KUFiPQTnKx8h>.
- Masahiro Sakurai on Creating Games. (2023, 28.3) *Audio as Fiction and Non-fiction [Audio]*. <https://youtu.be/6Ns03q8noEA?si=NZpf7cNLNYDeN6c1>.
- Merriam-Webster.com dictionary. Hentet 11 mai, 2024. *Suspend (one's) disbelief*. <https://www.merriam-webster.com/dictionary/suspend%20%28one%27s%29%20disbelief>.
- Nashville Film Institute. (hentet 14.05.2024). *Sound Design: Everything You Need To Know*. <https://www.nfi.edu/sound-design/>.
- Ng, Patrick & Nesbitt, Keith. (2013). Informative Sound Design in Video Games. *ACM International Conference Proceeding Series*. 10.1145/2513002.2513015.
- Oxford English Dictionary. (Juli, 2023). *Foley, n*. <https://doi.org/10.1093/OED/2473998421>.
- Sergio Ronchetti. (*How To Use Fmod – Creating Variation*). <https://www.youtube.com/watch?v=GmEcjNU5XzY>

Spill

- Hairy (Hairy Studios, 2023)
- Fatal Fury: City Of the Wolves (SNK, 2025)
- Sonic The Hedgehog (Sega, 1991)
- Super Mario Bros (Nintendo, 1985)
- Minecraft (Mojang Studios, 2009)
- Super Smash Bros. Ultimate (Nintendo, 2018)

6. Utstyr

Hardware:

- *Windows 11 Home Laptop*
- *Universal Audio Volt 2*

Software:

- *Ableton Live 11 Suite*
- *Audacity*
- *FMOD Studio*
- *Unity*

Hodetelefoner:

- *Sony MDR-7506*

Mikrofoner:

- *Røde Podmic*
- *Zoom H5 Multi-Track Handy Recorder*

Gjenstander (foley):

- *Bringebærgelé*
- *Gummihanske*
- *Svinekotelett*
- *Kjøttboller*
- *Potetgull*
- *Saltstenger*
- *Gulrøtter*
- *Jordbær*
- *Kiwi*
- *Tomat*
- *Melkesjokolade*
- *Kanne med Red Bull*

- *Drikkeglass*
- *BIC Pen*
- *Kjøkkengryte*
- *Kettlebell*
- *Whiskeyglass*
- *Gaffel*
- *Laptop*
- *Stemme*
- *IKEA Bergenes mobilholder*
- *DualSense trådløs kontroll*
- *Elektrisk gitar*
- *Pocket bok*

Annet:

- *Studio 2, Fjordgata*

7. Vedlegg

SpillOgLyd.zip

