

# Reeksamen: DADIU Praktik Rapport

I foråret 2021 ansøgte jeg om at påtage mig rollen som audiodesigner i min praktik på DADIU. I ansøgningen skulle jeg svare på en række spørgsmål i forhold til min motivation for at deltage i DADIU-forløbet. Jeg tilkendegav at jeg var nysgerrig på de virtuelle verdener der findes i computerspil, og hvilke potentialer der ligger indenfor lyd-design. Hvad er det for designprocesser der forudsætter disse virtuelle verdener? Dernæst gjorde jeg det klart at min interesse for spiludvikling naturligvis kom fra et audiodesignmæssigt perspektiv, og at jeg anså DADIU som en unik mulighed for netop at arbejde med non-lineær lyd i virtuelle miljøer.

Et andet vigtigt aspekt i DADIUs forløb er hvordan man fungerer i en interdisciplinær hold-dynamik. Min antagelse forud for forløbet var at man ville have mulighed for at inspirere og lære af hinandens forskellige fagligheder. På et interdisciplinært hold så jeg mig selv i rollen som audiodesigner; som en person der ville være i stand til at snakke med alle de forskellige fagligheder på holdet. Ligeledes forestillede jeg mig at være i stand til at kunne give konstruktiv, reflekteret kritik af den overordnede designproces, med udgangspunkt i min egen interdisciplinære faglighed, mellem grundforståelse for lyd og computerkode til,

Tabel 1 DADIU projektforsøgs plan

W35	W36	W37	W38	W39	W40	W41	W42
ASKOV	MGP#1	MGP#1	Competence 01	Competence 02	MGP#2	MGP#2	Competence 03
W43	W44	W45	W46	W47	W48	W49	W50
PREPROD	GG W1	GG W2	GG W3	GG W4	GG W5	GG W6	Reflections

humanistisk forståelse for forholdet mellem mennesker og teknologi.

Jeg påbegyndte mit DADIU-praktikforløb på Askov Højskole i efteråret 2021. Introforløbet på Askov Højskole havde til formål at ryste deltagerne sammen og give os et overblik over DADIU-praktikperioden, samt hvilke forventninger der ville være for de seks deltagende hold over den kommende periode på 15 uger (Tabel 1). Derudover fik man mulighed for at snakke sammen med ens eget hold, og få afgjort hvem der skulle påtage sig lederpositioner indenfor de forskellige fagligheder (Tabel 2). Ligeledes fik vi her allerede fastsat vores navn for holdet som "Wizbrew Entertainment" (Figur 1).

Tabel 2 Rollefordeling på Wizbrew Entertainment

Antal	Rolle
1	Game Instruktør
1	Producer
5	Programmer
5	Kunstnere
4	Game/Level Designers
1	QA/UR
1	Audiodesigner
1	Komponist

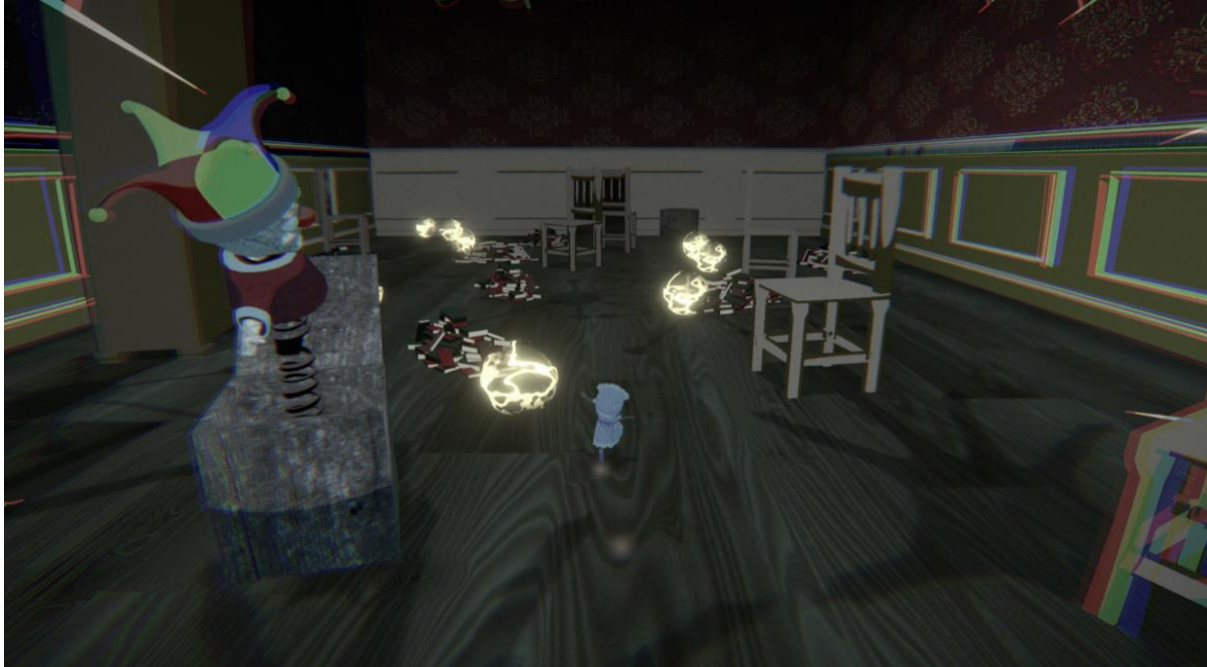


Figur 1 Wizbrew Entertainment Logo

Efter DADIU's intro-arrangement på Askov Højskole blev vores hold, Wizbrew Entertainment, sendt til Aalborg hvor vi ville mødes i CREATE bygningen på det lokale

universitet. Her skulle produktionen af *Mini-Game-Projekt-1*(MGP1) forgå over de næste par uger.

## Mini Game 1: Late to Bed



Figur 2 Eksempel på en korridor i spillet Late to Bed

Under MGP1 udviklede vi et spil indenfor genren Arcade Endless Runner, Spillerens interagerer i karakteren af en lille pige iført nattøj. Udgangspunktet for spillets historie er at den lille pige er ved at komme for sent i seng, og på sin vej gennem det gamle mørke hus kommer nogle mareridtsagtige fangarme der forsøger at trække pigen ind i mørket. Spillerens opgave er sørge for pigen ikke bliver fanget af mørket ved at løbe ned gennem de frygtindgydende og forvrængede korridorer i det gamle hus. I korridorerne er der imidlertid en masse forhindringer der gør det svære for spilleren at undslippe mørket. Heldigvis findes der rester af lys der kan give spilleren forøget hastighed, og gøre det lettere at undslippe (Figur 2).

Spilleren kan bevæge karakteren vha. WASD-tasterne, hoppe på mellemrums-tasten og lave et "dash" på shift-tasten, hvor den lille pige hurtigt rykker sig i en bestemt retning. Det der karakteriserer en Endless Runner er at der ikke findes noget endeligt mål for spilleren at



Figur 3 Late to Bed Game-Over Screen

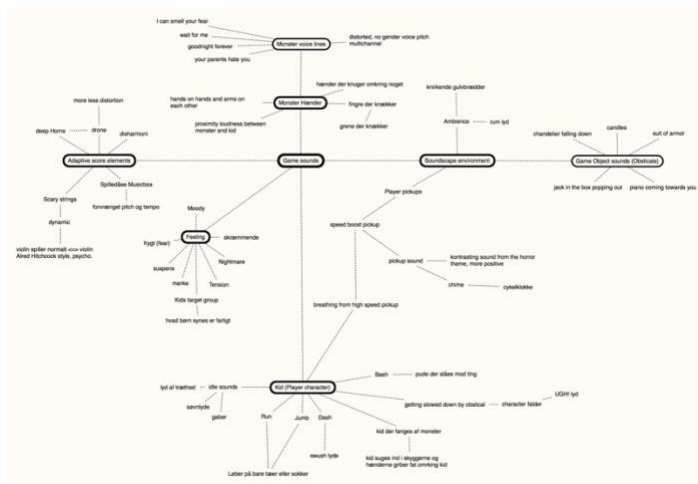
opnå. I stedet for handler det om at spilleren skal nå så langt som muligt før de opsluges af mørket.

Når spilleren opsluges af mørket, ophører spillet mens en "game-over screen" (Figur 3) oplyser spilleren den distance de har opnået i deres gennemspilning, samt deres bedste hidtidige distance, det man populært ville kalde

”highscore”. Game-over-screen giver spilleren mulighed for igen at spille, og potentielt slå sin forrige rekord. Derudover har spiller også mulighed for at afslutte spillet.

Udgangspunktet for spillets udviklingen var nogle målsætninger defineret af spil direktøren. Disse overordnede målsætninger var definerende for spillets overordnede ”game feel”. Altså hvilken følelse der gennem spillet forsøges at blive vækket i spillerens oplevelse. ”Tension” eller spænding var den primære målsætninger. Dertil kom vores designmæssige udfordringer; ”childs fear of the dark” og ”procedural 3D level” der særligt var opstillet for hhv. kunstnere og programmers.

I min rolle som lyddesigner havde disse målsætninger og designmæssige udfordringer betydning i forhold til lyddesignet stilistiske fokus på gys og suspens. Ligeledes havde målsætningerne også indflydelse på implementeringen af lydende. I den indledende fase af produktionen var det vigtigt hurtigt at få et overblik over potentielle lydlige aspekter i spillet. Altså, hvilke elementer i spillet der kunne have behov for at blive lydliggjorte. For at danne et overblik gik jeg mellem de forskellige fagligheder og spurgte ind til deres overvejelser og ideer.



Figur 4 Late to Bed Mindmap

F.eks. ville jeg snakke med designere og kunstnere i forhold til hvilken grafisk stil de havde valgt, hvilke 3D objekter de havde tænkt sig at modellere, samt hvilke overordnede stemninger de forsøgte at opnå gennem deres designs. Derfra ville jeg snakke med Game og Level designerne i forhold til spillerens bevægelser og andre spillemekaniske egenskaber, heriblandt, hvilke objekter skulle kunne opsamles, hvorvidt der kunne være potentiale for feedbacklyde etc. Herefter kunne jeg så diskutere med spil-direktøren hvordan hans vision i

forhold til monstre fra mørket var, og hvilke lyde der kunne tænkes at være og hvordan de kunne understøtte det fortællermæssige aspekt.

Gennem mine undersøgelser fra de forskellige fagligheder og deres ideer, kunne jeg eksternalisere det som et mindmap (Figur 4). Mit mindmap gav mig et overblik over de forskellige lydligt potentielle aspekter jeg ville kunne arbejde videre med. Mindmappet anskueliggjorde forskellige relationer mellem spillets mekanikker, spilkarakterer og spilmiljøets lyde. Jeg har videre i rapporten udvalgt to arbejdsopgaver der demonstrerer nogle af mine faglige kompetencer som lyddesigner. Først min refleksive lyddesign tilgang til udviklingen af spillet soundscape. Dernæst mine overvejelser i forhold til lyddesignet af monsteret fra mørket, og hvordan jeg gennem eksperimenterende vokalsyntese forsøgte at give monsteret en stemme.

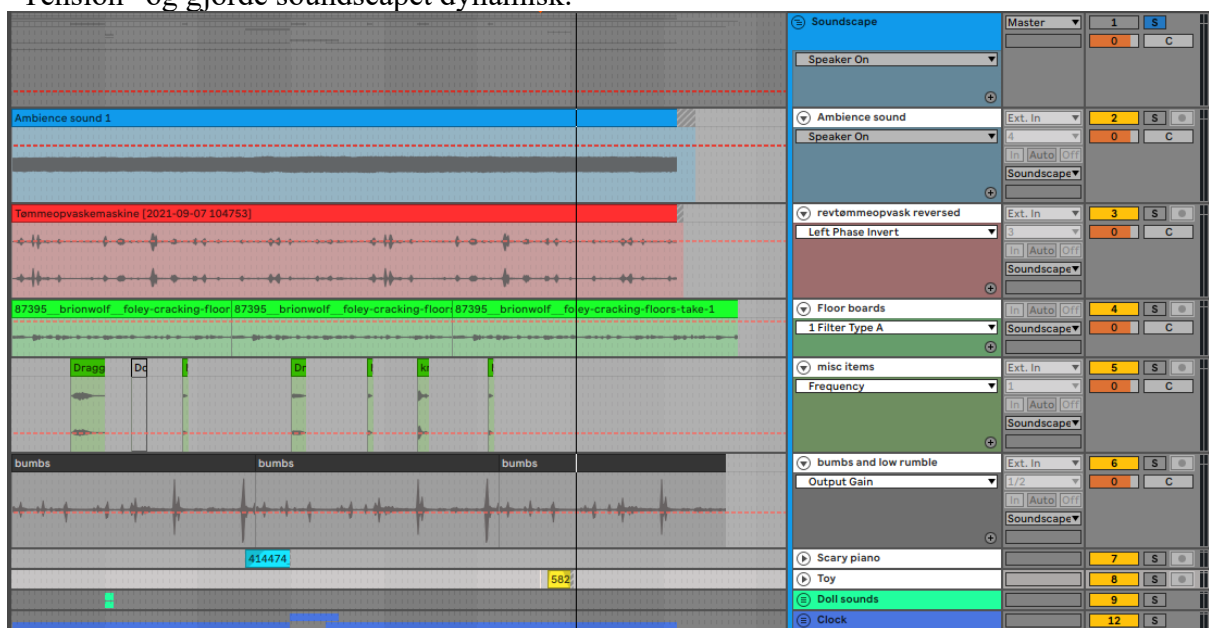
## Late to Bed: udviklingen af Soundscapet

”Soundscape environment” er i mindmappet sat i sammenhæng med rum lyd og knirkende gulvbrætter i forhold til et udgangspunkt for lyddesignet.

Hensigten med soundscapet i "Late to Bed" var at konstituerer de uhyggelige korridorer spilleren løber igennem. Soundscapets funktion i spillet var at give spilleren en anspændt og uhyggelig stemning. I forhold til lyddesignet var jeg inspireret af mine erfaringer som barn, og de lyde der var så fremtrædende om natten når alt lyset var slukket. Her kunne jeg særligt genkalde mig lyden af tikkende ure i min mormors hus, samt lyden af knirkende gulvbrædder. Derudover gav spillerens karakter, den lille pige, mig associationer til legetøj og dukker. I lyddesignet arbejdede jeg således med hvordan disse kunne iscenesættes uhyggeligt, f.eks. hvis man forvrængede de mekaniske dukke stemmer.

Mange af disse lyde måtte jeg finde på online lydbiblioteker, men der var dog enkelte lyde jeg selv lavede fra bunden. Jeg anvendte en optagelse af mig selv der tømmer opvaskemaskine. Ved at anvende en optagelse af noget bekendt og hverdagsligt gav det noget dynamik i soundscapet. For at gøre det mindre bekendt og mere uhyggeligt kunne jeg efterbehandle optagelsen ved at ændre tonehøjde og afspille optagelsen bagfra. Det hverdagslige kunne således blive mere obskurt og ukendt.

Rumtonen der kører som en konstant drone lavede jeg på min eurorack synthesizer hvor jeg forsigtigt regulerede på nogle filter parametre der gav dronen dynamiske træk. Disse dynamiske træk var med til at lægge en form for ildevarslende undertone der understøttede "Tension" og gjorde soundscapet dynamisk.



Figur 5 Ableton Arrangement af Soundscapet til Late to Bed

Jeg arrangerede lydende i Ableton Live (Figur 5) hvorfra jeg kunne generere en samlet lydfil som der kunne arbejdes videre med i lydimplementerings-middleware-programmet Audiokinetic Wwise, som jeg senere i rapporten vil komme nærmere ind på.





*Figur 6 Late to Bed Monstrene fra Mørket*

Lyden af Monstrene fra mørket (Figur 6) var en af de vigtigste lyde i spillet, idet lyden havde to vigtige anvendelser. Den var både æstetisk repræsentativ for spillets atmosfærer, og samtidig havde den en vigtig, funktionel anvendelse. Den var æstetisk i den forstand at lyden af monsteret skulle i forlængelse af spil-direktørens vision og målsætninger give spilleren en følelse af "tension". Den var funktionel i den forstand at lyden af monsteret skulle kunne fungere som en indikator for spilleren, der kommunikerede hvor monsteret befandt sig i forhold til spilleren selv. I lyddesignet tog jeg udgangspunkt i monsterets udformning i forhold til dets mange hænder, der forsøgte at gribe ud efter spilleren, samt dets mørke-røg-skyggeagtige visuelle stil. Jeg tog i mit lyddesign udgangspunkt i at bruge lyde fra hænder der stryges mod hinanden, fingre der knækkes, og klap.

I lyddesignet af monsteret havde jeg brug for at lave en lyd, der indikerede monstrenes bevægelse. Denne lyd blev i mit lyddesign præget en klistret sammensætning af knækkende fingre og lyden af en banan der blev spist optaget med en shotgun-mikrofon med høj gain setting, for at opnå alle de klistrede mundlyde der kunne illustrere en form for bevægelsesdynamik hos monsteret.

Dernæst havde jeg brug for at monstrene havde en form for stemme eller brøl. I mit ableton projekt forsøgte jeg at konstruere en "Hand voice", altså bygge en form for stemmeagtig lyd ud af de forskellige håndlyde jeg havde til rådighed. Min idé var at anvende forskellige håndlyde hvor de blev pitched forskelligt og med forskellige afspilningshastigheder, for så efterfølgende at efterbehandle lydene gennem nogle bandpass filtre der ville fremhæve nogle frekvenser mere resonante end andre. Ved at tilføje de forskellige bandpass filtre gjorde det, at jeg kunne give monstrenes stemme en unik karakter. Denne teknik er inspireret af formant-syntese og vowel-syntese, der fokuserer på hvordan man kunstigt kan skabe lyde, der efterligner menneskets vokallyde.

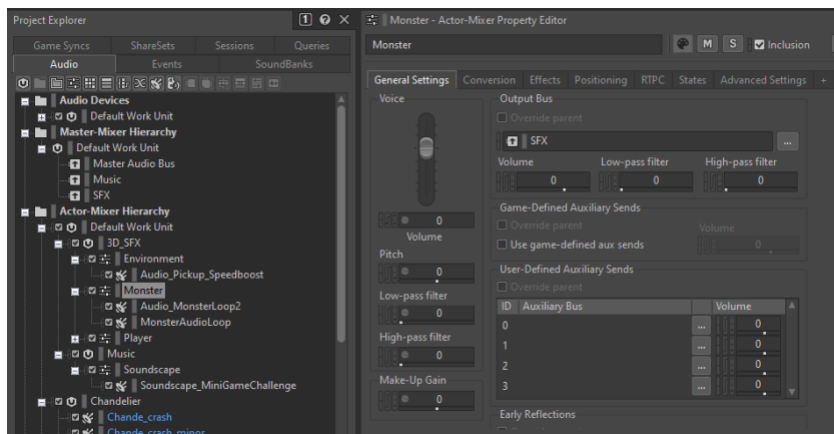
Tabel 3 Vowel Synthesis partialer

Phonetic Symbol	Example Word	$F_1$ (Hz)	$F_2$ (Hz)	$F_3$ (Hz)
/ow/	bought	570	840	2410
/oo/	boot	300	870	2240
/u/	foot	440	1020	2240
/a/	hot	730	1090	2440
/uh/	but	520	1190	2390
/er/	bird	490	1350	1690
/ae/	bat	660	1720	2410
/e/	bet	530	1840	2480
/i/	bit	390	1990	2550
/iy/	beet	270	2290	3010

F.eks. kunne jeg vha. ovenstående (Tabel 3) udvælge en vokal lyd som /u/ og indstille tre bandpass filtre med hhv.  $F_1=440$ ,  $F_2=1020$ ,  $F_3=2240$ . og således opnå at mit lydråmateriale (i dette tilfælde lyden af hænder) ville få en mere unik, stemmelignende kvalitet.

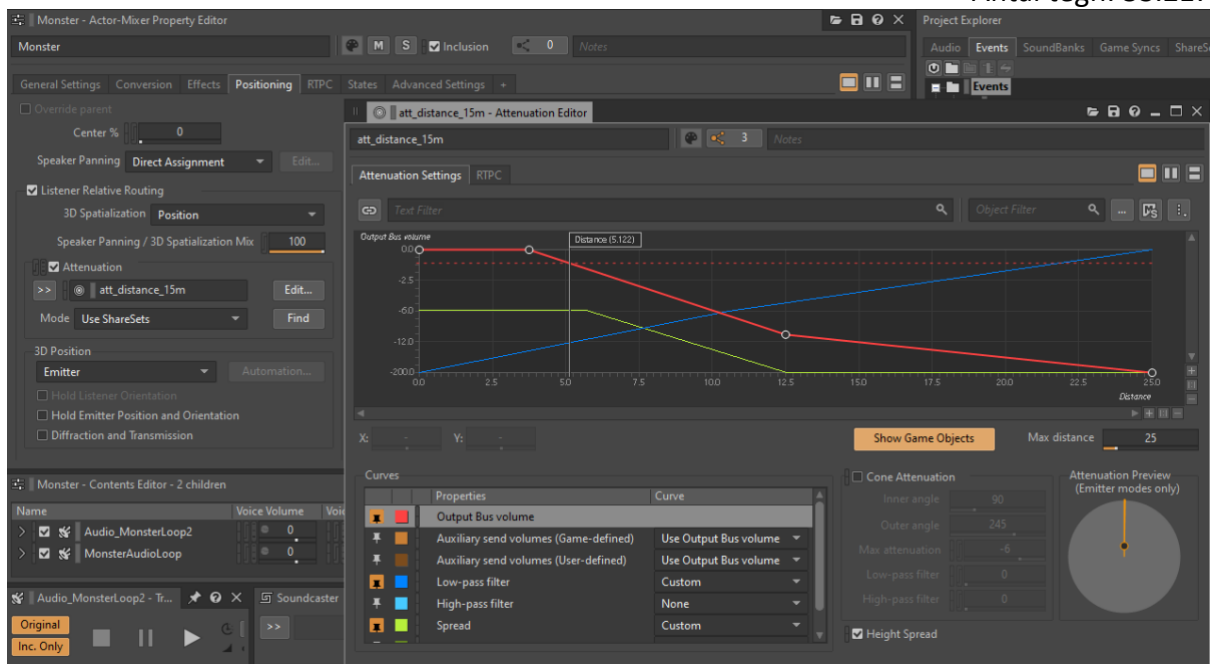
## Implementering af monstrenes lyd med Wwise

Efter lyddesignet af monsterlyden var færdiggjort skulle den implementeres i unity projektet, vha. Audiokinetic Wwise. Wwise er det man indenfor spil-industrien vil klassificere som et middleware, noget der er mellem lyddesignerens lydfiler og unity game engine. Wwise kan også kaldes for en "sound engine" der er forbundet med den overordnede "game engine". Først og fremmest overføres lydfilerne af monsterlyden jeg havde lavet i Ableton til Wwise.



Figur 7 Wwise setup af monsterlyden

Først indsættes monsterlydfilerne under actor-mixer hierarchy under en "actor mixer" jeg kalder for "monster". "monster" actor mixer sættes ind under en mixer bus "SFX" der er under Wwise "Master Audio Bus" (Figur 7). Dernæst opsættes logikken i Wwise for at kunne afspille lydende ved at kalde events. I Events laver jeg et "work unit" hvorunder jeg opsætter et MonsterPlay og et MonsterStop event. Disse to events kan hhv. afspille eller stoppe afspilning af monsterlyden.



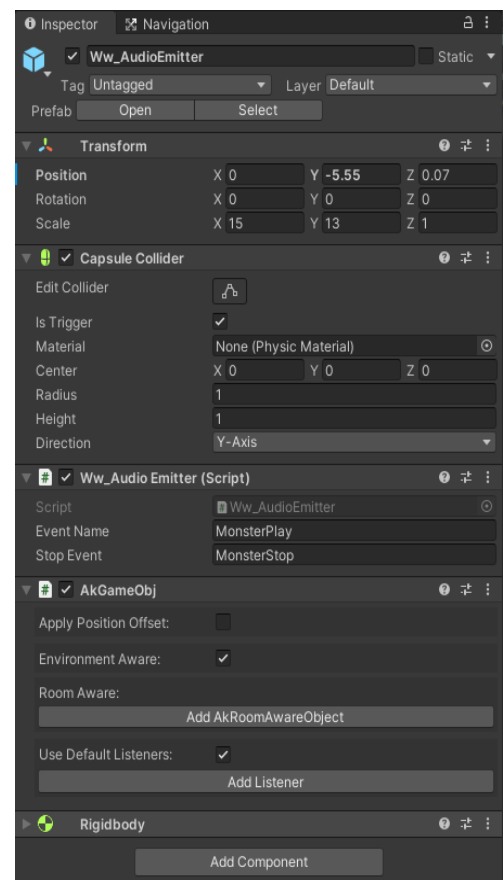
Figur 8 Monsterlydens attenuation kurve

I Wwise er alle lyde der indsættes som standard en 2D lyd. Det vil sige at lyden vil blive hørt af spilleren som om den er helt tæt på og ikke har nogle spatiale aspekter. For at gøre monsterlyden til en 3D lyd i Wwise tillægges den en "attenuation curve" (Figur 8). Dette er en kurve der afhængigt af spillerens (lytterens) position til lyd-emitter vil lade udvalgte parametre fade ud, for på den måde at give lyden pseudo-spatiale kvaliteter. I tilfældet med monsterlyden er der påsat en attenuation curve, der forandrer parametrene for lydens volume, filtrering, og stereo-image.

Når logikken og lyden er opsat i Wwise, skal der genereres en soundbank. En soundbank er en fil Wwise genererer, der indeholder alt, der skal til for at afspille og finde en lydfil, og de parametre der er opsat. Det er denne soundbank, der kan aflæses i Unity, og hvori de forskellige lyde kan blive afspillet ved at tilknytte c#-scripts til de game-objekter hvorfra lyden skal afspilles. I tilfældet med monsterlyden skal et c#-script tilknyttes på Unity-gameobjekt "KillWall" hvortil yderligere et gameobjekt kaldet "Ww\_AudioEmitter" der indeholder en capsule-collider funktionalitet. Denne bestemmer hvorvidt spilleren er indenfor den attenuation kurve der er defineret i Wwise, hvilket har afgørende betydning for hvordan spilleren opfatter den afviklede lyd. For at Wwise ved at det pågældende Unity-gameobjekt skal kunne afspille lyd, tilføjes et "AKGameObj" der fortæller Wwise, at dette objekt skal kunne afspille eller ændre parametre af en lyd i Wwise (Figur 9).

Et uddrag fra scriptet ses en funktion der beskriver hvordan capsule colliders interaktion med den øvrige "game world" afgøre hvorvidt monsterlyden er hørbar for spilleren eller ej.

```
Private void ontriggerEnter(Collider other)
{
    If (other.tag != "Player" || !IsInCollider) {return; }
    IsInCollider = true;
    AkSoundengine.PostEvent(Eventname, gameObject);
}
```



Figur 9 Ww\_AudioEmitter Unity-gameobjekt & AkGamObj

Først specificerer funktionen, at den aktiveres når noget kommer indenfor collider i kraft af at det er en "ontriggerEnter" funktion. Inde i funktionen er der dernæst et konditionalt statement, hvor det specificeres, at hvis det er et Unity-gameobject, der er tagget med "Player", så igensætter det et event der starter lyden i Wwise ved at compile `AkSoundengine.PostEvent(Eventname, gameObject)`. Eventname variabelen i koden kan varieres ved at skrive navnet på det pågældende event man ønsker at afspille i Wwise. Tilfældet med funktionen i scriptet kan Eventname erstattes med `MonsterPlay`.

## Feedback af MGP1

Mit lyddesignarbejde i kombination med min implementering af lyden ind i spillet gennem Wwise gør, at Monsterlyden kan videresende information til spilleren. På grund af Monsterlyden så ved spilleren hvor langt væk eller tæt på monsteret er fra dem. Kombinationen af den uhyggelige monster-lyd og at lyden kommer tættere og tættere på, skulle gerne give spilleren en fornemmelse af tension, når de forsøger at løbe hurtigere væk fra monsteret. Da MGP1 var færdigt, havde vi en præsentation og evalueringssession med nogle folk fra DADIU. Her fik jeg en masse konstruktiv kritik på netop min monsterlyd i forhold til hvordan den fungerede i spillet. Meget af feedbacken omhandlede hvordan



monsteret synes at være ret stille i forhold til andre monstre såsom xenomorph monstret fra Alien eller "the human centiped" fra filmen af samme navn.

Mit lyddesign af monsteret var domineret af dens klistrede hånd og fingreknækkende lyde, der ikke fungerede skræmmende i samme omfang, som hvis monstret brølede med et udgangspunkt i et mere naturligt fremfor kunstigt stemme syntese. I retrospekt tænker jeg at meget af det anonyme monster lyddesign skyldes at monsteret ikke var synderligt defineret. Baseret på konceptart hvad jeg havde nået frem til med mine samtaler om monstre med spil direktøren, var det mest baseret på hænder der kom ud af en mørk skygge-røg.

Jeg anså feedbacken fra MGP1 i forhold til monsteret som et perfekt afsætningspunkt fra MGP2 da vi havde fået at vide, at det i højgrad skulle ansues som et redesign af MGP1. Vi kunne altså regne med at MGP2 igen ville være et Endless Runner Arcade og igen ville have et element af et monster eller noget lignende, der ville være i hælene på spilleren.

## Competence-week

I DADIU-forløbet er jeg nået til uge 38-39, der var kompetence-ugerne af forløbet. Her havde jeg mulighed for at høre om de andre lyddesigneres oplevelser med MGP1, samt få feedback fra dem og vores mentor Bjørn Jacobsen fra CUJO Sound. Derudover blev tiden i disse uger anvendt på at lære mere om hvordan Wwise integrering fungerede med Unity.

Noget jeg synes var interessant, når jeg mødte de andre lyddesignere, var hvordan deres teams havde arrangeret sig forskelligt i forhold til hvor meget agens man som lyddesigner havde i forhold til netop implementeringen. Hvor jeg på mit eget team selv havde taget ansvar for at lave scripts hvori lyden blev implementeret, fandt jeg at andre hold havde afsat en af deres programmører til at være en slags lydimplementerings programmør der lavede al koden til at afvikle lyd i spillet. Dette frigjorde lyddesigneren til netop så at designe lyden der skulle bruges i spillet. Jeg fandt dog at fordelene ved at jeg både stod for alt med implementering sammen med alt lyddesign var at jeg ikke kom ud i problemer hvor lyd blev afspillet uden at jeg vidste præcis hvor i koden og på hvilke Unity-gameobjekter det befandt sig.

## Mini Game 2: Dysmemoria



Figur 10 MGP2 Dysmemoria

Efter competence-week gik vi ind i uge 40-41, hvor vi startede produktionen af MGP2(mini Game Project 2) (Figur 10). I MGP2 havde vi mere fokus på at spillet skulle være tæt forbundet med et bagvedliggende narrativ. Spilleets narrativ omhandler en mand kaldet James vi først møder som patient på et psykiatrisk hospital. James er i gang med en behandling for hans chok eller katatoniske, psykiske lidelse. Inde i James hoved har han det samme mareridt, der gentager sig igen og igen. Som James i spillet forsøger spilleren at undslippe de frygtindgydende manifestationer af James' undertrykkede minder, alt imens man løber gennem den surrealistiske forvrængede verden inde i James' underbevidsthed. Et nyt element der kom til i MGP2 var hvordan spilleren i løbet af spillet ville finde nogle lysende puslebrikker der er fragmenter af James hukommelse og giver spilleren et dybere kendskab til baggrundshistorien for spillets narrativ.

Det yderligere fokus på fortællingen i MGP2, samt den feedback jeg fik på lyddesignet af monsteret fra MGP1, gjorde at jeg i starten af udviklingsprocessen af spillet i fik fastsat et mere defineret design og beskrivelse af monsteret. Dette skete i tæt samarbejde med de øvrige fagligheder på holdet. Hvor monsteret i MGP1 var de uhyggelige, mørke hænder, der kom ud af mørket for at gribe efter spilleren, var udgangspunktet for monsteret i MGP2 alle James' frygtelige minder i en stor sammenblanding. I spillets fortælling er James endt på den psykiatriske afdeling efter at han i beruset tilstand kom til at køre sin gravide kone ned. Som konsekvens af dette er James endt i en katatonisk tilstand, hvor han igen og igen oplever uheldet den pågældende aften.

Designet af monsteret tager udgangspunkt i hændelsen og skal forestille en form for blanding af James' døde kone og barn, samt dele af en bil ligesom den han kørte dem ned i (Figur 11). Lyden af monsterets bevægelse er således, ligesom det første monster, uden ben.

Lyden af monsters brøl skal denne gang tage udgangspunkt i konen og barnet, for herefter at blive sammenblandet med motorlyde for at understrege monsterets visuelle design.

For at lave lyden af monsteret ønskede jeg i stedet for den formant syntese tilgang jeg havde anvendt i MGP1 at anvende optagelse af min egen stemme der brølede, og så anvende en mere traditionel layering-teknik og post processing. Jeg havde mulighed for at låne et lydstudie på CREATE hvor jeg vha. mit lydkort, hovedtelefoner, og en condenser mikrofon fik optaget forskellige brøl, skrig, gurglen og andre mundlyde der kunne give tekstur til monsterlyden. Min hensigt var at lave en monsterlyd der skulle lyde mere frygtindgydende og uhyggelig end monsteret i MGP1.



Figur 11 Monster 3D model from Dysmemoria

## Voiceline optagelser

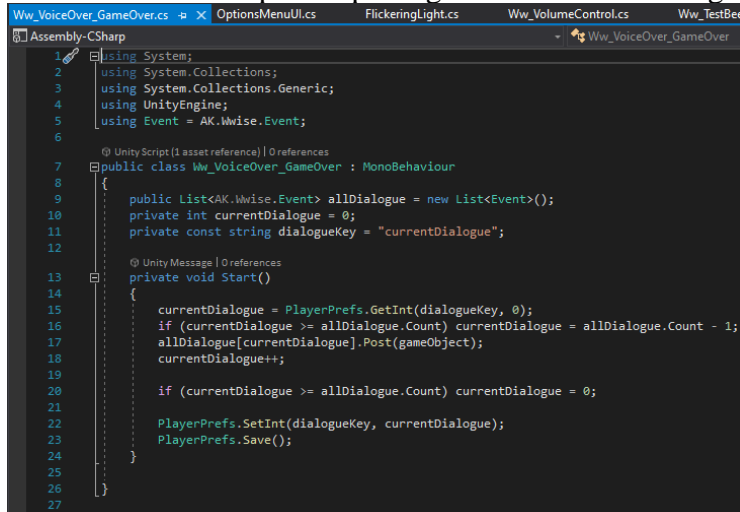
Noget af den kritik vi havde fået for vores MGP1 spil var hvorvidt, vi kunne gøre det mere interessant at dø i spillet. Idet spillet var en endless runner, ville det ofte forekomme at spilleren ville dø. Indtil videre havde vi en game over screen, hvor spilleren kunne se deres distance og deres hidtidige rekord distance. I MGP2 forsøgte vi at arbejde med hvordan vi kunne gøre game over mere interessant for spilleren. Med udgangspunkt i fortællingen om James, ønskede vi at game over screen skulle føles for spilleren som om de vågner fra mareridtet et kort øjeblik og hører deres psykiater fortælle dem at de skal gå længere ind i deres sind.

For at kunne opnå denne game over screen blev der skrevet 8 voicelines, der skulle kunne blive afspillet hver gang spilleren gik til game over screen. Jeg lavede en opstilling i lydstudiet, hvor jeg også havde optaget brølene til monsterlyden. Jeg kunne igen anvende den samme kondensator mikrofon, hertil havde jeg mit lydkort og to hovedtelefoner, både jeg og min voiceactor havde mulighed for at monitore. Fremgangsmåden med at arbejde med voiceline optagelser var, at voiceactor skulle læse samme linje op tre gange inden vi gik videre til en ny linje. På den måde var jeg sikker på, at hvis der var nogle problemer med en linje, at jeg i så fald ville kunne redigere den.

Udover det tekniske fandt jeg også, at det var vigtigt, at lade personen der skal indtale sin stemme slappe godt af og føle sig veltilpas i situationen. Jeg fandt også at jeg var i stand til at instruere voiceactor i forhold til kadence, lydstyrke, og hastighed af udtale der medvirkede til en højere grad af tydelighed og forståelighed af optagelserne.

I Implementering af optagelserne var det vigtigt at spilleren kun oplevede de otte voicelines efter hinanden, og at der således ikke var gentagelser af tidligere voicelines. Optagelserne måtte gerne komme tilfældigt, der måtte bare ikke forekomme gentagelser i

løbet af spillet. For at imødekomme dette var det nødvendigt at etablere en logik for afspilningen af optagelserne. For at imødekomme denne opgave snakkede jeg sammen med lead-programmøren på holdet om hvordan man potentielt kunne opnå et sådant afspilningssystem. Som følge af vores samtale udviklede han et C#-script (Figur 12), der kunne håndtere netop at afspilningen ville være tilfældig uden gentagelser.



```

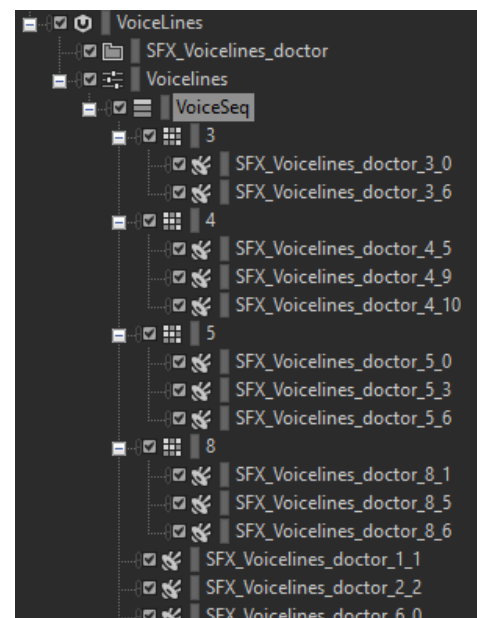
1 using System;
2 using System.Collections;
3 using System.Collections.Generic;
4 using UnityEngine;
5 using Event = AK.Wwise.Event;
6
7 [UnityScript (1 asset reference) | References]
8 public class Ww_VoiceOver_GameOver : MonoBehaviour
9 {
10     public List<AK.Wwise.Event> allDialog = new List<Event>();
11     private int currentDialog = 0;
12     private const string dialogueKey = "currentDialog";
13
14     [UnityMessage | References]
15     private void Start()
16     {
17         currentDialog = PlayerPrefs.GetInt(dialogueKey, 0);
18         if (currentDialog >= allDialog.Count) currentDialog = allDialog.Count - 1;
19         allDialog[currentDialog].Post(gameObject);
20         currentDialog++;
21
22         if (currentDialog >= allDialog.Count) currentDialog = 0;
23
24         PlayerPrefs.SetInt(dialogueKey, currentDialog);
25         PlayerPrefs.Save();
26     }
27 }

```

Figur 12 Lead-programmers Script til afvikling af voicelines

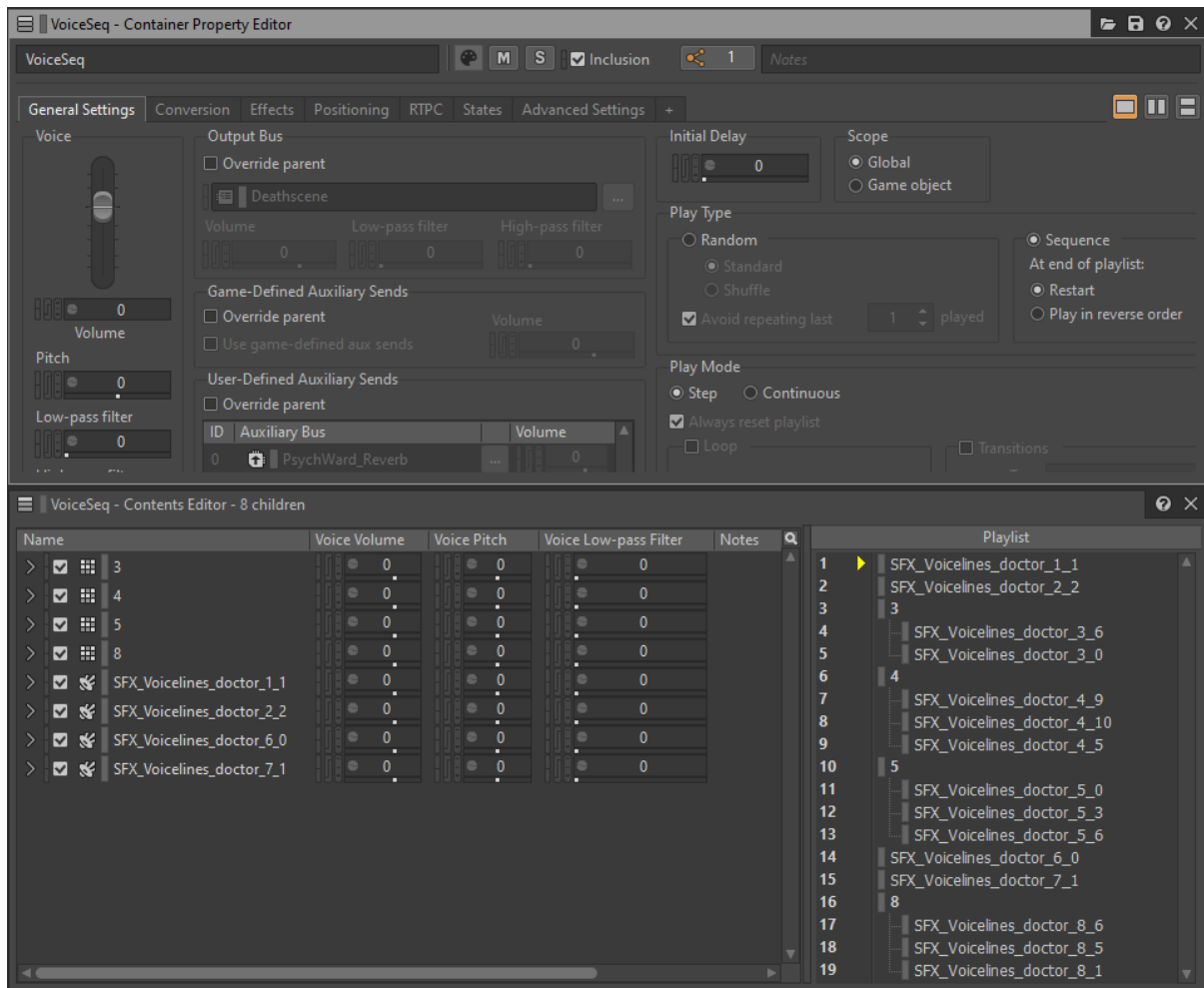
Min lead-programmørs script ville køre hver gang Game Over Unity-scenen ville blive loadet i spillet. Så snart scriptet compiler ville de eksekverer Start() funktionen. Når funktionen havde udvalgt et lydklip der ikke havde været afspillet endnu, ville det blive postet til Wwise via linjen `allDialog[currentDialog].Post(gameObject);`. Herefter ville den afspillede voiceline gøres utilgængelig førend alle andre voicelines havde været afspillet 1 gang. Koden min programmør havde skrevet ville tage udgangspunkt i et array som er defineret i linjen `public List<AK.Wwise.Event> allDialog = new List<Event>();`. Idet arrayet var public, ville det blive vist i Unity som et GUI i "inspectoren". Her kunne jeg indsætte Wwise events der ville blive anvendt i scriptet.

Efter at min lead-programmørs script var implementeret i forhold til at få afviklet voicelines, fandt jeg en metode hvor jeg i Wwise alene ville kunne opnå det samme resultat som det C#-script min lead-programmør havde lavet. I stedet for at lave et C#-Script for at håndtere logikken, kunne jeg have anvendt en Wwise sequence-container (Figur 13). I Wwise lavede jeg en workunit der hed "Voicelines" og under denne en actor-mixer med samme navn der var forbundet til SFX mixer bussen. Hierarkisk under voicelines actor-mixer indsatte jeg en Sequence-container med navnet "VoiceSeq". Under denne havde jeg uploadet de voicelines jeg ønskede at anvende. Hvis jeg havde mere end en optagelse af en voiceline, ville jeg arrangere dem under en Wwise random-container. Dette gjorde jeg med voicelines 3, 4, 5, og 8. I "VoiceSeq"-sequence-containeren (Figur 14) kunne jeg så opsætte en playliste som det ses i VoiceSeq – Contents Editor. For hver gang



Figur 13 Wwise Sequence Container

VoiceSeq belv kaldt i et script i Unity ville den rykke en position videre playlisten, og når den havde nået den sidste voiceline, starte forfra i sekvensen.



Figur 14 Wwise VoiceSeq Sequence container og playliste der henviser til voiceline samples

Med andre ord må det konstateres, at når det kommer til at konstruere den bagvedliggende logik for afvikling af lyden i Unity, er der ofte mere end én måde at gøre det på. Jeg kunne have valgt at gå videre med løsningen hvori logikken beroede sig i Wwise's objekter. Dog fandt jeg at min programmørs løsning var mere fleksibel, end den jeg kunne have opnået i Wwise. Selvom lyden aldrig ville have gentaget sig førend de alle var blevet afspillet, så var metoden i Wwise ikke i stand til at springe i rækkefølgen som min lead-programmørs script var i stand til. Det er sandsynligt, at der findes endnu flere måde dette problem kunne have været løst på, om det så ville have været gennem mere effektiv kode eller alternative objekter i Wwise. Dette eksempel på at der ofte er flere måde at tilgå problemet, er også med til at understrege vigtigheden af at samarbejde med de øvrige fagligheder. I dette tilfælde, særligt en programmør.

## Evaluerings af Mini game Projekterne

Det kan diskuteres hvorvidt MGP2's øgede fokus på fortællingen havde en positiv indvirkning på udviklingen af lyddesignet til det nye monster. Kunne det tænkes at lyddesignet for monsteret var blevet for komplekst? Jeg vil mene, at selvom lyddesignet for monsteret gav mening fra et fortællingssynspunkt, så må det være svært for at en spiller at få



tilstedeværelsen af monstret til at give mening. Ligeledes kan hensigtsmæssigheden af implementering af voicelines i spillets gameover-screen diskuteres i forhold til hvorvidt den opnår den ønskede effekt hos spilleren. Gør disse optagelser det faktisk sjovere at dø? Hvad kunne vi have gjort for faktisk at komme tættere på denne målsætning?

I det foregående afsnit har jeg redegjort for MGP1 og MGP2 gameplay og narrativ, samt givet et indblik i udvalgte designproblemer jeg har arbejdet med i udviklingen. Fra DADIU projektets side var hensigten med at vi skulle udvikle de to minigames tydelig. Vi havde som hold forstået, hvad det vil sige at samarbejde omkring udviklingen af et computerspil og havde fået en forståelse af hinandens roller, og hvad vi som hold var i stand til at udføre. Ligeledes havde vi fået opstillet en form for ledelses-hierarki hvorigennem mange af de afgørende, vigtige beslutninger kunne tages på et ledelses-niveau, fremfor at alle på hele holdet skulle inddrages hver gang.

For mit eget vedkommende som lyddesigner følte jeg også at jeg igennem udviklingen af de to spil havde fået en bedre forståelse for den generelle arbejdsgang, i forhold til at udføre lyddesign og implementere lyddesignet ind i "game engine". I forhold til lyddesign var jeg blevet opmærksom på at i starten af spiludviklingen, når man får skabt sig et overblik over hvilke lyde, der skal være i spillet, det giver god mening at få besluttet hvor mange af disse lyde, man selv kan udføre med en optager i et rum, og hvor mange af lydene man må finde i tilgængelige lyd-biblioteker. Noget af det jeg igennem de to minigames havde savnet, var mere håndgribelige metoder og teknikker i forhold til lyddesign-planlægning. Specifikt noget der ville give mig et bedre overblik over flere potentielle muligheder for et givent lyddesign.

Da vi snart gik ind i uge 44 hvor produktionen af vores graduation game ville begynde, blev vi opfordret til at overveje hvad vi gerne ville prøve kræfter med og lære i udviklingen af vores sidste spil. Særligt i forhold til soundscape ønskede jeg i graduation game at lærer mere om AK.Ambient og Ak.Room objekterne i Wwise og at finde ud af hvordan disse kunne anvendes i et virtuelt miljø, baseret på procedural generation principper. Hertil synes jeg også at det kunne være spændende at arbejde med dynamiske vejrsystemer i forhold til udviklingen af soundscapes. Jeg var også interesseret i at arbejde med mere intrikat lyddesign til mere komplekse objekter, altså våbenlyde eller bil lyde. Med andre ord, game objekter der havde en vis kompleksitet, der kunne understøttes med lyddesign, som var mere end bare én lyd, men kunne bestå af flere lyde, der ville blive afspillet efter hinanden for at understrege artefaktets f.eks. mekaniske kompleksitet. Jeg var også interesseret i at optage flere foley lyde til anvendelse i spillet.

Kort tid før vi påbegyndte produktionen af graduation spillet, fik vi tilknyttet en komponist fra Esbjergs elektroniske musikkonservatorie. Til orienteringsmødet fik vi at vide at det var op til det enkelte hold og dets lyddesigner, at afgøre hvordan man ønskede at udfører samarbejdet mellem komponist og lyddesigner. Jeg tænkte, at det ville være spændende at arbejde med relationen mellem et adaptivt musik-stykke, der ville reagere på hændelser i spillet således at musikken potentielt kunne fungere som en yderligere understregning af vigtige aspekter i spillet.

## Graduation Game: Ascent Trial of the Mountain



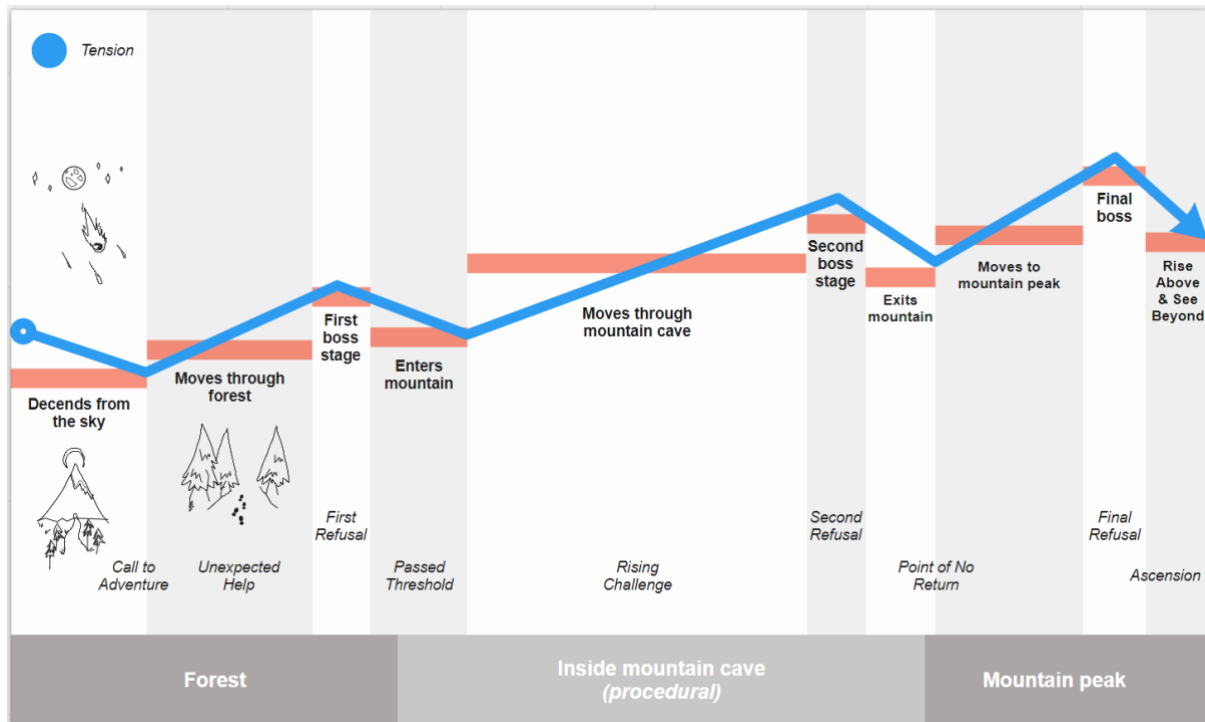
Figur 15 Bjerget fra Ascent Trial of the Mountain

Vores graduation spil starter med at spilleren ser en meteor styrte ned i en grotte. Her møder spilleren sin avatar; en mytisk sjæleskikkelse med en maske der minder om en ugles ansigt. Så snart sjælen kommer ud af grotten ses et højt bjerg tårne sig over landskabet, og på toppen ses noget magisk lys (Figur 15). Sjælen må kæmpe sig vej gennem bjergets indre grotter og overvinde de misundelige maskeløse for at komme igennem "The Trial of the Mountain". Spillerens mål er at nå bjergets top og overvinde bjergets sidste udfordringen før end sjælen kan opnå retten til at komme tilbage i himlen.

Spilleren er udstyret med et sværd til at kæmpe mod dets fjender. På vej op igennem bjerget vil spilleren, ud over at kæmpe mod fjender opdage stentavler med bidder af historien, samt tempel ruiner hvor der er magiske runer der kan give spilleren en fordel i sin søgen mod toppen af bjerget.

Når spilleren nedkæmper sine fjender, opsuger han deres sjæle. Disse kan senere anvendes når spilleren bruger "Soul charge" der giver spilleren mere liv. Hvis spiller mister alt sit liv, sendes spilleren tilbage til begyndelsen af spillet og må da atter kæmpe sig vej op gennem bjerget.

## Produktionen af Graduation Game



Figur 16 Grafisk model for handlingen af Ascent Trial of the Mountain

Udgangspunktet for produktionen af spillet var denne grafiske model (Figur 16), ud fra hvilken spillets overordnede handlingsforløb kan læses som en klassisk berettermodel. Ud fra denne model kunne jeg fastsætte, at der i alt skulle laves tre soundscapes; forest, cave, og mountain peak. I starten og i slutningen af spillet skulle der udvikles lyde til cutscenes. Ligeledes skulle der udvikles lyddesign til spillerens karakter, samt de forskellige fjender og bosses.

Efter lead-mødet hvor vi havde diskuteret spil-direktørens vision, havde jeg mit første møde med vores tilknyttede komponist. På dette møde fremlagde jeg for komponisten efter bedste evne hvad spil-direktørens vision gik ud på, og hvilke potentialer jeg umiddelbart så for musikken. Min komponist var generelt mest interesseret i hvilken overordnet stemning musikken skulle have. Ifølge spil-direktøren skulle musikken give en følelse af eventyr, nysgerrighed, spænding, gerne ved brug af instrumenter som fløjter og trommer. I forhold til implementering af musikken i Wwise, får jeg at vide at min komponist tidligere i sit forløb har prøvet kræfter med Wwise, og at han er meget interesseret i at lære mere om at anvende programmet. Jeg foreslår komponisten at se Wwise 201 guide videoer om Wwise indbyggede interaktive musik system. I forhold til spillets combat-aspekt ønsker jeg naturligvis noget musik der kan understøtte at spilleren er i sværdkamp, og snakker med komponisten om at arbejde med musik i flere forskellige intensitets niveauer.

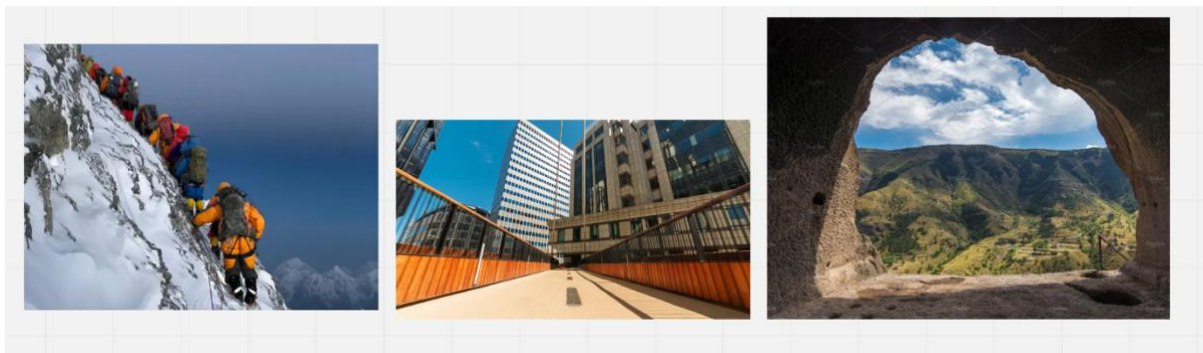
## Soundscape lyddesign og implementering

Fra de tidligere minigames havde jeg erfaret, at når det kom til udviklingen af soundscapes gjaldt det om hurtigt at få lavet en prototype på et soundscapes, der så kunne ændres i løbet af processen. En prototype af et soundscape behøver ikke at indeholde et ekstremt detaljeret, virkelighedstro soundscape, men i stedet indeholde de vigtigste, mest fremtonede aspekter. Min metode til at udvikle disse soundscapes starter med en erfarings og netnografisk tilgang. I forhold til f.eks at jeg skulle udvikle et soundscape til en skov, så kan jeg genkalde mig lyde

jeg har hørt når jeg selv har været i en skov. Hvislende blade og grene i blæst, fuglekvidren, og lyden af noget der rusker i et buskads. Jeg supplerer så mine erfarede lyde fra min hukommelse med optagelser jeg kan finde på Youtube. Der findes mange videoer der iscenesætter natur og skov-lyde i lange videoer til afslapningslytning. Derudover findes der også videoer fra andre computerspil, som også anvender diverse natur og skovlyde. Min tilgang til at anvende disse videoer var at udføre en soundscape analyse i forhold til at identificere hvilke lyde jeg kunne høre i optagelserne. Det var også her jeg opdagede hvordan mange af de 10 timer lange videoer med naturlyde ofte bestod af et meget kortere loop, idet jeg kunne genkende lyde der gentagne gange blev afspillet. Hvis man ikke lyttede til det gennem en soundscape linse, ville man sandsynligvis ikke opdage noget, men idet jeg havde et klart analytisk formål og tilgang til videoerne, var det meget tydeligt.

Min første prototype af et soundscape til "forest" bestod af en let vindlyd i træer, fuglekvidren, og fuglevinger der letter. Efter at jeg havde vist denne første iteration til min spil-direktør ville han gerne tilføje lyden af knagene træer. Dette gav mig en association til de levende træer i Lord of the Rings i Fangorn-skoven, der i Peter Jacksons film har mange af disse knagene træ-lyde. Denne lyd er et godt eksempel på en lyd jeg sjældent har oplevet når jeg selv har gået rundt i en skov, men som i computerspil og fiktions-film er fremtrædende idet den gør skoven og dens træer mere levende. For at imødekomme min spil-direktørs vision med lyden af skoven, fandt jeg en masse kunstigt fremstillede foley lyde af træ der blev brækket eller tvistet, og som var optaget med gode mikrofoner.

Denne nye version af soundscapet sendte jeg ud på vores holds fælles kommunikationskanal, således at vores kunstner-gruppe kunne lytte til mit soundscape og anvende det som inspiration i deres arbejde. Jeg anvendte samme metode for de to øvrige soundscapes, Cave og Mountain Peak. Dog var mit eget erfaringsgrundlag med disse to ikke så velkendt som med skoven, hvilket betød at jeg måtte tage mere udgangspunkt i min netnografiske soundscape analyse. Særligt i forhold til "cave"-soundscapet hvor jeg havde indsat en form for drypstenshule lyde med vand der drypper, oplevede jeg at kunstnere lod sig inspirere af dette og visuelt understøttede mine lyde ved at indsætte 3D modeller af



Figur 17 Billeder anvendt af QA/UR afdelingen til at teste mine soundscapes

drypsten, samt små vandpytter inde i grotterne.

De første soundscape prototyper af de tre miljøer var meget grove og kunne ikke loope, da jeg ønskede at finjustere soundscapes førend de blev implementeret i Wwise. Jeg sendte derfor disse prototyper til kvalitets tests ved QA/UR afdelingen. Her bad jeg dem specifikt at undersøge om de tre soundscapes fik folk til at føle, at de var til stede i de miljøer som de forskellige soundscapes skulle repræsentere, og hvilke følelser de tre soundscapes derudover vækkede i folk, der hørte dem. Dette resulterede i en rapport hvori jeg fik indblik i hvad QA/UR havde testet og deres fremgangsmåde, og hvilke resultater de var nået frem til.

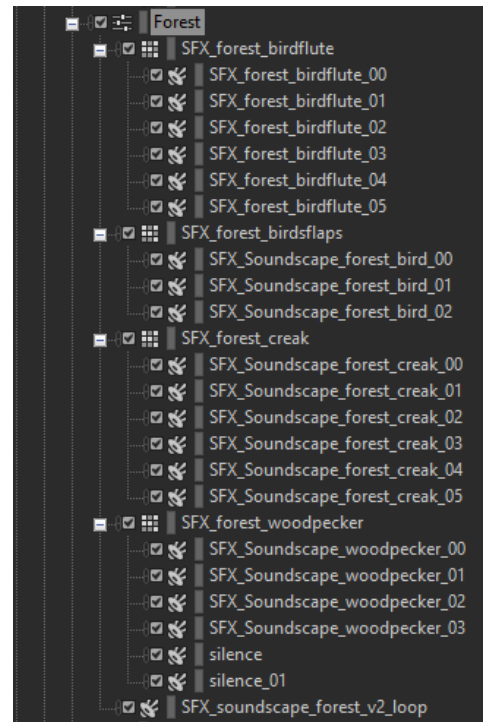
Fremgangsmåden for deres tests var at afspille mine soundscapes i kombination med at vise deltagerne i deres tests billeder af virkelige miljøer (Figur 17).

I deres tests var de nået frem til at alle mine soundscapes prototyper overordnet fungerede ønskværdigt, bortset fra mit soundscape til "mountain peak", her skulle workshopdeltagerne have opfattet lyden af vindsus, som have spatialt reflektive kvaliteter som om vinden kom fra en grotte, og ikke som om vinden var ude i det åbne. Som resultat af denne feedback forsøgte jeg mig med nogle alternative, kolde vindlyde.

Da jeg nu havde tre velfungerende soundscape prototyper, kunne jeg forberede dem til at blive implementeret i Wwise. Implementering af de forskellige soundscapes i Wwise bestod i at opdele dem i en generel baggrundslyd der kunne loope, og nogle tilfældigt kaldte lyde. I Forest soundscapet havde jeg et loop "SFX\_soundscape\_forest\_v2\_loop" der blev suppleret af fire random containere med forskellige former for fugle fløjt, fuglevinger, eller lyden af en hakspætte.

De fire random container havde parametre i forhold til hvor ofte en lyd ville blive afspillet, samt en indbygget sandsynligheds interpolering der medvirkede til at gøre afspilningerne yderligere tilfældige. Ligeledes var alle random-containers og baggrundsloops organiseret hierarkisk under en actor-mixer "Forest" hvori soundscapet var specificeret som en 3D positioneret lyd (Figur 18). På samme måde som i MGP-spillene havde soundscapet også en attenuation kurve, der determinerede lydens aftagen i volumen og frekvens afhængig af distancen lytteren er fra lydkilden.

I forhold til integrationen af forest soundscapet ind i unity projektet havde jeg igen snakket med lead-programmer i forhold til at jeg ønskede at anvende 3D spatialiseret lyd med henblik på at få skovens lyde til at virke mere dynamiske for spilleren. Min Lead programmer foreslog, at vi kunne kombinere det script der genererede skoven med at generere lydkilderne til skovens soundscape, så de dækkede området spilleren ville bevæge sig indenfor. Igennem DADIU-forløbet havde vi prøvet kræfter med procedural generation i forhold til spillets levels, og generering af træer i skoven var en del af denne udfordring. Jeg forsøgte først at tillægge lyden således at hvert træ i skoven ville fungere som en emitter af lyden, men dette viste sig at kræve for meget CPU i forhold til hvad spillet kunne klare. I stedet fik vi forest generatoren til at generere usynlige game objekter med lydkilde emitters på med en vis afstand fra hinanden. På billedet ses skoven ovenfra og de grønne kugler markerer attenuation kurverne visuelt i game-world (Figur 19).



Figur 18 Forest Soundscapet organiseret  
Hierarkisk i Wwise

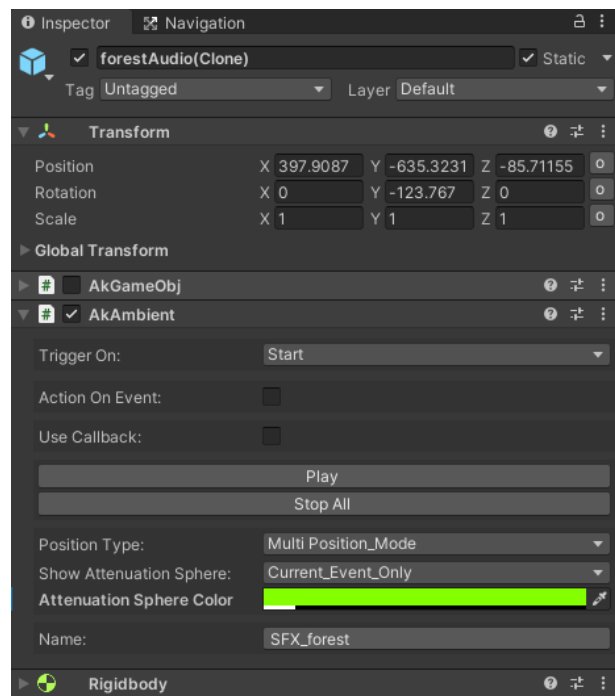




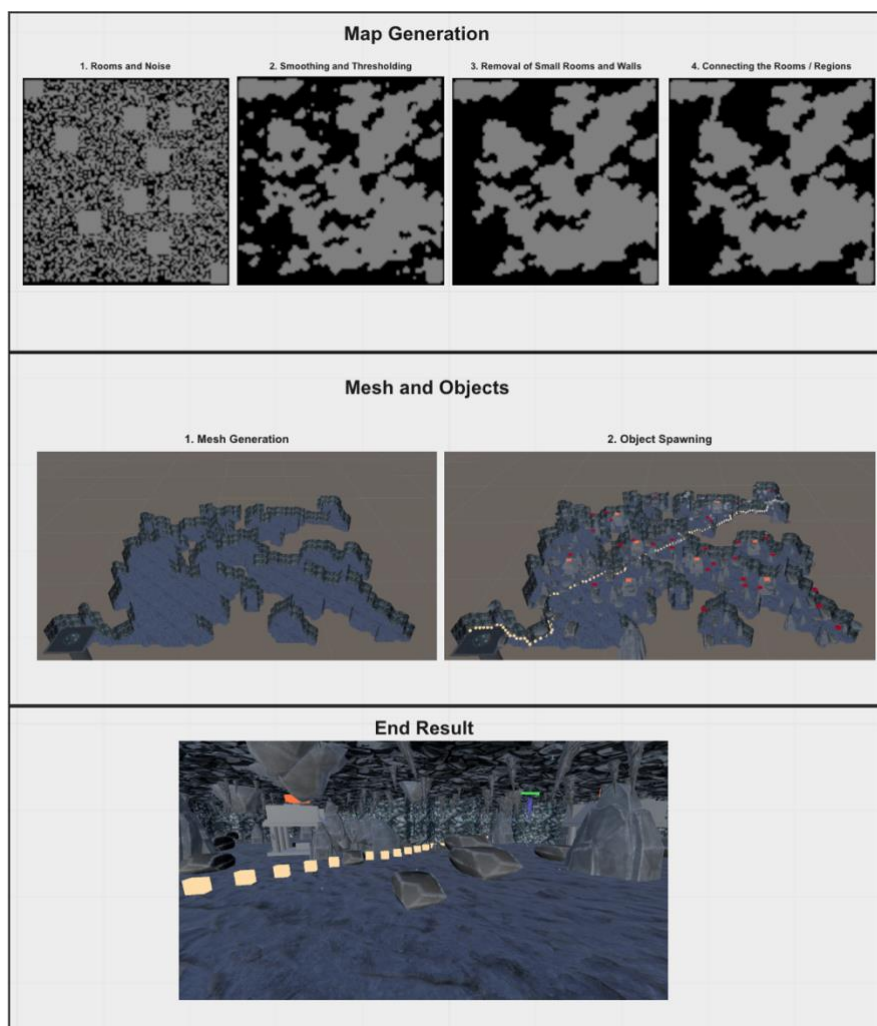
Figur 19 Implementering af Forest soundscapet, de grønne kugler illustrer attenuation for soundscapet

Jeg fandt ud af at jeg kunne udnytte Ak.Ambient i Wwise unity integration i forhold til at spare yderligere på CPU. Normalt hvis man normalt integrerer en lyd i Unity, der skal findes flere steder, kræver det at lyden afspilles alle stederne. I tilfældet med skovens soundscape ville det betyde otte voices, der alle ville trække på CPUen af spillet. Ved at udnytte Ak.Ambient og dens MultipositionMode kan disse otte voices reduceres til én voice der så stadig kan afspille fra de otte positioner det er indsat i (Figur 20).

Denne implementeringsmetode anvendte jeg også i forhold til at integrerer "Cave" soundscapet i dets proceduralt genererede grotter. I grotterne kan denne teknik siges at være endnu mere afgørende, idet der før spilleren kommer til åbningen af grotten, ikke findes en grotte. For hver gang en spiller ville spille spillet igennem, ville grotterne blive genereret forskelligt. Dette var en udfordring i forhold til at indsætte lyd-gameobjekter, idet jeg aldrig kunne regne med hvordan et levels layout ville komme til at være. Det blev derfor afgørende at jeg var i stand til at kunne læse og forstå koden og arbejde sammen med programmørerne for at finde ud af hvordan procedural generation fungerede i forhold til grotterne, og hvilke objekter der ville blive genereret inde i grotterne.



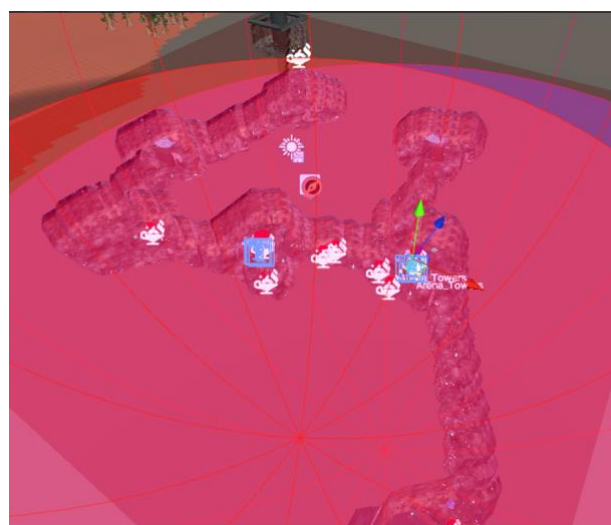
Figur 20 Wwise Multiposition mode i et AkAmbient script



Figur 21 Procedural Generation udviklingen af grotterne inde i bjerget

I ovenstående figur er vist hvordan programmører anvendte perlin noise og imageprocessing til at generere grotterne (Figur 21). Der ville i genereringen af levels forekomme rum i grotterne. I disse rum kunne forskellige objekter tilfældigt blive placeret ud fra nogle algoritmer programmørerne havde determineret. For hver af de grotter som spilleren skulle igennem, var det fastsat, at der altid ville forekomme runetempler, hvor spilleren ville have mulighed for at få boostet aspekter af deres gameplay. Idet jeg vidste at disse altid ville være i grotterne kunne jeg anvende dem som emitters af cave soundscapet (Figur 22).

For derudover at gøre grotterne mere troværdige havde jeg i grotterne et usynligt game-object, der fungerede som trigger i forhold til at aktivere eller deaktivere en sendFX i Wwise, hvorpå lyde der ville blive emittet inde i grotterne ville få



Figur 22 indsat emitters af cave soundscapet i de proceduralt genererede grotter

tillagt en rumklang. På den måde kunne jeg give spilleren fornemmelsen af at være i grotten spatialt, i forhold til at iscenesætte at lydende blev reflekteret af grottens vægge, præcis som det også sker i den fysisk-virkelige verden.

Jeg kunne forestille mig at det at implementere i lyde i en proceduralt genereret game world stiller nogle helt andre krav til lyddesignerens kompetencer end hvis man implementerer lyde i et statisk miljø, der er mere fastsat. I mit tilfælde vil der, i princippet, altid være en sandsynlighed for, at et game objekt der kan forekomme i de genererede grotter, ikke bliver genereret. Som lyddesigner ville det betyde at mit soundscape ikke ville blive hørt. Jeg er sikker på, at der er andre måder, hvorpå jeg kunne have integreret soundscape, hvor jeg ville være sikker på, at det ville blive afspillet hver gang, men jeg ønskede at arbejde med 3D spatiale lyde.

## Stenelevatoren Lyddesign



Figur 23 Stenelevatoren



Figur 24 Morter optagelses setup

Spilleren transporteres vha. en stenelevator mellem de forskellige grotter inde i bjerget. Så snart Spilleren træder ind i stenelevatoren fungerer denne som en måde at igangsætte genereringen af den næste grotte.

Stenelevatoren er et perfekt eksempel på et objekt hvor lyden er altafgørende for at spilleren kan indleve sig i spillet.

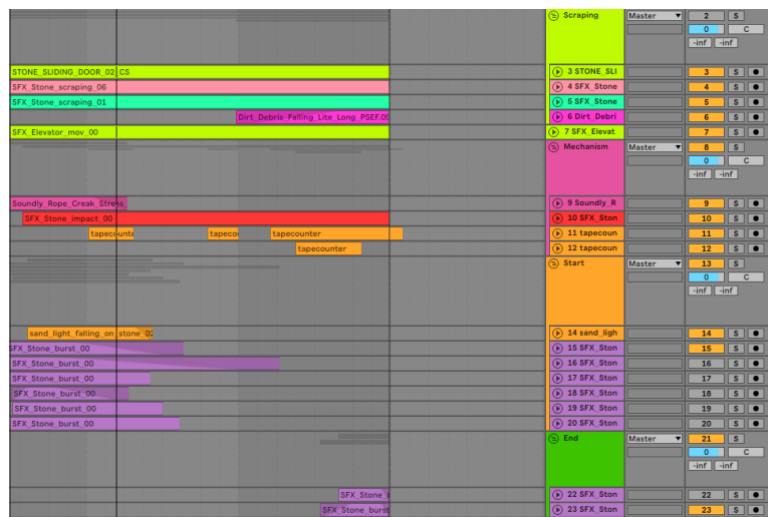
Stenelevatorens består af en 3D cylinder, der skubbes op igennem et rør (Figur 23). Uden et lyddesign virker stenelevatoren tyngdeløs og kunstig.

I udviklingen af lyddesignet for stenelevatoren var jeg inspireret af en scene fra filmen "Asterix og Obelix: Mission Cleopatra" hvor der inde i en pyramide findes en fungerende stenelevator. Mekanikken bag denne elevator var reb der udlignes af mekaniske tandhjul. Lyddesignet af stenelevatoren skulle ikke alene understrege elevatorens materialitet, men også dens bagvedliggende mekanik. Således kan lyddesignet af elevatoren inddeles i lag af hhv. illusorisk mekanik og materialitet.



Det illusorisk mekaniske bestod af lyde af tandhjul fra urværker, samt lyde af reb der strammes. I forhold til lyden af det materielle havde jeg lånt en morter af spil-direktøren som jeg havde påsat en kontaktmikrofon og en kondensator mikrofon (Figur 24). Ved at anvende de to mikrofoner kunne jeg mixe imellem vibrationerne fra morteren selv og morten i rummet. Ligeledes prøvede jeg forskellige teknikker af at lade pistilen falde ned i morteren, eller lade den rulle på overfladen.

I efterbehandlingen kunne jeg anvende de bedste optagelser og udstrække dem, og sætte dem i flere lag oven i hinanden for at give lyden noget mere dynamik (Figur 25). Det var vigtigt, at lyden af sten der skraber mod sten kunne loope, så uanset hvor lang elevatoren



Figur 25 Stenelevator lyddesign eksempel på layering

For at gøre optagelsen til et loop byttes der nu om på rækkefølgen, så slutningen på optagelsen spiller først. Dette betyder at når starten på optagelsen loopes tilbage til slutningen på optagelsen, så er overgangen naturlig. For så at gøre overgangen mellem slutningen og starten på optagelsen usynlig, anvendes en logaritmisk crossfading der bevarer volumen konstant og gør overgangen sømløs. Denne procedurer gentages for alle de sten skrabbings loops der bliver lagt ovenpå hinanden i lyddesignet.

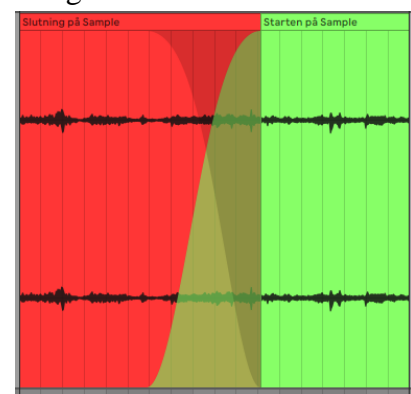
Helheden af lyddesignet består således af stenskrabbingsloops kombineret med mekaniske lyde af tandhjul og reb der strammes. Dertil kommer tilføjelsen af en startlyd og en slutlyd på elevatoren. Disse er med til at understrege elevatorens tyngde og hvordan mekanismen påbegynder og sten skrabbningen begynder rent æstetisk. Derudover fungerer de som en form for signifikationer der informerer spilleren om at elevatoren kører, og at elevatoren stopper når den har nået toppen.

## Sværdkamps lyddesign

Udover de udfordringer der var ved at designe lyd til en proceduralt genereret spil verden, var den anden store udfordring for vores hold at udvikle et sværdkamps-system der fungerede og gav spilleren en god oplevelse. Der krævede samarbejde mellem alle de forskellige fagligheder på holdet. Animationerne af karakterens forskellige sværdkamps-positioner blev

var i spillets "gameworld" eller dens hastighed, så ville lyddesignet altid tilpasse sig.

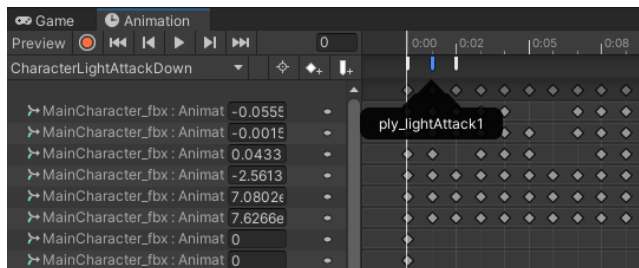
Metoden for at lave et loop af stenskrabningen bestod i først at udvælge et lydclip fra optagelserne. Dernæst vælges et tidspunkt i optagelsen hvor stenskrabningen virker konstant uden for mange svingninger i tonehøjde eller rytmisk forstyrrende elementer. Her opdeles lyden i to dele. I figuren har jeg illustreret dette ved at farve starten af optagelsen grøn, og farve slutningen af optagelsen rød (Figur 26).



Figur 26 Stenskrabnings loop eksempel

udført af vores kunstnere og sendt videre til programmørerne der integrerede disse animationer ind i Unity projektet. Game og level designerne testede hvordan sværdkampen føltes for spilleren og justerede parametre for skade, liv, hvor ofte man kunne slå etc. En af de største udfordringer vi havde, var at sikre os at sværdkampen føltes engagerende for spilleren. Det var vigtigt at sværdkamp ikke føltes som om man bare stod og viftede over en fjende med sit sværd, men at det i stedet havde en følelse af tyngde, og at man kunne mærke hvordan hvert et slag ramte. Med andre ord var det sværeste ved at designe sværdkampen at udtrykke de kausale effekter, som vi i vores virkelige verden tager for givet. I den virtuelle verden eksisterer der kun kausale forhold, hvis de er blevet udtrykket og designet algoritmisk.

For animationerne betød det at udover at lave sværdkamps-animationer, skulle der også laves animationer for hvordan fjenden blev ramt og reagerede kropsligt på dette. Programmørerne argumenterede også for at anvende et system der gjorde det muligt at justerer på kadencen af sværdkampsbevægelserne. Altså, hvordan man i et sværd-hug vil gøre noget af bevægelsen langsommere og noget af den hurtigere med henblik på at give en fornemmelse af ekstra kraft for hugget. Som lyddesigner var jeg naturligvis interesseret i at understøtte alt dette med et lyddesign, der skulle være tilpasset præcist til animationerne.



Figur 27 Unity Animation event triggerer en funktion `ply_lightAttack1` Der kalder en lyd i Wwise

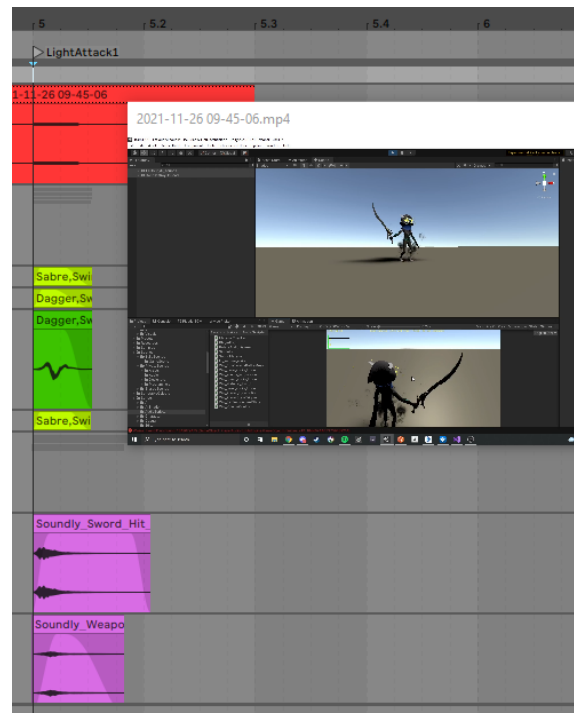
Den største udfordring var dog at jeg ud fra den designproces vi havde valgt, var afhængig af at de øvrige afdelinger færdiggjorde deres elementer, før at jeg kunne begynde at lave lyddesign til animationerne. Dette skyldes at programmørerne havde valgt at arbejde med "easings". Den programmør der havde udviklet scriptet til at kunne udføre disse easings blev en bottleneck for den overordnede

udvikling af sværdkamps-aspekterne. Selvom alle animationerne var produceret af kunstnere og det meste var testet og justeret af game og level, så kunne jeg ikke tilpasse mit lyddesign førend easing systemet var blevet udviklet og alle animationerne var tilpasset igennem det.

Da easing systemet endelig kom på plads og animationerne var blevet behandlet, kunne jeg påbegynde udviklingen af lyddesignet til sværdkamp. Spilleren havde mulighed for at kunne slå med sværdet på i alt fire måder. Der var to former for light attacks og to former for heavy attacks. Min metode for lyddesignet var at optage videoklip inde fra spillet, af figuren der slår, hvorpå jeg i starten af hver animation havde indsat et Unity animation event der ville kalde en beep lyd i Wwise. Dette ses illustreret i (Figur 27), hvor et animationsevent kalder en funktion `ply_lightattack1()`, der kalder et Wwise event, der sidst kalder en sværdlyd. På den måde kunne jeg se hvornår animationen startede og kunne derfra tilpasse mit lyddesign præcist til de bevægelser, jeg kunne se i videoklipet, hvor den korrekte easing hastighed for bevægelsen var inkluderet (Figur 28).



Videoerne med beeps kunne være blevet arbejdet med direkte i Ableton. Jeg anvendte beepet som startpunkt for mit lyddesign, og anvendte en lydpakke med lyden af sværd der klirrer og slås mod hinanden, samt sus af sværdet der bevæger sig gennem luften. Jeg lavede forskellige versioner af disse sværdkamps-lyde til de fire forskellige slag spilleren kan udføre med sværdet. Dernæst kunne jeg i Wwise erstatte beep-lyden med mine egne lyddesigns. I forhold til lyddesignet af stenelevatoren kan dette lyddesign ikke skaleres på samme måde. Hvis der laves ændringer i animationen, eller i den easing processering der kommer efter, da vil lyddesignet atter skulle tilpasses manuelt og en ny version af lyden implementeres i unity. Fordelen ved at anvende den teknik var imidlertid, at lyddesignet var skræddersyet til animationerne og den tidslighed der var i spillet, hvilket fint var med til at understøtte det overordnede mål om en følelse af kausalitet og at sværdets påvirkning kunne mærkes af spilleren selv.



Figur 28 Easing animation optagelse anvendt i Ableton til lyddesign

## Konklusion / Speciale forberedende

Jeg har i denne rapport beskrevet de tre spil jeg i løbet af det projektorienterede forløb på DADIU har været med til at udvikle. Derudover har jeg udvalgt forskellige faglige problemstillinger jeg blev stillet overfor i de tre forskellige udviklingsprocesser, med henblik på at disse fremhæver et så bredt udvalg af de teknikker og metoder jeg har anvendt for at imødekomme disse problemstillinger. Ligeledes har jeg understreget hvilke aspekter fra de to MGP projekter jeg videre har kunnet anvende i forhold til graduation spillet.

Konkret har jeg demonstreret mine faglige kompetencer som lyddesigner gennem lyddesign teknikker af lydclip i lag, FX-processering af sammensatte lyde, optagerteknikker i forhold til voicelines, samt lydeffekter. Dertil kommer de mere interdisciplinære færdigheder i forhold til at kunne forstå og udvikle med kode i forhold til Wwise, og Unity integrering af lyd i spillet. Dertil kommer den reflektive lyddesign tilgang i forhold til hvordan disse lyde fungerer for spilleren i spillets kontekst, hensigtsmæssigheden af de designede lyde, og de associative processer i forhold til inspiration og referencer til lyddesignet.

Noget der gik igen i alle tre spil var hvordan jeg kunne anvende min faglighed i forhold til at udvikle soundscapes. I mine undersøgelser af forskellige miljøer der skulle genskabes i den virtuelle verden, kunne jeg anvende soundscape analytiske aspekter i forhold til f.eks. at afdække hvilke typer af lyde der findes i grotter, og hvilke lyde der forekommer oftere end andre, samt forholde mig til hvilke lyde der er essentielle i forhold til at rekonstruere en grottes ambience.

I forhold til hvad man kunne tage videre fra DADIU-projektet til et potentielt specialeprojekt, tænker jeg at det kunne være interessant at fordybe sig mere i hvilken indflydelse soundscapes har i forhold til spillerens oplevelse. Man kan undersøge hvordan soundscapes kan tænkes som værende funktionelle som mere end blot lydbaggrund, men

også besidde kvaliteter i forhold til f.eks. feedback og signifiers for spilleren. Et opgavespørgsmål kunne være; Hvordan udvikler man soundscapes på en reflekteret måde i forhold til at understøtter feedback af andre modaliteter for spilleren? Tænker klart at man for at kunne undersøge et sådant spørgsmål må have kendskab til forskellige måde hvorpå man kan tilgå soundscape analyse. Potentiel relevant litteratur indenfor dette område kunne bl.a. være Maurice Schafers "the tuning of the world" (1977) med udgangspunkt i en generel bred definition for soundscapes. Herunder kunne man også tage udgangspunkt i nogle af hans lytte øvelser. Dertil kunne man anvende Gallagher's "Field recording and the sound of spaces " (2015). Her tænker jeg særligt i forhold til hans skelnen i forhold til hvordan soundscapes kan være repræsentationer eller performative. Endelig kunne man diskutere soundscapes i computerspil i forhold til det æstetiske dilemma som Barry Truax fremhæver i sen tekst "Sound, Listening and Place: The aesthetic dilemma" (2012).

For at sætte disse soundscapetekster i en mere computerspilspræget kontekst kunne man underbygge disse med f.eks. Stockburgers tekst "The game environment from an auditive perspective" (2003). Her diskuterer han hvilke potentialer der kunne være i soundscapes i forhold til f.eks. feedback for spilleren indenfor computerspil. Ligeledes kunne man her supplere med Grimshaw's tekst "Sound and Player Immersion in Digital Games" i forhold til hvordan soundscapes besidder potentielle indlevende kvaliteter der i en designproces bør tages højde for. Tænker også at man kunne Karen Collins tekst "Gameplay, Genre and the functions of game audio" ville være vigtig i forhold til at kunne stille soundscapet i relation til et spils genre og gameplay aspekter, idet man kunne forestille sig at soundscapets potentielle funktioner ville være afhængig af spillets genre og gameplay. Der burde således i en reflektiv udviklingsproces af soundscapet tages højde for dette.