

Muse Box

Projekt i kursen TNM113 – Procedurell Ijuddesign för användargränssnitt
Linköpings universitet

Adam Nilsson (adani731)
Johan Liu (johli066)

1. Inledning	3
1.1 Syfte	3
1.2 Frågeställningar	3
1.3 Verktyg och plattform	3
2. Teori	5
2.1 Sonifiering	5
2.2 Ljudsyntes och manipulation	5
3. Beskrivning, motivering och diskussion av den färdiga ljuddesignen	7
3.1 Ljuddesignen	7
3.2 Ljudeffekter och inspelning	7
3.3 Ljudens konstruktion	7
Komprimering	7
Toner och ackord	8
Bastrumma	8
Hi-hat	8
High tom och cowbell	8
Open Sound Control (OSC)	8
Mappning och ljudförändring	9
Tillgänglighet, användbarhet, och tillämpbarhet	9
Slutsats	10
Referenser	11

1. Inledning

1.1 Syfte

Detta projekt syftar till att skapa ett interaktivt och procedurellt genererat visuellt gränssnitt som ger återkoppling via ljud. Syftet är att utforska möjligheterna med sonifiering och att skapa en interaktiv upplevelse där olika ljud genereras och manipuleras baserat på inkommande data.

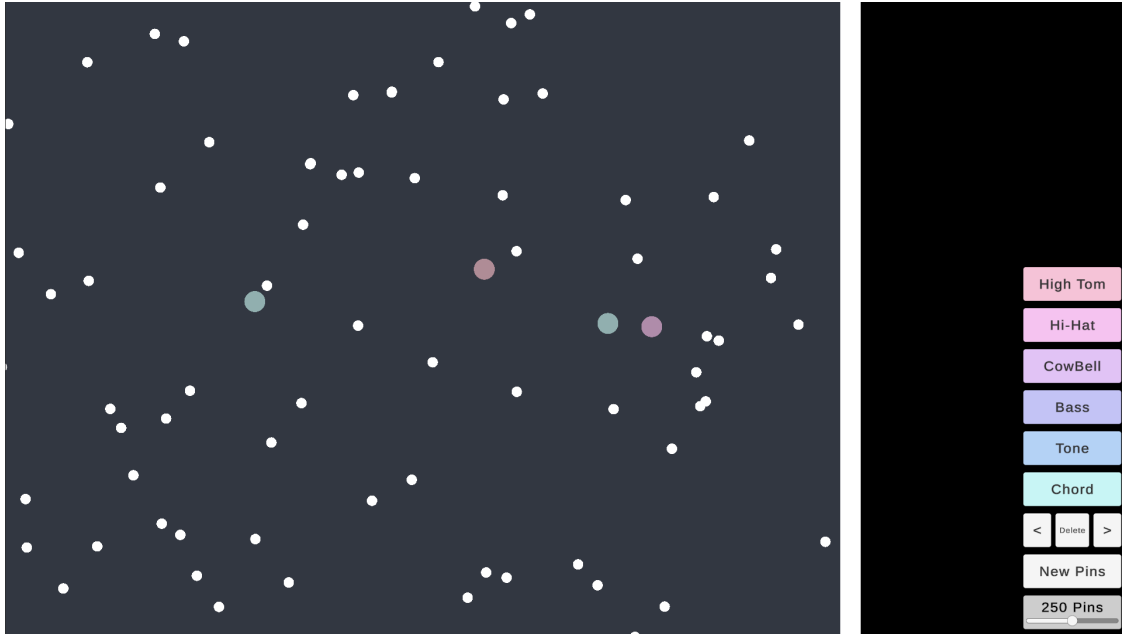
1.2 Frågeställningar

Frågeställningarna som undersöks inkluderar:

- Hur kan data effektivt mappas till ljudparametrar?
- Vilka ljudsynteser och filter kan användas för att skapa önskade ljudtexturer?
- Hur kan ljuddesignen säkerställas vara både funktionell och estetiskt tilltalande?

1.3 Verktyg och plattform

Projektet kombinerar användningen av både SuperCollider och Unity som program. SuperCollider ansvarar för allt ljudarbete medan Unity används för att skapa gränssnittet som ska styra ljudbilden. Gränssnittet är utformat som ett inhägnat område i 2D, med en rad knappar till höger om skärmen, som visas i **Figur 1** nedan. När användaren klickar på dessa knappar skapas så kallade "Synthbollar" som rör sig runt i inhägnaden och kolliderar med vita, statiska och cirkulära objekt, kallade "stift". Antalet stift kan justeras med ett nedre reglage och nya stift skapas genom att användaren trycker på knappen "New Pins". Ljud skapas i SuperCollider beroende på hur dessa bollar och stift interagerar, vilket resulterar i en ljudupplevelse som skulle kunna vara "musik".



Figur 1: Gränssnittet för projektet i Unity. De färgade cirkarna är synthbollar, de vita cirkarna är stift.

2. Teori

Ljuddesign involverar skapandet och manipuleringen av ljud för olika ändamål, inklusive musikproduktion, film, spel och interaktiva installationer. I detta projekt läggs fokus på sonifiering, där data omvandlas till ljud för att ge information på ett hörbart sätt. Ljuddesignen måste också ta hänsyn till perceptuella faktorer, såsom hur tonhöjd, rytm och timbre uppfattas.

Ljud kan användas för att skapa stämning och förmedla känslor, men också för att ge feedback och information i interaktiva system¹. I detta projekt används ljud för att skapa en immersiv upplevelse, samtidigt som användaren får information om systemets tillstånd och händelser genom ljudsignaler som sker vid specifika händelser i gränssnittet.

I projektet används flera tekniker för ljudsyntes, inklusive additiv/subtraktiv syntes och brusgenerering. Dessa tekniker gör det möjligt att skapa en bred palett av ljudtexturer som kan användas för att representera olika data och händelser.

2.1 Sonifiering

Sonifiering är processen att omvandla data till ljud för att kommunicera information på ett auditivt sätt. En av de stora fördelarna med sonifiering är att den kan utnyttja hörselsystemets förmåga att upptäcka förändringar och mönster i ljud, vilket kan vara svårt att uppfatta visuellt. Detta gör sonifiering särskilt användbart för övervakning och analys av komplexa data².

Ett centralt koncept inom sonifiering är mappning, där specifika dataelement kopplas till ljudparametrar såsom frekvens, amplitud och varaktighet. En väl genomförd mappning kan göra det möjligt att snabbt och intuitivt förstå data genom att lyssna på den. I detta projekt används OSC-meddelanden för att dynamiskt styra parametrar i syntharna, vilket gör det möjligt att realtidssonifiera data.

2.2 Ljudsyntes och manipulation

Ljudsyntes är processen att generera ljud från grunden genom elektroniska eller digitala metoder. Det finns flera typer av ljudsyntes, inklusive:

- **Additiv syntes:** Skapar ljud genom att addera enkla vågformer, vanligtvis sinusvågor, för att bygga upp komplexa ljud. Detta är användbart för att skapa harmoniskt rika toner och komplexa ljudtexturer³.
- **Subtraktiv syntes:** Involverar filtrering av en rik ljudkälla, såsom en sågtands- eller fyrkantsvåg, för att ta bort oönskade frekvenser och forma ljudet. Detta är en vanlig teknik i analog syntes och används för att skapa en mängd olika ljud, från bastrummor till lead-synthar⁴.
- **Brusgenerering:** Använder vitt, rosa eller andra typer av brus som grundljud, vilka sedan formas med filter och enveloper för att skapa ljud som hi-hats och cymbaler⁵.

Dessa tekniker tillämpades för att skapa de olika syntarna som representerar toner och percussiva element.

3. Beskrivning, motivering och diskussion av den färdiga ljuddesignen

3.1 Ljuddesignen

Ljuddesignen i projektet omfattar en rad synthar som genererar olika ljud, inklusive toner, ackord, bastrumma, hi-hat, high tom och cowbell. Dessa ljud valdes för dess simplicitet och direkta koppling till musik och snabba impulsrörelser, det vill säga de kollisioner som sker i gränssnittet med bollar och de cirkulära stiften. För att undvika att användaren blir överbelastad med långa och jobbiga ljud som smälter in i varandra, valdes korta och snabba ljud som på ett tydligt men ändå nyanserat sätt förmedlar om vad som sker i gränssnittet.

Ljuden valdes även baserat på deras musikaliska och perceptuella påverkan. Till exempel valdes en bastrumma för att skapa en stark rytmisk grund, medan toner och ackord används för att bygga mer melodiska och harmoniska strukturer. Att använda olika ljud ger upphov till en upplevelse som är både rikare och mer varierad.

3.2 Ljudeffekter och inspelning

All ljud skapades och manipulerades i SuperCollider. Syntharna genererar ljuden i realtid, vilket även ger flexibiliteten att förändra och anpassa ljuden dynamiskt. Inga ljud spelades in från externa ljudkällor utan det fokuserades mest på syntens och digital signalbehandling.

3.3 Ljudens konstruktion

Som det nämndes under avsnitt 2.2, så skapades ljuden genom olika syntesmetoder, såsom additiv och subtraktiv syntes. Sinusvågor, sågtandsvågor och brus tillämpades för att skapa olika ljudtexturer. För att modulera ljuden användes envelopes och lågpasfilter för att forma ljudens frekvensspektra och dynamik. Nedan följer mer detaljerade beskrivningar för varje del i SuperCollider-koden.

Komprimering

En *componder* (compressor + expander) definierades för att behandla den totala ljudsignalen. Den reducerar dynamikområdet genom att sänka volymen på starka ljud och höja volymen på svaga ljud, så att ljudbilden (soundscape) blir mer balanserad.

Detta är särskilt användbart i en interaktiv miljö där ljudstyrkan kan variera kraftigt beroende på användarens input. Genom att använda en compander säkerställs det att ljudbilden blir sammanhängande och inom intervallet $[-1, 1]$, vilket förhindrar distorsion. Detta förbättrar användarupplevelsen genom att skapa en jämnare och mer behaglig ljudmiljö.

Det skapades en synthdefinition som dynamiskt bearbetar inputljudet. Funktionen *Compander* används för att kontrollera ljudnivån för att sedan sänka eller höja värdena beroende på volymen, för att sedan ersätta den eventuella befintliga ljudsignalen på samma buss genom funktionen *ReplaceOut*.

Toner och ackord

Toner och ackord skapades med olika vågformer och filtertekniker. Synthdefinitionen *tone* använder en sågtandsvåg som är rik på övertoner och därför bra för att skapa komplexa ljudtexturer. Dessa vågor filtrerades med ett resonant lågpassfilter för att skapa en varmare ton och för att reducera oönskade frekvenser, vilket ger ett klarare och mer definierat ljud.

För ackorden användes sinusvågor, som genererar renare toner med färre övertoner än sågtandsvågor. Tre sinusvågor adderades tillsammans för att bilda ett ackord, och ljudet filtreras sedan med ett resonant lågpassfilter för att forma ljudet ytterligare. Genom att använda renare vågformer för ackorden, säkerställdes en harmonisk klarhet som var både behaglig och lätt att lyssna på i ett interaktivt sammanhang.

Bastrumma

För bastrumman användes en kombination av sinusvågor, rosa brus och olika enveloper för respektive grundljud. Envelopen för sinusvågen har en längre reasetid för att bevara den låga frekvensen något längre. Sinusvågen skapar kroppsljudet av trumman, medan det rosa bruset lägger till attackljud, vilket ger ett distinkt och kraftfullt trumslag. Detta val gjordes för att skapa ett ljud som är både djupt och slagkraftigt, vilket är viktigt för att ge rytmiskt stöd i miljön.

Hi-hat

Till hi-hat användes vitt brus filtrerat med ett högpassfilter och flera rektangulära vågformer. Vitt brus ger hi-haten dess distinkta "sizzle", medan de rektangulära vågformerna lägger till tonala element som skapar ett ljud som låter rikare och mer dynamiskt.

High tom och cowbell

För high tom användes en sinusvåg med en snabb pitch-modulering för att skapa ett dynamiskt percussivt ljud. Cowbell-ljudet skapades med rektangulära vågformer, vilket gav ett metalliskt ljud som sticker ut och ger variation. Dessa ljud valdes för deras distinkta och igenkännliga karaktär, vilket bidrar till en mer varierad och engagerande ljudupplevelse.

Open Sound Control (OSC)

För att koppla samman gränssnittet med SuperCollider användes OSC, vilket är en specifikation för datatransport som möjliggör realtidskommunikation med meddelanden mellan olika applikationer och hårdvaruenheter. Med denna metod kunde data samlas in i Unity, beroende på hur gränssnittet beter sig, och sedan skickas till SuperCollider där informationen genererar ljud baserat på

meddelandet. I Unity användes ett OSC-bibliotek där de önskvärda funktionaliteterna redan var implementerade. Data skickades som en sträng till en OSC-lyssnare i SuperCollider. För att kunna skicka flera dataelement i samma meddelande separerades de med mellanslag i strängen och splittrades sedan med funktionen *split(\$)* i SuperCollider för att sparas i en array.

Mappning och ljudförändring

Mappningen i detta projekt innebär att inkommande OSC-meddelanden styr olika parametrar i syntarna. Till exempel mappas data till att välja ackord och oktav, samt att trigga olika ljud. Denna dynamiska mappning gör att ljuddesignen kan anpassas i realtid baserat på användarens interaktion, vilket ger en flexibel och responsiv ljudmiljö som kan förändras och anpassas efter användarens handlingar och dataflöden.

Det första ordet i OSC-meddelandet motsvarar ifall ett ackord, ton, eller de övriga ljuden (bastrumma, hi-hat, high tom och cowbell) ska skapas. Om ordet är "chord" genereras alltså ett ackordljud. Samtliga grundackord som användes i projektet lagrades i en lista, där varje ackord representeras av tre MIDI-nummer. Ett ackord väljs slumpmässigt och korrigeras genom att välja oktav, baserat på ytterligare information från OSC-meddelandet. I Unity beräknas kollisionsvinkeln mellan synthbollen och stiften och skickas med i samma meddelande. Denna vinkel mappas linjärt med funktionen *linlin* så att vinkeln [0, 360] mappas till [0, 4] och avrundas till närmaste heltal. Om siffran är 0 spelas grundackordet, och om den är 1, 2, 3 eller 4 spelas ackordet en till fyra oktaver högre än grundackordet. I MIDI-noter motsvarar en oktav 12 steg. För att sätta oktaven på ackordet används grundackordet från listan och MIDI-numren ökas med 12 gånger antalet oktaver (0 till 4).

Om det första ordet är "tone" genereras en ton. Även här används informationen från kollisionsvinkeln för att skapa tonen. Vid applikationsstart har ton synthen MIDI-numret 60 och ökar/sänks beroende på kollisionsvinkeln. Intervallet för detta värde (0 till 360) mappas till intervallet [-4, 4] med *linlin*-funktionen. Denna siffra adderas sedan till variabeln *~currentTone* för att justera frekvensen av vågformen i ton-synthdefinitionen. På så sätt skapas en melodi som klättrar med upp till fyra steg vid varje kollision, från den föregående tonen.

Om det första ordet i meddelandet varken är "chord" eller "tone", kommer det att vara antingen "bassDrum", "cowbell", "hihat" eller "hightom". Dessa kommer endast att skapa de respektive syntharna som ordet beskriver. Här sker ingen ytterligare mappning för att förändra ljudet vidare, eftersom dessa typer av ljud är utformade för att behålla sin ursprungliga karaktär. Även om det finns olika varianter av exempelvis hi-hat-ljud, bedömdes den valda hi-haten vara tillräckligt bra för projektets behov och behövde därför inte modifieras ytterligare.

Tillgänglighet, användbarhet, och tillämpbarhet

Ljuddesignen är tillgänglig och användbar i olika sammanhang, såsom interaktiva installationer, musikproduktion och även spelutveckling. Ljuden är lätta att utökas eller ändras på för att anpassa ljuddesignen för olika behov. Tillämpbarheten är hög, eftersom systemet är flexibelt och kan integreras med andra programvaror och hårdvaror via OSC.

Slutsats

Projektet har visat hur data kan mappas till ljudparametrar för att skapa en interaktiv och auditiv upplevelse. Genom användningen av SuperCollider och Unity har en dynamisk miljö skapats där användaren påverkar ljuddesignen genom ett visuellt gränssnitt. OSC har använts för att överföra data från Unity till SuperCollider, vilket har möjliggjort en effektiv realtidsmappning. Denna metod har visat sig vara flexibel och kraftfull för att kontrollera ljudparametrar baserat på användarens interaktioner i gränssnittet.

Genom att tillämpa olika syntesmetoder såsom additiv och subtraktiv syntes samt brusgenerering, har en bred variation av ljudtexturer genererats. Varje typ av ljud, från toner och ackord till percussiva element som bastrumma och hi-hat, har utformats med specifika syntes- och filtertekniker för att uppnå de önskade ljudtexturerna.

Fokus har lagts på att skapa ljud som är korta och tydliga för att undvika överbelastning och att säkerställa att de bidrar till en estetiskt tilltalande ljudmiljö. Genom att använda en compander har också dynamiken i ljudbilden kunnat balanseras, vilket förbättrar användarupplevelsen.

Vidare har projektet utforskat potentialen för sonifiering i interaktiva system, vilket kan ha tillämpningar inom diverse områden såsom inom utbildning, medicin, datavisualisering och så vidare. Genom att omvandla data till ljud kan komplex information presenteras på ett intuitivt och lättillgängligt sätt, vilket kan underlätta analys och beslutsfattande.

I framtida arbeten skulle det potentiellt vara intressant att undersöka hur denna metod kan skalas upp och anpassas för mer komplexa datauppsättningar och hur användarupplevelsen kan förbättras ytterligare genom andra ljuddesigntechniker samt gränssnittsutveckling. Dessutom kan integrationen med andra teknologier och sensorer utforskas för att skapa ännu mer dynamiska och interaktiva ljudmiljöer.

Projektet har därmed inte bara uppnått sina ursprungliga mål, utan också öppnat upp för vidare forskning och utveckling inom området interaktiv ljuddesign och sonifiering.

Referenser

1. Farnell, Andy. 2010. *Designing Sound*. N.p.: MIT Press.
2. Hermann, Thomas, and Andy Hunt. 2011. *The Sonification Handbook*. Edited by Thomas Hermann, Andy Hunt, and John G. Neuhoff. N.p.: Logos Verlag.
3. Tagi, Eldar. 2023. "Synthesis Methods Explained: What is Additive Synthesis?" *Perfect Circuit*. <https://www.perfectcircuit.com/signal/what-is-additive-synthesis>.
4. "What is subtractive synthesis?" 2023. *Native Instruments Blog*.
<https://blog.native-instruments.com/subtractive-synthesis/>.
5. "Pink Noise vs. White Noise: What's the Difference?" n.d. *Yogasleep*. Accessed June 7, 2024. <https://yogasleep.com/blogs/give-sleep-a-chance-blog/pink-noise-vs-white-noise>.