

Tack

Stort tack till min texthandledare Kim Hedås och lärare Erik Peters för alla kloka råd och vägledning! Även stort tack till Herman Wikner* som hjälpt mig bygga användargränssnittet i React.js!

*Hermans GitHub finns tillgänglig på: <https://github.com/hermanwikner> (hämtad 15 april 2021).

Innehåll

Introduktion

Denna text kompletterar mitt självständiga arbete, *Radio Diabetes*, en interaktiv komposition/installation som genererar musik av blodsockervärden. Jag har byggt ett SuperCollider¹-program som skapar själva musiken och en webbplats där man kan lyssna på den, läsa om projektet och ladda upp sina egna blodsockervärden. När en deltagare laddar upp sina värden slussas dessa direkt vidare till SuperCollider-programmet, som i sin tur inkluderar dem i musiken, antingen direkt eller att de blir schemalagda i en kö. Musiken strömmas till webbplatsen (och vidare till lyssnaren) via en internetradiostation. På så sätt utgör musiken ett kontinuerligt flöde som deltagare och åhörare hör samtidigt: det finns ingen början, mitten eller slut, utan endast ett *nu*.

Bakgrund

Idéen om att göra musik av blodsockervärden föddes dagen då jag fick en *Freestyle Libre*-mätare², en så kallad kontinuerlig blodsockermätare, eller Continuous Glucose Monitor (CGM). Denna typ av blodsockermätare skiljer sig från tradionella mätare — som man är tvungen att sticka sig i fingret och på så sätt mäta blodsockret med — i att den regelbundet gör mätningar, vilket ger en kontinuerlig kurva över ens blodsockervärden. Kurvorna påminde mig om hur ljudsignaler ofta representeras visuellt³ och i ett tidigt experiment gjorde jag en direkt översättning av mina blodsockerkurvor till ljudfiler: en så kallad *audifiering*. Dessa ljud använde jag som samplings i mitt stycke *Värden och en vagg*⁴, som var ett av arbetsproverna jag sökte till Kungliga Musikhögskolan med.

Jag utvecklade vidare och förbättrade mitt första program som jag hade skrivit för att översätta mina kurvor till ljudfiler, så att vem som helst skulle kunna använda programmet och översätta sina egna värden till ljud. Jag byggde också en wavetable-synth i SuperCollider som använde dessa ljudfiler som källmaterial. Detta instrument har jag använt i ett antal olika kompositioner som jag skrivit under min skoltid. Båda dessa program (översättaren och wavetable-synthen) har jag publicerat på min Github⁵. En del av denna kod återanvände jag även i detta projekt.

Några tidigare exempel, historiska och nutida

Här har jag valt ut ett par exempel med musik och ljudinstallationer komponerad utifrån biologiska signaler och mätvärden för att ge ett historiskt perspektiv och nutida sammanhang.

Alvin Lucier ses som en pionjär inom detta fält, då han redan 1965 skrev stycket *Music*

¹noauthor'supercollider'nodate.

²abbott'glukosovervakningssystem'nodate.

³Bestående av en horisontell tidsaxel och en linjär vertikal axel

⁴jondell'varden'2017.

⁵jondell'kj-jondell'diabetes-synth'2021.

for *Solo Performer*⁶. I detta verk låter han uppmäta alfavågor med elektroder kopplade till huvudet av interpreten/utövaren. Dessa vågor förstärks sedan och spelas upp i 16 högtalare, som i sin tur exiterar diverse slagverk. Lucier kommenterade om stycket, i ett seminarium 2001 citerat av Straebel och Thoben:

I thought, “I don’t have a structure for this.” I mean, I’m a composer. I should impose some kind of structure, but then I thought, no, brain waves are a natural phenomenon. They should just flow out ...⁷

Detta belyser Luciers generativa förhållande till formen, och den direkta kopplingen mellan den uppmätta signalen och ljudvågorna är ett tidigt exempel på audifiering.

På senare tid har ett antal genrer eller rörelser uppkommit med utgångspunkt i sonifieringen av biologisk data, med namn som *protein music*⁸, *DNA music*⁹ och *gene music*¹⁰. Dessa genrer utgår samtliga från att låta musiken styras av dataset bestående av proteinveckning, DNA-sekvenser eller gener, till exempel mappningen av kvävebaser till tonhöjd¹¹. Det finns ett visst vetenskapligt anspråk i de texter som skrivs om dessa olika typer av musik, att sonifieringen kan ge en typ av insikt i de dataset som används som en visualisering *inte* skulle ge¹². Detta anspråk har dock nyanserats och kritiserats på senare tid, av till exempel forskaren och kompositören Peter E. Larsen¹³ som menar att sonifieringen knappast leder till någon djupare insikt än ett vanligt stapeldiagram. Jag vill lyfta Dr. David Deamers och Riley McLaughlins *Insulin A & B Chains* (1983)¹⁴ och Dr. Nobuo Munakatas *Hugging Tightly: Human RNase Inhibitor* (2009)¹⁵ som två exempel av *DNA*, eller *protein music*, som jag inspirerats av rent estetiskt.

Den 2 april 2021 släpptes Luca Yupanquis album *Sounds of the Unborn*¹⁶, som spelades in redan innan hon var född. Yupanquis föräldrar — musikerna Elizabeth Hart och Iván Diaz Mathé — hade med hjälp av elektroder kopplade till Harts kropp spelat in deras ofödda barns *in utero*-rörelser och sedan låtit sonifiera dessa inspelade signaler med ett system utvecklat av Sam Cusumano och hans företag *Electricity for progress*¹⁷. Elizabeth Hart berättar i en intervju i *Kulturnytt i P1*¹⁸ hur musiken kommit till av ett rent experimenterande, att föräldrarna till en början inte hade haft en tanke att ge ut det som ett album. Hart berättar vidare att hon *inte* tror att musiken säger något speciellt om de ofödda, att det inte varit föräldrarnas avsikt, och avfärdar frågan om huruvida processen endast varit ett PR-trick med att säga: “... for us, I think it is a pretty great album. Mu-

⁶lucier'music'1965.

⁷straebel'alvin'2014.

⁸king'pm'1996.

⁹k'kawazoe'study'2001.

¹⁰munakata'gene'1995.

¹¹shi'electronic'2007.

¹²king'pm'1996.

¹³larsen'more'2016.

¹⁴deamer'insulin'1983.

¹⁵munakata'hugging'2009.

¹⁶yupanqui'sounds'2021.

¹⁷noauthor'electricity'nodate.

¹⁸eklund'duo'2021.

sically, I am proud of what was achieved, musically.” Skivbolaget *Sacred Bones Records*, som släppt albumet, säljer systemet som Hart och Diaz Mathé använt under namnet *MIDI Biodata Sonification Device*¹⁹ och Cusumano, som utvecklat systemet, driver ett forum²⁰ där användare av detta system uppmuntras dela med sig av olika sonifieringar de gjort.

radioqualia är ett kollektiv bestående av de två nya zeeländska konstnärerna Honor Hargar och Adam Hyde som utforskar användandet av radio och internetradio som konstmedium²¹. Hargar och Hyde startade samarbetet 1998 och har gjort projekt som bland annat *Free Radio Linux*²² — en radioutsänd²³ datoriserad uppläsning av de fyra miljoner rader kod som Linux operativsystemskärna då bestod av, med start den 3e februari 2002 — och *Radio Astronomy*, en sonifiering/audifiering av signaler uppmätta med ett radioteleskop, som i realtid utsänds dels via AM, FM och kortvågsradio, dels via konstnärernas hemsida. I en text²⁴ av Jacques Perron som behandlar just *Radio Astronomy* citeras den tyska dramatikern Bertolt Brecht, som i sin text *The Radio as an Apparatus of Communication* från 1927 skrev:

Radio is one sided when it should be two. It is purely an apparatus for distribution, for mere sharing out. So here is a positive suggestion: change this apparatus over from distribution to communication. The radio would be the finest possible communication apparatus in public life, a vast network of pipes. That is to say, it would be if it knew how to receive as well as transmit, how to let the listener speak as well as hear, how to bring him into a relationship instead of isolating him. On this principle the radio should step out of the supply business and organise its listeners as suppliers.²⁵

Radio

En del av *Radio Diabetes* består av, som namnet antyder, en internetradiostation, som strömmar ut den genererade musiken. Just radion som medium påverkar själva innehållet, det vill säga musiken, i olika avseenden, till exempel dess temporalitet och interaktivitet. Marshall McLuhan beskriver i sin bok *Understanding media: the extensions of man*²⁶ radion som ett *hot medium*, något han menar är ett medium som “...extends one single sense in *high definition*... the state of being well filled with data.”²⁷, men som följdaktligen inte ger mycket utrymme för deltagande. McLuhan kontrasterar detta med exempel av *cool media* som telefonen, där innehållet skapas helt av deltagarna (*high* och *low definition* ska alltså inte blandas ihop med ljudkvalitet i detta fall). Den första utgåvan av *Understanding media* kom 1964, en tid då dessa medier kanske var mer kategoriskt *hot* eller *cool*. I Jonas Anghammars C-uppsats *Nya medier möter gamla radion: publikmedverkan i public service-radion nu och då*²⁸ från 2010 går det att läsa om hur graden av interaktion i Sveriges

¹⁹noauthor'midi'nodate.

²⁰noauthor'biodata'nodate.

²¹noauthor'r'nodate.

²²noauthor'free'2002.

²³Både rundradio- och internetradiosändning.

²⁴perron'radioqualia'2003.

²⁵brecht'radio'nodate.

²⁶mcluhan'understanding'2003.

²⁷mcluhan'understanding'2003.

²⁸anghammar'nya'2010.

Radios programutbud ökat markant bara sedan 90-talet, något som kopplas samman med den snabba digitaliseringen som skett. Anghammar skriver i ett citat av Nils Enlund från 2008 att "...publiken nu själva skapar stor del av medieinnehållet eftersom den tillåts göra det." Kanske undergår radion, som medium, en nedkylning, en del i utvecklingen av förgörelsen av tid och rum som McLuhan menar att elektrifieringen av vår värld har påkallat. "With instant electric technology, the globe itself can never again be more than a village..."²⁹.

Sonifiering

I det andra kapitlet av antalogin *The Sonification Handbook*, skrivet av Walker och Nees, ges en möjlig definition av sonifiering som: "... a subtype of auditory displays that use non-speech audio to represent information. [...] 'the transformation of data relations into perceived relations in an acoustic signal for the purposes of facilitating communication or interpretation.'" ³⁰ Som tidigare nämnts görs ofta ett vetenskapligt anspråk om användandet av sonifiering som något med potential att ge åhörare i allmänhet, och forskare i synnerhet, en insikt³¹ som en annan representation av informationen inte hade gett, eller kunnat ge.

I mitt projekt tar jag avstånd från idén om att användandet av sonifiering på något sätt ska skänka insikt till datan som jag låter gestalta i musiken; tvärtom hoppas jag att den musikaliska gestaltningen *inte* kommer avslöja något om de bakomliggande mätvärdena. Jag kommer ändå göra en kortare sammanfattning av olika *sonifieringstekniker*- och *kategorier* som redogörs i *The Sonification Handbook*, för att sätta ord på det jag har gjort i detta projekt.

Den vanligaste formen av sonifiering, skriver Hermann, Hunt och Neuhoﬀ i introduktionen³² till antalogin, är troligtvis den så kallade Parameter Mapping Sonification (PMSon), som använts nästan synonymt med begreppet *sonifiering*. PMSon är användandet av en *mappningsfunktion* (eller kort och gott *mappningar*) för att sammankoppla ett datavärde eller -attribut till en eller flera akustiska parametrar. I det femtonde kapitlet i antalogin, *Parameter Mapping Sonification*³³, beskriver Grond och Berger de tre olika topologierna av mappningar som är möjliga: *en-till-en*, *en-till-flera* (divergent) och *flera-till-en* (konvergent). Grond och Berger redogör också för de olika användningarna av PMSon, där de slutligen namnger den rent konstnärliga tillämpningen som *musifieringen* (en. *musification*) av datavärden.

Audifiering³⁴.

I *Radio Diabetes* har jag till största del använt en blandning av *en-till-en* och *en-till-flera* topologier av PMSon, men även inslag av *audifiering*.

²⁹ mcluhan'understanding'2003.

³⁰ hermann'theory'2011.

³¹ Ordet *insikt* är (något ironiskt) ett bra exempel på hur invand användningen av visuell representation är för *förståelsen*.

³² hermann'introduction'2011.

³³ hermann'parameter'2011.

³⁴ hermann'audification'2011.

Diabetes

Diabetes mellitus är en auto-immun sjukdom som huvudsakligen finns i två olika varianter: typ 1-diabetes och typ 2-diabetes. Insulinproduktion/insulinresistens, insulinbehandlad. Detta projekt riktar sig främst till typ 1-diabetiker, då vi i Sverige finns x antal typ 1-diabetiker, och i världen (siffra). Jag debuterade med typ-1 diabetes sista veckan i augusti 2001, precis efter att jag hade börjat första klass i grundskolan. Sedan dess har jag dagligen tagit insulininjektioner och mätt mitt blodsocker, inför varje måltid och däremellan. Jag har haft otaliga episoder av lågt och högt blodsocker (hypo- och hypoglycemi). Som diabetiker kan j Sjukdomen är kronisk, överdödlighet, komplikationer

Blodsockervärden

Blodsocker mäts i de flesta delar av världen i mmol/L, men enheten mg/dL existerar också. Variationen i blodsocker hos en icke-diabetiker är mellan 4 och 6 mmol/L [källa]. Hos en diabetiker kan detta värde variera från under 1 till över 30 mmol/L. I Sverige används huvudsakligen två olika CGM: dels Freestyle Libre-sensorn, dels Dexcom G6-sensorn. Freestyle Libre-sensorn har ett spann på att mäta från lägst 2,2 till 27,7 mmol/L (annars visar den *LO* respektive *HI*). Freestyle Libre-sensorn mäter kontinuerligt var 15:e minut, och sensorn byts varannan vecka.

Blodsockervärden som mäts med dessa två sensorer går att ladda upp, både automatiskt och manuellt, till en plattform som heter *Diasend*³⁵. Via *Diasend* går det att exportera sina värden, begränsat till en valfri period — men åtminstone två veckor lång — till en Excel-fil. Det är denna Excel-fil som man sedan kan ladda upp till *Radio Diabetes*. Tack vare *Diasend*-plattformen går det att exportera värden på ett format som är oberoende av vilken sensor man använder.

Förhållandet till mätandet

I sin text *Det autoimmuna jaget — om att sätta gränser*³⁶ skriver Mats Arvidson om kravet som diabetiker på disciplin och prestation.

Prestation, utmattning (bornemark...utmattning...)? Krav och värden...

Ett sentiment som ofta förekommande ("jag är **inte** min diabetes, mina blodsockervärden...", t.ex. artikel i *Hälsportalen*(???))

Datainsamling

Enligt *Dataskyddsförordningen* (GDPR) är biometrisk data en känslig personuppgift, och därför kräver insamlingen av den ett uttryckligt samtycke från varje deltagare att denna är införstådd i hur datan behandlas. Jag har försökt vara så transparent som möjligt i hur

³⁵.

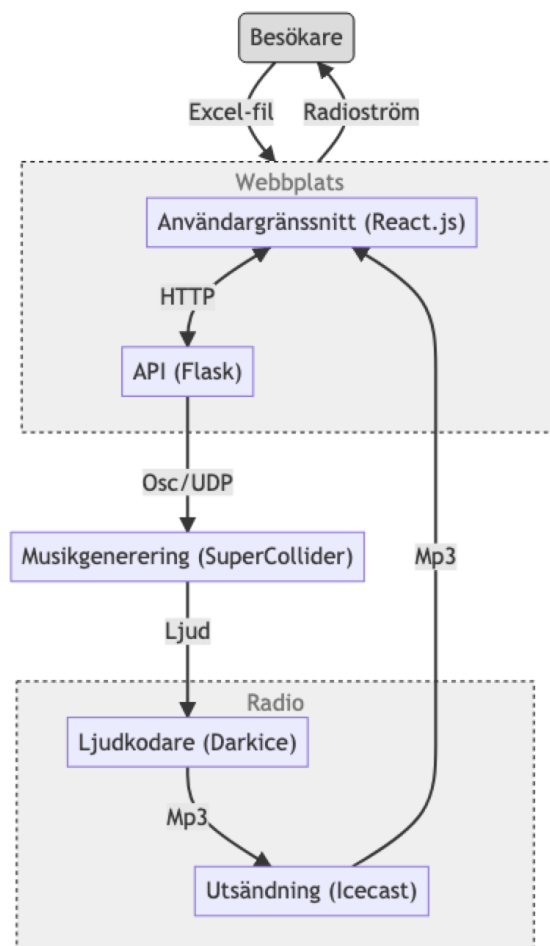
³⁶arvidson'det'2016.

datan behandlas, genom att dels dela **all** källkod som jag använder, och även i den kommunikation jag lagt ut på webbplatsen och i övriga dokument berörande projektet (såsom denna text). All data som samlas in anonymiseras/avidentifieras så fort som möjligt och den är inte sparad någonstans utöver arbetsminnet som SuperCollider använder. I enlighet med Vetenskapsrådets text *God forskningssed*³⁷ är sekretess, anonymitet och integritet av största vikt i detta projekt, även fast det är ett konstprojekt och inte ett forskningsprojekt. Jag har inget kommersiellt intresse i insamlingen av datan, jag delar den inte med någon extern part heller, och allt deltagande är valfritt. Min ambition är **inte** att samla data för sakens skull, utan att diabetiker ska kunna dela med sig av sina värden utan att de på något sätt bedöms eller värderas: helt enkelt, att själva delandet och deltagandet i sig står i fokus.

³⁷vetenskapsradet'god'2017.

Process

Radio Diabetes består som tidigare nämnt av två delar: ett musikgenererande program (SuperCollider) och en webbplats (se skärmdump på nästa sida). Här följer en teknisk beskrivning av detta system, som illustreras i figur ?? nedan.

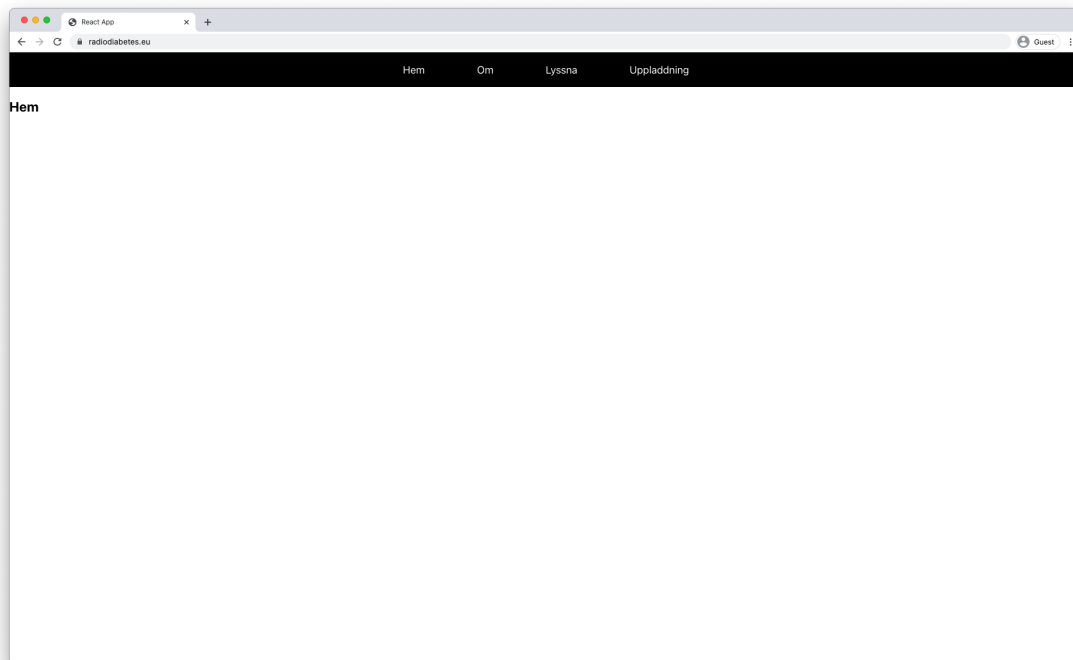


Figur 1: Översiktsdiagram av system

Stack

För att driftsätta systemet valde jag att hyra en så kallad Virtual Private Server (VPS), som tillåter mig att hyfast obegränsat installera, konfigurera och köra de program som jag behöver. Som operativsystem valde jag Ubuntu 20.04, eftersom det är det jag har installerat på min hemmaserver, som jag använt under utvecklingsfasen av projektet. Som server till användargränssnittet använder jag Nginx och till Flask-API:et använder jag Gunicorn. SuperCollider, Darkice — som kodar ljudströmmen till ett sändningbart MPEG-format — och Icecast har jag valt att köra samtliga som så kallade *services*, vilket underlättar vid uppstart och omstart, utifall systemet eller någon del av det skulle krascha. Att de körs som *services* underlättar också loggning, till exempel kan jag låta SuperCollider skriva ut

viktiga händelser, som tidpunkten när ett nytt datapaket mottagits av programmet. Dessa utskrifter har jag senare åtkomst till via kommandot `journalctl`.



Figur 2: Skärmdump av hemsida (temporär)

Webbplats

Webbplatsen finns tillgänglig på domänen³⁸: <https://radiodiabetes.eu/>. I figur ?? ovan visas en skärmdump av webbplatsen tagen den 13 april 2021.

Många webbplatser är uppdelade i en *front end* och en *back end*, och nedan följer en redogörelse av hur jag implementerat dessa för *Radio Diabetes*.

Front end

För att bygga webbplatsens *front end* använde jag ett JavaScript-bibliotek/verktyg som heter React.js, som jag fick hjälp av min vän Herman Wikner att koda. Webbplatsen består huvudsakligen av tre undersidor med information, en plats för att ladda upp filerna, och en kontrollmeny till radion, där besökaren kan spela och pausa musiken. Jag ville att musiken skulle spelas oavbrutet när en besökare navigerar på webbplatsen, och det var en av huvudanledningarna till att jag valde att använda React.js istället för endast HTML och CSS. Överlag ville jag ha en så smärtfri så kallad *User Experience (UX)* som möjligt, för att göra deltagandet så enkelt som möjligt och för att göra installationen så tillgänglig som möjligt. I slutändan är det musiken som ska stå i fokus.

³⁸jondell'radio'nodate.

Kommunikationen till Flask-API:et sker via HTTP-metoderna POST och GET, där React.js skickar filinnehållet till Flask-API:et via POST-metoden, och får tillbaka ett svar om filen överförts utan problem eller inte.

Back end

Behovet av ett så kallat Application Programming Interface (API) kommer av att jag behövde en mellanhand för att överföra informationen från de uppladdade filerna till mitt SuperCollider-prpgram. Från tidigare experiment med mina blodsockervärden hade jag redan skrivit Python-kod för att extrahera mätvärden från de Excel-filer som man kan exportera från *Diasend*. Därför återanvände jag en del av denna kod, och omstrukturerade denna för att kunna köras som ett Flask-program³⁹. När API:et tar emot filinnehållet från användargränssnittet via POST-metoden så skapar den direkt två listor bestående av inlästa tidsvärden och blodsockervärden från filen. Filen sparas alltså aldrig på servern. API:et gör också en del felhantering genom att försäkra att datavärdena är formaterade på ett korrekt sätt och att de är inom det möjliga intervallet. Sedan skickar det vidare listorna med värden till SuperCollider-programmet, via Osc.

Sammanfattningsvis har alltså API:et tre huvudsakliga funktioner: att *parsa* (eller tolka) de delade filerna, felhantera korrupta filer eller korrupt data, och upprätta en Osc-kommunikation med SuperCollider.

SuperCollider-system

När en deltagare laddar upp en fil med sina mätvärden så skickas de vidare, via Flask-API:et, till SuperCollider-programmet, som spelar upp en tack-hälsning för att ge en direkt återkoppling till deltagandet. Varje instans av mätdata — det vill säga varje bidrag till, eller varje interaktion med, installationen — gestaltas av en specifik musikalisk funktion, för att på sätt ge deltagaren en koppling och förståelse till hur dennes bidrag påverkar musiken. I SuperCollider-programmet representeras varje sådan instans av mätdata av ett *objekt*, som innehåller attribut som bland annat: register, skala, speltid, panorering, klangkälla/*SynthDef*, och tillhörande *Pattern*. Dessa attribut är antingen direkt bestämda av mätdata (en så kallad *mappning*) eller bestämda utifrån de andra aktiva objekten, till exempel förhåller de sig till redan “upptagna” register. Skapandet av dessa objekt gör programmet så fort det mottagit ett nytt paket av mätdata från Python-servern. Samtidigt avgör programmet om objektet ska spelas upp direkt, eller om det ska läggas på kö.

Kösystemet

Behovet av ett kösystem kommer ur scenariot att flera deltagare laddar upp värden inom en relativt kort tidsram. Om detta händer *utan ett kösystem*, kan systemet reagera på två sätt: antingen spelas alla objekt upp samtidigt, med risk för att överrösta varandra och bli

³⁹Flask är ett webbramverk för Python, det går alltså att använda Python-kod i Flask.

en kakofoni, eller så ersätter de varandra, vilket skulle leda till att ens bidrag inte skulle höras mer än en väldigt kort stund. En kompromiss är att använda ett kösystem, så att antalet samtidigt spelande objekt begränsas, och att dessa spelas *minst* en given längd tid, men inte längre än en annan bestämd tid, om det står väntande deltagare på kö. Allt som allt har jag begränsat antalet samtidigt spelande objekt till *tre* stycken, och när dessa är fyllda ställs antingen ett nytt bidrag på kö — om inte de spelande objekten har varit aktiva i den givna minimumtiden — eller så ersätter det nya bidraget det äldsta spelande objektet. Kösystemet är i tekniska termer alltså ett First In, First Out (FIFO)-system.

Tillvägagångssätt

Eftersom jag har byggt ett hyfsat komplext system, med många av varandra beroende delsystem, har jag stött på en del oförutsägbaheter under resans gång, något jag vill reflektera över här.

En förutsättning till att jag skulle få allt att fungera var att utveckla systemet modulärt, det vill säga att de olika beståndsdelarna (se diagram i i figur ?? för referens) är fristående, med in- och utdata som passerar genom dem. För mig har själva SuperCollider-programmet alltid varit en central och prioriterad del av hela systemet, då det är den som faktiskt genererar musiken som jag har komponerat, och därför började jag med att bygga det. Till en början lät jag programmet använda en hårdkodad lista med mätvärden som indata, då jag inte hade byggt färdigt webbplatsen och API:et. Detta påverkade mina konstnärliga val i SuperCollider, då jag hade ganska lite variation i mätvärdena. Något problematiskt hade jag inte heller bestämt några begränsningar i vad mitt SuperCollider-program skulle generera för musik, dessutom var frihetsgraden väldigt hög, så det var svårt att veta hur mycket tid jag skulle lägga på den delen, och svårt att känna mig nöjd. I slutändan bestämde jag mig för att utveckla delarna parallellt istället för sekvensiellt.

Just utvecklingen av webbplatsen och programmeringen i React.js och Flask blev kanske den största flaskhalsen i projektet, och det mycket på grund av att jag aldrig byggt en dynamisk webbapplikation tidigare. I planeringsfasen hade jag lite svårt att bestämma mig för vilken sorts arkitektur jag skulle använda mig av — en enklare sida med endast Flask, CSS och HTML, eller React.js och Flask — och det var ett ganska sent beslut som jag tog att faktiskt använda React.js, som medförde många fördelar med UX i åtanke, men krävde mer tid. Just planeringsmässigt tror jag att projektet lidit av att jag dels haft väldigt svårt att avgöra hur lång tid saker skulle ta, att jag inte heller vetat hur många timmar jag haft till förfogande, att jag inte haft tydliga milstolpar och mål, att jag kanske varit för ambitiös. Genomföringsmässigt har det också lidit av mitt vacklande självförtroende och en gnutta utmattning. Eftersom det är ett konstnärligt projekt så tror jag det kanske är omöjligt att formulera tydligare mål och milstolpar än de jag haft, men jag har haft en groende känsla att jag fastnat lite i detaljerna.

En liten motstridig erfarenhet som jag hade med utvecklingen av SuperCollider - programmet var att jag hade en ganska stark idé om att inte göra “enkla” *en-till-en*

mappningar av mätvärdena till tonhöjd. Det var en väldigt kontraproduktiv idé, rotad i ambitionen av att musiken inte skulle bli transparent och avslöjande om de bakomliggande värdena. När jag väl provade att göra den mappningen så var effekten väldigt annorlunda: jag tyckte inte alls att det var för transparent, eller ens tråkigt, jag gillade hur det lät. Utan att bli allt för efterklok och självkritisk så önskar jag ändå att jag inte hade haft så många förutfattade meningar om hur musiken i slutändan skulle låta.

Sammanfattningsvis vill jag dra slutsatsen att jag kunde planerat in striktare deadlines för vissa tekniska aspekter — som att bli klar med webbplatsen vid ett visst datum — men mindre planerande kring det estetiska utfallet, att kanske tro mer på min process.

Musiken

Musiken genereras av SuperCollider-programmet som jag har skrivit. Jag har fyra olika lager, instrument eller musikaliska funktioner tillgängliga:

1. Ett fundament bestående av en GSM-avkodad formant-filtrerad “kör”.
2. Sinusvågor och elektroniska störningsljud som spelar rytmiska arpeggion.
3. En wavetable-pad
4. Ett samplings-baserat slagverksliknande instrument.

Alla instrument som spelar arpeggion, det vill säga 2 och 4 i tabellen ovan, spelar i en pentatonisk durskala, där skalstegen bestäms av de inkommande mätvärdena. När de skapas tilldelas en transponering som bestämmer vilket omfång/register de ska spela i, samt en panorering i stereofältet. Viss variation i envelop och duration har jag skapat med en stokastisk variabel, i andra ord en slumpgenerator, men annars bestäms alla parametrar av mätvärdena. Att jag valde just att låta dem spela i en pentatonisk skala var för att jag ville ha en förhållandevis enkel harmonik och melodik, kontrasterat med vissa av de mer komplexa, hårda klangerna.

Ljudmaterialet i de samplingsbaserade instrumenten har jag tagit från diverse inspelningar som jag har gjort genom åren: jag ville att de skulle ha en personlig koppling till mig. Den GSM-avkodade ljudfilen som utgör “kören” skapade jag av *misstag* genom att läsa in en annan ljudfil som om den vore GSM-kodad i programmet Audacity. Effekten är en hård och övertonsrik klang, som jag låter passera genom ett formant-filter för att skapa ett *människolikt* ljud. Denna spelar en följd av fyra ackord, som upprepas utan variation. Däremot varieras andra attribut av detta instrument. De andra samplingarna jag använt är utklipp av elektroniska störningsljud — återvunna från mitt förstaårsprojekt *Calling out of context*, men i ny tappning — och en inspelning av min induktionshäll. Samtliga inspelningar jag använt har en viss *elektrisk* karaktäristik, vilket kommit att utgöra min ljudpalett.

Wavetable-padden, det vill säga punkt nummer 3 i tabellen, är en sorts audifiering av de inkommande mätvärdena: dessa omvandlas till så kallade *wavetables* som används av en Wavetable-UGen i SuperCollider. På så sätt kan jag kontrollera tonhöjd men att själva klangfärgen bestäms helt av mätvärdena. Detta är en SynthDef som jag utvecklat under en längre tid, men som jag anpassat till *Radio Diabetes* för att direkt omvandla mätvärdena till wavetables. Tidigare har jag använt ett annat program för att göra själva omvandlingen, men det var inte möjligt för detta projekt, då det hade krävt att jag sparat ned värdena som ljudfiler. Detta var jag som sagt tvungen att undvika, av säkerhet- och integritetssjäl.

Rumslighet

Varje objekt ges en unik position i stereofältet, och på så sätt en plats i rummet i musiken.

Presentationen av radioströmmen genom hemsidan som jag har utformat påverkar också den upplevda rumsligheten i musiken.

Ett planerat konserttillfälle kommer att ske den 20 maj i Lilla Salen på Kungliga Musikhögskolan. Då spelas ett utdrag ur radioströmmen upp, som den hörs i realtid. I och med de rådande restriktionerna så kommer ingen publik kunna närvara, utan konserten strömmas vidare till en publik i etern. Själva konserttillfället blir därför en sorts manifestation av radioströmmen i tid och rum.

Temporalitet

Radio Diabetes utspelar sig — i sin helhet — någonstans mellan de tidsskalor som Curtis Roads i sin bok *Microsound*⁴⁰ benämner *infinite* och *supra*, den senare definierar Roads som: “A time scale beyond that of an individual composition and extending into months, years, decades and centuries”⁴¹. Eftersom min installation spelas upp i en radiosänding, undviker jag *radiotystnad*, och ett konstant flöde av musik strömmas därför ut. Utan tystnaden som markör för en början eller slut blir lyssnandet fokuserat på skeenden i *nuet*. Varje lyssnare börjar *in media res* i musiken, och har en unik lyssningsupplevelse då det är väldigt osannolikt att två besökare börjar och slutar lyssnandet samtidigt. En deltagare kommer ha ytterligare en annan tidsuppfattning, då denne kanske väntar på att höra sitt bidrag: som antingen spelas upp direkt eller ställs i kö.

Roads beskriver⁴² hur *macroform* kan uppstå på två sätt, det ena kallar han *top-down* och det andra *bottom-up*. I *Radio Diabetes* bestäms denna form av de underliggande processer som utgör musiken, och därför är den ett exempel av *bottom-up*. En ambition jag haft har varit att tidsbestämma olika möjliga skeenden. Med andra ord, att musiken påverkas av tiden på dygnet, till exempel att visa instrument spelas endast under vissa timmar, eller dagar. Då skulle jag möjligtvis behöva förhålla mig till en *top-down*-metod, att planera ut när vissa saker skulle hända, eller i vilken ordning. Men detta är än så länge bara en framtida eventualitet.

Generativitet

I texten *Defining Sound Toys: Play as Composition*⁴³ beskriver Dolphin en form av interaktivt, audiovisuellt kompositionssystem som han kallar *sound toys*. *Sound toys* bjuder in till en högre grad av lekfullhet och fritt interagerande och reagerande än vad jag gör i *Radio Diabetes* — och därför kanske inte min installation klassificeras som en sådan — men mycket av det Dolphin nämner i sin text är högst relevant för detta projekt. Dolphin

⁴⁰roads'microsound'2004.

⁴¹roads'microsound'2004.

⁴²roads'microsound'2004.

⁴³collins'defining'2014.

refererar bland annat till Umberto Ecos text *The Poetics of the Open Work*, och hur det han kallar *sound toys* faller in någonstans i gränslandet⁴⁴ mellan *open work*, *kompositionsverktyg* och *instrument*. På många sätt är mitt *Radio Diabetes* också en blandning av dessa tre kategorier, där *instrument* möjligtvis går att byta ut mot *en samling instrument* eller *orkester*.

⁴⁴collins' defining'2014.

Slutsatser

En av de främsta slutsatser jag dragit av att arbeta med detta projekt är att *saker tar längre tid än vad jag hade tänkt*⁴⁵. I planeringsfasen hade jag många idéer, och en stark motivation och ambition, som under projektets gång tyvärr dalat. Jag hade en vision om hur saker skulle låta, och hur de skulle fungera, men av många anledningar — kanske främst att *saker inte blir som man tror/tänkt* — så har jag varit tvungen att omforma stora delar av projektet. Men jag vill understryka att även om jag kanske varit “för” ambitiös eller ivrig, så är jag väldigt nöjd med att ha byggt en infrastruktur och arkitektur, en grund till plattformen, som jag i framtiden kan fylla med flera av mina musikaliska idéer. Nedan följer en redogörelse för de utvecklingsmöjligheter jag anser finnas.

Utvecklingsmöjligheter

1. Jag hade en tanke om att göra det möjligt för deltagare att lämna meddelanden, antingen i formen av en gästbok på webbplatsen eller att sätta upp ett telefonnummer dit man kunde ringa och lämna meddelanden som spelas upp i radion. I det senare fallet ville jag uppmuntra delandet av upplevelser kring att leva med diabetes.
2. En teknisk ambition jag har är att sätta upp två parallella radioströmmar, som jag kan växla mellan när jag vill underhålla och/eller utveckla SuperCollider-systemet. Som systemet fungerar nu måste jag stänga av radion för att göra förändringar i SuperCollider, något som skapar *radiotystnad*. Med två parallella strömmar kan jag slussa lyssnarna till den ena medan jag jobbar på den andra.
3. Rent rumsligt hade jag också en idé om att på något sätt binauralisera musiken — kanske med hjälp av en generaliserad HRTF — vilket skulle försätta lyssnaren i ett gemensamt akustiskt virutellt rum. Likt föregående punkt hade det kunnat möjliggöras genom att använda två strömmar, där den ena är binauraliserad och den andra vanlig stereo.
4. Jag vill internationalisera webbplatsen genom att erbjuda en engelsk översättning av all text och bjuda in diabetiker från hela världen att delta. I praktiken är det redan idag möjligt för diabetiker utanför Sverige att delta, så länge deras sjukvårdssystem använder *Diasend*. Men ett möjligt problem är att vissa länder inte använder mmol/L som måtenhet, och då skulle jag behöva konvertera från mg/dL som används på vissa platser och som har ett helt annat omfång och storlek.
5. Jag påbörjade ett försök att också visualisera delade mätvärden, med ett program som jag skrev i *openFrameworks*. Själva programmet blev jag klar med, men att inkludera *openFrameworks*-programmet i den infrastruktur jag byggde upp blev för tidskrävande och därför sköt jag upp den idéen.

⁴⁵Även känt som *Hofstadters lag*.

Referenser

Bilagor

Ordlista

Audifiering	en. <i>audification</i> . Att direktöversätta en data-serie till ljudkurvor.
Back end	Allt som rör sig "under huven", server-side programmering.
Front end	Användargränssnittet i webbutveckling.
Pattern	Ett verktyg i SuperCollider för att generera
Sonifiering	en. <i>sonification</i> . Att översätta en dataserie.
Stack	Alla delar som utgör en plattform eller ett system. Finns olika vanligt använda.
SuperCollider	Ett programmeringsspråk och plattform för syntes av ljud, och musikalisk programmering.

Förkortningar

API	Application Programming Interface.
CGM	Continuous Glucose Monitor.
FIFO	First In, First Out.
PMSon	Parameter Mapping Sonification.
UX	User Experience.
VPS	Virtual Private Server.

Radio Diabetes (2021)

Ett inspelat utdrag från själva radioströmmen, detsamma som spelades upp i Lilla Salen, är bifogat med denna text. Inspelningen skedde den 20 maj 2021.