

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CIÊNCIA DA COMPUTAÇÃO

EDUARDO AQUILES AFFONSO RADANOVITSCK

**mixDroid: Compendo através de
dispositivos móveis**

Prof. Dr. Marcelo Soares Pimenta
Orientador

Prof. Dr. Damian Keller
Co-orientador

Porto Alegre – RS – Brasil, julho de 2011

SUMÁRIO

LISTA DE FIGURAS	3
LISTA DE TABELAS	4
RESUMO	5
ABSTRACT	6
1 INTRODUÇÃO	7
2 CONCEITOS E FUNDAMENTOS PARA ENTENDER O MIXDROID . .	8
2.1 Música Ubíqua	8
2.2 Mixagem	9
2.3 Metáforas	10
2.4 Marcação Temporal	10
2.5 Concurrent Task Tree	10
3 CONTEXTUALIZAÇÃO	12
4 PROTÓTIPOS	13
4.1 Descrição sucinta	13
4.2 Descrição detalhada	14
4.2.1 mixDroid v0.1	14
4.2.2 mixDroid v1.0	15
4.2.3 Teste de espacialização	19
4.2.4 Teste de geração de som em tempo real	20
4.2.5 mixDroid v1.5	21
4.3 Modelagem CTT e exemplo de uso do mixDroid	22
5 EXPERIMENTOS	23
5.1 Estudo 1: Avaliação Qualitativa Com Usuários Leigos	23
5.1.1 Amostra	23
5.1.2 Procedimento Experimental	24
5.1.3 Resultados	25
5.2 Estudo 2: Avaliação Qualitativa Com Usuário Experiente	25
5.2.1 Resultados	25
6 DISCUSSÃO FINAL	26
REFERÊNCIAS	27

LISTA DE FIGURAS

Figura 4.1:	Primeiro protótipo do mixDroid	14
Figura 4.2:	mixDroid 1.0 - Atividade Iniciar	15
Figura 4.3:	Modelagem CTT da atividade “Definir nome da composição”	16
Figura 4.4:	Implementação da atividade “Definir nome da composição”	16
Figura 4.5:	Modelagem da atividade compor (parte1)	16
Figura 4.6:	Modelagem da atividade compor (parte2)	16
Figura 4.7:	Implementação da atividade Compor	17
Figura 4.8:	Implementação da atividade Compor — ação mapear som para botão	17
Figura 4.9:	Listar composições salvas	18
Figura 4.10:	Escutar composição salva	18
Figura 4.11:	Protótipo de espacialização	19
Figura 4.12:	Modelagem CTT em mais alto nível do mixDroid	22

LISTA DE TABELAS

Tabela 5.1:	Perfil dos voluntários	24
Tabela 5.2:	Tempo (em minutos e segundos) necessário para realizar as tarefas e impressões dos usuários quanto às ferramentas	24

RESUMO

Esse trabalho visa investigar a aplicação de uma nova metáfora para interação em contextos de computação ubíqua, sobretudo em aplicações de computação musical: a metáfora “*time tagging*” ou marcação temporal. Esta metáfora se apoia na informação temporal disponível através do próprio som e não de representações gráficas. O processo evolutivo de construção e as principais características do *mixDroid* — um protótipo que foi implementado em telefones celulares que suportam o sistema operacional livre *Android* para realizar experimentos com esta metáfora — são descritos. Experimentos envolvendo usuários reais foram realizados e os resultados são apresentados, confirmando a aplicabilidade da metáfora nas atividades musicais em contexto ubíquo e pervasivo.

Palavras-chave: Computação musical, música ubíqua, IHC, android.

mixDroid: composing music using mobile devices

ABSTRACT

This work investigates the applicability of a new interaction metaphor to be used in the ubiquitous music context, specially in computer music applications: the “time tagging”. This metaphor is based on the temporal information available in the sound itself and not on any graphical representation. The evolutionary process of building and the main characteristics of the *mixDroid* — a prototype implemented for cell phones running the open source operational system Android to perform studies with this metaphor — are described. Were performed experiments involving real users and the results are presented, confirming the applicability of the metaphor on musical activities in ubiquitous and pervasive contexts.

Keywords: musical computing, ubiquitous music, HCI, android.

1 INTRODUÇÃO

O objetivo deste trabalho é a investigação da metáfora “*time tagging*” (ou marcação temporal) e o desenvolvimento de protótipos em dispositivos móveis para realizar experimentos com ela. Este trabalho é uma parte da tese de doutorado (de Luciano V. Flores) sendo realizada no LCM/INF/UFRGS (FLORES, 2010) e cujo objetivo final é ampliar a base de usuários em música ubíqua, eliminando a necessidade de conhecimentos específicos musicais, de habilidades na manipulação de ferramentas pouco intuitivas e de orçamentos proibitivos para a estruturação de ambientes de produção musical.

Este objetivo inclui algumas metas específicas, entre as quais:

- Pesquisar quais são os recursos disponíveis e as limitações nos dispositivos feitos para outros propósitos (p.ex., celulares, desenhados para permitir comunicação e acesso a informação) quando utilizados em computação musical a fim de se estabelecer uma expectativa de quão complexo um sistema voltado a computação musical pode ser;
- Utilizar a plataforma livre Android (2009) — um sistema operacional de código aberto voltado a dispositivos portáteis, principalmente celulares — como forma de oferecer uma alternativa aos orçamentos proibitivos que sempre acompanham os ambientes de produção musical. Embora celulares com Android ainda sejam relativamente mais caros, seus preços estão caindo rapidamente com a popularização da plataforma;
- Aplicar a metáfora de marcação temporal no desenvolvimento de um protótipo (o mixDroid) para verificar se a necessidade de conhecimentos específicos musicais pode ser contornada;
- Realizar experimentos com o protótipo a fim de validar a metáfora.

Após essa introdução, o capítulo 2 apresenta conceitos básicos para o entendimento do trabalho desenvolvido. No capítulo 3 são citados alguns trabalhos relacionados à área. No capítulo 4 são apresentados os protótipos desenvolvidos durante a pesquisa e no capítulo 5 é apresentado um experimento realizado com usuários reais envolvendo o mixDroid. O capítulo 6 apresenta um fechamento sobre o trabalho.

2 CONCEITOS E FUNDAMENTOS PARA ENTENDER O MIXDROID

2.1 Música Ubíqua

A *música ubíqua* é o resultado da conjunção de sistemas que permitem atividades musicais, utilizando múltiplas interfaces para a manipulação de dados e a geração de som, viabilizando o acesso simultâneo de múltiplos usuários num contexto ubíquo. Este tipo de contexto envolve tecnologias de sistemas distribuídos (*musical networks*), computação móvel (*mobile music*), sistemas cooperativos e tecnologia assistida. O termo já vem sendo empregado na área de *mobile music* (HOLMQUIST, 2005), mas ainda focado no potencial tecnológico apenas para acesso e compartilhamento de música. É nossa intenção buscar a expansão deste conceito, incluindo atividades musicais criativas (e.g., composição e performance) que possam tirar proveito de contextos ubíquos.

Uma das principais motivações do nosso trabalho é permitir que qualquer pessoa – inclusive o leigo em música – possa utilizar sistemas musicais de forma criativa, colocando a ênfase na qualidade do som e na naturalidade das interfaces. Ao incluir os dispositivos móveis como parte da interface musical, a intenção é de eliminar algumas das barreiras que impedem que os não-músicos se expressem musicalmente. Podemos citar como exemplos o alto custo financeiro e a exigência de um desenvolvimento psicomotor que demanda vários anos de estudo. Para ser acessível a leigos propomos que, na interação com ambientes ubíquos musicais os usuários utilizem os próprios dispositivos móveis comuns de informação e comunicação que já possuem e com os quais já estão familiarizados. No entanto, é importante ressaltar que música ubíqua não é apenas “música feita com dispositivos móveis”. Nesta definição se encaixa a *mobile music* (GAYE et al., 2006). Nossa proposta não é colocar o dispositivo como foco da pesquisa, mas utilizar dispositivos móveis como parte integrante de sistemas musicais ubíquos.

Colocando estes conceitos em termos mais precisos, podemos dizer que a música ubíqua é música feita no contexto da computação ubíqua (WEISER, 1995) – englobando computação móvel, conceitos como independência e heterogeneidade dos dispositivos (ou seja, interfaces múltiplas com características diferenciadas), infraestrutura pervasiva (incluindo *middleware*, redes sem fio, e sensores múltiplos), percepção do contexto (*awareness*), e mobilidade ou portabilidade. Ao incorporar os conceitos da UbiComp, a música ubíqua tira a ênfase do dispositivo (o equivalente tecnológico do instrumento musical) para colocá-la na atividade musical. As fontes sonoras e as interfaces são incorporadas ao ambiente, tornando-se parte dele (por exemplo, através de conexões sem fio). Desta feita, as atividades musicais podem ser integradas a outras formas de interação social e passam a formar parte de uma rede mais ampla de interações entre agentes e objetos.

O nosso grupo propôs um quadro conceitual que permite situar o fazer musical no contexto mais amplo das atividades humanas (KELLER et al., 2009). Nosso enfoque abrange duas áreas de pesquisa: o estudo das interações entre agentes e objetos, fundamentado nos conceitos de Gibson (1986) e o estudo das interações entre agentes, com suporte no trabalho de Leontiev (1978), descritos abaixo:

- **Atividade:** Pela teoria de Leontiev podemos examinar os processos humanos em três níveis: atividades e seus motivos são classificados como o nível mais alto. A atividade configura um conjunto de interações dos agentes com o ambiente com um objetivo específico.
- **Ação:** Leontiev define ações e seus objetivos associados como um nível intermediário no estudo da atividade. Uma atividade consiste em uma sequência de ações. Cada ação é uma unidade espaço-temporal que abrange agente e objetos.
- **Operação:** No nível mais baixo da definição de Leontiev está a análise das operações que servem como meios para desenvolver uma ação. A principal diferença entre ação e operação é que a operação serve aos fins específicos da execução de uma ação e só pode ser estudada dentro desse contexto.

Os conceitos que são discutidos no parágrafo anterior servem para compreender os diferentes aspectos da atividade musical. No entanto, na hora de aplicar esse conhecimento em casos específicos precisamos ter cuidado para não misturar ferramentas de implementação com ferramentas conceituais. No suporte para desenvolvimento da plataforma Android há vários exemplos desse problema:

- **Atividade:** na plataforma Android também existe o uso da palavra atividade definida como uma interface visual para um objetivo específico que o usuário pode realizar (ANDROID: APPLICATION FUNDAMENTALS, 2009). Seus significados se assemelham, uma vez que em Android a atividade seria o nível mais alto de abstração do aplicativo.
- **Ação:** embora não exista essa definição em Android, podemos usar esse termo para descrever as ações que o usuário pode realizar no contexto de uma atividade (na definição da plataforma). Por exemplo, dada a atividade de seleção de arquivos de som, a ação consiste na sequência de abrir a pasta, escolher e abrir o arquivo.
- **Operação:** No nível mais baixo da definição de Leontiev está a análise das operações que servem como meios para desenvolver ações. Novamente, não existe o conceito em Android, mas podemos fazer uma relação com a forma de interação disponível para cada ação. No exemplo acima, uma operação seria pressionar com o dedo o espaço da tela em que o arquivo a ser selecionado está localizado.

2.2 Mixagem

Com o objetivo de entender em mais profundidade os mecanismos utilizados pelos músicos na hora de realizar uma mixagem, aplicamos os conceitos gerais propostos por Leontiev ao contexto específico da atividade musical (KELLER et al., 2009). Nesse contexto específico, como é bem conhecida no âmbito musical, a mixagem é o procedimento de combinar sons previamente gravados (amostras sonoras) com o objetivo de obter um

novo resultado a partir da distribuição temporal do material sonoro. Nos editores digitais existentes, geralmente são utilizadas interfaces que permitem a seleção e o posicionamento de arquivos sonoros no eixo temporal. Os dados sonoros são representados por gráficos de amplitude dispostos em trilhas alinhadas verticalmente.

A atividade de mixagem pode ser separada em atividades específicas para possibilitar seu estudo e implementação. Estas atividades são:

- **Sequenciamento:** (ou sequenciação) é a atividade de organizar eventos numa linha temporal. Ele pode ser realizado em tempo real ou fora do tempo real (como acontece nos editores gráficos de partituras e nos sequenciadores com entrada de dados a partir de mouse).
- **Espacialização:** abrange tanto a manipulação da amplitude do fluxo sonoro em função da posição dos sistemas de transdução (geralmente caixas acústicas ou alto-falantes) quanto à simulação de pistas psicoacústicas de localização das fontes sonoras (KELLER et al., 2010). O primeiro caso constitui a difusão, podendo ser realizada através do controle manual do nível de amplitude das fontes em relação a um sistema multicanal, ou através de algoritmos de automação envolvendo sistemas matriciais de amplificação (ou seja, amplificadores que controlam múltiplos canais de forma independente). O exemplo mais simples de controle de amplitude em relação aos dispositivos de transdução é o panorama ou a panorâmica (*panning, pan*) que consiste na distribuição de uma fonte monofônica em dois canais - geralmente à esquerda e à direita do ouvinte - através de um único dispositivo de controle - *knob* ou manivela rotatória.

2.3 Metáforas

Na computação, metáforas são analogias ou similaridades para se representar ou modelar atividades e/ou objetos do mundo real num ambiente computacional. Podemos citar a conhecida metáfora da “área de trabalho” que leva para o mundo virtual a ideia de se ter uma mesa de trabalho, onde podem existir diversos documentos ou objetos dispersos. Um exemplo de metáfora fora da área da computação são os balões de conversa existentes em revistas em quadrinhos que representam tanto falas quanto pensamentos das personagens.

2.4 Marcação Temporal

A metáfora visual de representação em trilhas utilizada pela maioria dos editores para ambientes desktop não é facilmente transposta aos sistemas portáteis. Como alternativa, adotamos a metáfora da marcação temporal (KELLER et al., 2010). Nesta metáfora, a reprodução do som fornece as pistas perceptuais que o usuário utiliza ao colocar as amostras sonoras no eixo temporal. Portanto, a interface pode se limitar a um número reduzido de teclas que são acionadas para acessar cada uma das amostras sonoras. Um exemplo de sua aplicação está nos gravadores antigos onde a tecla de gravar é pressionada e não existe nenhum retorno visual sobre o som que está sendo gravado, o compositor se baseia somente nos sons sendo executados no momento.

2.5 Concurrent Task Tree

O método CTT (*Concurrent Task Tree*) para definição e modelagem de tarefas ou ações (PATERNO; MANCINI; MENICONI, 1997), é baseado em atividades que o usuário pode efetuar com o sistema. Cada atividade representa um nodo, e cada ação dessa atividade é um nodo filho. Essa modelagem se encaixa fortemente no modelo de desenvolvimento para Android. A fim de demonstrar isso, foi feita uma modelagem utilizando essa notação, que está disponível em (MODELAGEM EM CTT DO MIXDROID, 2009). Cada atividade é mapeada na implementação para uma classe onde sua interação com o usuário é definida.

3 CONTEXTUALIZAÇÃO

O projeto de pesquisa MuJava (MUSIC IN JAVA - PROJETO EM COOPERAÇÃO ENTRE A UFAC E UFRGS. PROJETO DE PESQUISA CNPQ 571443/2008, 2009), do qual este trabalho faz parte, possui outros subprojetos que são da mesma área (música ubíqua), entre eles pode-se citar:

- MusicXML Player
- Arpeggiator

O projeto MusicXML Player (MUSICXML PLAYER - PROJETO DE PESQUISA CNPQ, 2009) visa desenvolver um *player* tanto em Java SE (para uso em computadores) quanto em Java ME (para uso em dispositivos móveis) que suporte o formato MusicXML. Esse formato foi desenvolvido para definição de sons e composições. É um formato que possibilita ajuste fino de todas propriedades sonoras. O uso do MusicXML como forma de salvamento das canções no mixDroid foi adiado por trazer uma complexidade que até o momento não agregaria valor suficiente para justificar sua implementação.

O projeto Arpeggiator (ARPEGIATTOR - PROJETO DE PESQUISA CNPQ, 2009) é um projeto iniciado antes do mixDroid que visa explorar a plataforma JavaME, porém resultados encontrados e apresentados no XXI Salão de Iniciação Científica da UFRGS (2009) mostram que a plataforma é muito limitada comparando-se com Android.

Alguns aplicativos existentes no mercado oferecem funcionalidades semelhantes ao trabalho aqui desenvolvido, entre eles:

- *Tape Machine* (2010) – permite gravar sons do ambiente e editá-los (Mas não grava sons do próprio aparelho).
- *ChordBot* (2010) – permite efetuar uma composição sequenciando-se acordes de instrumentos predefinidos (não permite sequenciar sons gravados pelo usuário).
- *ULoops Studio* (2010) – permite, entre outras coisas, efetuar o sequenciamento de sons, porém não efetua espacialização e a música produzida é enviada para um servidor para geração de um arquivo final.

4 PROTÓTIPOS

Como a plataforma Android era relativamente nova, pouco difundida e explorada, foi feito um processo evolutivo de desenvolvimento baseado em protótipos para que fossem adquiridos fundamentos na plataforma. Como primeira etapa, era necessário explorar a plataforma a fim de se descobrir como se obtém acesso a recursos de som, como é feito o gancho para escutar eventos de interação e como produzir um aplicativo básico para a plataforma. Seu resultado é o protótipo *mixDroid v0.1*. Após essas descobertas, se concluiu que o melhor caminho a seguir seria continuar no mesmo rumo, adicionar a possibilidade de “gravar” a composição e escutá-la. Essa etapa culminou com o *mixDroid v1.0*. Foi decidido explorar a metáfora de espacialização, e para isso foi requisitado um protótipo que executasse dois sons simultâneos e que fosse possível espacializar os dois. Para isso foi desenvolvido um protótipo descartável totalmente novo a fim de validar a ideia. Para a próxima etapa era desejada a possibilidade de integração da atividade de espacialização no *mixDroid*. Para atingir esse objetivo a atividade de escutar a composição no *mixDroid* foi modificada. Essa iteração do desenvolvimento culminou com o *mixDroid v1.5*. Nesse momento, o *mixDroid* está salvando a composição internamente em um XML com referências dos sons tocados, seus tempos e espacialização. É desejado exportar a composição para um arquivo único a fim de publicar a composição. A fim de atingir esse objetivo, era primeiro necessário descobrir como efetuar a reprodução e gravação de arquivos sonoros em modo bruto (também chamado de baixo nível, onde se trabalha com *bytes* que representam a onda sonora). Essa etapa da pesquisa gerou outro protótipo descartável. A integração dessa atividade no *mixDroid* ainda não foi realizada.

4.1 Descrição sucinta

Resumindo o parágrafo anterior, os seguintes protótipos foram elaborados no decorrer da pesquisa:

1. *mixDroid v0.1* - Apenas uma tela;
2. *mixDroid v1.0* - Possibilidade de sequenciar uma composição e escutá-la;
3. Protótipo de Espacialização: Teste com dois sons para validar a ideia de implementação;
4. *mixDroid v1.5* - *v1.0* + Possibilidade de ajustar a espacialização da composição;
5. Protótipo de geração de som em tempo real: Teste para verificar como efetuar gravação e leitura de som em tempo real.

4.2 Descrição detalhada

A seguir cada protótipo será explicado em mais detalhes, incluindo figuras.

4.2.1 mixDroid v0.1

A primeira versão do mixDroid era composta de apenas uma atividade (conforme Figura 4.1) e serviu de base para conhecer da plataforma de desenvolvimento. A interface inclui nove botões: cada um mapeando um arquivo sonoro predefinido. Na tela existe um campo onde é possível visualizar um contador progressivo e um botão para disparar o contador. O protótipo não possui nenhum tipo de salvamento da interação realizada com o dispositivo.

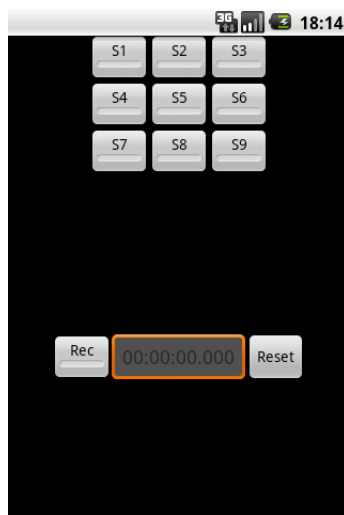


Figura 4.1: Primeiro protótipo do mixDroid

4.2.2 mixDroid v1.0

A versão 1.0 do mixDroid possui as seguintes funcionalidades:

- Gravação do sequenciamento de até nove sons definidos pelo usuário; e
- Escuta das composições salvas.

Nessa versão, o mixDroid abrange as seguintes atividades: Iniciar, Sequenciar, Ouvir. Atividades auxiliares - isto é, que não aplicam a metáfora de marcação temporal - foram definidas: Definir Nome da Composição, Escolher Composição Salva para Ouvir, Escolher Som Para os Botões. A atividade Iniciar possui a seguinte ação: escolher qual atividade realizar (mapeada para um item do menu). Sua implementação está vista na Figura 4.2.



Figura 4.2: mixDroid 1.0 - Atividade Iniciar

A atividade Sequenciar é composta pelas seguintes sub atividades:

- Nomear a composição;
- Compor;
- Mapear sons a botões;
- Escutar sons; e
- salvar Composição;

A atividade (a) foi mapeada numa série de ações identificadas por “Definir nome da composição”. Sua modelagem em CTT está descrita na Figura 4.3, e a implementação na Figura 4.4.

A atividade (b) “Compor” – a principal atividade dentro do sequenciamento – é realizada através das seguintes operações: iniciar, marcar, concluir. Para iniciar a atividade, o usuário pressiona o botão gravar, e após pode apertar livremente os nove botões disponíveis para marcar o posicionamento das amostras no eixo temporal. Para terminar de realizar a marcação temporal, é pressionado novamente o botão de gravar. Essas operações

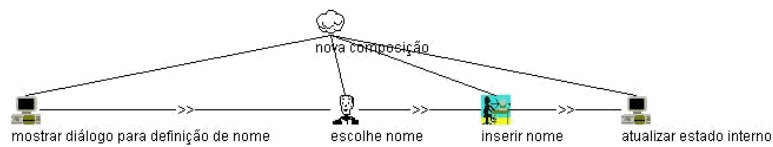


Figura 4.3: Modelagem CTT da atividade “Definir nome da composição”

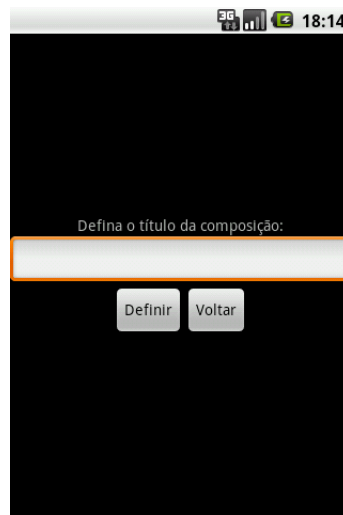


Figura 4.4: Implementação da atividade “Definir nome da composição”

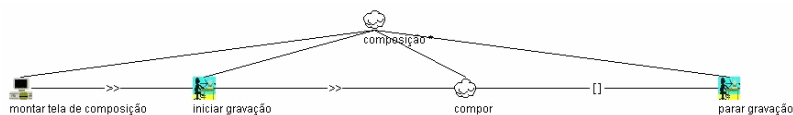


Figura 4.5: Modelagem da atividade compor (parte1)

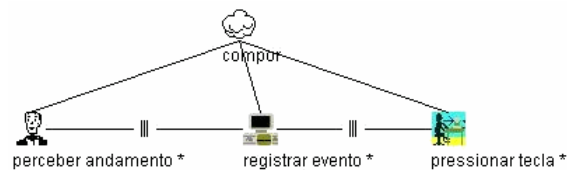


Figura 4.6: Modelagem da atividade compor (parte2)

se assemelham muito a utilização de um gravador simples para gravar uma composição – o compositor pressiona a tecla de gravar e após isso executa a obra. Sua modelagem em CTT é vista nas Figuras 4.5 e 4.6, e sua implementação na Figura 4.7.

A atividade (c) “Mapear sons a botões” foi desenvolvida com o auxílio de outra aplicação disponível gratuitamente chamada AndExplorer (ANDROID FREE FILE MANAGER, 2009) que possibilita escolher um arquivo disponível no sistema de arquivos do aparelho. A mudança no mapeamento ocorre da seguinte forma: O usuário efetua um toque longo no botão (no caso do emulador o gesto pode ser implementado como um clique longo no botão) e é iniciada a atividade. Através do AndExplorer é possível efetuar a navegação pelo sistema de arquivos do dispositivo e selecionar um arquivo de som para



Figura 4.7: Implementação da atividade Compor

mapear no botão. Uma imagem de sua implementação pode ser vista na Figura 4.8.

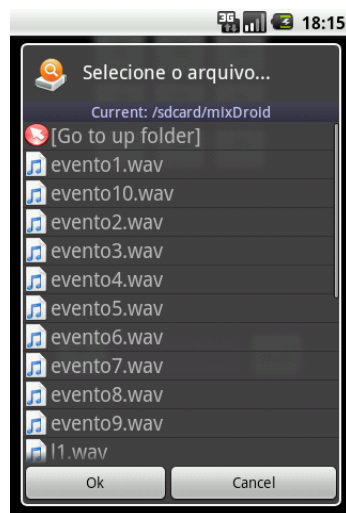


Figura 4.8: Implementação da atividade Compor — ação mapear som para botão

A atividade (d) “escutar sons” é realizada através de um clique simples no botão desejado da atividade Compor. Ele toca o som mapeado atualmente para ele.

A atividade (e) é realizada através de um menu, onde é exibida a opção de salvamento. É gerado um XML desenvolvido especificamente para a aplicação, o qual é salvo internamente no sistema.

A atividade “Ouvir”, última das atividades implementadas nessa versão do mixDroid, é composta pelas seguintes ações: escolher composição salva para ouvir e escutar a composição. A primeira ação foi implementada através de uma atividade auxiliar na qual são listadas todas as composições salvas no dispositivo e o usuário escolhe qual escutar (Ver Figura 4.9). A segunda ação é realizada através de uma tela em que estão dispostos 3 elementos: um botão para reproduzir e parar a composição, um botão para reinício da reprodução e um campo onde pode ser visualizado o tempo decorrido desde o início da composição (Figura 4.10).



Figura 4.9: Listar composições salvas

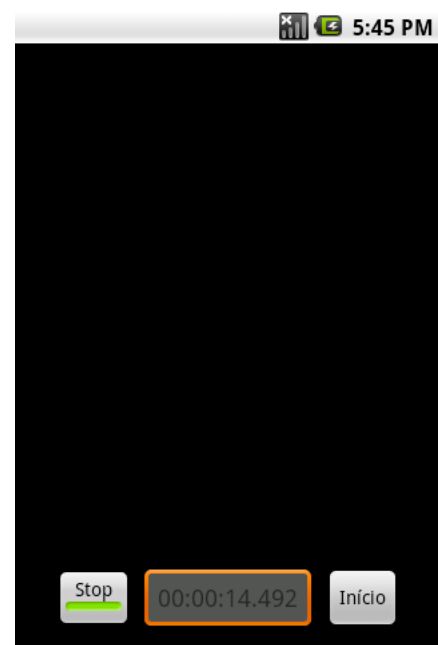


Figura 4.10: Escutar composição salva

4.2.3 Teste de espacialização

Esse protótipo foi desenvolvido para estudo de como realizar a mudança de espacialização incorporada na versão 1.5 do mixDroid. Foram estudados: criação dinâmica de elementos visuais (Botão na tela), elementos sonoros (objeto MediaPlayer), além de movimentação dinâmica dos botões (Através de toques na tela). Sua implementação é bem simples e está apresentada na Figura 4.11.



Figura 4.11: Protótipo de espacialização

Cada botão representa um som gerado pelo sistema. A posição dele na tela representa a espacialização atual do som. Movendo-se o botão no eixo horizontal muda-se a panorâmica do som entre canal esquerdo e direito. Movendo-se o botão no eixo vertical é possível trabalhar com a amplitude do som onde na parte superior da tela o som é emitido com amplitude de 100% e na parte inferior com amplitude de 0%.

4.2.4 Teste de geração de som em tempo real

O desenvolvimento desse protótipo teve como objetivo realizar um estudo sobre:

- Programação paralela e concorrente na plataforma;
- Leitura (do disco) e escrita (na placa de som) de arquivos de som em baixo nível (WAVE, 16 bits, 44.1kHz).

Através desse protótipo foi possível aprender como programar com *threads* na plataforma, além das diferentes maneiras de enviar som em baixo nível para o dispositivo de som. Como processamento sonoro é uma tarefa custosa para o sistema e não é possível bloquear a interação com o usuário pois ele pode querer gerar mais eventos para o sistema, o uso de programação concorrente no desenvolvimento de todos os outros protótipos foi necessário.

Primeiramente, foi necessário separar toda a computação relacionada a geração de som da *thread* que recebe notificações de interação com a tela pois se essa segunda *thread* fica certo tempo sem se notificar com o sistema a plataforma acha que o processo está em *loop* e termina com ele.

Foi preciso também separar a computação da leitura de cada arquivo do disco para que o processamento pudesse ser assíncrono - enquanto uma *thread* lê um arquivo a outra pode salvar no *buffer*. Uma outra *thread* lê o que está disponível em cada um dos *buffers* e combina num *buffer* final que será escrito para a placa de som, tarefa realizada por outra *thread*.

Além das *threads* citadas acima, os *buffers* são uma zona de exclusão mútua pois se a *thread* produtora e a consumidora lessem ao mesmo tempo poderiam encontrar um estado inconsistente.

A leitura e escrita de arquivos de som em baixo nível será incorporada em versões futuras do mixDroid para tornar possível a exportação da composição.

4.2.5 mixDroid v1.5

Essa versão é uma extensão da versão anterior, incluindo a seguinte atividade: Espacializar. A espacialização disponível no mixDroid 1.5 compreende a mudança de panorâmica e de intensidade. Essa atividade é realizada de forma semelhante à atividade “Ouvir”, tendo como mudança principal a interface. Além dos controles existentes anteriormente, são exibidos botões que representam os sons que estão sendo reproduzidos no momento. Esses botões aparecem na tela no momento que o som específico começa a executar e desaparece ao seu término. A ação que o usuário realiza para mudar a espacialização é mover os botões nos eixos vertical e horizontal da tela.

4.2.5.1 Panorâmica

A panorâmica determina a amplitude do som em relação aos canais esquerdo e direito. A posição padrão inicial de cada som é centralizado com volume de 50% e a localização do controle é no centro da tela. Mover o botão para a direita representa mover a fonte sonora para a direita, portanto o som do lado esquerdo será reduzido proporcionalmente a distância do eixo horizontal da tela. Raciocínio análogo pode ser feito ao mover o botão para o lado direito.

4.2.5.2 Intensidade

A amplitude do som é mapeada da seguinte maneira. Movendo-se o botão para a parte superior da tela, o som é reproduzido com 100% da amplitude - ou seja, atinge zero decibel de atenuação. Movendo-se para baixo, o som é reduzido até amplitude zero. Portanto, temos um eixo vertical central em que a amplitude original do som é reduzida em 50% (valor padrão inicial).

4.3 Modelagem CTT e exemplo de uso do mixDroid

Antes do desenvolvimento do mixDroid, foi produzida uma modelagem em CTT do seu funcionamento, apresentada em parte nas Figuras 4.3, 4.5, 4.6 e 4.12.

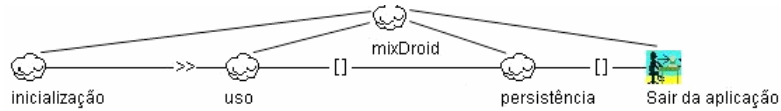


Figura 4.12: Modelagem CTT em mais alto nível do mixDroid

A seguir, será apresentada uma descrição completa das ações e operações que compõem uma atividade possível no mixDroid:

1. O usuário inicia o aplicativo e pode optar pela seguinte atividade: Iniciar uma nova composição (que leva o usuário para a sequenciação).
2. A sequenciação é procedida da seguinte maneira: O usuário define o nome da composição e em seguida é redirecionado para a atividade de sequenciar.
3. A atividade de sequenciar pode ser modelada como a Figura 4.5: O usuário percebe o andamento da composição, decide se pressiona uma tecla (gera um novo evento sonoro), e o sistema devolve o *feedback* para o usuário. A interação do usuário com o sistema é feita através de nove botões que são mapeados para sons escolhidos previamente pelo próprio usuário. Quando o usuário pressiona um botão, este muda de estado indicando sua ativação e o evento sonoro mapeado a ele é reproduzido. Ao término da execução do som, o botão volta para o estado inativo. Durante toda a gravação é exibido em um *display* na tela um contador do tempo transcorrido. A qualquer momento o usuário pode interagir com o sistema para fazer a gravação da composição.
4. Em seguida o usuário deseja realizar a espacialização da composição salva. Para realizar a espacialização, o sistema exibe as composições salvas no dispositivo e o usuário seleciona uma delas. Após essa seleção realiza-se a atividade “espacializar”.
5. Nela o sistema reproduz a composição definida. Enquanto o usuário percebe seu andamento, pode realizar intervenções no sistema movendo os botões para posições distintas na tela. A tela representa a posição espacial do som. Por exemplo, quanto mais à direita está o botão menos o som será reproduzido no canal esquerdo e vice-versa. Movendo-se o botão para a parte superior da tela o som é reproduzido com uma intensidade mais alta e movendo-se para baixo a reprodução ocorre com uma intensidade mais baixa. O sistema percebe a posição corrente do botão e realiza os ajustes necessários para que o *feedback* seja em tempo real. Ao término da sessão, o usuário pode salvar o estado atual da composição ou descartar as alterações.
6. Por último, o usuário pode decidir gerar um arquivo final único da composição, e para isso foi modelada a atividade de salvar a composição. O sistema exibe uma lista com todas as composições salvas no dispositivo, o usuário seleciona uma delas e o sistema produz (renderiza) o arquivo final.

5 EXPERIMENTOS

5.1 Estudo 1: Avaliação Qualitativa Com Usuários Leigos

Foi realizado um estudo comparativo com seis usuários tendo como objetivo verificar se eles entenderiam como a atividade de compor deve ser realizada. Para isso foi utilizada a versão 1.0 do mixDroid.

Nossa questão para avaliação era: “Uma interação na forma de ‘gravação de performance musical’, conforme implementada pelo mixDroid, é adequada para realizar tarefas de composição musical?”

Planejamos um experimento comparativo da adequação à tarefa, entre duas ferramentas: o mixDroid e o Kristal (KRISTAL AUDIO ENGINE, 2010). Kristal é uma ferramenta de edição sonora multi-pista gratuita para uso não-comercial. A escolha do Kristal foi motivada por dois fatores: 1 – O aplicativo realiza a função de mixagem; e 2 – Uma nova composição já inicia com trilhas vazias visíveis.

Nosso teste aproxima-se da abordagem “rápida e suja” sugerida por Preece (2007) e foi principalmente qualitativo. As medidas comparativas foram o desempenho (eficiência medida através do tempo total para realização da tarefa) e a impressão (satisfação subjetiva) dos usuários. A metodologia envolveu testes com usuários (“ensaios de interação”), em sessões individuais filmadas enfocando a tela do computador, com intuito de capturar a interação com o *software* e de registrar os comentários dos usuários. Cada sessão foi seguida de questionários de satisfação (incluindo um espaço para comentários livres).

Para a realização do experimento escolhemos a execução do mixDroid no emulador do Android. Dessa maneira, variáveis como fator de forma, *affordance* e mobilidade não entraram no estudo. A exclusão dessas variáveis permitiu uma comparação entre os mecanismos de interação visual – utilizados no Kristal – e os mecanismos de ancoragem sonora utilizados no mixDroid (KELLER et al., 2010). O dispositivo de interação se manteve o mesmo para as duas aplicações, limitando-se ao mouse.

5.1.1 Amostra

Seis sujeitos participaram no experimento com as características especificadas na Tabela 5.1. Os dados foram obtidos através do preenchimento de um questionário e através da documentação em vídeo para determinação dos tempos de realização das tarefas. As perguntas que envolviam uma gradação da resposta foram limitadas ao intervalo fechado de 0 a 5, sendo 0 a pior resposta e 5 a melhor.

Os participantes foram divididos em dois grupos. Um grupo iniciou a atividade de mixagem no mixDroid e o outro grupo utilizou primeiro o Kristal. Dessa forma, eliminamos o efeito da exposição prévia (com possível impacto no aprendizado) nos resultados do experimento.

Tabela 5.1: Perfil dos voluntários

Voluntários			Interesse por		Conhecimento em		
Id	Sexo	Idade	Computador	Música	Composição Musical	Kristal	Android
1	M	22	5	5	0	0	1
2	M	27	5	5	3	0	1
3	M	24	4	4	1	0	1
4	M	23	5	5	2	0	3
5	M	22	5	4	0	0	4
6	M	27	5	3	2	0	3

5.1.2 Procedimento Experimental

Uma composição-modelo de 20 segundos previamente gravada foi reproduzida para os voluntários, juntamente com a exibição de uma “partitura” num formato simples. A composição utilizou quatro amostras sonoras que foram identificadas como *evento 1*, *evento 2*, *evento 3* e *evento 4*. Cada evento foi representado por um número na partitura. Quando dois eventos tocam concomitantemente, foi utilizada uma simbologia para sua representação. A simbologia em questão são colchetes envolvendo os números dos eventos. A partitura apresentada aos voluntários possuía a seguinte inscrição:

$$1 \ 3 \ 4 \ 2 \ [4 \ 1] \ 4 \ 2 \ 1$$

Em um nível mais abstrato, a primeira tarefa que os usuários precisavam realizar foi a mesma: identificar como os sons são carregados para o programa. Em um nível mais concreto: no Kristal a tarefa foi identificar como os sons são carregados nas trilhas disponíveis; no mixDroid a tarefa foi identificar como os sons são carregados para os botões. Foi tomado o tempo para a assimilação das operações necessárias, o qual é apresentado na Tabela 5.2. A segunda tarefa foi recriar a composição escutada. Novamente, em um mais alto nível a tarefa é idêntica para as duas plataformas. Especificamente, no Kristal era necessário dispor os sons nas trilhas para reproduzir corretamente a composição; no mixDroid era necessário somente pressionar os botões no tempo correto. Os tempos para a execução da tarefa estão relacionados na Tabela 5.2. Na mesma tabela podem ser vistas as notas que os usuários deram para as ferramentas, juntamente com sua opinião pessoal de qual atendeu melhor os requisitos da tarefa.

Tabela 5.2: Tempo (em minutos e segundos) necessário para realizar as tarefas e impressões dos usuários quanto às ferramentas

Usuário(id)	Tarefa 1		Tarefa 2		Nota		Ferram. ideal
	Kristal	mixDroid	Kristal	mixDroid	Kristal	mixDroid	
1	00:15	01:00	04:06	00:50	2	4	mixDroid
2	00:30	00:50	08:10	00:42	3	5	mixDroid
3	00:14	01:30	03:30	00:33	4	3	mixDroid
4	00:27	01:00	10:00	02:00	4	3	mixDroid
5	00:20	00:30	06:10	00:37	4	4	Kristal
6	00:10	00:33	10:11	00:35	4	3	Kristal

Também foram recolhidas sugestões dos usuários que serão utilizadas para um refinamento da interação no mixDroid.

5.1.3 Resultados

Uma análise geral das duas tarefas mostra que os tempos de execução não estão relacionados à familiaridade com as interfaces dos ambientes. Se este fosse o caso, as duas tarefas teriam sido realizadas em tempos menores numa única plataforma. Na tarefa 1, todos os sujeitos obtiveram médias de execução mais baixas com o Kristal. Já na tarefa 2, as melhores médias corresponderam ao uso do mixDroid.

Pôde-se observar pelos vídeos que, após a identificação dos passos, os usuários realizaram as operações mais rapidamente — ou seja, houve aprendizado rápido no uso de ambas as interfaces. Se observou que a maior dificuldade nos usuários do mixDroid deveu-se à ação de manter o botão pressionado para selecionar o som — essa é uma ação não comum para usuários de desktop.

Na avaliação subjetiva, a média do mixDroid (3,667) foi levemente superior à média do Kristal (3,4), indicando que para as tarefas propostas não houve predisposição dos sujeitos a favorecer um ambiente em particular. A mesma conclusão pode ser observada pelas escolhas dos usuários sobre qual ferramenta é melhor para a realização dessa tarefa proposta. Quatro pessoas preferiram o mixDroid, e duas o Kristal.

5.2 Estudo 2: Avaliação Qualitativa Com Usuário Experiente

O processo de validação da metáfora de marcação temporal envolveu o uso do mixDroid para a criação de uma obra musical completa. O principal objetivo da validação foi testar se a técnica de interação seria adequada durante um ciclo completo de mixagem: a partir do estado inicial – definido como uma coleção de amostras de som, até o estado final – um conjunto de sons organizados no domínio tempo.

O sujeito escolhido para o estudo (um dos membros da nossa equipe) tem formação musical e vinte anos de experiência com tecnologia. A escolha foi baseada na necessidade de integrar um usuário experiente ao ciclo de desenvolvimento. As decisões composicionais são difíceis de avaliar e a qualidade do produto musical geralmente envolve avaliações subjetivas. A integração entre composição e validação parece ser a forma mais direta de evitar essas dificuldades.

O processo de mixagem abrangeu várias sessões de uso. O mixDroid foi utilizado no modo de emulação em um *laptop*. Várias dúzias de amostras foram utilizadas, com durações variando entre alguns centissegundos até aproximadamente dois minutos.

5.2.1 Resultados

O sujeito compôs a obra estéreo *Green Canopy: On the Road*, de sete minutos de duração (KELLER, 2009). A peça faz uso de sons coletados em uma localidade da Amazônia Peruana - nas margens do rio *Yanayacu*. Mais detalhes dos procedimentos composicionais podem ser encontrados em (KELLER et al., 2010).

6 DISCUSSÃO FINAL

Neste subprojeto, aplicamos uma metáfora de interação que fornece uma forma intuitiva de execução de mixagens em contexto ubíquo: a marcação temporal (KELLER et al., 2010). Para testar essa idéia, implementamos um protótipo para telefones celulares que suportam o sistema operacional livre Android. O protótipo mixDroid 1.0 permite combinar sons em tempo real através de um teclado virtual que segue o padrão do teclado numérico encontrado nos dispositivos móveis. Cada tecla é vinculada a um som escolhido pelo usuário. A mixagem é feita pressionando as teclas do celular, sendo que o posicionamento de cada som no eixo temporal corresponde ao tempo de acionamento da tecla. O estudo de uso realizado apontou por um lado para a aplicabilidade desta metáfora na criação sonora, e por outro para a eficiência desse mecanismo de interação na realização de mixagens por parte de usuários leigos.

Os objetivos iniciais da pesquisa foram atingidos em relação ao desenho de novos mecanismos de interação. O protótipo mixDroid 1.5 pretende oferecer ao compositor a possibilidade de realizar uma composição muito mais rica explorando mais ainda o padrão proposto de marcação temporal.

O estado atual do mixDroid possibilita que se realize a composição de uma obra, efetuando o sequenciamento e a espacialização dos sons. A ferramenta também oferece a possibilidade de escuta das obras salvas no dispositivo. Para a seguinte fase do projeto, planejamos implementar as seguintes funcionalidades:

- Exportação da composição para arquivo padronizado;
- Suporte para composição cooperativa;
- Importação e exportação de composições;
- Visualização e edição da composição salva para pequenas correções do usuário.

Através da exportação de uma obra para um arquivo padronizado, podemos prover ao usuário de um método fácil de distribuir suas composições. Através de atividades de composição cooperativa (MILETTO et al., 2005, 2009), podemos colocar a atividade composicional no contexto das atividades sociais, fornecendo suporte para o trabalho criativo em rede. A importação e a exportação das composições permitem o compartilhamento dos dados em diversas plataformas. Esse tipo de mecanismo de persistência é uma das formas de apoio à composição cooperativa.

REFERÊNCIAS

ANDROID: application fundamentals. [S.l.]: Google Inc., 2009. Disponível em: <<http://developer.android.com/guide/topics/fundamentals.html>>. Acessado em abril 2010.

ANDROID Free File Manager. [S.l.]: LYSESOFT, 2009. Disponível em: <<http://www.lysesoft.com/products/andexplorer/index.html>>. Acessado em agosto 2009.

ANDROID.COM. [S.l.]: Google Inc., 2009. Disponível em: <www.android.com>. Acessado em abril 2010.

ARPEGIATTOR - Projeto de pesquisa CNPq. Disponível em: <<http://lcm.inf.ufrgs.br/svn/arpejador/>>. Acessado em junho 2010.

CHORDBOT. [S.l.]: Contrasonic, 2010. Disponível em: <<https://market.android.com/details?id=com.chordbot>>. Acessado em junho 2011.

FLORES, L. V. **Defesa de Proposta de Tese**: uma infraestrutura para o design da interação musical com dispositivos móveis cotidianos. 2010. Tese (Doutorado em Ciência da Computação) — UFRGS, Porto Alegre, RS, Brasil.

GAYE, L. et al. Mobile music technology: report on an emerging community. In: NEW INTERFACES FOR MUSICAL EXPRESSION, 2006., Paris, France, France. **Proceedings...** IRCAM — Centre Pompidou, 2006. p.22–25. (NIME '06).

GIBSON, J. **The Ecological Approach to Visual Perception**. Hillsdale: Lawrence Erlbaum Associates, 1986.

HOLMQUIST, L. E. Ubiquitous music. **interactions**, New York, NY, USA, v.12, p.71–ff, July 2005.

KELLER, D. Green Canopy: on the road [obra musical]. In: BRAZILIAN SYMPOSIUM ON COMPUTER MUSIC (SBCM), 12. **Proceedings...** USP/SBC, 2009.

KELLER, D. et al. Música Ubíqua: conceito e motivação. In: ..., Curitiba. **Anais...** PPGM/UFPR, 2009. p.539–542.

KELLER, D. et al. Anchoring in Ubiquitous Musical Activities. , New York, NY, USA, 2010.

KRISTAL Audio Engine. [S.l.]: Kristal, 2010. Disponível em: <<http://www.kreatives.org/kristal/>>. Acessado em agosto 2010.

LEONTIEV, A. **Activity, Consciousness, and Personality**. [S.l.]: Prentice Hall, 1978.

MILETTO, E. M. et al. CODES: a web-based environment for cooperative music prototyping. **Org. Sound**, New York, NY, USA, v.10, p.243–253, December 2005.

MILETTO, E. M. et al. Music Creation by Novices should be both Prototypical and Cooperative - Lessons Learned from CODES. In: BRAZILIAN SYMPOSIUM ON COMPUTER MUSIC (SBCM), 12., Recife, PE, Brazil. **Proceedings...** USP/SBC, 2009.

MODELAGEM em CTT do mixDroid. [S.l.]: LCM-UFRGS & NAP-UFAC, 2009. Disponível em: <<http://mujava.googlegroups.com/web/mixDroid-ctte.zip>>. Acessado em novembro 2009.

MUSIC in Java - Projeto em cooperação entre a UFAC e UFRGS. Projeto de Pesquisa CNPq 571443/2008. Disponível em: <<http://groups.google.com.br/group/mujava/>>. Acessado em junho 2010.

MUSICXML Player - Projeto de pesquisa CNPq. Disponível em: <<http://lcm.inf.ufrgs.br/svn/musicxml-player/>>. Acessado em junho 2010.

PATERNO, F.; MANCINI, C.; MENICONI, S. ConcurTaskTrees: a diagrammatic notation for specifying task models. In: **Anais...** Chapman & Hall, 1997. p.362–369.

PREECE, J. **Design de interação além da informação homem-computador**. Porto Alegre: Bookman, 2007.

TAPE Machine. [S.l.]: Samalyse, 2010. Disponível em: <<https://market.android.com/details?id=com.samalyse.tapemachine>>. Acessado em junho 2011.

ULOOPS Studio. [S.l.]: ULoops Labs, 2010. Disponível em: <<https://market.android.com/details?id=net.uloops.android>>. Acessado em junho 2011.

WEISER, M. Human-computer interaction. In: BAECKER, R. M. et al. (Ed.). . San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995. p.933–940.

XXI Salão de Iniciação Científica UFRGS. Disponível em: <<http://www.propesq.ufrgs.br/sic2009/resumos/Salao2009/CET2009.pdf>>. Acessado em junho 2010.