

Project SBAM

- People with music

Uddannelse:	Multimedie Designer
Uddannelsessted:	Odense Tekniske Skole og TietgenSkolen
Årgang:	Juni 2003
Opgave:	Afsluttende hovedopgave
Vejleder:	Søren Priisholm
Elev:	Rasmus Berlin 270979-1985
Virksomhedskontakt:	Stud. datalogi Bent Bisballe Jensen
Projekt periode	Fra 06/01-2003 til 23/5-2003
URL:	http://sbam.aasimon.org

Project SBAM

- People with music

Indledning

Forord

Jeg har arbejdet som gæste DJ på forskellige klubber igennem ca. 4 år og holder meget af det. Jeg kender kun DJs, som kan lide deres arbejde, for det er så stressende og tit underbetalt, at brød de sig ikke om jobbet, havde de stoppet for længe siden.

Da det irriterede mig en dag hvor dårlig WinAmp3's crossfader-funktion var, kom jeg til at overveje, i hvor simpel en algoritme at en professionel DJs mix kunne beskrives. Jeg undrede mig længe over, at jeg aldrig havde set, et program benytte den teknik, jeg udtænkte. Da jeg skulle til at finde et emne til hovedopgave på multimedie designer uddannelsen, valgte jeg at undersøge emnet nærmere. Jeg testede en del software og fandt til min forbløffelse ud af at min idé lod til at være ny. Jeg fandt også en del meget professionelt udstyr til DJing af MP3-filer, men som alle var bygget på gamle idéer.

Et firma solgte en speciel pladespiller, som man kunne tilslutte computeren. Pladespilleren sendte så signaler til computeren om, hvor på pladen nålen var. Med dette udstyr kunne man søge i en MP3 fil med en pladespiller som interface. Udstyret var fascinerende, men virkede på mig meget overflødigt. Jeg mente at der da måtte være nogle nye muligheder i et nyt medie.

Den første idé var i virkeligheden at lave programmet som et plugin til WinAmp. Dette viste sig dog hurtigt at have et par problemer, så man først sent i projektet ville kunne vurdere om det overhovedet var muligt. Og projektet bevægede sig over imod et stykke komplet software.

Jeg var klar over, at jeg ikke længere ville kunne klare opgaven helt alene. Jeg opsøgte derfor en gammel skolekammerat, som jeg før havde lavet udviklingsopgaver med i fritiden. Jeg vidste, at han havde startet firmaet FleXoft, som nu var ved at lukke igen; og han nu var datalogistuderende på Århus Universitet. Da firmaet var lukket, og han havde fået en masse fritid, sagde han ja til et samarbejde om denne opgave. Hans arbejde på opgaven ville bestå i at være programmør. Mens mindre dele af programmeringen, design, planlægning og alt andet ville være min opgave.

Udviklingen har gået enormt godt med utroligt få problemer undervejs, og vi har begge været enormt glade for samarbejdet. Da jeg selv starter på datalogi/multimedie på Århus Universitet efter sommerferien, bliver der måske også mulighed for at lave endnu flere projekter sammen.

Plan for projektet

Igennem projektet har de forskellige designfaser kørt parallelt, så de hele tiden dynamisk har kunnet påvirke hinanden. Konceptet og en grundlæggende idé om projektets mål dannedes i tiden op til projektperioden. Idéen, designs og udvikling af produktet har så kørt sideløbende med hinanden, og da deadline for rapportafleveringen nærmede sig, kørte det desuden parallelt med rapportskrivningen. Derfor kan nogle af rapportens afsnit måske virke foregribende.

Skolens deadline for projektet er i øvrigt ikke deadline for hele projektet. Udviklingen vil fortsætte i sommerferien, hvor vi regner med efter sommeren at have et færdigt funktionelt produkt. Og selv efter dette fortsætter udvikling og test af produktet, da jeg selv regner med aktivt at bruge softwaren til min fremtidige DJing.

Grundlæggende idé	Videreudvikling på idé	Fortsættelse af udvikling
	Program design	Fortsættelse af udvikling
	Funktionelt design	Fortsættelse af udvikling
	Grafisk design	Fortsættelse af udvikling
	Hardware design	Fortsættelse af udvikling
	Rapport	
Projekt start	Rapport aflevering	Mundtlig eksamen

Indholdsfortegnelse

Project SBAM.....	3
Indledning.....	4
Forord	5
Plan for projektet	6
Indholdsfortegnelse.....	7
Problemformulering.....	8
Idéen.....	10
DJens job.....	11
Kernen i DJens job	11
Med en DJ på job	11
Automatisering af DJ-jobbet.....	13
Diskussion af arbejdsopgaver	13
Sortering og opsummering	16
Diskussion af fordele og ulemper.....	17
Kan DJen helt fjernes?.....	19
Software.....	20
Stjerneanalyse.....	21
Program design.....	24
Initialisering.....	24
Mixer	27
Cue	29
Tråde	30
Libraries.....	30
Funktionelt design.....	32
Mix-variation.....	36
Grafisk design	37
Nærhed.....	37
Grafiske symboler.....	38
Licens	39
Hardware & Udvidelser	40
Hardware designs	41
Mål	41
Den billige	41
Den smarte.....	41
Den professionelle.....	42
Udvidelser.....	43
Lyseffekt eller tekstscroller	43
Jukebox/Ønske-system	43
Afslutning	44
Konklusion.....	45
Perspektivering.....	46
Ordforklaring	47
Litteraturliste.....	49
Testet DJ-software	50
Bilag 1: Første skitse af design.....	51
Bilag 2: Billede af arbejdsprocessen.....	52

Problemformulering

Hvorfor

Før 1990 har DJs kun spillet vinyl, i dag står det stadig usikkert hvilket medie der er fremtiden, men det kunne meget vel være at computeren vil erstatte turntables¹ og mixere².

I en standard DJ CD-kuffert kan der være 100 CD'er, hvilket sjældent er nok. På en 80 GB harddisk vil der kunne ligge over 800 CD'er i OGG-filer af høj kvalitet.

En del af DJens arbejde er monotont arbejde som kræver fordybelse og timing, hvor andet kræver planlægning og overblik. De monotone dele af arbejdet må være mulige at automatisere, og derved give mere overskud til DJens planlægning.

Det software der findes på markedet til DJs, er alt sammen beregnet på beatmixing³; men DJs inden for en del genrer kan meget sjældent beatmixe mellem numrene.

Hvad

Jeg ønsker at udvikle et DJ-system for DJs, som ikke har behov for beatmixing. Systemet skal lette DJens arbejde på en måde, som gør ham istand til at udføre sit arbejde bedre ved at få større overblik. Det skal være muligt at gå til, uanset om man er gammel DJ, eller man er helt ny i faget. Systemets funktionalitet og design skal være innovativt, og ikke bundet af de "gamle" medier, men derimod udnytte nogle af dette nye medies muligheder.

Hvordan

For at være istand til at udvikle dette ret omfattende stykke software, har jeg slået mig sammen med en tidligere software udvikler for FleXoft, stud. datalogi Bent Bisballe Jensen fra Århus Universitet til de dele af opgaven, som omfatter programmering og programdesign.

¹ Se ordforklaring.

² Se ordforklaring.

³ Se ordforklaring.

Idéen

DJens job

Kernen i DJens job

Før vi kan se på, hvordan DJens job kan automatiseres, er det en god idé at kigge på, hvad kernen af en DJ's job i virkeligheden er.

DJen's job må være at skabe en god fest eller en god stemning til festen gennem musikken. Det hører man i hvert fald sagt tit i forbindelse med DJ'ens job. Men hvad er det så der skaber denne gode stemning til en fest?

Først og fremmest er man nødt til at have en stor viden indenfor musik. Det er jo vigtigt at vide noget om de virkemidler man har. Desuden er man nødt til at have en mening om, hvad der er god musik. Som oftest er det også DJens job at eje og medbringe musikken.

Thomas Knak fortæller i Ud & Se april 2003 om folks angst for stilhed. Udover alkoholen er det også den konstante musik, som gør at folk i byen er mindre nervøse og i bedre humør, man er nærmest bange for stilhed. En af DJ'ens vigtige opgaver er at holde gang i musikken; og holde beatet i gang, hvis det er meningen at publikum skal kunne danse.

Det er så DJens job at vælge den helt rigtige musik til at sætte den stemning man ønsker, og til at tilfredsstille publikums smag og ønsker i det omfang der harmonerer med resten.

Det er også tit en vigtig opgave for en DJ at have en vis publikumskontakt. Det gælder både for høre deres ønsker, men i høj grad også at formidle musikken ud. Publikum kan have spørgsmål til musikken, og man har mulighed for at promovere musikken. Især hvis man ønsker at virke som trendsætter, er det vigtigt med publikumskontakt.

Det er DJens opgave konstant at vurdere om noget skal ændres eller tilpasses. Hvis et nummer får folk til at forlade dansegulvet, er det DJen der bør vurdere, hvad der skal til for at lokke folk tilbage. Og omvendt, hvis noget får publikum til at gå ud på dansegulvet, er DJens opgave at forsøge at holde dem der.

Dertil kommer en masse faktorer, som er med til at lægge en stemning, som DJen desværre ikke altid har kontrol over. Det være sig ting som mængden af publikum, publikums opførsel, udvalg i baren, barpriser, indgangspriser, udsmykning, lys og røg.

De arbejdsopgaver, som er krævet for at DJen kan udføre sit job, men som ikke er en del af jobbet's kerne, vil jeg betegne som DJens sekundære arbejdsopgaver. Det kan f.eks. være ting som at slæbe DJ-kufferten hen til klubben.

Med en DJ på job

Jeg vil forsøge at beskrive et eksempel på en arbejdsdag for en DJ meget detaljeret, dette skal senere være grundlag for en analyse af hvilke arbejdsopgaver, der kan automatiseres på en computer; men er også en analyse af målgruppen og situationen for produktet.

DJen lægger hjemmefra en plan for, hvad han vil spille. Hvor nøjagtigt han gør dette kan variere, måske har han blot en forestilling om hvilke genrer han vil spille og måske nogle idéer om hvilke numre, men det kan også være at han lægger en helt nøjagtig plan, en playliste, for hvilke numre, han vil spille, og i hvilken rækkefølge. Hvis han laver en nøjagtig playliste, kan det være at han også noterer cue-tider⁴ for de enkelte numre. Desuden vælger han hvilke CDer, han vil tage med sig, og putter dem i sin DJ-kuffert.

⁴ Se ordforklaring.

På selve aftenen tager han så hen på klubben medbringende DJ-kufferten. Han er der tidligt så han kan sætte udstyret op; eller hvis det er sat op, så tjekke at det er sat ordentligt op og fungerer. Når han ved, at alting fungerer som det skal, venter han på, at klubben åbner.

Når klubben så åbner (eller lidt før) starter han musikken. Den første CD findes frem af kufferten og sættes på, det rigtige musikstykke vælges, lydniveauet tjekkes og normaliseres⁵, og der trykkes på play.

Det næste nummer vælges eller aflæses fra playlisten, CD'en findes, sættes på, og det ønskede nummer vælges. Cue-punkterne findes, dette gøres ved først at finde punktet nogenlunde ved afspilning og trykke på pause, derefter bruger man en stutter-funktion⁶ til at finde punktet præcist. Man gennemlytter nummeret for at finde ud af præcist, hvordan man vil mixe det. Man tjekker lydstyrken og normaliserer. Til sidst venter man så på, at det første nummer nærmer sig slutningen, så gør man sig klar til at starte det næste. Når det rette tidspunkt kommer trykker man play og foretager mixet, dvs. man skruer op for det nye nummer og bagefter ned for det første. Denne procedure gentages så ca. hvert femte minut resten af aftenen, en gang for hver stykke musik.

I løbet af aftenen kommer nogle fra publikum op og ønsker et par numre. Hvis DJ'en har nummeret med, er det nu hans opgave at overveje, om han synes det ville være godt at spille. Hvis han gerne vil spille det, skal han finde ud af hvornår det ville passe at spille det i forhold til den anden musik. Mange DJs vil som oftest afslå ønsker, men ønsker kan stadig bruges som inspiration og som indikator af hvilken retning, publikum ønsker musikken skal gå.

Andre fra publikum kommer op og spørger hvilket band, der bliver spillet, eller hvad en bestemt sang hedder. Det fortæller DJ'en selvfølgelig. Nogle af dem kommer måske også bare op for at sige at de godt kan lide netop det nummer, som bliver spillet. Så ved DJ'en at han gør noget rigtigt.

Hen på morgenen bliver DJ'en bedt af klubbens ejer om at skrue ned for musikken, så de kan meddele, at det er sidste runde i baren. Helt til sidst bliver han bedt om at spille aftenens sidste nummer og runde festen af.

Efter en god nat pakker han sammen, får betaling for sit arbejde og tager hjem.

⁵ Se ordforklaring.

⁶ Se ordforklaring.

Automatisering af DJ-jobbet

Grundlaget for, at man kan foretage en automatisering af en proces, er at der er tale om gentagelser, som ikke kræver menneskelig intuition, vurdering eller kreativitet, men som kan opstilles i et simpelt matematisk sprog.

I DJens job ligger der, som man kan se i forgående afsnit, en række gentagelser, som ikke kræver kreativitet eller intuition, men som kan gøres på præcis samme måde hver gang. Men der er så andre dele af jobbet, som kræver et menneske, hovedsageligt de dele som er DJ-jobbets kerne og blev beskrevet i afsnittet "Kernen i DJens job".

Min påstand er at ved at automatisere dele af DJens job, giver man ham en mulighed for at koncentrere sig om de dele af jobbet der kræver hans viden, intuition og kreativitet. Derudover giver det ham en mulighed for at slappe mere af og derved blive mindre stresset.

Det næste nummer vælges eller aflæses fra playlisten, CDen findes, sættes på, og det ønskede nummer vælges. Cue-punkterne findes, dette gøres ved først at finde punktet nogenlunde ved afspilning og trykke på pause, derefter bruger man en stutter-funktion⁷ til at finde punktet præcist. Man gennemlytter nummeret for at finde ud af præcist, hvordan man vil mixe det. Man tjekker lydstyrken og normaliserer. Til sidst venter man så på, at det første nummer nærmer sig slutningen, så gør man sig klar til at starte det næste. Når det rette tidspunkt kommer trykker man play og foretager mixet, dvs. man skruer op for det nye nummer og bagefter ned for det første. Denne procedure gentages så ca. hvert femte minut resten af aftenen, en gang for hver stykke musik.

Diskussion af arbejdsopgaver

Med baggrund i det citerede afsnit, vil jeg diskutere skridt for skridt, hvordan en automatisering vil kunne gavne DJen til en nemmere arbejdsproces.

Huske playlister

Mange DJs vælger at forberede sig hjemmefra ved at skrive en playlist til hele eller en del af festen og derved lette arbejdet med at finde den rigtige musik. Playlister på en computer gør ikke blot det, men hjælper også med at finde musikstykkerne frem. Og med et lager af gode playlister kan man have værktøjer til hurtigt at sammensætte god musik.

Desuden kan computeren hjælpe dig med at huske hvilke numre, rent faktisk er blevet spillet den enkelte aften, disse lister bruges når der skal betales KODA-afgift, og bruges også tit på klubbers hjemmesider til at vise eksempler på musikken på stedet, og af DJen selv, når han søger arbejde.

Her vil automatisering kunne gavne DJen en del.

Finde musikken frem

Hvis du arkiverer din musik på din computer efter et godt system, så bliver dette system ikke ødelagt mens du Djer, da du ikke skal flytte filerne for at afspille dem. Hvorimod en kuffert med CDer hurtigt kan blive rodet, hvis det hele skal gå stærkt, og man ikke får lagt Cderne tilbage på de rigtige pladser. Desuden kan man bruge søgefunktioner på

⁷ Se ordforklaring.

computere til f.eks. at finde frem til et nummer, hvor du kender titlen, men har glemte kunstneren.

Her vil der være nogle fordele for DJen ved automatisering, hvis han holder et ordentligt system i sine musikfiler.

Finde cue-tider

Cue-tider er de tidspunkter hvor nummeret skal startes og stoppes for at lave et godt mix. Her er nøjagtighed ret vigtig for at lave et mix som lyder godt. Disse bør findes meget præcist, hvis man ønsker at lave et godt mix.

Der findes to meget gennemtestede måder at finde cue-tider på. Den analoge, hvor man med en plade på en pladespiller kører fingrene rundt og slipper pladen igen for i hovedtelefonerne at lytte sig frem til det rigtige sted at starte pladen, inden nummeret startes på lydanlægget. Og den digitale, hvor man bruger en stutter-effekt, som gentager en lille bid af musikken igen og igen i hovedtelefonerne, og så kan man søge sig langsomt frem til det helt rigtige punkt.

Jeg kan ikke se andre muligheder end at benytte sig af den normale digitale metode i vores software.

Huske cue-tider

Hver gang en DJ skal sætte et nyt nummer på, så skal cue-tiderne findes, så nummeret er klar til at blive mixet. Da der er en meget stor sandsynlighed for at DJen har spillet dette nummer før, så virker det som dobbeltarbejde at skulle finde dem igen.

At kunne gemme disse informationer, vil gøre det muligt at springe denne del af arbejdsprocessen over senere. Der er derfor meget store tidsbesparelser at hente for DJen.

Vælge type af mix

I nogen tilfælde vil man være interesseret i at mixe to numre lidt anderledes end man plejer, man kan f.eks. vælge at man vil lade den første nummer spille færdigt og først der starte introen på det næste nummer. Det vil kunne være interessant, hvis det er et populært nummer med en god intro, og gøres for at lokke folk på dansegulvet. Et sådan valg ville man før kunne tage mere eller mindre "on the fly", i vores software vil det være nødvendigt at fortælle computeren at dette skal ske.

Der er her tale om en lille forlængelse af arbejdsprocessen, den er dog næsten ubetydelig lille.

Normalisere

Som med cue-tiderne, skal DJen hver gang et nyt nummer sættes på tjekke lydstyrken og normalisere, så alle musikstykkerne lyder lige høje.

Det er ret simpelt at normalisere, man skal finde det sted hvor musikken er højest, det er normalt de steder hvor stortrommen og/eller lilletrommen slår. Her skal man så tilpasse så lydstyrken er så høj som muligt uden at overstyre, også kaldet peake, altså bliver så høj at lydsignalet bliver forvrænget.

Det er ret enkelt at få computeren til automatisk at sætte en normaliseringsværdi for nummeret, som så kan bruges under afspilning, så alle numre spiller ved samme lydstyrke. Hele denne arbejdsproces kan altså automatiseres.

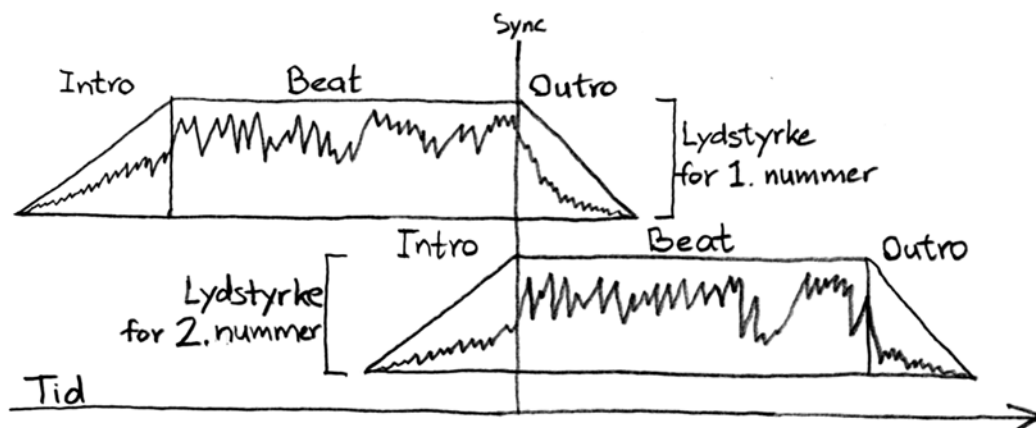
Huske normaliseringsværdi

Ved at gemme normaliseringsværdien efter, at den er fundet første gang, vil man ikke efterfølgende behøve at finde den, desuden vil en evt. manuel korrektion af værdien på denne måde også kunne gemmes..

Fortage mix

Her er det så det begynder at blive svært. Der vil altid være dem som mener at et menneske kan lave et bedre mix end en computer. Den påstand vil jeg såmænd heller ikke forsøge at afkræfte. Men spørgsmålet er om alle DJs er så teknisk dygtige at en computer ikke ville kunne gøre det mindst lige så godt, og så kan DJen koncentrere sig det mere om at udvælge den rigtige musik.

Der findes mange måder at mixe fra et stykke musik til et andet, men en af dem jeg selv har brugt meget og som jeg på mange måder opfatter som den optimale, vil jeg prøve at beskrive: Det kræver at man ved hvor lang tid introen varer på det nummer man skal mixe til, og hvor lang tid outroen varer på det man mixer fra. Intro og outro kalder jeg den tid hvor der intet rigtigt beat er i musikken først og sidst i nummeret. Ved at lægge tiden på intro og outro sammen ved man hvornår man skal starte det nummer man vil mixe til. Lad os sige at introen på det nummer, man vil mixe til, er 15 sekunder og outroen på nummeret, man vil mixe fra, er 10 sekunder, så skal man starte nummeret man vil mixe til, når der er 25 sekunder tilbage af det første nummer. Man hører så introen langsomt bygge op mens det andet nummer er ved at slutte. Når beatet på første nummer slutter, da starter beatet på det andet nummer. Det første nummers outro lader nu nummeret glide ud, mens folk danser til det nye. Denne type mix vil jeg fra nu af kalde cue-mixing, da jeg ikke kender til at den skulle have noget andet navn. Alle de metoder jeg personligt bruger når jeg DJer, er afarter af denne type mix, og kan alle beskrives ud fra denne "algoritme".



Hele mixingen ville kunne fortages automatisk udfra de instruktioner som DJen har givet, dvs. at han ikke behøver at være aktiv under selve mixingen. Det er en stor fordel ikke at være slave af tiden, men være istand til at udføre sine arbejdsopgaver forud. DJen vil hurtigt mærke stressfaktoren blive mindre.

Beatmixing

Der findes dog én anden meget brugt måde at mixe på, den kaldes beatmixing. Denne type mixing benyttes hovedsageligt af techno-, house-, hiphop- og RnB-DJs, hvorimod DJs som spiller blandet pop, rock, metal, goth og meget andet ikke har så meget glæde af denne type mix.

Der er flere grunde til at jeg har valgt ikke at beskæftige mig med beatmixing i dette projekt. Hovedsageligt er det fordi det kræver en ret avanceret lyd-analyse at finde beatet i et stykke musik og tilpasse dette til et andet beat, og jeg regnede ikke med at kunne mestre det på et halvt år. Desuden lagde jeg mærke til under min research af eksisterende DJ-værktøjer, at de alle forsøgte at gøre netop beatmixing muligt. Da der er mange DJs som har nogle helt andre behov var det naturligt at lave noget, der henvender sig til dem. Beatmixing er dog en funktion, som kunne være interessant at tilføje i en senere version af softwaren.

Gøre klar til næste nummer

Fordi hele mixingen af flere numre frem kan klares af computeren, hvis bare playlisten er skrevet, har DJen en mulighed for at forlade mixerpulten, som han ikke tidligere havde. Det er nu muligt at holde en lille pause, gå på toilet, eller hente noget at drikke, uden at risikere at musikken går i stå.

Dette vil give meget mere humane arbejdsbetingelser for DJs.

Sortering og opsummering

Gentagende processer	Krav om menneskelig interaktion
Huske playlister	Skrive playlister
Finde musikstykker	Vælge musikstykker
Huske cue-tider	Finde cue-tider
Normalisere	Tage imod ønsker ⁸
Huske normaliseringsværdi	Publikumskontakt
Fortage mix	Tilpasse playlisten løbende
Gøre klar til næste nummer	

I figuren har jeg forsøgt at opstille DJ-jobbet i de dele, som jeg mener at man med fordel ville kunne automatisere, og dem, som kræver menneskelig interaktion.

⁸ Grunden til at dette punkt står under krav om menneskelig interaktion er at publikum tit vil bruge det som en undskyldning for en personlig kontakt med DJen, desuden er det tit lige så vigtigt at få en idé om hvad publikum generelt ønsker som et konkret musikønske.

Diskussion af fordele og ulemper

Vægt

Det er tit meget tungt for en DJ at slæbe rundt på CDer, især hvis man skal spille længe og være i stand til at opfylde ønsker, kræver det at man har rigtig mange CDer med. Der kan være 100 CDer i en almindelig DJ-kuffert, og 500 hvis man bruger plastlommer til CDerne. Man kan dog ikke slæbe kufferten særlig langt, hvis den er helt fyldt, da den vejer op imod 15 Kg.

Alt efter hvilken type hardware man vælger, vil en computer veje mellem 3 og 20 Kg. Men fordi vægten af en harddisk er uafhængig af hvor meget der ligger på den, behøver man som DJ ikke længere vælge hvilke CDer man skal have med, man kan i stedet bare tage hele samlingen med. Man behøver altså ikke længere at stå og mangle noget musik, som man har valgt at lade ligge derhjemme. Med den størrelse moderne harddiske har i dag, har man med en enkelt harddisk mere kapacitet end flere fyldte DJ-kufferter kan rumme.

Brugervenlighed

Ved at fjerne en del af de tekniske dele af DJing, opnår man større brugervenlighed og tilgængelighed, der gør det nemmere for såvel nye som erfarne DJs. En mindre teknisk og mere automatiseret indgang til DJing vil bringe fokus hen på musikken fremfor teknikken.

Der bør være et simpelt interface med kun den funktionalitet som er nødvendig for at udføre et godt stykke arbejde som DJ, og værktøjerne skal virke naturlige at arbejde med. Så vil det føles rarere at stå bag en computer med SBAM end bag en DJ CD-afspiller.

Frihed

Ved at computeren udfra en playlist, som DJen bygger op, mixer musikken for DJen, får DJen en frihed, han aldrig har haft før. Han har mulighed for at gå på toilet, hente noget at drikke i baren, tale uforstyrret med publikum eller bare koncentrere sig det mere om at vælge den helt rigtige musik.

Denne øgede frihed vil også gøre det mere menneskeligt for de DJs, der DJer på en klub i 10 timer i træk, og hjælpe til med at undgå, at de får stress.

Menneskelige fejl

Man kan nemt komme til at begå fejl når man mixer numre sent på aftenen og natten, især hvis man ikke er så øvet som DJ. Men hvis computeren overtager alle de kritiske dele af arbejdet, så mindskes risikoen for fejl enormt. Man kan helt fjerne crossfader⁹ og knapper som eject, pause og lignende, som kan have pinlige konsekvenser, hvis man trykker på dem på et galt tidspunkt.

Dette kan være betydeligt både for nye DJs og deres arbejdsgivere, desuden kan det kun være en fordel for selv den mest erfarne DJ, at han ikke længere kan begå fejl.

⁹ Se ordforklaring.

Tekniske fejl

Jo mere teknik, jo flere tekniske fejl, desværre. Når man vælger at bruge en computer i stedet for en CD-afspiller, opstår der pludselig en hel ny række af muligheder for fejl, som man må tage hensyn til. Man er nødt til at tjekke at hardware, operativsystem, drivere og software fungerer som de skal.

Et andet problem kan være strømafbrydelser. Efter en strømafbrydelse tager det måske 10 sekunder at få startet en CD-afspiller igen, så musikken kan fortætte. Derimod kan det tage flere minutter at boote en maskine, logge ind og starte softwaren. Hvis der er sandsynlighed for strømafbrydelser bør man nok enten anskaffe sig en UPS eller have en CD-afspiller og en CD parat i det uheldige tilfælde, at strømmen skulle ryge.

Lydkvalitet

OGG er som MP3 et format som pakker lyden med en forringelse af lydkvaliteten, det er dog muligt at indstille hvor stort et tab man vil have, mod at filen så bliver større. Selvom en OGG fil aldrig ville være helt lige så god som originalen på CD, kan man dog stadig med de rigtige indstillinger opnå en kvalitet, som opfylder ens behov. Man må derfor, når man ripper¹⁰ sin musik ind på computeren, sørge for at gøre det i en kvalitet, som er tilstrækkelig. Men jeg mener sagtens man kan opnå en kvalitet, hvor kvalitetstabet er helt uden betydning.

Den bedste måde at teste om kvaliteten er god nok er ved at sammenligne. Man sætter udstyret op og spiller først originalen, derefter sætter man OGG-filen til at spille over samme anlæg ved samme lydstyrke. Hvis man ikke kan høre forskel, så må kvaliteten være god nok¹¹.

Slid

Ved kun at medbringe sin musik på computer slipper man for det enorme slid på Cderne, det ellers kan give at slæbe rundt på dem, når man DJ'er. Så man kan slippe for at have det dårligt med at slide sine dyrebare og sjældne skiver.

Tyveri

Noget af det mest forfærdelige en DJ kan forestille sig er at få stjålet sine CDer. Uanset om det er en langfingret gæst som snupper en enkelt sjælden CD ud af kufferten, eller det er en tyv som stjæler en hel kuffert fuld, er det en forfærdelig tanke.

Hvis man bruger en computer og SBAM, ville man, hvis man fik stjålet noget, få stjålet hele sin computer. Men man skal huske at sætte det i forhold til hinanden. Hvis man får stjålet sin computer, så koster det fra 5.000,- op til måske 20.000,- at få en ny. Hvis ikke man har backup af sin musik, så kan man forberede sig på at bruge noget tid på at lægge det ind igen, men det er ikke mistet. Og når man køber ny computer får man tit mere for pengene end man fik, da man købte den gamle, så det er en god mulighed for at opgradere. Hvis man derimod mister sin kuffert med CDer, så risikerer man ikke at kunne skaffe al musikken igen. Derudover, hvis det "bare" var en kuffert med 100 CDer, koster det stadig ca. 12.000,- at erstatte, og hvis det var en kuffert med 500 CDer i plastlommer, koster det ca. 60.000,-.

¹⁰ Se ordforklaring.

¹¹ I litteraturlisten vil man desuden kunne finde link til "The Digital Audio Format Test".

Kan DJen helt fjernes?

Det er meningen at SBAM skal gøre DJs til bedre DJs, ikke gøre dem overflødige, det ville være meget trist hvis DJen blev 100% erstattet af en maskine. Den personlige kontakt imellem publikum og musik ville forsvinde, og valget af musik ville blive taget udfra tilfældigheder og hitlister, og ikke udfra et personligt valg, men udfra en vurdering af en masse forskellige faktorer - musikken ville miste personlighed.

MTV har faktisk valgt at lade en computer bestemme hvilken musik der bliver spillet i hovedparten af deres sendetid. De vælger simpelthen efter nogle forskellige kriterier en tilfældig video og spiller den. Det giver intet flow i musikken, ingen langsom udvikling af genrer over nogle numre, desuden mister man lidt muligheden for at være trendsetter, hvis man hovedsageligt spiller det musik, som er på hitlisterne i forvejen.

Jeg mener helt klart ikke at man kan undvære DJen. Uden DJen ville der ingen ordentlig kvalitetskontrol være, der ville ikke være en trendsetter, der ville ikke være nogen til at give musikken et naturligt flow, og hver aften med den samme computer ville nemt komme til at ligne hinanden. Så det er vigtigt at understrege at dette stykke software er designet så DJs kan blive bedre DJs, og ikke for at vise at de kan undværes.

Software

Stjerneanalyse

Hensigt

Udvikler

Hensigten med vores software er at stille et værktøj til DJing på computer til rådighed for vores brugere.

Der er tale om en form for player, som kan mixe numre ind over hinanden. Den slags software findes der ingen egentlig design-genre indenfor. Nogle forsøger enten at ligne en almindelig player med udvidet funktionalitet, men de fleste vælger dog at forsøge at ligne det hardware, altså DJ CD-afspillere, som de erstatter.

Bruger

Brugeren er højest sandsynlig enten en erfaren DJ, som nu gerne vil til at DJe på computer, eller en computerbruger, som gerne vil forsøge sig som DJ.

Som erfaren DJ er man vant til DJ CD-afspillere, DJ pladespillere og DJ mixere. Man har en masse musik på CD og vinyl, og ved hvordan man mixer musik sammen. Hensigten for en erfaren DJ ville være at blive istand til at DJe fra en computer for at kunne opnå alle de fordele, som det bringer med sig.

Som ny DJ er hensigten med at benytte netop DJ software fremfor de traditionelle værktøjer, at føle sig mere sikker med øgede brugervenlighed og nedsatte risiko for menneskelige fejl. Desuden har denne bruger måske allerede al sin musik liggende på sin computer.

Mål

Udvikler

Målet er at udvikle et effektivt stykke open-source software, som kan udfordre de eksisterende kommercielle programmer på markedet. Det er et ønske at fremvise noget funktionalitet som ikke er set før, i håb om at skabe fokus på nogle mangler, og et mål er måske at kunne påvirke markedet. Det er også et mål at forsøge at sætte en ny standart for hvordan man tænker på DJing på computer, hvor den traditionelle tankegang er at gå efter at emulere eksisterende hardware, til at være mere nytænkende og innovativ. Desuden er det vigtigt at vise at man sagtens kan lave open-source software, som er nytænkende og nyskabende og ikke bare kopier af andre produkter.

Bruger

Brugernes mål med at bruge produktet er at sætte sig istand til at DJe ved hjælp af en computer. Valget af software tages hovedsageligt ud fra funktionalitet og brugervenlighed, men man skal også kunne føle sig hjemme i programmet. Det skal virke cool samtidig med at det skal virke professionelt.

Ressourcer

Udvikler

Rent økonomisk er der ingen penge til rådighed til udviklingssiden. Derfor må projektet køre på frivillig basis af udviklerne Rasmus Berlin og Bent Bisballe Jensen, som står til rådighed for dette projekt. I og med at projektet bliver lavet som en hovedopgave for Rasmus Berlin på multimediedesigner uddannelsen, har vi ca. 5 måneder til rådighed til

udvikling af en prototype og udarbejdelsen af denne rapport. Derudover har vi en sommerferie til at færdiggøre arbejdet, derefter skulle der gerne ligge et stykke færdigt, funktionelt software.

På hardware siden har der været nødt til at blive foretaget nogle investeringer. Rent økonomisk har det været afgrænset af hvad min personlige økonomi har kunnet bære. Der er tale om investeringer for omkring 5000,- kr. Samme tidsfrist som for softwareudviklingen gælder for udviklingen af hardwareplatformen, som gerne skulle kunne tages i brug senest efter sommerferien, men skal dog være funktionel under den mundtlige eksamen.

Det betyder dog ikke at udviklingsarbejdet er et overstået kapitel. Udviklingen af fremtidige versioner af softwaren er allerede på tegnebordet, dog lagt på hylden indtil version 1.0 er helt færdigudviklet.

Bruger

Det kræver en række ressourcer hos brugeren af softwaren. Først og fremmest skal du have en egnet computer, det skal være en computer, der er hensigtsmæssig at slæbe rundt på, og som har et ordentligt lydkort, og en dertil hørende skærm, som ligledes er til at slæbe på. Læs mere i afsnittet "Hardware".

Desuden kræver det at man installerer Linux, indtil vores software bliver portet¹² til Windows. Det kræver lidt snilde og kendskab til computere at installere Linux, og det kræver noget harddiskplads.

Det kræver også at man ripper alle sine CD'er og lægger dem ind på sin computer. Og så kræver det, at man har mod på at sætte sig ind i noget nyt og forsøge at gøre tingene på en ny måde.

Tradition

Udvikler

Da dette er det første stykke software, som vi udvikler¹³, har vi ingen visuel identitet. Men ved at vælge at lave vores GUI i GTK+ 2.0, kommer det til at ligne andet open-source software, der også benytter sig af GTK, og derved får man en fælles visuel identitet med andet open-source software.

Som jeg skriver i afsnittet "Hensigt", så er det mest almindelige at gøre indenfor DJ software at efterligne noget eksisterende hardware.

Vi har med vores projekt forsøgt at komme væk fra idéen om at efterligne fysisk hardware, og i stedet koncentrere sig om de muligheder computeren giver os. Derfor mener jeg også at vi skal have et design som afspejler dette. Jeg har valgt at lægge mig lidt op af WinAmp's design, da den på mange måde har sat standarden for hvordan players på computere bør se ud. Dog er funktionaliteten så meget anderledes end både en player og en DJ CD-afspiller, at der er brug for at finde en ny måde at illustrere funktionerne på, for at opnå størst mulig brugervenlighed.

Bruger

Som sagt er brugeren højst sandsynlig enten en erfaren DJ, som nu gerne vil til at DJ'e på computer, eller en computerbruger, som gerne vil forsøge sig som DJ. De har hver især en idé om hvordan tingene bør se ud og fungere, de vil være forberedte på at det måske er lidt anderledes, men hvis det er alt for anderledes er der chance for at de springer fra igen. Hvis det afviger for meget fra noget man kender, kan det virke forvirrende, og så er man tit hurtigt til at give op, og holde sig til det, man kender.

¹² Se ordforklaring.

¹³ Vi har lavet projekter sammen før, men aldrig noget som er blevet færdigt.

Man skal altså finde et design, som folk, der er vant til WinAmp, kan nikke genkendende til, men som også folk, der er vant til DJ CD-afspillere, kan genkende. Samtidig skal det naturligt indkorporere al det øgede og ændrede funktionalitet.

Situation

Brugerens første møde med vores produkt kunne være på nettet, hvor brugeren støder på vores hjemmeside efter at have søgt efter software til DJing. Brugeren vil på den måde have en del tid til at undersøge programmet. Det kræver at der er noget forklarende og direkte information på sitet. Derefter skulle han gerne downloade softwaren og afprøve det selv. En anden mulighed er at han ser en anden DJ benytte netop vores software på en klub, og bliver interesseret den vej.

Men der, hvor softwaren for alvor skal stå sin prøve, er på klubberne. Hvis det viser sig ikke at være hensigtsmæssigt eller giver andre problemer her, så vil softwaren miste sin eksistensberettigelse, og DJs vil holde op med at bruge det. Viser det sig derimod at være praktisk og ubesværligt, så vil DJing på computer vinde frem.

Men hvordan er situationen bag DJ-pulten på en natklub? Det er fra sent om aftenen til langt ud på natten, der er mørkt og tilrøget og masser af larm og høj musik. Med jævne mellemrum bliver man afbrudt af en publikum, der vil spørge om noget. Den skærm, man har med, kan være alt fra en 5" sort-hvid tv-monitor til en 15" TFT-skærm.

Det er vigtigt at man kan ramme knapperne under stress og at man kan se alle de vigtige informationer på selv en meget lille skærm. Samtidig skal programmet kunne vises på en skærm med ret lav opløsning, helt ned til 640x480 eller i hvert fald 800x600.

Motiv/Trend

Jeg har valgt at gå efter et seriøst og professionelt look gennem et meget enkelt og modernistisk design. En modernistisk designdialekt er desuden den mest oplagte til et overskueligt grafisk program-interface.

Program design

Initialisering

Initialiseringsfasen ligger i starten af `main()`, og er den del som udføres med det samme programmet starter.

main.c

```
int main (int argc, char *argv[]) {
    if (!g_thread_supported ()) g_thread_init (NULL);
    else fprintf(stderr, "Threading not supported!\n");

    gdk_threads_init ();
    gtk_init (&argc, &argv);

    //Parameter parsing
    if(argc < 2) {
        fprintf(stderr, "Usage: %s [oggfile]\n", argv[0]);
        exit(1);
    }
    sprintf(cue.dec.fname, argv[1]);

    //Model
    model();

    //Load playlist
    loadplaylist();

    //Set timere
    gtk_timeout_add(500, timer_500ms, NULL);
    gtk_timeout_add(10, timer_10ms, NULL);

    //Initialisering slut.

    gdk_threads_enter ();
    gtk_main ();
    gdk_threads_leave ();

    fprintf(stderr, "Shutting down libao\n");
    ao_shutdown ();
    return 0;
}
```

Parameter parsing

De parametre, som programmet er blevet kørt med, tjekkes og lagres. Hvis programmet startes fra en kommandolinje, kan brugeren tilføje nogle parametre, som kan fortælle programmet forskellige ting. F.eks. kunne programmet måske i fremtiden tage en playliste med linjen "sbam -playlist min_playliste.sbam"; programmet ville så starte med at laade denne playliste ind.

Model

`model()` som udføres fra `main` i initialiseringen, ligger i sin egen fil: `model.c`. Her bygges vinduerne: mixer, playlist, cue, options og about op, og til sidst tegnes de.

model.c

```

//Funktioen der bygger mixeren op af GTK widges, byggestenene i GTK.
static void make_mixer() {
    //Et nyt GTK vindue laves.
    dlg[DLG_MIXER].widget = gtk_window_new(GTK_WINDOW_TOPLEVEL);
    //Størrelsen af vinduet sættes til 380x180 pixels.
    gtk_widget_set_size_request(dlg[DLG_MIXER].widget, 380, 180);
    //Titlen på vinduet sættes til "SBAM-Mixer".
    gtk_window_set_title(GTK_WINDOW(dlg[DLG_MIXER].widget), "SBAM-Mixer");
    //Vinduet sættes til ikke at kunne ændre størrelse.
    gtk_window_set_resizable(GTK_WINDOW(dlg[DLG_MIXER].widget), FALSE);
    //En container af typen "fixed" laves.
    dlg[DLG_MIXER].container = GTK_CONTAINER(gtk_fixed_new());
    //Containeren lægges ind i vinduet.
    gtk_container_add(GTK_CONTAINER(dlg[DLG_MIXER].widget),
        GTK_WIDGET(dlg[DLG_MIXER].container));

    //En progress bar laves.
    dlg[DLG_MIXER].ctls[DSP_PROGRESS] = gtk_progress_bar_new();
    //Progress baren sættes til en størrelse af 315x10 pixels.
    gtk_widget_set_size_request(dlg[DLG_MIXER].ctls[DSP_PROGRESS], 315, 10);
    //Progress baren lægges ind i containeren i punktet (10,10).
    gtk_fixed_put(GTK_FIXED(dlg[DLG_MIXER].container),
        dlg[DLG_MIXER].ctls[DSP_PROGRESS], 10, 10);

...her er flere widgets...

    //En toggle knap14 laves.
    dlg[DLG_MIXER].ctls[BTN_PLAY] = gtk_toggle_button_new();
    //Et billede laves fra filen "FILE_PLAY".
    dlg[DLG_MIXER].ctls[IMG_PLAY] = gtk_image_new_from_file(FILE_PLAY);
    //Billedet lægges ind i knappen.
    gtk_container_add(GTK_CONTAINER(dlg[DLG_MIXER].ctls[BTN_PLAY]),
        dlg[DLG_MIXER].ctls[IMG_PLAY]);
    //Knappens størrelse sættes til 40x30 pixels.
    gtk_widget_set_size_request(dlg[DLG_MIXER].ctls[BTN_PLAY], 40, 30);
    //Knappen lægges i containeren i punktet (10,120).
    gtk_fixed_put(GTK_FIXED(dlg[DLG_MIXER].container),
        dlg[DLG_MIXER].ctls[BTN_PLAY], 10, 120);
    //Funktioen " mixer_toggle_button_handler" knyttes til knappen,
    //når den sender signalet "toggled", dvs. at den er blevet trykket på.
    gtk_signal_connect (GTK_OBJECT (dlg[DLG_MIXER].ctls[BTN_PLAY]),
        "toggled",
        GTK_SIGNAL_FUNC (mixer_toggle_button_handler),
        GINT_TO_POINTER (BTN_PLAY));

...her er flere widgets...

}

//Funktioen som laver og viser all vinduerne.
void model() {
    //Kører funktionen ovenover, som bygger mixeren.
    make_mixer();
    //Viser mixer-vinduet.
    gtk_widget_show_all(dlg[DLG_MIXER].widget);

...her er flere vinduer, der laves...

}

```

¹⁴ En knap, som kan trykkes ind, og bliver inde indtil, man trykker den ud igen.

Load playlist

`load_playlist()` udføres lige efter `model()` i `main`, og selve funktionen ligger i sin egen fil: `playlist.c` Her hentes playlisten frem og alle de vigtigste informationer om hvert element bliver placeret i en linket liste. Hvert element i listen inderholder information om kunstner, titel, længde o.s.v., men vigtigst af alt cue-punkterne, og til sidst ligger en pointer (`next`), som peger på det næste element i listen.

`decode.h`

```
typedef struct pl_entry {
    char *title;           //
    char *artist;         //
    unsigned int length;  //
    unsigned int size;    //
    _decode_params dec;   // Data om nummeret
    unsigned int in_fade; //
    unsigned int in_cue;  //
    unsigned int out_fade; //
    unsigned int out_cue; //

    struct pl_entry* next; //Pointer til næste element i listen
} pl_entry;
```

Grunden til vi har valgt at lægge disse informationer i en linket liste, er at det her er meget nemt at tilføje og fjerne elementer, samt flytte rundt på deres rækkefølge ved blot at ændre på hvor pointeren `next` peger. Desuden er der ikke nogen grænse for hvor meget listen kan udvides.

Hvis man i stedet havde placeret informationerne i et array, havde man været nødt til at reservere en bestemt mængde hukommelse til arrayet; hvis den reserverede plads så ikke var nok, ville man blive nødt til at oprette et nyt, større array og flytte alle elementerne over i det nye. Desuden hvis man fjerner et element i et array, så er man nødt til at flytte alle de efterfølgende elementer et tak frem i arrayet.

Vi har valgt have playlisten i XML format, og læse den med `libxml`. Dette giver gode muligheder for at udvide playlistens funktionalitet senere.

Set timere

Sidst i initialiseringsfasen sættes to timere. Disse sørger for at et stykke kode bliver udført med et specifikt interval.

Den ene timer, som er på 500 ms, dvs. at den udføres hver halve sekund, har til opgave at sikre at f.eks. progressbaren i mixer-vinduet og position slideren i cue-vinduet bliver opdateret, så man kan følge med i hvor langt numrene er nået.

Den anden timer er på 10 ms og udføres altså 100 gange per sekund. Denne vil have til hovedopgave at stå for kommunikationen med trådene¹⁵ i programmet. Trådene skal opfattes som helt individuelle programdele. Den eneste måde de kan tale med hinanden, er ved at lægge beskeder. Timeren tjekker så 100 gange i sekundet om, der er beskeder fra trådene, og sørger for at udføre det, som trådene beder om.

F.eks. når mixertråden kan se at tiden er inde til at starte den næste tråd, sætter den variabelen `mixer.startnext` til pointeren til næste nummer på playlisten. Timeren læser denne variabel, da der står noget i den, udfører den, hvad den skal. I dette tilfælde starter den en tråd med det nye nummer. Bagefter sætter den variabelen til `NULL`, så den ikke bliver startet endnu en gang, og så variabelen er klar til blive brugt til næste besked.

¹⁵ Hvad en tråd er vil blive uddybet i senere i dette afsnit, i underafsnittet "Tråde".

I dette tilfælde er det ikke så vigtigt, men hvis timeren skulle kunne udføre beskeder, som kunne komme meget hurtigt efter hinanden, så ville det kræve at beskederne blev sat i en kø. En sådan kø ville være en linket liste, hvor beskederne blev lagt i til sidst. Timeren ville så tage den første besked, udføre det, slette den og gå videre til næste besked. Det er muligt, at vi vil implementere en rigtig kø ved lejlighed.

Mixer

Efter initialiseringen venter mixeren¹⁶ på input fra brugen, før den gør mere.

Klargøring

Når brugeren så trykker "play", startes en ny tråd. Programmet henter informationer om nummeret, som skal spilles¹⁷ fra playlistens linkede liste, og filen åbnes. Desuden åbnes en forbindelse til lydkortet.

Afspilning

Et loop spiller filen igennem, hvor der bliver taget højde for ændringer i lydstyrke under afspilningen. Nye numre bliver mixet ind på de rigtige tidspunkter, og deres lydstyrke reguleres, så numrene langsomt blandes med hinanden. Desuden bliver der tjekket, om der skal fades ned eller op, hvis brugeren har trykket på fade-knappen.

decode.c

```
while (mixer.playing && !eof) {
    //Hvis der ikke skal fades, spil da almindeligt.
    if (mixer.fade_vol == 0) mixer.vol = 1.0;
    //Ellers, hvis der skal fades op eller ned, så gør dette.
    //Hvis der skal fades ned, vil mixer.fade_vol være lig 1 til sidst,
    //og den vil stoppe med at fade.
    //Hvis den skal fades op, vil mixer.fade_vol være lig 0 til sidst,
    //og den vil stoppe med at fade.
    else if (mixer.fade_vol < 0) {
        mixer.vol += 0.00001; mixer.fade_vol += 1;
    }
    else if (mixer.fade_vol > 1) {
        mixer.vol -= 0.00001; mixer.fade_vol -= 1;
    }
    //Hvis mixer.fade_vol er 1 pauser dekodningen.
    if(mixer.fade_vol != 1) {
        //Indlæsframe.
        long ret = ov_read (&vf, pcmout, MAX_FRAME_SIZE, 0, 2, 1,
                           &current_section);
        //Hvis der ikke er noget så stop, ellers fortsæt.
        if (ret == 0) eof = 1;
        else {
            //Er vi er nået til at skulle starte mix af næste nummer, så start.
            if(dec->frm == next_frm) {
                mixer.startnext = pl_current = pl_current->next;
            }
            //Er nummeret ikke nået sit første cue-punkt endnu, så spil ikke.
            if(dec->frm < pl->in_fade) {
                if(state != 0) {
                    fprintf(stderr, "[id:%d]{frm:%d}_\n", (int)pl, dec->frm);
                    state = 0;
                }
            }
        }
    }
}
```

¹⁶ Det jeg betegner som "mixeren" består af mixer kontrollerne (i.e. vinduet), vores mixertråd, samt de beskeder som GTK håndterer for os.

¹⁷ Der hentes desuden også informationer om næste nummer, for at programmet kan vide hvornår mixet skal foretages.

```
    }
    fade = 0.0;
}
//Er nummeret imellem de 2 første cue-punkter, fade da op.
else if(dec->frm < pl->in_cue) {
    if(state != 1) {
        fprintf(stderr, "[id:%d]{frm:%d}/\n", (int)pl, dec->frm);
        state = 1;
    }
    fade += fade_inc;
}
//Er nummeret er forbi sit 4. cue-punkt, stop da afspilningen.
else if(dec->frm > pl->out_fade) {
    if(state != 2) {
        fprintf(stderr, "[id:%d]{frm:%d}_\n", (int)pl, dec->frm);
        state = 2;
    }
    fade = 0.0;
    eof = 1;
}
//Er nummeret mellem de 2 sidste cue-punkter, fade da ned.
else if(dec->frm > pl->out_cue) {
    if(state != 3) {
        fprintf(stderr, "[id:%d]{frm:%d}\\\n", (int)pl, dec->frm);
        state = 3;
    }
    fade -= fade_dec;
}
//Er vi midt i nummeret, afspil da normalt.
else {
    if(state != 4) {
        fprintf(stderr, "[id:%d]{frm:%d}**\n", (int)pl, dec->frm);
        state = 4;
    }
    fade = 1.0;
}
//De forskellige lydstyrker ganges på PCM dataen.
float mult = mixer.vol * mixer.master * fade;
l = *(short*)&pcmout[0] * mult; // v
r = *(short*)&pcmout[2] * mult; // h
//De nye værdier lægges tilbage i pcmout
memcpy(pcmout, &l, 2);
memcpy(pcmout + 2, &r, 2);
//Vores frame afspilles
ao_play (device, pcmout, ret);
//Det undersøges om vores frame er højere end peak-værdien
if(mixer.peak < *(short*)&pcmout[0])
    mixer.peak = *(short*)&pcmout[0];
if(mixer.peak < *(short*)&pcmout[2])
    mixer.peak = *(short*)&pcmout[2];
//Vi går til næste frame
dec->frm ++;
}
}
```

Afslut

Når nummeret så er færdigt afspillet, lukkes forbindelsen til lydkortet, og tråden slår sig selv ihjel.

Cue

Klargøring

Når en fil åbnes i cue-editoren, åbnes filen og dekodes. Hele nummeret ligger nu altså pakket ud i hukommelsen.

Afspilning

Nummeret begynder nu at blive afspillet i et loop i `decode.c`, hvor der bliver taget hensyn til parametrene: `cue.playing`, `cue.speed`, `cue.stutter` og `cue.dec.frm`, som bliver sat af knapperes handler-funktioner i filen `controller_cue.c`.

`cue.playing` bestemmer, om filen bliver afspillet eller ej. Denne er 0, hvis stopknappen er trykket ind, og 1 i alle andre tilfælde.

`cue.speed` bestemmer ved hvilken hastighed, musikken bliver afspillet. Denne er 0, hvis pause er trykket ind, og 1 ved almindelig afspilning. Trykkes på "play baglæn" sættes denne til -1, og nummeret spilles baglæns; hvor "play hurtigt" sætte den til 2, og nummeret spilles i dobbelt tempo. O.s.v.

`cue.stutter` bestemmer, om stutter-effekten er slået til. Denne er 1, hvis stutter er slået til, og 0 i alle andre tilfælde.

`cue.dec.frm` bestemmer, hvor i musikken vi befinder os. Under almindelig afspilning bliver denne langsomt større, men den kan også bruges til at hoppe frem og tilbage i nummeret. Den bruges af stopknappen til at hoppe tilbage til starten af nummeret. Den bruges af "go to cue"-knapperne til at hoppe til cuepunkterne, og af "set cue"-knapperne til at sætte cue-tiderne. Og den bruges af "step"-knapperne til at hoppe skridt frem og tilbage i nummeret.

`decode.c`

```
//Hvis cue.playing ikke er 0, så skal der afspilles.
if(cue.playing) {
    fprintf (stderr, "[Playing]");
    //Så længe vi stadig afspiller, og vi ikke er nået udover enderne så
    //kør følgende loop.
    while((dec->frm < bufisz) && (dec->frm > -1) && (cue.playing)) {
        //Hvis stutter funktionen er slået til, så afspil bidder af
        //100000 frames frem til det aktuelle frame-nummer.
        if(cue.stutter) {
            int i;
            for(i = -100000; i < 0; i+=4) {
                if((dec->frm + i > 0) && (dec->frm + i < bufisz)) ao_play
                    (device, pcmbuf + dec->frm + i, 4);
            }
        }
        //Ellers afspil det aktuelle frame, og hop frem til næste, men
        //tag højde for cue.speed.
        else {
            ao_play (device, pcmbuf + dec->frm, 4);
            dec->frm += 4 * cue.speed;
        }
    }
    //Stop nummeret, og hop tilbage til starten.
    cue.playing = 0;
    dec->frm = 0;
    fprintf (stderr, "[done]\n");
}
```

Tråde

Et almindeligt simpelt program består af en række instruktioner, som udføres efter hinanden, dette kan man kalde for et program med én tråd. Men et program som benytter multi-threading har i samme program flere tråde, som bliver udført parallelt. Det vil sige at programmet kan udfører flere ting på samme tid¹⁸.

I vores tilfælde har vi brug for at vores GUI hele tiden kan bruges og bliver opdateret, samtidig skal programmet kunne spille op til 2 individuelle numre i mixer-vinduet og et tredje i cue-vinduet. Derfor er vi nødt til at benytte multi-threading.

Man kan forstå et multi-threading program som en samling af tråde. Selve programmet har noget hukommelse og nogle ressourcer såsom eksterne filer og grafik til rådighed. Trådene er så de dele, der udfører kode. Trådene deler så programmets hukommelse¹⁹ og ressourcer. De enkelte tråde udfører deres arbejde parallelt og individuelt, de har intet med hinanden at gøre bortset fra at de deler computer og hukommelse. Derfor, hvis tråden skal tale med en anden tråd i programmet, bliver den nødt til at ligge en besked i en variabel, som den anden tråd kan læse, og så skal den anden tråd sørge for at læse beskeden.

For at visualisere dette kan man forestille sig trådene som selvstændige individer, der udfører arbejdet på en travl fabrik. De kan ikke tale sammen fordi de hele tiden render rundt og ikke ved, hvor de andre er, derfor lægger de beskeder til hinanden. Programmet er så fabrikken, hvor trådene arbejder.

Libraries

Et library er en samling af færdigskrevet kode, som man kan bruge i sit eget program. Man bruger normalt libraries til at forenkle avancerede processer. Programmøren, som bruger andres færdigskrevne libraries, behøver ikke at sætte sig ind i, hvordan de fungerer internt. Han kan nøjes med hvordan "interfacet" er, altså hvilke parametre funktionerne tager, og hvad de gør.

F.eks. `ov_read` i `vorbisfile` tager en række parametre som fortæller hvordan man ønsker sit lydoutput fra en åbnet fil. Det er egentlig alt programmøren behøver vide. I virkeligheden laver denne funktion en masse ting, for den skal først læse filen, derefter analysere den pakkede data og dekode den, og til sidst skal den så levere lyddataene på den rigtige måde.

Vorbisfile

Vorbisfile er det library, som muliggør at læse OGG-filer. Libraryet muliggør alle former for læsning af OGG-filer. Vigtigst af alt kan man med `vorbisfile` decode OGG-filerne; men man kan også læse deres tags, også kaldet comments.

Vi har valgt `vorbisfile`, som er det officielle OggVorbis library, fordi vi vil bruge OGG-filer i vores system. Den vigtigste årsag til at vi har valgt at bruge OGG-filer er at det er et gratis og åbent format. Hvor MP3-formatet nu koster penge at få lov at lave players til. En anden vigtig grund er et OGG er kendt for at have den bedste lyd kvalitet af alle de forskellige komprimerede formater.

Der kan findes dokumentation til `vorbisfile` på:

<http://www.xiph.org/ogg/vorbis/doc/vorbisfile/>

¹⁸ I de fleste tilfælde vil de ikke blive udført parallelt men der vil blive skiftet frem og tilbage mellem dem så hurtigt så det for både programmør og bruger vil virke som samtidig.

¹⁹ De har dog hver deres egen stak til mellemregninger og lignende.

LibAO

Dette library er et simpelt audio output library. Det gør det muligt at sende rå lyddata via OSS (Open Sound System), ESD (ESoundD), ALSA (Advanced Linux Sound Architecture), Sun audio system (brugt på Solaris, OpenBSD, og NetBSD) eller aRts (Analog Realtime Synthesizer) til lydkortet. Med dette library slipper vi altså for at skrive specifikt til en af lydsystemerne. Dette library eksisterer ikke til Windows, og hvis/når programmet skal portes til Windows, så skal der findes en erstatning til dette.

Vi har valgt dette library, fordi det opfylder vores krav til funktionalitet og generalitet. Desuden er det udviklet af de samme folk, som står bag oggvorbis, der har ry for at lave gode, stabile libraries.

Der kan findes dokumentation til libao på:

<http://docs.maraudingpirates.org/libao-0.8.2/>

GTK+

Dette er et library til at lave grafiske interfaces. Libraryet indeholder en masse grafiske elementer, som er byggestene til et GTK+ interface. Dette library eksisterer til Linux og andre Unix-lignende platforme samt Windows. Det er derfor ikke noget problem at porte disse dele til Windows senere.

Vi har blandt andet valgt GTK+ på baggrund af dens muligheder for at blive portet til Windows. Desuden giver især GTK+ 2.0 et ret flot design til dit program.

Der kan findes dokumentation til GTK+ på:

<http://developer.gnome.org/doc/API/2.0/gtk/>

GDK

GDK kræves af GTK+, og står for funktionaliteten i interfacet og for at tegne til skærmen. Disse funktioner har sit eget library. Som GTK+, så findes GDK også til Windows.

Der kan findes dokumentation til GDK på:

<http://developer.gnome.org/doc/API/2.0/gdk/>

Glib

Dette er et generalt library, som indeholder en masse forskellige ting. Vi bruger Glib til multi-threading. Grunden til vi har valgt at bruge netop Glib til dette er at den ligesom GTK+ også findes til Windows, og det gør det lettere at porte programmet senere.

LibXML

Dette library kan parse en XML-fil. Dvs. den læser en XML-fil, fortolker den, og oversætter den til et C dataformat, som er nemt at læse for programmet. Ved at bruge libxml behøver vi ikke opfinde vores eget playlist-format, og skrive vores egen parser.

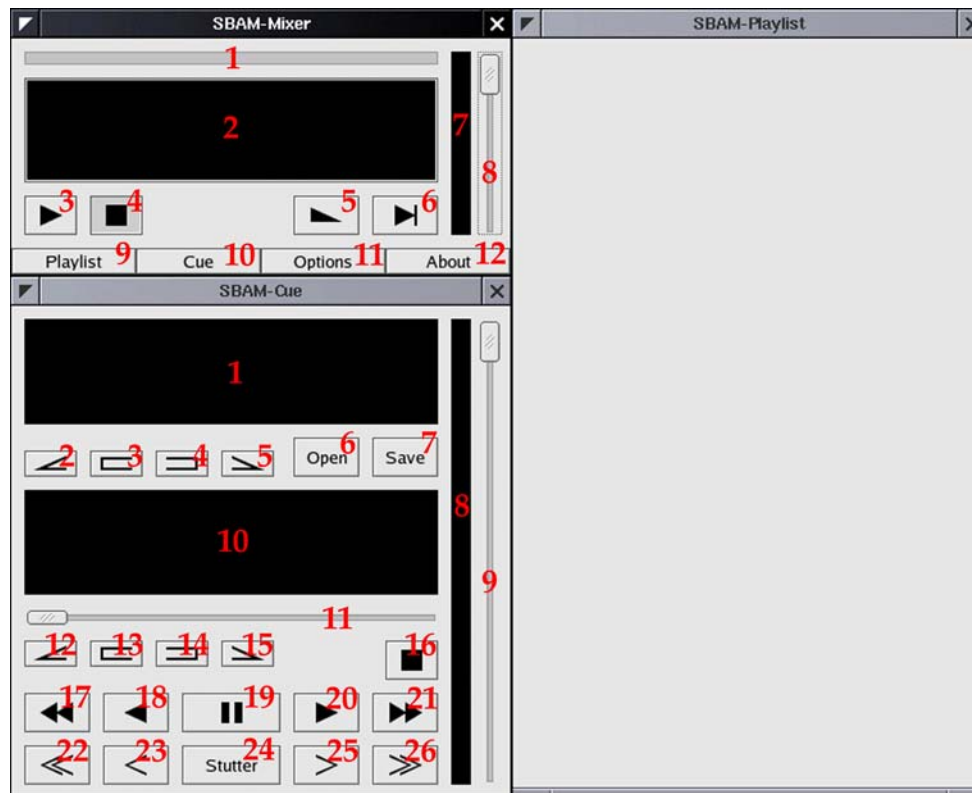
Vi har valgt at bruge netop XML, fordi det giver gode muligheder for at udvide funktionaliteten i playlisten senere.

Der kan findes dokumentation til libxml på:

<http://xmlsoft.org/APIchunk0.html>

Funktionelt design

Jeg har forsøgt at analysere hvilke funktionaliteter, der var nødvendige for at lave et godt stykke DJ-software. Dette har været en løbende analyse af, hvad der er nødvendigt eller praktisk, og hvad der er overflødig i forhold til, hvilke muligheder computeren giver os. Derfor afviger dette screenshot en anelse fra første designudkast, som kan findes i bilagene.



Billedet viser et screenshot fra SBAM version 0.1

Mixer

1. Progressbar, viser positionen i nummeret. Den er placeret her for hele tiden grafisk at illustrere hvor meget der er tilbage af nummeret, så DJen kan forberede sig på hvornår det næste mix vil blive foretaget.
2. Display, viser elapsed, remaining og total tid samt titel og kunster. Dette svarer til et display på en CD-afspiller, men kravene til indhold er lidt anderledes.
3. Play knap, starter afspilning af playlisten. Knappen bliver nede under afspilning, og hopper først op når der trykkes på stop.
4. Stop knap, stopper øjeblikkelig afspilningen af playlisten og hopper tilbage til starten af nummeret. Denne knap bør kun bruges i forbindelse med test af lydanlægget og når den er på pause. Knappen bliver nede så længe play knappen ikke er trykket ned.
5. Fade knap, fader musikken ned og sætter den på pause. Knappen bliver nede indtil der trykkes på den igen, desuden skifter ikonet på knappen til et "fade-op"-symbol. Når der trykkes på knappen anden gang, startes musikken igen og den fades op.

6. Next knap, starter øjeblikkeligt et mix over i næste nummer på playlisten. Denne knap er beregnet på hurtigt at kunne skippe videre fra et upopulær eller fejlindsat nummer.
7. Volume meter, viser lydstyrken grafisk. Bruges til grafisk at vise DJen lystyrken, så han kan vurdere om et nummer skulle være fejlnormaliseret, så master volumen skal korrigeres.
8. Master volume slider, ændrer lydstyrken i masteroutputet fra programmet.
9. Playlist knap, åbner playlist-vinduet. Knappen bliver nede så længe playlist-vinduet er åbent. Trykkes knappen anden gang lukkes playlist-vinduet.
10. Cue knap, fungerer som overstående knap blot for cue-vinduet.
11. Options knap, fungerer som overstående knap blot for options-vinduet.
12. About knap, fungerer som overstående knap blot for about-vinduet.

Playlist

Playlisten er ikke færdig designet, og præcis hvordan den kommer til at se ud afhænger af hvilke muligheder der ligger i GTK's liste-funktion. Playlisten kommer højest sandsynligt til at indeholde følgende elementer:

1. En liste med 3 kolonner: mix-in type, kunstner og titel og til sidst mix-out type. Hver række i denne liste repræsenterer så en entry på playlisten. Om muligt får denne liste drag and drop egenskaber, så man nemt med musen kan ændre rækkefølgen af numrene. Desuden vil der komme en form for indikator af hvor på listen vi er nået, og hvilke numre, der er blevet spillet. Præcist hvordan man skal vælge de forskellige mix-typer er endnu ikke helt afgjort, det kommer igen til at afhænge af mulighederne i GTK. Men evt. kan mixtypen fungere som en knap, der skifter til næste type, når den bliver trykket på.
2. En knap til at tilføje numre/playlister til playlisten.
3. En knap til at fjerne numre fra playlisten.
4. En knap til at flytte markeringen af et nummer op. Med denne og de næste tre knapper vil man kunne arrangere rundt på playlisten²⁰, så man ikke er nødsaget til at tilføje numrene i den rigtige rækkefølge.
5. En knap til at flytte markeringen af et nummer ned.
6. En knap til at flytte det markerede nummer et nummer højere op på playlisten
7. En knap til at flytte det markerede nummer et nummer længere ned på playlisten.
8. En knap til at gemme den aktuelle playliste som en fil.

Se desuden afsnittet om mix-variation.

Cue

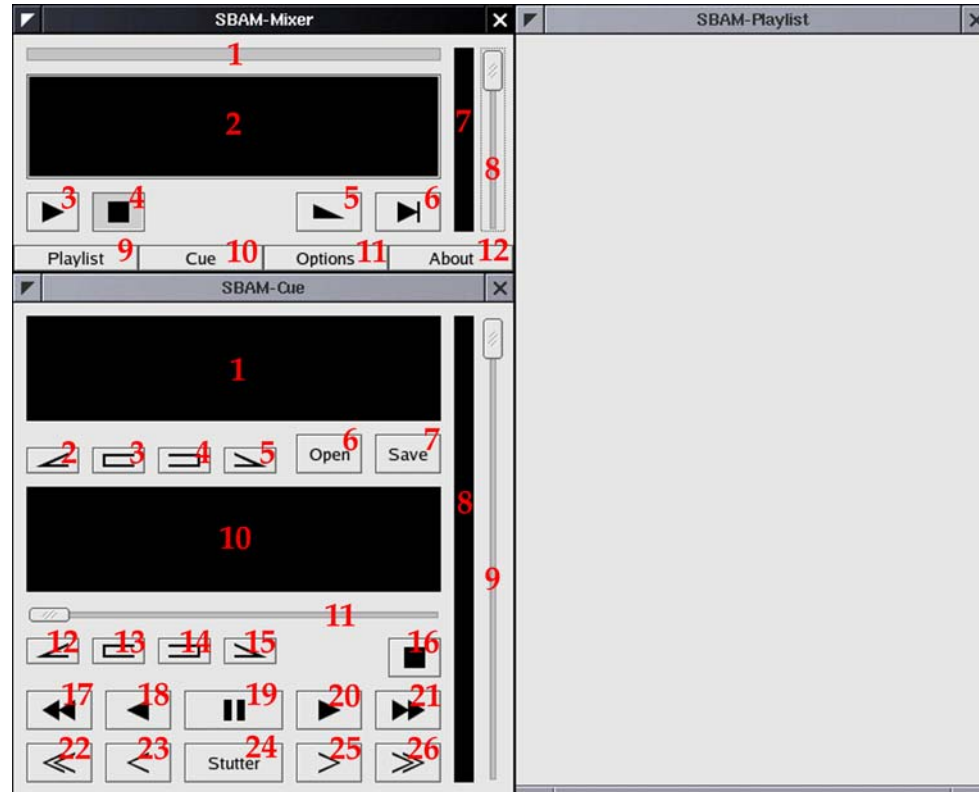
1. Display, viser som displayet i mixer-vinduet elapsed, remaining og total tid samt titel og kunster.
2. Set cue 1 knap, sætter "start fade in"-cuepunktet. Dette punkt placeres gerne lige inden et nummers intro starter; eller lidt inde i introen, hvis den er meget lang.
3. Set cue 2 knap, sætter "stop fade in"-cuepunktet. Dette punkt placeres gerne lige hvor nummeret trommerytme starter.
4. Set cue 3 knap, sætter "start fade out"-cuepunktet. Dette punkt placeres gerne lige efter nummerets trommerytme stopper; eller hvis nummeret fader ud til sidst, placeres punktet på det sted hvor lydstyrken akkurat er så lav, at det føles somom nummeret er slut.

²⁰ Uden brug af mus, men ved at bruge genvejstaster til knapperne. Eller hvis drag and drop ikke er understøttet.

5. Set cue 4 knap, sætter "stop fade out"-cuepunktet. Dette punkt placeres, hvor nummeret er slut. Hvis der er lang outro eller andet, så bruges dette punkt til at vise, hvor man vil have, at nummeret skal være slut.
6. Open knap, åbner en ny fil i cue-vinduet.
7. Save knap, gemmer cue-punkterne i ogg-filens tags.
8. Normaliserings meter, viser lydstyrken grafik, denne kan bruges under manuel korregering af normaliseringsværdien.
9. Normaliserings slider, viser normaliseringsværdien og bruges til manuel korregering af denne. Programmet vil dog placere en normaliseringsværdi udfra den målte lydstyrke i nummeret.
10. Waveform display, viser lydstyrken over tid for nummeret. Dette kan bruges til hurtigt at finde frem til, hvor nummeret starter og stopper; og hvis der er skjulte numre eller lignende, skulle det være til at se her. Der indtegnes her streger som markerer normaliseringsværdi, cuepunkter og nuværende position.
11. Position slider, kan bruges til lynhurtig søgning i nummeret. Den er placeret så den kan bruges sammen med waveform displayet.
12. Goto cue 1 knap, går til "start fade in"-cuepunktet. Man bruger hovedsageligt denne knap til at tjekke at man har placeret cue-punktet korrekt.
13. Goto cue 2 knap, går til "stop fade in"-cuepunktet. Man bruger hovedsageligt denne knap til at tjekke at man har placeret cue-punktet korrekt.
14. Goto cue 3 knap, går til "start fade out"-cuepunktet. Man bruger hovedsageligt denne knap til at tjekke at man har placeret cue-punktet korrekt.
15. Goto cue 4 knap, går til "stop fade out"-cuepunktet. Man bruger hovedsageligt denne knap til at tjekke at man har placeret cue-punktet korrekt.
16. Stop knap, stopper afspilningen og går til starten af nummeret. Denne knap er egentlig lidt overflødig, da der er andre måder at stoppe nummeret og komme til staten på; men da mange er vant til at have en knap med denne funktion, har jeg valgt at tage den med. Desuden kan det være rart at vide, man med sikkerhed er kommet helt tilbage til starten af nummeret hurtigt.
17. Play dobbelt tempo baglæns knap, afspiller nummeret baglæns i dobbelt tempo. Denne knap bruges til at spole i nummeret, men da den afspiller det samtidig, er man bedre klar over hvor i nummeret man befinder sig.
18. Play baglæns knap, afspiller nummeret i normalt tempo baglæns. Denne knap bruges ligeledes til at spole med.
19. Pause knap, pauser musikken. Denne knap bruges i forbindelse med spolning.
20. Play knap, afspiller nummeret i normalt tempo. Denne knap bruges både i forbindelse med spolning og til bare at aflytte et nummer.
21. Play dobbelt tempo knap, afspiller nummeret i dobbelt tempo. Denne funktion bruges til at spole frem i et nummer; men ved at afspille nummeret hurtigt, har man bedre styr på hvor i nummeret, man befinder sig.
22. Fast step baglæns knap, hopper et skridt tilbage i nummeret, når stutter-funktionen er slået til.
23. Step baglæns knap, hopper et lille skridt tilbage i nummeret, når stutter-funktionen er slået til.
24. Stutter knap, slår stutter-funktionen til. Stutter betyder at en lille bid af musikken bliver afspillet igen og igen. Denne funktion bruges, når man skal finde frem til et helt præcist punkt i musikken, f.eks. der hvor beatet starter. Når denne funktion er slået til, kan man benytte step knapperne til at søge præcist frem og tilbage.
25. Fast step forlæns knap, hopper et skridt frem i nummeret, når stutter-funktionen er slået til.

26. Step forlæns knap, hopper et lille skridt frem i nummeret, når stutter-funktionen er slået til.

Knapperne nr. 16, 17, 18, 19, 20, 21 og 24 bliver nede når de trykkes på dem, og de hopper så op når der trykkes på en af de andre af disse knapper.



Options

Options-vinduet er ikke færdig designet endnu. Dets funktion bliver at give mulighed for at ændre på diverse indstillinger i programmet. Vigtigst er det nok at man her skal have mulighed for at ændre og/eller tilføje mix-typer, så man kan få den automatiske mixing til at lyde præcist som man ønsker, og derved gøre det hele lidt mere personligt. Men også ting som hvor langt et stykke der skal spilles igen og igen, når stutter-funktionen er slået til vil kunne indstilles her.

About

About-vinduet er heller ikke designet færdigt endnu, dette vindue skal dog også kun indeholde nogle informationer som programmets navn, version, programmører og lignende.

Mix-variation

For ikke at risikere at mindske DJens indflydelse på mixingen af musikken, og for at gøre det muligt at variere sine mix efter hvilke numre, der bliver spillet. F.eks. ville en DJ måske være interesseret i på et tidspunkt at skabe opmærksomhed ved hjælp af intro, og ikke som normalt lade numrene flyde helt sammen. Små variationer i mixet kan også være DJen signatur, eller bare være med til at få en endnu bedre præstation ud af softwaren.

Måden denne variation vil blive implementeret er ved at lave nogle yderligere parametre, som der tages højde for under mixingen af numrene. De parametre som jeg forestiller mig at vi vil implementere i første omgang er følgende:

1. Hvilke 2 cue-punkter, som skal tilpasses hinanden.
2. Forsinkelsen mellem de to cue-punkter.
3. Lydstyrke ved start og stop af mix for første nummer.
4. Balance ved start og stop af mix for første nummer.
5. Lydstyrke ved start og stop af mix for andet nummer.
6. Balance ved start og stop af mix for andet nummer.

Der skal så være muligt at gemme disse indstillinger i nogle brugerdefinerede presets²¹, så man hurtigt kan vælge sine egne specielle mix-typer, eller man kan blot ændre for det enkelte mix. Desuden skal det være muligt at ændre default mixet.

På playlisten vil der for hver nummer være en knap til mix-variation for netop dette mix. Et klik vil bringe dialogen op, hvor mixet kan varieres eller vælges ud fra presets. Alle parametrene for mixet vil derefter blive gemt i playlisten, så hvis playlisten gemmes som fil og åbnes på en anden computer uden de samme presets, så vil mixet stadig lyde som det var ment.

Da man jo skal kunne flytte rundt på numrene i playlisten, får man det problem at man skal finde ud af, hvor mixet hører til henne. Skal mixet følge med nummeret før eller efter? En god løsning ville være at splitte parametrene for mixet op, så de parametre, der havde med det ene nummer at gøre knyttes til denne. Punkterne 1, 2, 5 og 6 vil altså høre til det nummer, der bliver mixet til. Mens punkterne 3 og 4 knyttes til nummeret der mixes fra. Man sikrer dermed, at indstillinger lavet for at tilpasse et bestemt nummer, følger netop dette.

²¹ Preset betyder sæt af forudbestemte indstillinger.

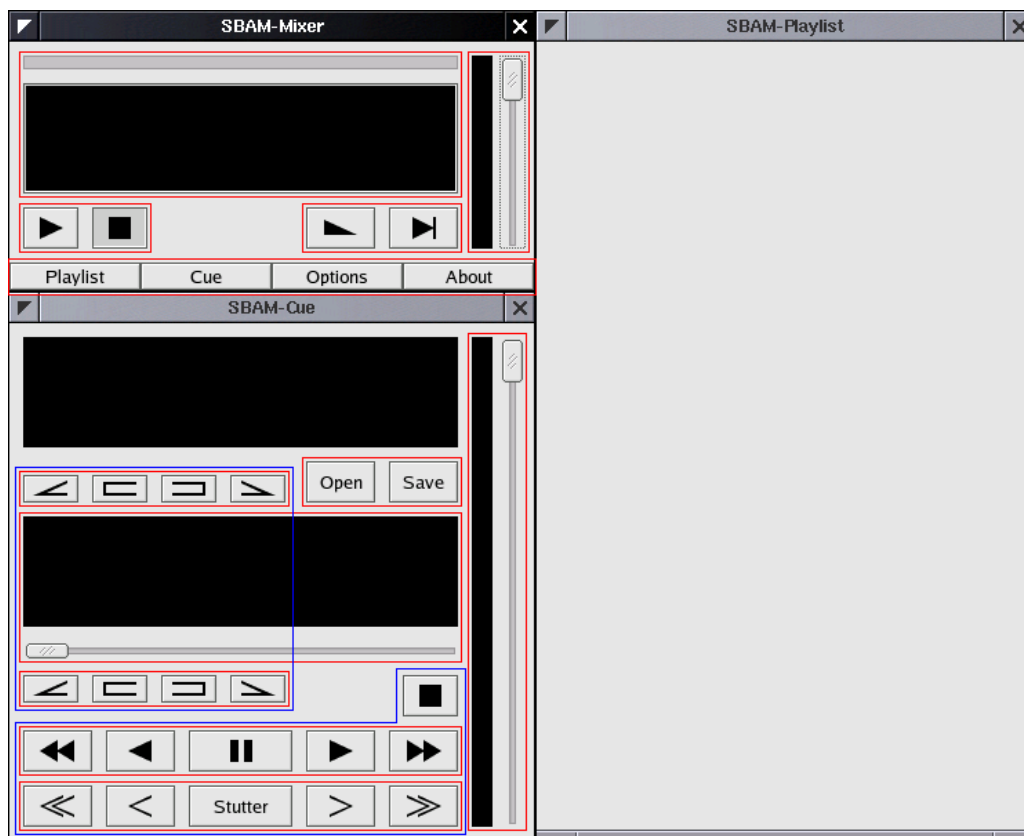
Grafisk design

Ved at bruge GTK til GUI har man lagt sig fast rent visuelt på en række punkter. F.eks. ser en knap ud på én bestemt måde i GTK, og det kan man ikke umiddelbart ændre på. Til gengæld ved man at alle vil genkende det som en knap, da den overholder de visuelle standarder vi er så vant til. Man har desuden stadig stor frihed til at placere knapper, sliders og lignende præcis som man ønsker. Man får med GTK et udmærket værktøj til at lave et godt interface.

Jeg har imens jeg designede det grafiske interface haft for øje at det skulle kunne bruges på små utydelige skærme samt sort/hvid skærme. Det er vigtigt at tænke på dette, da man ellers vil komme til at udlukke nogle hardwaremuligheder.

Nærhed


For at lave et godt interface har jeg brugt teorien om nærhed, dvs. at jeg placerer objekter, som har noget med hinanden at gøre, i nærheden af hinanden. Jeg forsøger også at placere dem i en logisk retning i forhold til hinanden, og skabe en form for logik i placeringen af knapperne. Brugeren vil på denne måde meget hurtigere forstå, hvordan programmet fungerer.



I figuren har jeg forsøgt at illustrere hvordan jeg har placeret knapper, som har noget med hinanden at gøre, i nærheden af hinanden.

Grafiske symboler

I mange tilfælde kan et lille symbol sige mere end et ord og desuden aflæses hurtigere. Derfor bliver grafiske symboler mere og mere en del af vores hverdag. Jeg har forsøgt at bruge nogle meget tydelige velkendte symboler på min knapper. Jeg har dog selv måttet finde på symboler som kan illustrere de 4 cue-punkter; men selvom der er tale om nye symboler, som folk ikke umiddelbart kender, er de stadig langt mere effektive end en tekstlig afbildning ville have været.

	Start på nummeret
	Beatet begynder på nummeret
	Beatet stopper på nummeret
	Slutning på nummeret
	Afspil nummeret baglæns
	Fade nummeret langsomt op til fuld styrke
	Fade nummeret ned
	Afspil nummeret hurtigt baglæns
	Afspil nummeret hurtigt
	Hop et stort skridt tilbage i nummeret
	Hop et stort skridt frem i nummeret
	Skift til næste nummer
	Sæt nummeret på pause
	Afspil nummeret
	Hop et lille skridt tilbage i nummeret
	Hop et lille skridt frem i nummeret
	Stop nummeret

Licens

Vi har valgt at udvikle vores projekt under GNU General Public License

De fleste licenser på software har til formål at fjerne Deres ret til at dele softwaren med andre og ændre i det. I modsætning hertil har GNU General Public License til formål at garantere Deres ret til at dele og ændre frit software - for at sikre, at det pågældende software er tilgængeligt for alle brugere.²²

Den økonomiske side

Der skal en hel del til for at kunne etablere sig på softwaremarkedet. Og da jeg ikke har kræfterne til at forsøge at markedsføre vores produkt, så synes jeg det er en del mere spændende at forsøge at sætte en dagsorden for branchen. Vi ønsker med vores valg af licens at kunne nå ud til et større publikum, og derved få skabt fokus på en alternativ måde at gribe tingene an på. Der er mere tilfredsstillelse i at lave et produkt som vil blive brugt, end at tjene nogle håndører.

Den politiske side

Hvis man i softwareudvikling brugte mere tid på at udveksle erfaringer og hjælpe hinanden med at udvikle det bedste system end på at konkurrere hinanden og beskytte sine idéer, så er jeg helt sikker på der havde været meget bedre software på markedet i dag. Det gælder for software som for idéer og viden, jo flere mennesker, der bliver involveret i dem, og jo flere, der udvikler på dem, jo bedre bliver de.

Hvorfor behøver man en licens?

Det kan virke underligt at man behøver en licens, som siger at man må bruge og kopiere softwaren; men det er faktisk meget vigtigt. Hvis ikke man har en licens, som sikrer at softwaren forbliver fri, så kan der lige pludselig være nogen andre, som indkasserer penge for vores idé.

1. De har ret til at kopiere og distribuere ordrette kopier af Programmets kildetekst, således som De har modtaget den, i ethvert medium, under forudsætning af, at alle kopier tydeligt og på behørig vis forsynes med en passende ophavsretmeddelelse og garantifraskrivelse; at alle meddelelser, der refererer til denne licens og den manglende garanti, bibeholdes; og at en kopi af denne licens vedlægges, når De distribuerer Programmet.

2. De har ret til at ændre Deres kopi eller kopier, eller en hvilken som helst del deraf, hvorved der skabes et Værk baseret på Programmet, samt ret til at kopiere og distribuere sådanne ændringer eller værk i henhold til paragraf 1 ovenfor, forudsat De også opfylder følgende betingelser

...

Disse krav er gældende for det ændrede værk i sin helhed. ...

²² Fra Christian Hansen's danske oversættelse af GNU General Public License

Hardware & Udvidelser

Hardware designs

Mål

Det hardware man vælger at bruge til at DJ'e på kunne i teorien være en hvilken som helst PC med den rigtige software installeret, men der er nogle forskellige egenskaber, som ville være praktiske til netop DJ-brug. F.eks. skal computeren stå på en mørk klub tit i nærheden af et lysshow, derfor er det ikke praktisk, hvis computeren er hvid. Man ville lægge alt for meget mærke til den, i stedet ville en sort computer være at fortrække. Hvis computeren skal transporteres til og fra klubben, så er det praktisk, hvis den ikke er for tung eller i for mange dele, og at den kan bæres på en praktisk måde. Desuden skal computeren helst være rimelig robust, så den ville kunne klare en evt. hård behandling.

Hvis vi kigger på specifikationerne for hardwaren, så kræver det ikke noget specielt voldsomt af CPU og RAM, dog vil en stor harddisk til pladesamlingen være nødvendig. Firewire, så man har mulighed for at tilslutte eksterne harddiske, ville ikke være af vejen. Et kvalitetslydkort vil sikre en ordentlig lyd kvalitet. Og man skal have en CD-ROM drev for at kunne rippe musikken.

Den billige

Den billigste løsning er helt klart at samle en mini-computer selv. Hvis man så bygger den ind i en DJ-kuffert eller flightcase²³, så er den nem at tage med sig på klub. Det er også den løsning, jeg selv har valgt.

Jeg har valgt et VIA Epia-M9000 bundkort. Det har jeg valgt, fordi det har en størrelse af kun 17 × 17 cm, desuden har det både firewire og USB 2.0. Der sidder en 933 MHz C3 CPU på, hvilket er mere end kraftig nok til at dekode 3 OGG-filer på én gang.

Jeg har valgt at sætte 512 MB RAM i computeren, hvilken er rigeligt. Desuden købte jeg en 120 GB harddisk, her er der plads nok til alle mine CD'er i god kvalitet, og stadig plads til at jeg kan få mange, mange flere albums.

Der sidder et VIA on-board lydkort på bundkortet. VIA er kendt for at lave chips til Terratec's lydkort, som er de bedste i Verdenen. Er man alligevel ikke tilfreds med lyden på dette on-board lydkort, så kan man udvide med et rigtigt Terratec lydkort i bundkortets ene PCI slot, eller man kan tilslutte et USB-lydkort.

Med en lille flexATX strømforsyning og en sort DJ-kuffert har jeg tænkt mig at bygge det hele om til et handy system. Der skal så i låget side en 15" TFT skærm.

Den færdige maskine er en maskine med stor kapacitet og velegnet til DJ-brug for omkring 8000,-. Det er nogenlunde samme pris som en dobbelt DJ CD-afspiller.

Den smarte

En lidt dyrere løsning, men som til gengæld ville veje og fylde meget mindre, ville være at købe en bærbar computer. Deres harddiskes kapacitet er sjældent voldsomt stor, men så må man have en ekstra ekstern harddisk med, hvis det skulle være nødvendigt.

²³ Se ordforklaring.

Den professionelle

Den professionelle løsning ville være at købe en computer til montering i rack. DJ-pulten er oftest bygget i et 19" rack i forvejen, så en industriel rack-computer ville passe lige ind. Alt efter hvordan pladsen er ved DJ-pulten og om den til tider skal kunne pakkes væk, beslutter man sig for enten at have en almindelig TFT-skærm eller en rack-monteret sammen med computeren.

Denne løsning vil være professionel og robust, desværre vil desværre også være relativt dyr, og vil kun være egnet til steder med en forholdsvis stationær DJ-pult. Selvom den vil kunne bygges ind i en solid flightcase og transporteres.

Udvidelser

Lyseffekt eller tekstscroller

En spændende ting at udvide systemmet med ville være en måde at vise publikum hvilket nummer, der bliver spillet. På den måde vil man kunne promote sin yndlingsmusik, samt udvide sit publikums viden om musik. Desuden ville det kunne se flot ud, hvis det blev gjort på den rigtige måde.

En mulighed ville være at anskaffe sig et diode-display af den slags som nogle gange ses i små butikkers butiksvinduer. Den ville så kunne sluttet til computeren og sættest til at scrolle kunstnerens navn og titel på nummeret ud til publikum.

En anden mulighed ville være at tilslutte en projektor. Den skulle så enten tilsluttes DJ-computeren, hvis denne har dual-display, eller den skulle tilsluttes en dedikeret computer, som får lydinput og information om numrene fra DJ-computeren. Denne projekter ville så kunne vise visuelle effekter synkroniseret til musikken samt information om kunstner og titel. Hvis den placeres i forbindelse med dansegulvet, vil den samtidig virke som en ekstra lyseffekt.

Jukebox/Ønske-system

På nogle klubber vælger DJen at ligge en seddel frem, som publikum kan skrive musikønsker på. Dette gøres af flere grunde, for det første behøver man ikke bruge så lang tid på at tage imod ønsker, desuden har man dem bagefter på skrift, så man ikke glemmer dem igen.

Denne proces ville også kunne digitaliseres og gøres endnu smartere. Hvis man stillede en ekstra lille computer op ude ved publikum, denne skulle så indeholde en database over alle de albums, som DJen havde, så kunne publikum bladre DJens musik igennem og ønske herudfra. Dette vil også give publikum en større forståelse for, hvis DJen ikke kan spille deres ønske, fordi han ikke har det. Ønske-systemet kunne være tilsluttet DJ-computeren via netvært, og DJen vil så kunne se de ønskede numre direkte på sin egen skærm.

Afslutning

Konklusion

Gennem vores arbejde de sidste 5 måneder, har vi fået udviklet en tidlig version af det program som var vores mål. Selvom programmet ikke er færdig, og ikke al funktionalitet er tilføjet, kan man stadig se konceptet tegne sig. Denne tidlige version viser at det er muligt at automatisere mixingen, imod at DJen blot skal indsætte 4 cue-tider på forhånd. Den viser også en praktisk måde, hvorpå man hurtigt kan finde frem til disse cue-punkter.

Præcis hvor stor tidsmæssig besparelse der ligger i teknikken for DJen, kan vi stadig ikke se, det ville kræve en færdig version at teste dette. Men man kan se et koncept tegne sig, hvor DJen har en langt større frihed, et koncept som går på planlægning af musikken fremfor timing.

Denne måde at gribe konceptet om DJing på computer an på, lader til at være ny og innovativ, og det er da også en idé om at skabe et effektivt værktøj gennem nyskabelse, som har drevet projektet frem. Vi har også fået mulighed for at afprøve mange nye ting, som vi aldrig har arbejdet med før dette projekt.

Jeg har med vores interface forsøgt at matche vores innovative funktionalitet for at give en brugerflade, som er selvforklarende og nem at gå til. Og den umiddelbare respons på programmet har da også netop vist dette.

Selvom programmet ikke er helt færdigudviklet, så er alle elementerne på plads til at bygge resten. Det som startede som en idé om, hvad der var teoretisk muligt, har efterhånden taget form til et færdigt produkt.

Perspektivering

Projektet vil blive videreudviklet hen over sommerferien, så der forhåbentligvis ligger en fuldt funktionel version klar sidst på sommeren. Vi mangler stadig at bygge hele interfacet til playlisten og implementere funktionaliteten til den. Vi mangler at kode, så cue-tiderne bliver gemt direkte i OGG-filens tag ved at trykke save, og vi mangler at bygge fileselection-vinduet til open. Derudover mangler muligheden for mix-variation samt nogle mindre tilrettelser.

Når de rettelser er sket, så er programmet funktionelt, og jeg har tænkt mig selv at tage det i brug. Så vil det også hurtigt vise sig, hvor det kan forbedres, og vi kan begynde at kode på næste version.

For at jeg kan tage programmet i brug, er jeg ved at bygge en computer specielt til formålet. Denne er langt fra færdig, og der vil også ligge en opgave i at få den bygget. Det kunne desuden også være spændende at prøve at udvikle nogle af de idéer beskrevet under afsnittet udvidelser.

Det vil blive interessant at se om programmet vil få flere brugere, og hvilken respons man vil få fra dem. Jeg er spændt på at se, om der vil komme ønske om yderligere funktionalitet, eller brugerne umiddelbart kan alt, de ønsker. Men allermest glæder jeg mig til selv som DJ til at have den ekstra frihed, som vores software giver.

Ordforklaring

Vinyl

Professionelt udtryk for plader (LP'er, 12", 7" m.m.), da disse er lavet af vinyl, har man valgt denne fællesbetegnelse.

Turntable

Professionelt udtryk for pladespiller.

Mixer

En form for forforstærker, som har flere kanaler. Disse kanaler kan der så skrues op for individuelt, hvorpå de bliver lagt sammen og sendt til et master-output. Dette output går så til en effektforstærker eller PA-anlæg. At kunne lægge to lydkilder sammen og variere deres lydstyrke individuelt er det essentielle værktøj for en DJ.

Beatmix

En bestemt teknik til mixing af to numre. Man beatmatcher de to numre, dvs. man får de to numres beat til at være lige hurtige ved at ændre tempo på det ene af dem og derefter tilpasser så de ligger perfekt oveni hinanden. Når to numre er beatmatched, kan man mixe fra det ene til det andet uden at man bemærker at det ny nummer begynder, det bliver nærmest en fortsættelse af det første.

Denne teknik kræver at al den musik, man spiller, har nogenlunde samme tempo og helst ingen temposkift, og det kræver at der er stykke tid med kun beat i starten og slutningen af numrene. Desuden fjerner man introer og outroer fra numrene med denne teknik. Af disse grunde er denne teknik kun almindelig indenfor bestemte genrer musik.

Mix

At bruge en mixer til at skabe en overgang mellem to stykker musik.

Cue-tid

Cue-tider er de tidspunkter, hvor nummeret skal startes og stoppes for at lave et godt mix. Ordet stammer fra DJing på turntable, hvor at cue pladen op betyder at finde det punkt, hvor pladen skal startes.

I SBAM er der 4 typer cue-tider. En cue-tid er det punkt hvor nummerets intro starter. En anden er der, hvor introen stopper, og beatet begynder. De sidste to er det samme bare gældende for outroen.

Normalisering

Normalisering betyder at indstille musikken så alle numre har samme lydstyrke, selvom de måske er indspillet forskelligt.

Stutter

En effekt på DJ CD-afspillere, som muliggør at finde punkter i musikken meget nøjagtigt. Funktionen virker ved at gentage den samme stump af musikken igen og igen, DJen kan så langsomt flytte sig frem og tilbage i nummeret, mens han stadig kan høre, hvor han er.

Crossfader

Når man mixer to numre, gøres det ved først at skrue op for den ene kanal, derefter skrue ned for den anden. For at kunne gøre denne proces nem og med kun en hånd, er de fleste DJ mixere udstyret med en crossfader. I venstre position vil første kanal være på fuld styrke mens anden kanal er slukket. Ved at flytte den langsomt mod højre, skrues man op for den anden kanal. Når crossfaderen er i midterpositionen, er begge kanaler på fuld styrke, og i højre position er første kanal slukket, mens anden kanal er på fuld styrke.

Rippe

Når man kopiere en musik CD ind på computeren kræver det at man læser dataene på CDen og omsætter det til waveform data, som computeren kan forstå. Dette kaldes at rippe CDen. Typisk vil man samtidig vælge at komprimere denne wave data til f.eks. OGG-filer, som fylder en del mindre end det ukomprimerede wave data.

Flightcase

En type solide og professionelle kasser af tyk krydsfiner med kanter og hjørner af metal, som normalt bruges til transport af lydudstyr.

Porte

At tage et stykke kode skrevet til et system, og tilpasse det til et nyt system. Det kan f.eks. være, når man tager et Linux program og omskriver det en smule, så det kan kompileres på en Windows maskine. Hvis ikke man har skrevet sin kode med øje for at det kan portes, kan det blive en meget vanskelig proces; har man derimod kodet det netop med dette for øje, kan det være ret smertefrit.

Litteraturliste

The C Programming Language

Second Edition

Af Brian W. Kernighan og Dennis M. Ritchie

ISBN 0-13-110362-8

Stjerneanalysen

Version 1.04

Af Lise Agerbæk

Xiph.Org

Building a New Era of Open Multimedia

Af the Xiph.Org Foundation

<http://www.xiph.org>**GNOME Development Site**<http://developer.gnome.org/>**The XML C library for Gnome**

Af Daniel Veillard

<http://xmlsoft.org/>**Hvorfor digitalt?**

Et forsøg på at give en forklaring på hvorfor elektronikken blev digital

Af Hans Siggaard Jensen

<http://www.detdigtaledanmark.dk/docs/tema1/siggaardoplaeg.doc>**BeatMatching 101**

Af Jonny Thrice

http://www.djforums.com/tutorials/beatmatch_101.php**The Digital Audio Format Test!**

Af Jens Adrian

<http://www.cdburner.ca/digital-audio-formats-article/digital-audio-test.htm>**Dare to compare!**

Af Monty, lead developer of the Ogg Project 20020719

<http://www.xiph.org/ogg/vorbis/listen.html>**GNU General Public License**

2. version, juni 1991

Af Christian Hansen

http://www.fundanemt.com/gpl_da.html

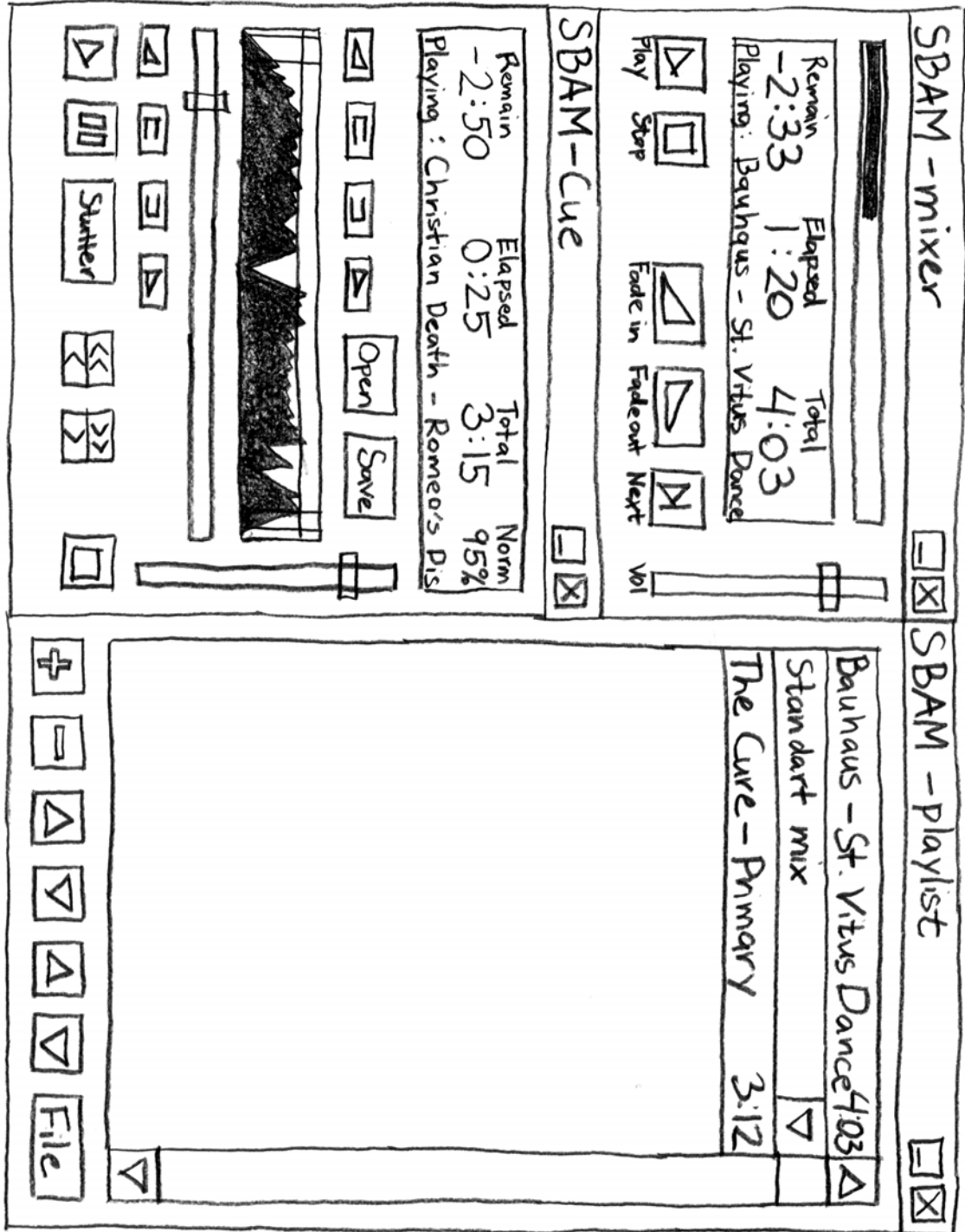
Testet DJ-software

For at få en idé om hvordan markedet for DJ-software var, testede jeg demoer af følgende DJ-software pakker:

AtomixMP3
BPM-Studio 4
BPM-Studio 2
PCDJ Red 5.2
MpegDJ DoublePlayer V2.80
Xaudio PartyPlayer
Og 9 forskellige DJ-plugins til WinAmp

Dertil kommer at jeg har læst om en række software pakker, som jeg desværre ikke kunne få adgang til at prøve.

Bilag 1: Første skitse af design



Bilag 2: Billede af arbejdsprocessen

The screenshot shows the Anjuta IDE with the following components:

- File Explorer (Left):** Lists files such as SBAM_GUI-0.1, source [src], controller_cue.c, controller_mixer.c, main.c, pixmap [pixmaps], dataness.png, play.png, and stop.png.
- Main Editor:** Contains C code for a GTK application. Key sections include:
 - `controller_cue.c`: Functions for setting slider sizes, connecting signals, and handling button clicks (Backwards, Forward, Stop).
 - `controller_mixer.c`: Functions for connecting signals and handling button clicks (Play, Stop).
 - `main.c`: The main function that initializes the application and connects the various components.
- Terminal (Bottom):** Shows the output of a 'make' command. It lists the compilation of various UI components:
 - Slider 13 Changed.
 - Slider 15 Changed.
 - Slider 15 Changed.
 - Slider 13 Changed.
 - Slider 13 Changed.
 - Toggle-Button 25 Pressed.
 - Toggle-Button 14 Pressed.
 - Toggle-Button 7 Pressed.
 - Toggle-Button 4 Pressed.
 - Button 32 Pressed.
 - Button 20 Pressed.
 - Button 20 Pressed.
 - Button 28 Pressed.
 - Button 28 Pressed.
 - Button 20 Pressed.
 - Button 20 Pressed.
 - Button 26 Pressed.
 - Button 26 Pressed.
 - Button 20 Pressed.
 - Button 20 Pressed.
 - Button 22 Pressed.
- SBAM-Mixer and SBAM-Queue Windows:** These windows show a graphical representation of the application's state, including a volume slider and playback controls (Play, Stop, Pause, Stutter).
- Status Bar (Bottom):** Displays 'Project: SBAM_GUI', 'Line: 0349', 'Col: 087', 'INS', 'Job: Name', and 'Mode: Unix (LF)'.