

UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO



Sketchument: ambiente de experimentação para criação de instrumentos musicais digitais

Dissertação de Mestrado

Aluno: Filipe Carlos de Albuquerque Calegario
Orientador: Geber Lisboa Ramalho
Co-orientador: Giordano Ribeiro Eulalio Cabral

(fcac@cin.ufpe.br)
(glr@cin.ufpe.br)
(giordanorec@gmail.com)

Recife, 22 de fevereiro de 2013

Agradecimentos

Agradeço aos meus pais, por todo amor, suporte e segurança que me transmitem, por minha educação e pela política de “se for livro, não precisa nem perguntar: pode comprar!”.

A Alissa, companheira de todas as horas, pela paciência e pelo apoio que me deu nessa jornada, me escutando e me aguentando; pelas carinhosas gargalhadas que tanto me fazem bem.

Ao meu irmão, minha primeira cobaia, com quem eu sempre posso contar.

À minha família, meu porto seguro.

A Geber, por nossas discussões empolgadas, cheias de rabiscos e densas em ideias; pelos inúmeros feedbacks e melhoramentos propostos.

A Giordano, pelos comentários incisivos, pelas produtivas reuniões e pela veemente crença no projeto.

A Jeraman, por nossas aventuras interativas, pela constante energia e vontade de tirar as ideias do papel.

A Eneida Ende, pela disponibilidade, pela gentileza e pelas palavras.

A André Araújo, Cello Mello, Felipe Machado, HD Mabuse, Luís Felipe Ramalho e William Paiva, pelas conversas e impressões sobre o projeto.

A Alexandre Braz de Macêdo, Armando Menicacci, Colin Lasser, David Aragão, Edgar Neto, Fábio Caparica, Ricardo Teixeira, Rodolfo Ferraz e Tanja Gandit, pelos feedbacks sobre o sistema.

A Josie, Mabuse, Aninha e Ciro, pelas discussões sobre métodos e processos.

Ao grupo MustIC, pela instigação nas reuniões e nos projetos.

Ao Centro de Informática, por me proporcionar um ambiente de nível internacional.

Ao Colégio de Aplicação e a todas as pessoas maravilhosas que lá conheci, pelo ambiente carregado de discussões e de criatividade, que mudou a forma como eu vejo o mundo e me fez ser o que sou hoje.

A Alvinho, Chico, Manel, Marcelo e Pedro, meus bróderes de Faringes, pelo convívio, pela experiência e pelo amadurecimento que me proporcionaram.

A Thiago e a Ciro, por mostrar que existem mais coisas para se pendurar no varal.

Além de Acauã Borges do Nascimento, Aline Franciele Correia da Silva, André Luiz Buarque Vieira e Silva, Anita Dubeux, Cecília Lima, Chico Carvalho, Cícero Ferraz, Clarissa Costa, Danilo Soares, Dário Soares, Diogo Guedes, Diogo Rodrigues Cabral, Djeeffther Souza

Albuquerque, Eduarda, Eduardo Kenji, Eduardo Loureiro, Emerson de Lira Espínola, Evandro Lins, Fabrizio Barbosa Farias, Felipe Machado Duarte, Frank Junior, Gabriel Bandeira, Gabriel Miranda, Gabriel Wanderley Albuquerque, Gileno, Guilherme Queiroga, Heitor Rapela, Hugo Cristo, Hugo Leonardo Nascimento Almeida, Ian, Jackeline, Jaziel Vitalino Souza e Silva, João Erik Melo, João Nogueira Tragtenberg, Jonas Federman, Jonysberg Peixoto Quintino, Josão das Nhega, Leonardo Menezes, Luca Bezerra, Lucas Silva Figueiredo, Luciana Andrade de Oliveira, Madre Teresa, Madyana Torres, Manassés Bispo, Marcelo Araújo, Marcio, Márcio Brito Barbosa, Marcio Galdino, Marcos, Mariana Dourado, Marina Mapurunga, Michael Thomas Rodrigues de Arruda, Patrícia M., Phillippi Oliveira, Pipo, Rafael Moura, Raissa Andrade, Ramon Victor A. Bezerra, Ricardo "Bicudo" Marcelino, Ricardo Couto, Ricardo Queiroz, Robertson, Rodolfo Siri Ferraz, Thaís Vidal, Thiago Amazonas, Thiago Diniz, Valdeir Silva de Souza Júnior e Vanessa Gomes, que ajudaram na evolução deste projeto.

E eu não poderia esquecer de Alominha, com seus pelos sedosos e seu sorriso maroto.

Resumo

Novas interfaces estão mudando a forma como interagimos com as máquinas e isto é particularmente importante para o campo da música. Estas novas tecnologias abrem um grande leque de possibilidades para a criação de Instrumentos Musicais Digitais (DMIs), que são sistemas compostos por módulo de entrada, módulo de saída e uma estratégia de mapeamento conectando tais módulos. Ao contrário dos instrumentos acústicos, que impõem restrições físicas à sua fabricação, os DMIs permitem uma maior liberdade na sua construção. Paradoxalmente, esta vantagem pode se tornar um problema, pois não existem métodos ou ferramentas para guiar o designer ou luthier na definição de qual mapeamento é mais adequado entre entradas e saídas de um DMI. Seguindo a definição que o usuário é a força motriz da inovação e que a tendência do Movimento Maker mostra que, atualmente, tal usuário não é apenas um consumidor passivo de produtos, por que não fornecer aos músicos e artistas o poder de definir suas próprias estratégias de mapeamento e assim construir seu DMI? Neste contexto, a prototipação, conceito bastante importante no Design, se mostra como uma possível solução para o problema do mapeamento, permitindo que os usuários possam experimentar a adequação de entradas, saídas e mapeamentos diversos na criação de seu próprio DMI. Alguns sistemas musicais já permitem prototipação de DMIs (ex.: Pure Data, Max/MSP, Chuck, SuperCollider etc.), porém a maioria ainda precisa de um alto nível técnico para se chegar a um resultado não trivial. Este Projeto apresenta o Sketchument, um ambiente de experimentação de fácil utilização, voltado para o público não técnico, que permite que o usuário prototipe e crie DMIs, usando múltiplas entradas, saídas e mapeamentos. De protótipos de baixa fidelidade a protótipos funcionais, passando por avaliações por questionário e entrevista, o Sketchument tem sido desenvolvido seguindo a mesma filosofia de prototipação que ele se propõe a prover. O processo cíclico de concepção, implementação e avaliação tem produzido importantes resultados a partir dos potenciais usuários, que são bastante úteis para dar suporte às decisões de projeto e, assim, permitir modificações e melhoramentos.

Abstract

New interfaces are changing the way we interact with machines and this is particularly important for the musical domain. These new technologies open a wide range of possibilities in the creation of Digital Musical Instruments (DMIs), artifacts with inputs (interface controllers) and outputs (sound synthesis modules) that connected according to a mapping strategy. Contrary to acoustic instruments, which impose physical constraints to their design and fabrication, the design of DMIs has more freedom. Paradoxically, this advantage is a problem, since there is no established method or tool to guide the DMI designer or luthier to define how interfaces gestures should be adequately mapped into sound variables. Given that the user is the driving force of innovation and the Maker Movement currently shows that the user is not only a passive consumer, why not allow the musicians and artists to define their own mapping strategies and built their DMI? In this context, the prototyping, very important concept in Design, appears as a possible solution for the mapping problem, enabling the users to experiment more adequate inputs, outputs and mapping for their own context. Some musical systems that could help users to prototype DMIs already exist (e.g. Pure Data, Max/MSP, Chuck, SuperCollider etc.) However, most of them do not provide an adequate level of usability in order to be fully explored by non-technical audience. This Project presents the Sketchument, an environment devoted to help non-technical users to easily prototype DMIs, using multiple input modes and allowing the integration to other useful technologies. From low-fi paper prototype, to functional ones, passing through movies, questionnaires, interviews, Sketchument has been developed following the same prototyping philosophy we intend to propose to its users. The cyclic process of design-implementing-evaluating has produced valuable feedback from potential users, that have been very useful to back design choices and to push modifications and enhancements.

Sumário

Agradecimentos	2
Resumo	4
Abstract	5
Sumário	6
Índice de Imagens	12
1. Introdução	14
2. Instrumentos Musicais Digitais.....	17
2.1. Descrição	17
2.1.1. Entradas ou Controles Gestuais	18
2.1.2. Saídas ou Produções Sonoras	19
2.1.3. Mapeamento	20
2.1.4. Feedbacks Hápticos e Visuais.....	22
2.2. Classificação	23
2.3. Análise da Construção de um DMI	24
2.4. Considerações sobre o Design de DMIs.....	26
2.4.1. Impressões e Comparações entre Instrumentos Tradicionais e Digitais.....	27
2.4.2. Fáceis de aprender, possíveis de se aprofundar.....	28
2.4.3. Design de DMIs e Design de Games.....	29
2.4.4. Metáforas de Interação e Transparência de Mapeamentos	29
2.4.5. Coerência Gestual	30
2.4.6. Exploração de Novos Paradigmas de Interação.....	30
2.4.7. Requisitos de Latência.....	30
2.4.8. Esforço Cognitivo e Complexidade vs. Expressividade.....	31
2.4.9. Diferentes Diversidades para DMIs	31
2.5. Avaliação de DMIs	32
3. Diretrizes	34
3.1. Usabilidade	34
3.2. Feedback	34
3.3. Diversidade	35

3.4. Configuração	35
3.5. Integração	35
3.6. Disponibilidade	36
4. Estado da Arte	37
4.1. Sistemas Modulares para Música	37
4.1.1. Sintetizadores Modulares Analógicos	37
4.1.2. Linguagens de Programação Textual	38
4.1.3. Sistemas Orientados a Fluxograma	39
4.1.4. Interfaces Físicas e Tangíveis	40
4.2. Sistemas de Mapeamento de Dados	41
4.2.1. OSCulator	42
4.2.2. junXion v5	43
4.2.3. Libmapper	44
4.3. Novas Formas de Interação e Música	45
4.3.1. Controles de Videogames	45
4.3.2. Dispositivos Móveis e Telas Multitoque	49
4.3.3. Reflexões sobre Paradigmas de Interação	52
5. Bases Conceituais	55
5.1. Movimento <i>Maker</i>	55
5.2. Experimentação e Prototipação	57
5.3. Metadesign	58
6. Descrição do Processo	60
6.1. Métodos de Inspiração	62
6.1.1. Experiência Pessoal	62
6.2. Métodos de Prototipação	63
6.2.1. Protótipos Preliminares	63
6.2.2. Protótipos para iPad	63
6.3. Métodos de Investigação	63
6.3.1. Análise da Literatura	63
6.3.2. Análise de Aplicativos de iPad	63
6.4. Métodos de Avaliação	65

6.4.1. Entrevistas	65
6.4.2. Questionário Online	66
6.4.3. Beta Test.....	67
6.5. Métodos de Ideação.....	67
6.5.1. Definição das Diretrizes do Projeto.....	67
6.5.2. Caixas Morfológicas.....	67
6.5.3. Descrição de Perfis de Usuários e Cenários de Uso.....	68
6.5.4. Lista Ordenada de Funcionalidades	68
7. Inspiração e Protótipos Preliminares	69
7.1. Experiência Pessoal.....	69
7.2. Protótipo em Papel.....	70
7.3. Protótipo em Keynote.....	71
7.4. Protótipo em Processing	71
8. Análise de Aplicativos	73
8.1. Aplicativos Musicais	73
8.2. Aplicativos Concorrentes	73
8.2.1. Tabletop	73
8.2.2. Reactable mobile	75
8.2.3. Beatsurfing.....	76
8.2.4. Audulus for iPad.....	77
8.2.5. Quadro Comparativo.....	78
8.3. Aplicativos com Semelhança de Interface	80
8.3.1. Adobe Proto	80
9. Caixas Morfológicas.....	82
9.1. Caixa Morfológica de Funcionalidades	82
9.2. Caixa Morfológica da Interface	85
10. Protótipos Iniciais no iPad.....	86
10.1. Tecnologias Utilizadas	86
10.1.1. UIGestureRecognizer.....	86
10.1.2. libPD	88
10.1.3. Core MIDI e OSC	89

10.2. Versão 0	90
10.3. Versão 1	90
11. Protótipo Versão 2	92
11.1. Objetos de Entrada	92
11.1.1. Objetos Gestuais	93
11.1.2. Objetos Tocáveis	93
11.1.3. Objetos Tangíveis	94
11.2. Objetos de Saída	95
11.2.1. Conjunto de Notas	95
11.2.2. Disparador de <i>Samples</i>	96
11.2.3. Fluxo de Notas	96
11.2.4. Afrobeat Machine	97
11.2.5. Objeto OSC	98
11.3. Mapeamento	99
12. Questionário Aberto	102
12.1. Opiniões sobre o Conceito	103
12.2. Opiniões sobre Projetos Relacionados	104
12.3. Impressões sobre a Interface	105
12.4. Opiniões sobre Perfis de Potenciais Usuários	106
12.5. Opiniões sobre Contextos de Uso	107
12.6. Pontos Positivos, Negativos e Sugestões	109
13. Entrevistas	111
13.1. Aspectos Gerais	111
13.2. Possíveis Usos Musicais para o iPad	112
13.3. Impressões sobre o Sketchument	114
13.4. Testes de Interface	115
13.5. Exemplos de Instrumentos Construídos	116
13.5.1. Pandeiro	116
13.5.2. Acompanhamento Afrobeat	117
13.5.3. Disparador de Acordes	118
14. Perfis de Usuários e Cenários de Uso	120

15. Protótipo Versão 3	123
15.1. Considerações	124
16. Beta Test.....	125
16.1. Testflight.....	125
16.2. Dados de Uso (Quantitativo).....	126
16.3. Feedbacks (Qualitativo)	127
17. Lista Ordenada de Funcionalidades	130
18. Conclusão	134
18.1. Críticas ao Processo	134
18.2. Considerações sobre o Protótipo Atual.....	136
18.3. Trabalhos Futuros	137
Referências	139
19. Anexo I: Arquitetura do Sketchument	148
20. Anexo II: Descrição da Caixa Morfológica de Funcionalidades	152
20.1. Definição	152
20.2. Plataforma	153
20.3. Níveis de Controle.....	153
20.4. Funcionalidades Musicais	153
20.5. Interface Musical	153
20.6. Opções de gravação	154
20.7. Edição	154
20.8. Tipos de Arquivos Exportáveis.....	154
20.9. Tipos de Projetos Exportáveis	154
20.10. Opções de Compartilhamento	154
20.11. Formas de Compartilhamento.....	155
20.12. Conexão com outros Aplicativos.....	155
20.13. Conexões com o Computador	156
20.14. Conexões com Outros iPhone/iPads	156
20.15. Integração com Hardware	156
20.16. Loops Pré-carregados.....	157
20.17. Formas de Entrada	157

20.18. Formas de Importação	157
20.19. Produz Som	157
20.20. Download de <i>Presets</i> da Internet.....	157
20.21. Customização	157
20.22. In App-Purchase	158
20.23. Colaboração.....	158
20.24. Modo de Edição/Modo de Execução	158
21. Anexo III: Descrição da Caixa Morfológica de Interface	159
21.1. Adicionar Objetos.....	159
21.2. Mover Objetos.....	159
21.3. Excluir Objetos	159
21.4. Editar Objetos	159
21.5. Redimensionar Objetos.....	160
21.6. Conectar Objetos	160
21.7. Desconectar Objetos.....	160
21.8. Ajuda	160
21.9. Ante-canvas	160
21.10. Modo de Execução	161
21.11. Detalhes de Interface	161
21.12. Menus	161

Índice de Imagens

Figura 2.1: Instrumento Musical Digital.....	17
Figura 2.2: Exemplo de classificação de DMIs pela semelhança da interface de controle com instrumentos acústicos (extraído de (Miranda & Wanderley, 2006)).....	24
Figura 4.1: Classificação de sistemas musicais que permitem a experimentação de novos instrumentos.....	37
Figura 4.2: À esquerda, Robert Moog tocando o sintetizador modular Moog 55 em meados dos anos 60. À direita, Donald Buchla posando junto ao sintetizador modular Buchla 100 (1963). .	38
Figura 4.3: MAX/MSP e Pure Data	39
Figura 4.4: Reactable.....	40
Figura 4.5: Molecule Synth	41
Figura 4.6: Ableton Live	41
Figura 4.7: OSCulator	42
Figura 4.8: junXion v5	43
Figura 4.9: Interface Gráfica para o libmapper	44
Figura 4.10: Nintendo WiiMote e Nunchuck.....	46
Figura 4.11: Microsoft Kinect	47
Figura 4.12: Rastreamento de pontos do corpo usando OpenNI e Kinect	48
Figura 4.13: V Motion Project, projeto musical interativo feito com o sensor Kinect	49
Figura 4.14: Fragmentação das versões do Android de dezembro de 2009 a dezembro de 2012	51
Figura 4.15: Concerto for iPad and Orchestra	51
Figura 4.16: kinect studies por Johannes Kreidler	52
Figura 4.17: Exemplos de aplicativos para iPad que simulam instrumentos e dispositivos existentes.....	53
Figura 4.18: Interface gráfica do aplicativo Rebirth for iPad	54
Figura 6.1: Métodos utilizados no processo de design do Sketchument	62
Figura 6.2: Diagrama de Análise de Aplicativos	64
Figura 7.1: Primeiro protótipo feito em papel e animado através de um vídeo em stop motion .	70
Figura 7.2: Protótipo feito em Keynote.....	71
Figura 7.3: Protótipo implementado em Processing usando Kinect como interface de entrada.	72
Figura 8.1: Aplicativo Tabletop.....	74
Figura 8.2: Aplicativo Reactable mobile.....	75
Figura 8.3: Aplicativo Beatsurfing	76
Figura 8.4: Aplicativo Audulus for iPad	78
Figura 8.5: Aplicativo Adobe Proto.....	80
Figura 10.1: Máquina de Estados do UIGestureRecognizer.....	88
Figura 10.2: Protótipo para iPad v0	90

Figura 10.3: Protótipo para iPad v1	91
Figura 11.1: Tela de edição da versão 2 do protótipo.....	92
Figura 11.2: Gestos discretos de deslizamento com um dedo	93
Figura 11.3: Gestos contínuos de toque e movimento pela tela	93
Figura 11.4: Objeto tocável	94
Figura 11.5: Redimensionamento do objeto tocável.....	94
Figura 11.6: Agrupamento de objetos tocáveis para joystick.....	95
Figura 11.7: Conjunto de notas	96
Figura 11.8: Conjunto de Samples.....	96
Figura 11.9: Objeto musical responsável por liberar um fluxo de notas	97
Figura 11.10: Afrobeat Machine no Pure Data desenvolvido por Jarbas Jácome	98
Figura 11.11: Afrobeat Machine	98
Figura 11.12: Objeto OSC responsável por enviar mensagens pela rede.....	99
Figura 11.13: Conexões discretas	99
Figura 11.14: Conexões contínuas	100
Figura 11.15: Destaque para conexões discretas.....	100
Figura 11.16: Destaque para conexões contínuas	100
Figura 11.17: Múltiplas conexões discretas com um objeto de geração sonora	101
Figura 11.18: Múltiplas conexões discretas com múltiplos objetos de geração sonora	101
Figura 12.1: Opiniões sobre o conceito agrupadas por categorias.....	104
Figura 12.2: Projetos relacionados ao Sketchument mencionados no questionário	105
Figura 12.3: Impressões sobre a interface mencionadas no questionário.....	106
Figura 12.4: Perfis de potenciais usuários mencionados no questionário	107
Figura 12.5: Contextos de uso agrupados por categorias	108
Figura 12.6: Número de participantes que utilizariam ou não o Sketchument.....	108
Figura 12.7: Contexto de utilização mencionados pelos participantes	109
Figura 12.8: Pontos positivos mencionados no questionário agrupados por categorias	109
Figura 12.9: Pontos negativos mencionados no questionário agrupados por categorias.....	110
Figura 13.1: Menu flutuante do protótipo versão 2	116
Figura 13.2: Pandeiro digital	117
Figura 13.3: Acompanhamento Afrobeat	118
Figura 13.4: Disparador de acordes.....	119
Figura 14.1: Relação entre Perfis e Cenários	122
Figura 15.1: Protótipo para iPad v3	123
Figura 15.2: Tela de ajuda adicionada a versão 3 do protótipo	124
Figura 16.1: Uso do aplicativo pelos beta testers	127
Figura 19.1: Diagrama do Fluxo de Telas do Sketchument.....	149
Figura 19.2: Pacote de classes relativo ao modo de edição.....	150
Figura 19.3: Pacote de classes relativo ao modo de execução.....	151
Figura 19.4: Pacote de classes auxiliares.....	151

1. Introdução

O surgimento de dispositivos móveis com alto poder de processamento, novos controles de videogame e o fácil acesso a novos sensores e a plataformas de desenvolvimento são alguns fatores que estão influenciando uma crescente popularização de novas formas de interação entre humano e computador (Saffer, 2008). Telas multitoque, controles com acelerômetros e posição rastreável, câmeras de profundidade com percepção em 3D, por exemplo, permitem interações inovadoras através de gestos, movimentos e manipulações diretas de elementos gráficos.

Novas interfaces trazem novas ideias de produtos, artefatos e paradigmas (McGlynn, Lazzarini, Delap, & Chen, 2012) e para o contexto musical não é diferente. Essas novas tecnologias de interface, combinadas, abrem um grande leque de possibilidades para a criação de novos Instrumentos Musicais Digitais (*Digital Musical Instruments*, DMIs), artefatos musicais cujas entradas (controles) e as saídas (síntese sonora) são independentes e interconectadas por uma estratégia de mapeamento (Miranda & Wanderley, 2006).

Ao contrário dos instrumentos acústicos, que possuem uma restrição física para sua concepção e construção, o design de DMIs é contemplado com uma maior liberdade. As estratégias de mapeamento, que são a alma dos instrumentos musicais digitais, são virtualmente incontáveis (Rovan, Wanderley, Dubnov, & Depalle, 1997). Paradoxalmente, esta vantagem pode ser enfrentada como um problema, já que não existe nenhum método estabelecido ou uma ferramenta que guie o designer ou luthier a definir quais parâmetros gestuais da interface devem ser adequadamente mapeados em variáveis musicais (Miranda & Wanderley, 2006).

Associando este fato com a ideia de que o usuário é a força motriz da inovação (Baldwin & Hippel, 2010; Bogers, Afuah, & Bastian, 2010; D. A. Norman, 2004), por que não fornecer aos músicos e artistas o poder de definir suas próprias estratégias de mapeamento e, por conseguinte, construir seu próprio instrumento dependendo das suas preferências, intenções e contextos de uso?

Dar poder ao usuário é uma estratégia que converge em algumas tendências atuais, como a cultura "*Do It Yourself*" ou movimento "*Maker*" (Dougherty & Frauenfelder, 2005) que já se espalha pela internet e também está presente no campo da música (Kirn, 2011; Steiner, 2005). O usuário de hoje em dia parece apresentar um desejo de ser mais proativo, participar da construção da solução, adaptá-la para seus interesses, em vez de serem simples consumidores passivos de produtos (Anderson, 2012).

Neste contexto, é importante lembrar que prototipação tem um papel importante no design (Dow, 2011), já que ajuda a identificar falhas nos artefatos, redirecioná-los, ajustá-los, entender como funcionam em determinados contextos e gerar novas ideias (Warfel, 2009). Além disso, a

prototipação faz com que as alternativas tornem-se concretas, facilita a comunicação de ideias e auxilia a formalizar e validar os objetivos (Dow, 2011; Gerber & Carroll, 2011; Gray, Brown, & Macanufo, 2010; Hartmann, 2009). Então, pode-se concluir que dar aos músicos a oportunidade de criar protótipos de DMI de maneira fácil pode ser uma abordagem promissora para o problema de mapeamento apresentado anteriormente.

Alguns sistemas voltados para a música já ajudam os usuários interessados a prototiparem seus próprios instrumentos (ex.: Pure Data, Max/MSP, Chuck, SuperCollider etc.) (McCartney, 2002; Puckette, 2002; Wang, 2008). Neste Projeto, tais sistemas são apresentados em dois grupos: Sistemas Modulares para Música, que permitem a interconexão de módulos pré-definidos com o intuito de gerar resultados musicais; e Sistemas de Mapeamento de Dados Musicais, que auxiliam no processo de mapeamento entre dados de controle de entrada e saída musical. Apesar de serem poderosas ferramentas, o usuário ainda necessita de um alto nível técnico para se chegar a um resultado não trivial, o que afasta músicos, designers e artistas que possuem pouco ou nenhum conhecimento em linguagens de programação e tecnologia digital.

Este Projeto apresenta o Sketchument, um ambiente de experimentação de fácil utilização, voltado para o público não técnico, que permite que o usuário prototipe e crie DMIs. Neste ambiente, módulos pré-determinados, representando controles de entrada e saídas musicais, podem ser combinados de forma ágil, gerando resultados imediatos.

Para o processo de concepção e desenvolvimento do sistema, foi usada uma abordagem de design centrado no usuário. A ideia é entender qual é a melhor forma de apresentar este ambiente ao usuário através de diversos ciclos de inspiração, investigação, ideação, prototipação e avaliação do sistema, trabalhando diretamente com o usuário.

Para a implementação do conceito do ambiente de experimentação, um protótipo funcional foi desenvolvido no iPad, escolhido como plataforma inicial por apresentar uma tela multitoque com dimensão razoável, boa precisão e baixa latência, além de estar se tornando bastante popular no meio musical. As diversas versões do protótipo foram submetidas à análise e à avaliação de potenciais usuários e os resultados serviram de base para a evolução do ambiente de experimentação.

Os resultados são encorajadores, incluindo uma avaliação favorável por parte dos usuários e diversos melhoramentos absorvidos das diversas avaliações realizadas.

A estrutura desta Dissertação será comentada a seguir:

O Capítulo 2 apresenta o contexto de aplicação desta pesquisa. Nele, é definido o modelo de Instrumento Musical Digital usado no Projeto, apresentando classificações possíveis para tais sistemas e seus componentes e fazendo uma reflexão sobre o seu design e a sua construção.

O Capítulo 3 apresenta as diretrizes usadas no desenvolvimento deste Projeto.

No Capítulo 4 é apresentado o Estado da Arte através de projetos relacionados divididos em dois grupos: Sistemas Modulares para Música e Sistemas de Mapeamento de Dados. Neste Capítulo ainda são descritas novas interfaces e discutidas as adequações de paradigmas de interação para elas.

No Capítulo 5 são apresentadas as Bases Conceituais do Projeto, nas quais se destacam o Metadesign, a tendência "Maker" e algumas reflexões sobre a importância da prototipação e da experimentação.

O Capítulo 6 descreve os métodos utilizados na concepção e desenvolvimento do Sketchument, apresentando alguns processos inspiradores de design.

Do Capítulo 7 ao Capítulo 18, são apresentados os resultados das aplicações dos métodos.

O Capítulo 19 apresenta as considerações finais e os trabalhos futuros.

2. Instrumentos Musicais Digitais

2.1. Descrição

Nos últimos anos, a crescente comunidade que pesquisa novas formas de interação musical vem adotando um modelo para descrever sistemas musicais interativos cujos controles de entrada são independentes das saídas sonoras. São os Instrumentos Musicais Digitais, ou DMIs. Nestes instrumentos, as entradas são mapeadas nas saídas seguindo algum tipo de estratégia definida pelo respectivo designer (Malloch, Birnbaum, Sinyor, & Wanderley, 2006).

Como uma crítica a esta nomenclatura, (Gurevich & Cavan Fyans, 2011) argumentam que nem todas as formas de interações musicais são instrumentais, propondo assim um termo para substituir Instrumentos Musicais Digitais: Interações Musicais Digitais. Apesar de considerar a discussão relevante, neste Projeto será usado o termo Instrumento Musical Digital como um artefato composto por partes que podem ser analisadas independentemente, com o qual o usuário interage com o objetivo de obter um resultado musical.

A Figura 2.1 apresenta um modelo genérico de DMI baseado no diagrama apresentado em (Rovan et al., 1997). Neste modelo, pode-se ver o módulo de controle gestual se relacionando com o módulo de geração sonora através de uma estratégia de mapeamento. Além disso, veem-se os feedbacks primário e secundário.

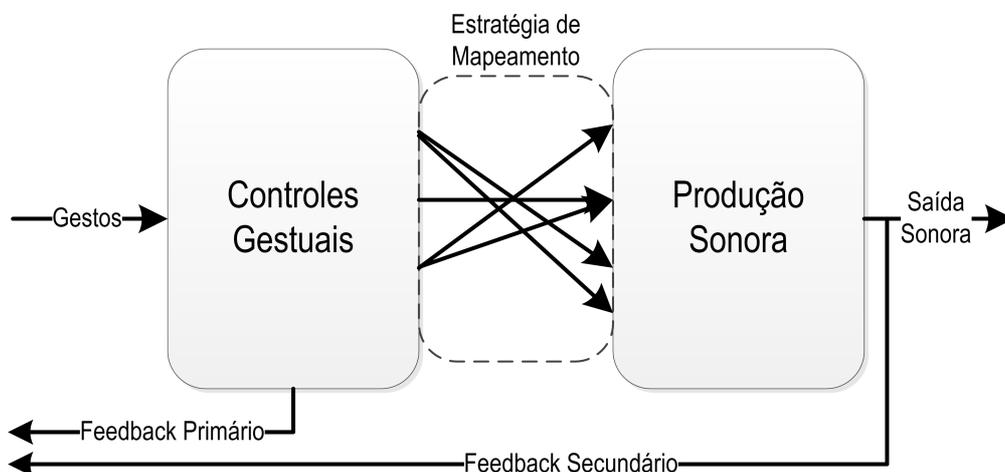


Figura 2.1: Instrumento Musical Digital

Para uma análise do modelo, o DMI será dividido em quatro grupos de componentes:

- Entradas ou controles gestuais;
- Saídas ou produções sonoras;
- Estratégias de mapeamento;
- Feedbacks.

Segue a descrição de cada grupo com algumas discussões relacionadas.

2.1.1. Entradas ou Controles Gestuais

Podem-se entender as entradas de um DMI como sendo um ou mais sensores que traduzem o gesto de controle feito pelo usuário para uma série de valores que podem ser interpretados digitalmente pelo computador.

Os valores brutos provenientes dos sensores precisam, na maioria das vezes, de um processamento antes de serem mapeados (Malloch, Sinclair, & Wanderley, 2008; Steiner, 2005). As operações mais simples envolvem converter um conjunto de valores de uma unidade para outra e operações mais rebuscadas se relacionam com a aplicação de filtros para suavizar os valores advindos do sensor ou para remover ruídos.

Com o intuito de refletir sobre as possíveis entradas de um DMI de maneira sistemática, (McGlynn, 2011) classifica os dados de entrada de acordo com a sua complexidade, separando-os em três grupos: brutos, simbólicos e gestuais, como se pode ver abaixo:

- Dados brutos: (*on/off*), posição de *faders*, coordenadas X, Y, Z dependendo do sistema de entrada;
- Dados simbólicos: poses predefinidas resultantes da associação de dados brutos;
- Dados gestuais: ações predefinidas associadas ao movimento dinâmico resultante da combinação de dados simbólicos ou dados brutos.

De forma geral, (Overholt, 2009) apresenta uma classificação de gestos formulada por (Cadoz, 1994) para categorizar as várias funções dos gestos da mão:

- Gestos ergóticos¹: aqueles que não carregam nenhum significado e que são caracterizados pela manipulação física de objetos. São os gestos mais usados no contexto de instrumentos musicais. Esta classificação pode ser subdividida em três funções:
 - Seleção: há a escolha de um valor dentro de um espaço discreto de valores possíveis. Por exemplo, a escolha das teclas quando se toca um piano ou da posição dos dedos aos se tocar uma flauta;
 - Excitação: a energia é adicionada ao sistema. Por exemplo, o movimento do arco de um violino ou o sopro para se tocar uma flauta;
 - Modulação: a forma dada a algum parâmetro de controle. Por exemplo, quando é feito um vibrato, que é a modulação da altura de uma nota ou o tremolo, que é a modulação da amplitude ou volume desta nota.

¹ Traduzido do inglês *ergotic gestures*

- Gestos semióticos: aqueles usados para comunicar informações significativas, como plegar para cima para informar que algo está correto. Com o uso de técnicas de visão computacional e reconhecimento de gestos, estes gestos podem ser incorporados no contexto musical interativo;
- Gestos epistêmicos: movimentos exploratórios para adquirir informação háptica ou tátil de um determinado artefato. Pode-se considerar como uma fase preliminar aos gestos ergóticos, no qual o usuário está experimentando o espaço para melhor entendê-lo.

Associando a classificação de controle gestual usada em (Apple Inc., n.d.; Microsoft Corporation, 2012) com a taxonomia de gestos em superfícies apresentada por (Wobbrock, Morris, & Wilson, 2009), pode-se descrever o controle por gestos a partir do seu fluxo de execução, como:

- Gestos discretos: a resposta ao seu acontecimento ocorre após o término da sua realização. Dentro dos gestos discretos, existem:
 - Gestos dinâmicos: gestos que estão associados a um movimento. Por exemplo, quando o usuário faz um símbolo de interrogação (?) com o dedo e o sistema apresenta uma tela de ajuda;
 - Gestos estáticos: também chamados de poses ou posturas, são gestos que não estão associados a um tipo de movimento, mas sim ao reconhecimento de uma posição estática feita pelo usuário.
- Gestos contínuos: onde a resposta ao seu acontecimento ocorre durante a sua realização. Por exemplo, quando o usuário com dois dedos faz o movimento de “pinça” em uma tela multitoque e o sistema aumenta o tamanho de uma foto.

Um destaque interessante para este tipo de classificação de gestos é a ligação com o tipo de controle do músculo. “A origem de todos os gestos é através da ação muscular, e os neurofisiologistas reconhecem dois tipos de controle dos músculos: controle corrente que são movimentos sustentados e podem ser modificados enquanto estão sendo realizados; e controle balístico, caracterizado por movimentos curtos e enérgicos” (Goldstein, 1998). Aparentemente, um paralelo pode ser traçado entre os gestos contínuos e o controle corrente e entre os gestos discretos e o controle balístico.

2.1.2. Saídas ou Produções Sonoras

As saídas de um instrumento musical digital são as formas de geração de som que ele possui. Nesta subseção, não será feita descrição das diversas formas de síntese sonora. Para mais detalhes nesta área, são recomendadas as seguintes leituras: (Cook, 2002; Miranda, 1998). Neste Projeto será abordada a geração de som em um nível de abstração mais alto, preocupando-se mais em definir as possibilidades de controle em cima da geração sonora.

Em (Schloss, 1990) é apresentada uma classificação para controles musicais baseada no nível de abstração que inclui propriedades do som, notas e processos musicais. Será utilizada esta

classificação no contexto do controle de possíveis saídas musicas de um DMI. Abaixo, tem-se a descrição dos níveis de controle:

- Nível de timbre: é o controle microscópico, no qual as propriedades do som podem ser manipuladas para geração de timbres. Trata-se de um nível que demanda controle contínuo dos parâmetros de síntese em diferentes dimensões. Por exemplo, quando um usuário de um sintetizador está experimentando novos sons, ele está controlando a saída sonora no nível de timbre;
- Nível de nota: é o controle com o qual se está acostumado ao pensar em um instrumento acústico como o piano. Quando uma tecla do piano é tocada, o músico está controlando a saída sonora no nível de controle de nota. Também pode-se chamar de nível de evento, no qual o usuário tem o controle de disparar eventos no sistema;
- Nível de processo: trata-se do nível macroscópico, no qual há o controle de um processo musical pré-definido. Uma situação que se enquadra neste nível é o disparo de *samples* por um DJ que está fazendo uma música baseando-se em gravações ou quando um maestro aumenta a dinâmica da música sendo tocada por uma orquestra.

Esta classificação é empregada por (Birnbaum, Fiebrink, Malloch, & Wanderley, 2005) para definir uma das dimensões utilizadas em uma forma de visualização voltada para comparação de DMIs.

2.1.3. Mapeamento

Em contraste com instrumentos musicais acústicos, que são limitados fisicamente pela sua forma de produzir som, o DMI permite uma maior liberdade, já que o som é sintetizado digitalmente por um módulo independente da interface de controle. Este fato permite uma ampla gama de possibilidades e torna a estratégia de mapeamento uma das partes mais importantes no momento de sua concepção (Sergi Jordà, 2005). Esta “nauseante infinidade de possibilidades” (Ryan, 1991) pode ser considerada um problema na realização dos mapeamentos e por isso será estudada aqui de forma detalhada.

As estratégias de mapeamento são a essência dos instrumentos musicais digitais (Rovan et al., 1997). Estratégias diferentes para o mesmo conjunto de entradas e saídas afetam como o intérprete reage musical e psicologicamente ao instrumento (Hunt, Wanderley, & Paradis, 2003). Isso leva a pensar que, dependendo do contexto ou da intenção do músico, tais estratégias podem ser alteradas, gerando o que pode se considerar outro instrumento.

Pode-se reduzir o universo do projeto de um DMI ao seguinte cenário: “Cada link entre o performer e o computador tem de ser inventado antes que qualquer coisa possa ser tocada” (Ryan, 1991). Ou seja, o projeto de um Instrumento Musical Digital não se restringe a se criar uma interface de controle e uma forma de gerar o som. O designer precisa se preocupar em realizar as ligações entre as partes, caso contrário não existe instrumento.

Classificam-se os tipos de mapeamento levando em consideração alguns critérios. Pela natureza do mapeamento, ele pode ser caracterizado em (Wanderley, 2006):

- Mapeamento implícito: as conexões entre os módulos de entrada e saída são resultado de algum processo, por exemplo: redes neurais, e tem uma “abordagem caixa-preta” (Nort & Wanderley, 2006);
- Mapeamento explícito: define a relação entre os módulos de controle e de geração sonora pela ligação entre dois conjuntos de variáveis. Ou seja, onde esta “relação pode ser expressa analiticamente” (Nort & Wanderley, 2006).

Como exemplo do mapeamento implícito, tem-se (Cont, Coduys, & Henry, 2004), que se baseiam no uso de redes neurais para gerar o mapeamento a partir do aprendizado de gestos do usuário. No projeto citado, a definição das estratégias de mapeamento se baseia em um processo iterativo, “onde o usuário ranqueia o desejo por um dado gesto, levando a um conjunto de gestos selecionados no final” (Steiner, 2005).

Já em relação ao mapeamento explícito, pode-se ter como exemplos os sistemas de mapeamento gráfico que serão descritos na Seção 4.2. Além desses sistemas, existem os projetos LoM (Nort & Wanderley, 2006) e MnM (Bevilacqua, Müller, & Schnell, 2005), que, apesar de usarem um processo de interpolação para ajudar o usuário a fazer o mapeamento, permitem tanto entender como as ligações estão sendo feitas, como ainda a edição posterior dessas estratégias (Malloch et al., 2008). E aí reside o benefício de um mapeamento explícito, que é ter conhecimento de como as conexões estão sendo realizadas e, se houver a necessidade, ter a possibilidade de adaptá-las para diferentes contextos de uso (Nort & Wanderley, 2006).

Pela quantidade de entradas e saídas, o mapeamento pode ser classificado (Rovan et al., 1997) como sendo:

- Um-para-um: relação na qual cada controle independente de gesto está associado a um parâmetro de saída musical. É o “mais simples dos mapeamentos, mas geralmente o menos expressivo” (Rovan et al., 1997);
- Um-para-muitos (mapeamento divergente): relação na qual uma variável de entrada gestual controla simultaneamente mais de um parâmetro musical. Neste mapeamento, poucos controles geram muitas saídas sonoras. Este fato pode dar certa expressividade na quantidade, mas peca em não permitir a manipulação de detalhes sonoros individualizados;
- Muitos-para-um (mapeamento convergente): relação na qual mais de um parâmetro de entrada está associado a apenas um parâmetro musical. Neste caso, certas variáveis de entrada podem influenciar (ou modular) os valores de outras variáveis também de entrada. Apesar de mais complexo que os demais mapeamentos apresentados anteriormente e de ter um domínio mais difícil, esta convergência “prova-se muito mais expressiva que os mapeamentos unitários mais simples” (Rovan et al., 1997).

Pela sua simplicidade, mapeamentos um-para-um não permitem uma interação profunda do usuário com o instrumento. Pesquisas revelam (Hunt et al., 2003) que o usuário sente-se mais compelido a usar uma interface musical mais desafiadora do que uma simples de controlar e fácil de entender.

Em uma tentativa de facilitar a criação de mapeamentos mais complexos que os um-para-um, foi pensada a utilização de camadas intermediárias entre os controles e as saídas musicais que elevariam o nível de abstração para a manipulação de parâmetros mais significativos que os valores brutos vindos dos sensores ou variáveis dos sintetizadores. Essas camadas teriam como objetivo processar os dados brutos agregando mais semântica aos seus comportamentos, por isso foram denominadas de camadas semânticas (Malloch et al., 2008).

Como existe uma tendência dos designers de instrumentos em fazer mapeamentos um-para-um quando estão desenvolvendo a interface, o uso de tais camadas semânticas pode permitir a simplicidade de mapear uma entrada diretamente em uma saída, deixando transparente para o designer toda a complexidade de funcionamento das conexões entre valores brutos dos sensores e sintetizadores (Hunt et al., 2003).

Pelo número de camadas de abstração entre as entradas e saídas, pode-se classificar o mapeamento como:

- Mapeamento direto ou de uma camada: conexão direta entre os valores brutos dos sensores aos parâmetros dos sintetizadores;
- Mapeamento multicamadas: conexão dos dados brutos a camadas intermediárias que processam os valores e agregam significado a eles.

Pode-se entender uma camada semântica associada a valores de sensores de movimento, por exemplo, como o posicionamento de um módulo de reconhecimento de gestos que dá significado ao conjunto de dados brutos. Assim, uma sequência de números aparentemente desconexos pode ser entendida como o movimento circular de uma mão. Esta forma de pensar o mapeamento se relaciona diretamente com a classificação de dados de entrada (brutos, semióticos e gestuais) já apresentada na Seção 2.1.1.

2.1.4. Feedbacks Hápticos e Visuais

Além do feedback sonoro proveniente do módulo de saída sonora, pode-se ter no DMI:

- Feedbacks hápticos, relacionados a referenciais táteis, geralmente produzidos por texturas, características superficiais do instrumento ou motores de vibração;
- Feedbacks visuais, que podem ser simples, como o uso de luzes e LEDs (*light emitting diode*, ou diodo emissor de luz) ou mais rebuscados como a geração de imagens e visualizações influenciadas pelo som do instrumento.

A ausência de feedbacks hápticos na maioria dos projetos de DMI é uma das principais críticas quando comparados aos instrumentos acústicos (Magnusson & Mendieta, 2007). Ao abrir

canais não usados de comunicação com o cérebro (Steiner, 2005), tais *feedbacks* ajudam a compreender melhor o funcionamento do sistema e proporcionam uma melhor utilização.

Dependendo da sua origem, a classificação do *feedback* pode ser:

- Primário: quando é gerado diretamente pelo controle gestual;
- Secundário: quando é gerado a partir de dados provenientes da geração sonora.

Além disso, pode-se detalhar o *feedback* usando como critério a produção do efeito:

- Passivo: inerente ao próprio dispositivo de interação, sem a necessidade de uso de nenhum atuador, como por exemplo “o som das teclas de um teclado de computador quando se escreve um texto” (Wanderley, 2006);
- Ativo: quando, para a sua geração, é necessário o funcionamento de algum atuador, como exemplo um alto-falante, motor, luz etc.

2.2. Classificação

De acordo com a semelhança com instrumentos acústicos, pode-se classificar os DMIs (Marshall, 2010; Miranda & Wanderley, 2006; Wanderley & Orio, 2002) como:

- Instrumentos Acústicos Aumentados: são instrumentos acústicos munidos de sensores capazes de captar dados da interação com o intuito de ampliar a expressividade do músico seja por geração paralela de som ou da modificação do próprio som produzido pelo instrumento;
- Simuladores de Instrumentos Acústicos: são DMIs projetados para reproduzir fielmente o funcionamento de um instrumento acústico. Por exemplo, o teclado eletrônico que simula o piano;
- Instrumentos inspirados em Instrumentos Acústicos: são DMIs que carregam algumas características inspiradas em instrumentos acústicos, porém não têm o objetivo de reproduzir funcionalidades destes instrumentos;
- Instrumentos Alternativos: são DMIs baseados em interfaces alternativas e inovadoras que não possuem semelhanças com instrumentos acústicos. Apesar desse total desligamento com a aparência de instrumentos convencionais, estes instrumentos podem se inspirar em objetos existentes para o seu projeto. As novas formas de interação com o computador se relacionam diretamente com esta categoria e apresentam desafios de projeto de interface que precisam ser melhor fundamentados e estudados.

Na Figura 2.2, versão traduzida da figura apresentada em (Miranda & Wanderley, 2006), estão alguns exemplos de DMIs existentes posicionados no contínuo desta classificação.

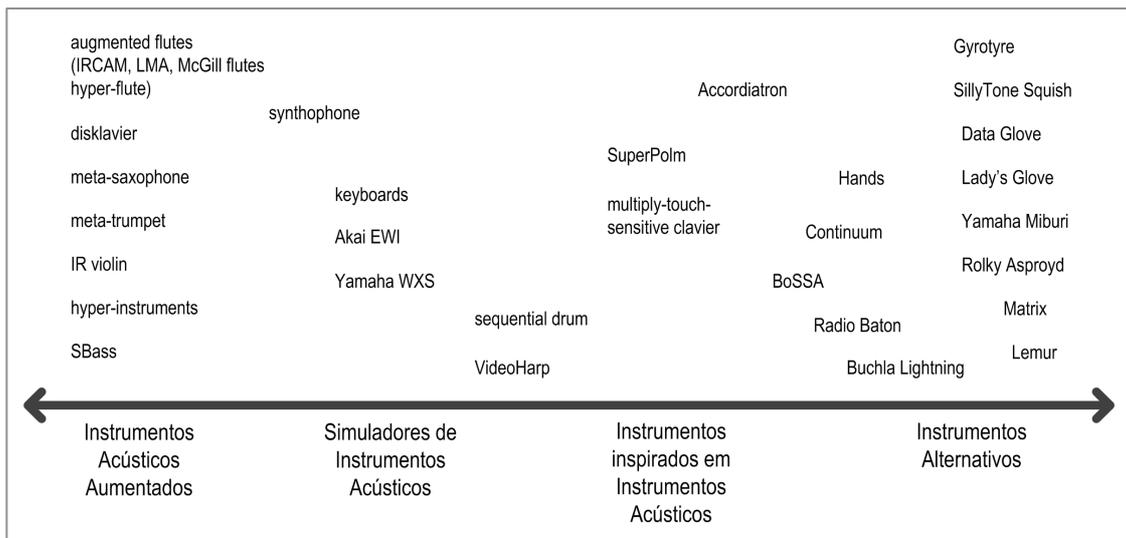


Figura 2.2: Exemplo de classificação de DMIs pela semelhança da interface de controle com instrumentos acústicos (extraído de (Miranda & Wanderley, 2006))

Em (Marshall, 2010), pode-se ver uma subdivisão da classe de instrumentos alternativos, proposta por (Mulder, 2000), que leva em conta controladores baseados em toque, por meio dos quais o usuário interage através do toque em uma superfície física; controladores de alcance expandido, permitindo ao usuário interagir através de pouco ou nenhum contato físico e que proporcionam mais liberdade de movimento, porém com algumas limitações de captação de gestos; e controladores imersivos, que capturam todos os gestos produzidos pelo usuário.

Além dessas classificações, (Malloch, 2008, pp. 4–7) propõe classificações por som, por interface e por interação e (Birnbbaum et al., 2005) apresentam uma ferramenta de visualização de dados que agrega vários critérios para comparação de DMIs e conseqüentemente pode ser usada como outra forma de classificação desses instrumentos.

2.3. Análise da Construção de um DMI

Nesta Seção, serão analisadas algumas sequências de passos encontradas na literatura que visam auxiliar o designer de DMIs a construí-lo. A palavra construção é aqui usada no sentido de implementação de ideias e intenções já concebidas pelo designer.

Esta liberdade de escolha no projeto de DMIs é de fato uma vantagem quando comparado às limitações físicas do projeto de um instrumento acústico. Pensar na estrutura, no material, na forma e como adequar tudo isso a uma maneira possível de se tocar é uma atividade que demanda muita técnica e experiência.

As escolhas de projeto da maioria dos instrumentos acústicos que se conhecem se basearam mais na capacidade sonora do instrumento do que na ergonomia propriamente dita. Como os amplificadores elétricos só vieram aparecer na primeira metade do século XX, os instrumentos precisavam potencializar a própria geração sonora, deixando muitas vezes de lado a preocupação com a ergonomia ou a facilidade de tocar.

A possibilidade de desatrelar o controle da produção sonora traz consigo um poder maior de experimentação e gera novas possibilidades de expressão musical. Além disso, os DMIs podem ter interfaces de controle mais ergonômicas sem que isso interfira no seu resultado sonoro.

Enquanto alguns autores (Wessel & Wright, 2002) destacam as vantagens da liberdade de mapeamento entre os DMIs, outros consideram, como mencionado anteriormente, uma “nauseante infinidade de possibilidades” (Ryan, 1991). Este grande número de possibilidades pode se tornar um problema durante o projeto do DMI pela ausência de guias ou ferramentas que restrinjam as opções e conseqüentemente a complexidade de se iniciar o projeto.

Para ajudar a enfrentar tal complexidade, o primeiro passo é preciso pensar o projeto do DMI de forma estruturada. (Miranda & Wanderley, 2006) sugerem os seguintes passos para a construção de um DMI:

- 1) Decidir quais gestos serão usados para controlar o sistema;
- 2) Decidir as estratégias de captura desses gestos, ou seja, quais os sensores mais adequados à natureza dos gestos escolhidos para traduzi-los em valores processáveis;
- 3) Definir as formas como os sons serão gerados no sistema, seja por algoritmos de geração sonora ou softwares de manipulação de sons pré-definidos;
- 4) Definir as estratégias de mapeamento entre os valores dos sensores e as variáveis de controle do módulo de geração sonora;
- 5) Decidir sobre o uso de feedbacks não sonoros.

De forma aprofundada, (Rudraraju, 2011) propõe um maior refinamento desses passos através de uma análise hierárquica da tarefa. Esta análise se baseia na separação do DMI em três componentes básicos (Wanderley & Depalle, 2004):

- 1) Interface física: contendo sensores, atuadores e o corpo físico do instrumento;
- 2) Sistema de síntese sonora: que cria tanto a saída sonora do instrumento como lida com todos os possíveis feedbacks, seja visuais ou hápticos que o instrumento proporciona;
- 3) Sistema de mapeamento: que permite a conexão entre a interface física e as variáveis do sistema de síntese.

Para cada parte, (Rudraraju, 2011) associa um conjunto de tarefas. Abaixo, seguem a apresentação, a interpretação e a descrição dessas tarefas:

1. Projetar a interface física

- 1.1. Escolher os sensores adequados para a detecção de fenômenos físicos ou gestos;
- 1.2. Escolher os atuadores capazes de gerar feedbacks físicos (feedback primário);
- 1.3. Escolher sensores para serem ligados diretamente às entradas do módulo de geração sonora;

- 1.4. Escolher atuadores para serem ligados diretamente às saídas do módulo de geração sonora (feedback secundário);
- 1.5. Escolher o material que comporá a estrutura da interface física agrupando o corpo, os sensores e os atuadores;
- 1.6. Projetar o formato e a aparência da interface física;
2. Projetar o sistema de síntese sonora
 - 2.1. Escolher um algoritmo de síntese sonora;
 - 2.2. Escolher um mecanismo de geração de feedback visual ou háptico;
 - 2.3. Escolher os parâmetros de entrada para síntese sonora ou variáveis musicais;
 - 2.4. Escolher os parâmetros de entrada relativos à geração de feedback visual ou háptico;
 - 2.5. Escolher os parâmetros de saída que poderão ser usados pela interface física para gerar feedbacks visuais ou hápticos;
3. Projetar o mapeamento
 - 3.1. Escolher um subconjunto de possíveis saídas;
 - 3.2. Escolher um subconjunto de possíveis entradas;
 - 3.3. Definir transformação do sinal de entrada;
 - 3.4. Criar um mapeamento entre uma entrada e uma saída.

Este processo não pode ser visto como linear, e sim como um ciclo de iterações seguidas de avaliações e modificações. Além disso, dependendo do contexto do projeto, estes passos podem acontecer em diferentes ordens (McGlynn, 2011).

É bom deixar claro que esta análise hierárquica da tarefa leva em conta a construção do DMI em todos os seus estágios, porém geralmente os passos não são seguidos como um todo, pois em algumas etapas podem ser aproveitados artefatos já existentes. Por exemplo, o uso de uma interface física já existente, como um controlador MIDI (*Musical Instrument Digital Interface*) ou um tablet com superfície multitoque, faz com que a etapa “Projetar a interface física” seja simplificada pelo uso de dispositivos já existentes.

Esta análise de atividades relacionadas a um projeto de DMI serve início para se pensar a construção desses novos instrumentos, porém não resolve o problema da infinidade de possibilidades de mapeamentos, já que trata-se de uma definição geral e não absorve as idiosincrasias de contextos de diferentes DMIs e as intenções de seus designers ou luthiers.

2.4. Considerações sobre o Design de DMIs

Depois de analisar a literatura, nesta Seção, é apresentada uma série de considerações sobre o design de instrumentos musicais digitais. A palavra design é usada neste contexto para descrever o processo de concepção e desenvolvimento do DMI, ou seja, a etapa que precede a construção propriamente dita do instrumento e se relaciona com o pensamento conceitual do

projeto. Este conjunto de reflexões tem por objetivo funcionar como arcabouço de informações para se começar a pensar no projeto do DMI.

2.4.1. Impressões e Comparações entre Instrumentos Tradicionais e Digitais

Com o intuito de investigar a relação entre o músico e o instrumento (podendo ser acústico ou digital) e entender situações nas quais os instrumentistas procuram novas formas de expressão na modificação ou até mesmo na construção de novos instrumentos, (Magnusson & Mendieta, 2007) realizaram pesquisa através de um formulário online aplicado a participantes que incluíam desde instrumentistas acústicos que nunca tiveram contato com o computador a músicos experimentais, como *live-coders*², que não tocam nenhum instrumento musical convencional. Com esta gama de músicos foi possível levantar opiniões diversas que dão um panorama geral das impressões e comparações que os músicos fazem entre instrumentos acústicos e digitais. Alguns desses resultados serão apresentados a seguir e podem auxiliar na tomada de decisões em projetos de DMIs.

O primeiro tema que iremos analisar dos resultados do questionário é a tradição. Enquanto alguns participantes julgam a falta de convenções sociais em relação aos novos instrumentos digitais um empecilho para o seu uso, outros citam que esta liberdade de tradição e legado proporciona um vasto terreno para a criatividade e a originalidade. Como ponto negativo dos instrumentos acústicos, foi dito que é muito difícil se fugir do *cliché* com estes instrumentos, pois a maioria das formas de tocar e de gerar novos sons já foi exaurida.

Muitos falaram da característica experimental e exploratória dos DMIs, possibilitando a geração de “qualquer som e qualquer interface”, como uma vantagem que permite a descoberta de novas formas de expressão musical. Porém, foi levantado que a maioria dos projetos de instrumentos musicais ainda é “escrava do histórico”, isto é, na sua maioria, os DMIs ainda continuam buscando apenas simular ou reproduzir as funcionalidades dos instrumentos acústicos.

Pode ser vista uma polaridade entre acústico e digital no que diz respeito à profundidade do músico em relação ao instrumento: de um lado, há um instrumento tradicional e legado, de propósito não restrito, que dá sensação de continuidade ao músico que queira se aprofundar no desenvolvimento de sua técnica; do outro, há um sistema construído para uma ocasião específica, que pode não se adequar a outras situações e desenvolvido em uma plataforma tecnológica que não dá garantia de continuidade, podendo entrar em desuso e levar consigo o

² Artistas que baseiam sua performance na escrita de códigos em linguagem de programação que geram sons e imagens em tempo real.

instrumento, transformando-o em “artefato arqueológico”. Merece citação um trecho interessante dos resultados de (Magnusson & Mendieta, 2007): “o instrumento digital pode ser criado para necessidades específicas, enquanto o instrumentista tem de se moldar ao instrumento acústico”.

Comentando alguns aspectos técnicos, um dos pontos positivos mencionados sobre o instrumento acústico é o seu feedback háptico, que proporciona ao músico a possibilidade de usar sua memória motora no ato de tocar. Já sobre o DMI, esta ausência de feedback é vista como um ponto negativo, que cria dificuldades em seu uso. Outra técnica comentada como ponto negativo é a latência que pode existir em instrumentos digitais e é inexistente em instrumentos acústicos.

Porém, as limitações de tonalidade e microtonalidades dos instrumentos acústicos são vistas negativamente por alguns dos participantes do questionário, dando vantagem aos instrumentos digitais que permitem que tais detalhes sejam implementados. Além disso, foi citada a capacidade de incluir memória, inteligência e automação nos DMIs, permitindo a geração de bons resultados musicais e podendo torná-los boas ferramentas a serem usadas por leigos ou iniciantes no mundo da música.

Citando algumas opiniões mais subjetivas existem alguns pontos negativos dos DMIs como a sua “falta de substância” e de proporcionar uma “experiência desencarnada”. Do outro lado, há os instrumentos acústicos, que foram mencionados como sendo um artefato com o qual é mais fácil expressar sentimentos e que de fato possuem limitações, porém são desafiadoras e inspiradoras para o caminho do instrumentista à virtuosidade.

De forma geral, verifica-se que há uma polarização entre acústico e digital, que provavelmente nunca deixará de existir. Os instrumentos acústicos fazem parte da nossa cultura e sempre existiram como base da música. Porém, a investigação de novos instrumentos digitais é de extrema importância para a descoberta de novas formas de expressão musical, possibilitando a transformação da música atual e até o aparecimento de novos gêneros e formas de se comunicar musicalmente.

2.4.2. Fáceis de aprender, possíveis de se aprofundar

Em (Wessel & Wright, 2002), é argumentado que a facilidade em usar um sistema musical não deve sobrepor o seu potencial expressivo. Os instrumentos tradicionais não são simples no começo do aprendizado, mas proporcionam um grande potencial musical. Como citado anteriormente, o projeto de instrumentos acústicos, na sua evolução, priorizou o poder sonoro do instrumento em revelar a forma como o instrumento é tocado. Com os instrumentos digitais, encontra-se o ponto de equilíbrio entre uma interface intuitiva que permite um início de aprendizado suave e um potencial musical evolutivo entre o músico e o instrumento. (Sergi Jordà, 2004a) dá o exemplo do piano como instrumento que oferece “baixa taxa de entrada sem limite para a virtuosidade”.

A área de IHC (Interação Humano-Computador) preza por criar sistemas cada vez mais fáceis e simples de usar. No contexto musical, esta aplicação pode não ser verdadeira para todos os casos. O esforço está intimamente relacionado com a expressão no ato de tocar instrumentos tradicionais (Ryan, 1991). Não apenas o esforço físico para tocar, mas também o esforço de aprender a tocar o instrumento.

2.4.3. Design de DMIs e Design de Games

Ligando-se diretamente com o tema anterior, salienta-se a relação entre desafio, frustração e tédio. E é neste ponto que há uma convergência entre o design de DMIs e o design de games (Overholt, 2009): se o jogo é muito fácil e não oferece desafio, logo o jogador fica entediado e sem incentivo para continuar a jogar; se o jogo é extremamente difícil, o jogador se sente frustrado e da mesma forma perde o interesse em prosseguir jogando (Koster, 2005). É preciso que haja um equilíbrio para tornar o jogador engajado. Esta noção de engajado está diretamente ligada à definição de Estado de Fluxo, descrita por (Csikszentmihalyi & Csikszentmihaly, 2002).

Se for feito o paralelo com o design de DMIs, verifica-se que o mesmo equilíbrio é de grande importância para o engajamento do instrumentista: “se o músico não achar um instrumento ou mapeamento suficientemente gratificante, isto não é conducente para que ele atinja um nível de controle profundo” (Overholt, 2009). Porém, o projeto de DMI se torna mais complexo por não existir, da mesma forma que no jogo, uma sensação de tarefa pontual a ser cumprida (como terminar um determinado nível ou alcançar determinada pontuação). Para instrumentos musicais digitais, é importante que se dê ao músico a sensação de estar jogando um jogo infinito, aquele em relação ao qual a diversão é continuar jogando (Carse, 1986). Se um DMI possuir um número limitado de possibilidades, o músico logo perceberá que o “jogo” não tem mais nada a oferecer e assim desistirá do instrumento.

2.4.4. Metáforas de Interação e Transparência de Mapeamentos

(Fels, Gadd, & Mulder, 2002) argumentam que, em função de toda uma carga cultural já existente envolvendo os instrumentos acústicos, existe uma facilidade de entender a relação do gesto com o som que está sendo gerado. Os autores defendem que o entendimento desta relação torna a performance mais expressiva se comparada com mapeamentos opacos ou difíceis de entender. (Wessel & Wright, 2002) recomendam o uso de metáforas para organizar e guiar como será a interação com o DMI. Os autores afirmam que o uso de metáforas espaciais de movimentos conhecidos como “pegar e arremessar”, “arrastar e soltar”, “mergulhar”, “esfregar” etc., é uma forma de trazer conhecimentos intuitivos que se tem do mundo para o design de instrumentos. Porém, destacam que a conexão precisa ser pensada com cuidado para que tenha coerência e faça sentido.

2.4.5. Coerência Gestual

Para satisfazer tanto o performer quanto a plateia, (Wessel, Wright, & Schott, 2002) defendem que haja uma correspondência entre o “tamanho” do gesto e o resultado sonoro. (Goldstein, 1998) define esta relação como coerência gestual. Pode-se entender da seguinte forma: quando o controle gestual é sutil, o resultado sonoro deve acompanhar este estímulo sendo, da mesma forma, leve e delicado. Já para movimentos enérgicos e fortes, o resultado sonoro deve sofrer mudanças drásticas.

2.4.6. Exploração de Novos Paradigmas de Interação

Alguns dos primeiros instrumentos eletrônicos já usavam as teclas de piano como interface: *Musical Telegraph* (1876), *Telharmonium* (1897), *Singing Arc* (1899). Na década de 1960, o teclado foi usado no primeiro sintetizador comercial da *Moog*. Comercialmente falando, os DMIs mais populares se baseiam no paradigma do teclado de piano. Um novo instrumento que se distancia deste padrão de teclado é o *Theremin* (1919), que possui duas antenas cujo campo elétrico é alterado com a proximidade das mãos, influenciando a frequência e amplitude do som produzido (Sinyor, 2006). Quanto ao uso do computador em performances musicais, constata-se uma “sintomática e frustrante” utilização de mouses e teclados MIDI (Sergi Jordà, 2004a).

Com a popularização de novas interfaces, surge a possibilidade de reflexão sobre novas formas de controle musical. Por estas interfaces inovadoras ainda não terem paradigmas de interação consolidados, o que acontece na maioria das vezes é que os designers absorvem um paradigma antigo e o utilizam sem ao menos adaptá-lo para as idiosincrasias das novas interfaces (McGlynn et al., 2012). Como exemplo, há os *tablets* com superfície multitoque, que apesar do vasto potencial de controle através de gestos inerentes à sua interface (Saffer, 2008), vários aplicativos musicais continuam a simular pianos (ou outros instrumentos tradicionais) ou utilizam o paradigma WIMP (*Windows, Icons, Menus and Pointers*) como principal forma de interação.

2.4.7. Requisitos de Latência

Latência é a duração entre um estímulo e a resposta a este estímulo. A latência pode ser estática, ou regular, quando o atraso permanece constante durante o tempo ou pode ser variável, recebendo o nome de *jitter*.

Considerando que a atividade musical é crítica em relação à precisão no tempo, a baixa latência é um requisito bastante importante no projeto de um instrumento musical. Instrumentos acústicos possuem pouca ou nenhuma latência (apesar de que em alguns casos há certo atraso em função da necessidade de acúmulo de energia para que haja o evento sonoro, como é o caso da tecla do piano, ou o sopro em uma tuba) (Magnusson & Mendieta, 2007).

Porém, uma situação mais complicada é a do instrumento digital, cujo sistema precisa realizar cálculos e processamentos para converter o movimento do usuário em dados, fazer a ligação dos dados de entrada com parâmetros de saída e, finalmente, sintetizar o som. Todos esses processos agregam atrasos, que podem prejudicar o uso do instrumento.

Alguns níveis de controle musical têm um pouco mais de tolerância a atrasos (Malloch et al., 2006) que outros. Geralmente, quando o controle musical é relacionado ao disparo de um evento, o atraso é crítico. Entretanto, controles mais voltados à manipulação contínua de propriedades do som ou processos pré-definidos têm um pouco mais de tolerância a atrasos.

(Sergi Jordà, 2005, pp. 220–222) faz uma intensa revisão da literatura sobre a percepção da latência no uso de instrumentos musicais. O autor afirma que o ouvido humano é capaz de perceber a diferença entre dois cliques disparados com diferença de tempo igual a 2 μ sec. Porém, quando o evento sonoro se torna mais complexo ou diferente, a tolerância auditiva aumenta drasticamente para 20 ms. Ou seja, dois eventos sonoros disparados com 20 ms de atraso são interpretados pelo nosso sistema auditivo como eventos simultâneos. O autor conclui que há uma margem de tolerância à latência entre 20 e 30 ms. Desta forma, há um requisito funcional direto para se levar em conta no projeto de um DMI: para controles musicais cujo atraso é crítico, precisa-se limitar a latência abaixo de 30 ms.

2.4.8. Esforço Cognitivo e Complexidade vs. Expressividade

É importante levar em consideração que, apesar da possibilidade de utilização de inúmeros sensores e entradas em um DMI, é preciso ter cuidado para que o emaranhado de dados não deixe o sistema caótico e difícil de gerenciar. Existem maneiras de diminuir o esforço cognitivo no projeto de DMIs, como o uso de camadas semânticas que abstraem a complexidade (ver seção 2.1.3), mas é sempre preciso manter a integridade dos dados e conexões (Malloch, 2008).

Porém, projetos de DMI não devem ser guiados pelo uso excessivo de entradas e sensores. (Wanderley, 2006) afirma que não existem provas de que uma interface mais complexa pode ser considerada uma interface mais expressiva. A expressividade emerge da transparência e do entendimento do instrumento por parte do músico (Fels et al., 2002).

2.4.9. Diferentes Diversidades para DMIs

No intuito de melhor entender a relação instrumento-performer, (Sergi Jordà, 2004b) apresenta uma classificação que leva em conta o quão diverso pode ser um instrumento no contexto de performances musicais. O autor divide as diversidades em três e associa cada uma ao impacto que tem nos performers:

- Micro-diversidade (*MicD*): é o mais detalhado nível de diversidade e se relaciona com as nuances de uma performance, ou seja, o quão diferente pode ser a performance de uma mesma obra, ou peça musical, tocada mais de uma vez pelo mesmo instrumento. A presença deste tipo de diversidade em um instrumento pode transformar um músico intérprete em um músico virtuoso;
- Diversidade média (*MidD*): se relaciona com o contraste entre performances, ou seja, o quão distinto é o resultado musical da interpretação de duas peças diferentes usando o mesmo instrumento. Esta diversidade pode transformar um iniciante ou curioso em um intérprete;
- Macro-diversidade (*MacD*): representa a flexibilidade estilística do instrumento, ou seja, como o instrumento se adapta a diferentes contextos musicais. Instrumentos com pouca *MacD* têm mais chances de dar origem a novos gêneros musicais e instrumentos com alta *MacD* têm mais chances de começarem a ter seu uso consolidado e se tornar populares.

Da reflexão sobre estas diversidades, o autor constrói alguns corolários que podem auxiliar a tomada de decisão em projetos de DMI. Abaixo, esses corolários foram traduzidos e reproduzidos de forma semelhante à proposta do autor:

- Um instrumento precisa ser capaz de tocar diferentes peças sem a necessidade de ser reprogramado; performers não são necessariamente programadores;
- Alguns instrumentos podem ter um alto *MidD* “natural”; outros talvez precisem trabalhar em diferentes configurações ou *setups*;
- Se diferentes peças precisam de diferentes configurações, todas essas configurações não devem contradizer as regras básicas de *playability* do dado instrumento. Caso contrário, não se está mais lidando com um instrumento e sim com um sistema variável de controlador+gerador;
- Se configurações são uma necessidade, a possibilidade de carregá-las *on-the-fly* vai gerar improvisação.

2.5. Avaliação de DMIs

Como parte do processo de concepção e construção de DMIs, deve-se pensar em como eles podem ser avaliados. Assim, durante o desenvolvimento, pode-se comparar versões de um mesmo instrumento para saber se suas funcionalidades se adequam ao objetivo do projeto proposto. Além disso, tais métodos de avaliação permitem comparações entre instrumentos e um entendimento mais profundo sobre os quais melhor se adequam a determinados contextos.

Em (O’Modhrain, 2011), são apresentadas algumas visões que devem ser levadas em conta durante a avaliação de um DMI. São elas: a visão do performer, ou instrumentista; a visão da plateia; a visão do designer; e a visão do fabricante.

Em (Stowell, Robertson, Bryan-Kinns, & Plumbley, 2009) verifica-se que nem todos os projetos de DMI apresentam uma forma de avaliação estruturada, o que faz com que o conhecimento que se tem acerca do projeto fique restrito aos seus pesquisadores e designer. Para tanto, os autores apresentam um método estruturado para avaliação da visão do performer em relação ao DMI, baseando-se em avaliações qualitativas usando análise do discurso. (Barbosa et al., 2011) incorporam tal método e associam-no à forma de visualização dos dados de um instrumento proposta por (Birnbaum et al., 2005), que facilita a comparação entre versões de um mesmo instrumento durante o desenvolvimento do projeto e entre diferentes instrumentos.

Levando em consideração a visão da plateia, que quanto melhor entende a relação de causa e efeito de um DMI, melhor aproveita o espetáculo, (Barbosa, Calegario, Teichrieb, & Ramalho, 2012) apresentam um método estruturado baseado em avaliação quantitativa para medir o entendimento de espectadores quanto ao funcionamento do instrumento através da distribuição online do vídeo da performance e a aplicação de um questionário com perguntas relacionadas.

3. Diretrizes

A fim de resolver o problema do mapeamento de DMIs, a abordagem apresentada neste Projeto diz respeito a dar poder ao usuário, que, através da experimentação, poderá ser capaz de encontrar o mapeamento adequado para seus contexto e intenção. Para isso, propomos um ambiente de experimentação, onde o usuário pode conceber, construir e experimentar DMIs.

Seguindo esta abordagem, um conjunto de diretrizes é necessário para ajudar o desenvolvimento do ambiente de experimentação. As seguintes diretrizes foram compiladas e serão explicadas com detalhes abaixo: usabilidade, feedback, diversidade, configuração, integração e disponibilidade.

É importante salientar que, embora estas diretrizes estejam relacionadas ao ambiente de experimentação de DMI, há uma relação clara entre o ambiente e o DMI criado. Isso acontece principalmente porque, como parte do processo de experimentação, o ambiente deve proporcionar meios para executar o DMI. Isto significa que, durante a execução, o DMI torna-se um subconjunto do ambiente. Conseqüentemente, algumas diretrizes do ambiente podem ser também aplicáveis aos DMIs construídos, mas neste trabalho vamos nos concentrar apenas no desenvolvimento do ambiente.

3.1. Usabilidade

Usabilidade nos permite criar sistemas "mais intuitivos e eficazes para uma pessoa que está tentando realizar uma determinada tarefa" (Isbister & Schaffer, 2008). Está relacionada com o quão fácil e satisfatória é uma interface no que diz respeito à aprendizagem, uso e memorização (Nielsen, 1993).

Além do domínio tradicional da Interação Humano-Computador, o conceito de usabilidade tem sido utilizado em contextos peculiares, como jogos (Isbister & Schaffer, 2008), páginas da web (Nielsen, 1993), e outros - incluindo os próprios DMIs (Orio, Schnell, & Wanderley, 2001).

Assim, se o intuito é que os usuários experimentem rapidamente diferentes combinações com o objetivo de encontrar o mais adequado para as suas necessidades, acreditamos que a usabilidade deve ser considerada como uma diretriz.

3.2. Feedback

Entre os aspectos que compõem a usabilidade, um deles parece ser bastante sensível no contexto musical: o feedback. Este ponto está relacionado ao fornecimento de uma resposta clara, perceptível e em tempo real para as ações e modificações realizadas pelos usuários e pode ser de diferentes formas: háptica (tátil e cinética), sonora e visual (luzes, imagens etc.)

Como demonstrado em trabalhos anteriores, o feedback pode aumentar a percepção do usuário que esteja interagindo com um sistema musical interativo (S Jordà, 2003) e também pode melhorar a precisão na hora de tocar um DMI (O'Modhrain & Chafe, 2000).

Consequentemente, pela sua natureza crítica no contexto musical, acreditamos que o feedback deve ser considerado como uma diretriz independente e autônoma para o ambiente aqui desenvolvido.

3.3. Diversidade

De acordo com a sua natureza (Miranda & Wanderley, 2006), todo DMI é composto de uma combinação de módulos de entrada, que permitem aos usuários interagirem com o instrumento, e os módulos de saída, que são responsáveis pelo resultado sonoro. Definindo a maneira como esses módulos são combinados, existe um conjunto de estratégias de mapeamento.

A diretriz “diversidade” lida com o espaço que o usuário terá para explorar essas possíveis combinações.

A experimentação pressupõe exploração e combinação e é recomendado se ter um número razoável de opções para tornar isto possível (Thomke, 2003). No entanto, no contexto deste projeto, a diversidade deve ser equilibrada para permitir a experimentação, porém não prejudicando a compreensão do usuário.

3.4. Configuração

Como o objetivo é criar um ambiente para a criação de DMIS, é importante assumir que não se tem a priori nenhum conhecimento sobre o contexto ou intenção específica de cada usuário. Por esse motivo, o usuário deve ter a possibilidade de personalizar suas próprias configurações, a fim de adequar o sistema as suas necessidades individuais.

3.5. Integração

Como já observado em trabalhos anteriores (Chagas, 1992; Schmeder & Freed, 2008), protocolos de comunicação especialmente concebidos para o contexto musical (como MIDI e OSC) são ferramentas muito poderosas e muitas vezes usadas projetos de DMIs.

Uma explicação comum para esta popularização vem de sua simplicidade (facilidade de uso) e "a promessa de interoperabilidade com um diversificado leque de aplicações", o que resulta em uma grande versatilidade e capacidade de integração (Schmeder & Freed, 2008).

Assim, o objetivo é permitir que o ambiente seja facilmente integrado com os sistemas já existentes, beneficiando de suas funcionalidades, a integração deve ser considerada como uma das diretrizes de desenvolvimento do ambiente.

3.6. Disponibilidade

De acordo com (Sergi Jordà, 2005), um problema muito comum em DMIs é a falta de disponibilidade de tais sistemas. Isto é, embora podendo ser intuitivo de se usar durante a performance, para a maioria dos DMIs é necessária a presença de um técnico para configurar o sistema (incluindo a instalação e calibração) antes que ele fosse usado pelo músico ou artista - um problema básico que pode tornar impossível a popularização de um DMI.

Como o ambiente aqui apresentado é destinado a usuários não técnicos, o sistema deve estar disponível de forma fácil, tanto do ponto de vista de aquisição, quanto do ponto de vista de instalação.

4. Estado da Arte

Nesta Seção, estão selecionados e apresentados alguns projetos relacionados, dividindo-os em dois grupos (ver Figura 4.1): Sistemas Modulares para Música e Sistemas de Mapeamento de Dados Musicais.

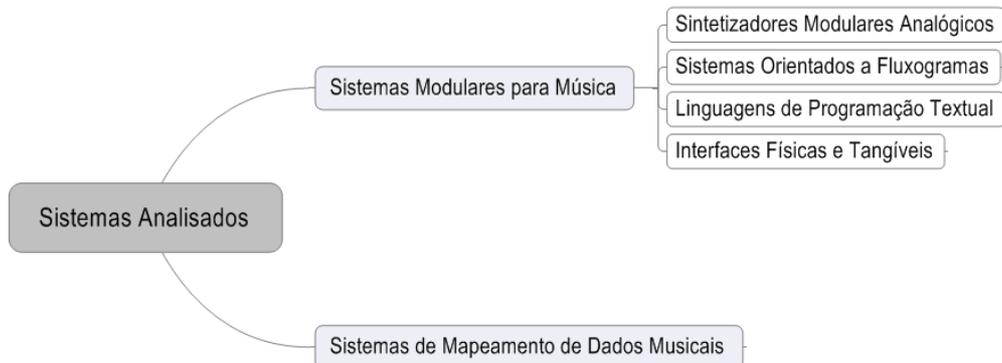


Figura 4.1: Classificação de sistemas musicais que permitem a experimentação de novos instrumentos

Algumas ferramentas, apresentadas abaixo, apesar de não terem sido originalmente criadas para servir como um ambiente de prototipação para DMIs, podem ser assim usadas. A maioria apresenta bons feedbacks, diversidade, possibilidades de customização e integração, porém, no que diz respeito à usabilidade e à disponibilidade, apresentam alguns problemas para o público de usuários não técnicos.

4.1. Sistemas Modulares para Música

Será utilizada a definição de Sistemas Modulares para Música (SMM) a fim de descrever projetos cujo funcionamento baseia-se na manipulação e interconexão de módulos para geração de um resultado musical. Os módulos podem ser entendidos como entidades que possuem uma funcionalidade específica, além de parâmetros de configuração, entradas e saídas. Através de conexões entre essas entradas e saídas, fluxos de dados são repassados entre um módulo e outro, ampliando as possibilidades de resultado através das combinações.

4.1.1. Sintetizadores Modulares Analógicos

Os primeiros SMMs remontam à época dos sintetizadores modulares analógicos, que eram instrumentos musicais eletrônicos cujas propriedades sonoras podiam ser modificadas pela interconexão entre módulos de processamento sonoro e alterações de seus parâmetros. Cada módulo tinha sua saída ligada fisicamente a entrada de outro através de fios (chamados *patch-cords*). O fluxo do sinal de áudio era transmitido por esses cabos e nos módulos era processado até finalmente chegar a uma caixa de som. Algumas referências são os sintetizadores controlados por voltagem de Robert Moog ou de Donald Buchla (Chadabe, 1975)

(ver Figura 4.2). Apesar de necessitar de um alto nível de aprofundamento para usar tais sistemas e por não estarem tão facilmente disponíveis para um público não técnico, através da experimentação sonora da combinação entre os módulos destes sintetizadores, surgiram novos sons e novos gêneros musicais (Sergi Jordà, 2005).

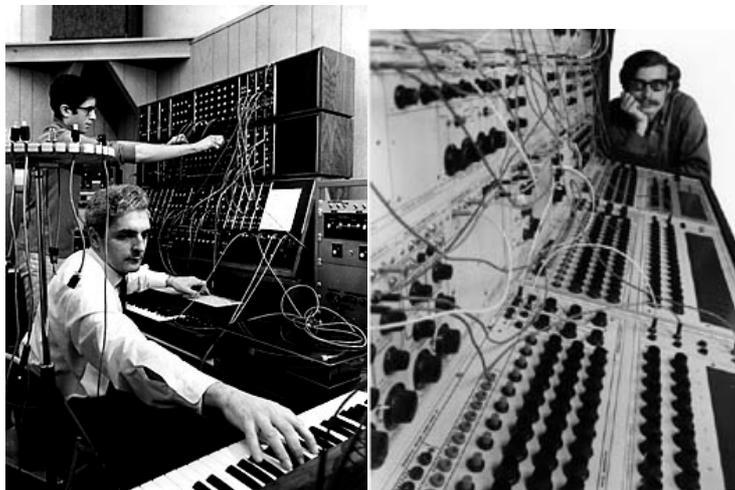


Figura 4.2: À esquerda, Robert Moog tocando o sintetizador modular Moog 55 em meados dos anos 60. À direita, Donald Buchla posando junto ao sintetizador modular Buchla 100 (1963).

4.1.2. Linguagens de Programação Textual

Nesta Seção, são apresentadas algumas linguagens de programação voltadas para a música, que carregam o conceito modular em suas implementações. Embora não possuam interfaces gráficas, as conexões entre os módulos são feitas através da codificação textual das ligações de entrada, processamento e saída.

A linguagem CSound (Boullanger, 2000), criada em 1986 por Barry Vercoe, é uma linguagem implementada em C voltada para síntese sonora, performance em tempo real e composição algorítmica.

A SuperCollider (McCartney, 2002) também permite síntese em tempo real e além de poder ser usada em composição algorítmica também é usada em *live coding*. É uma linguagem mais abrangente sendo também considerada uma linguagem de propósito geral.

A mais nova das três linguagens aqui apresentadas é a ChuckK (Wang, 2008). Além de permitir síntese em tempo real, de ser usada em *live coding* e composição algorítmica, a linguagem ChuckK foi desenvolvida de forma a potencializar seu uso pedagógico, apresentando uma forma fácil de ser escrita e lida. Foi projetada como um novo modelo de programação concorrente crítica ao tempo. Ultimamente, vem sendo bastante usada como ferramenta de rápida experimentação em computação musical.

Tais linguagens, por não possuírem interfaces gráficas, apresentam deficiências no que diz respeito ao feedback e conseqüentemente a usabilidade, mesmo sendo pedagogicamente projetadas, tornando-se mais difícil de serem usadas por um público não técnico.

4.1.3. Sistemas Orientados a Fluxograma

Como um subconjunto de sistemas modulares, existem os Sistemas Orientados a Fluxogramas (SOF), que é uma definição apresentada por (Jácome, 2007) para designar sistemas com interfaces gráficas que permitem manipulação e a conexão visual entre os seus módulos audiovisuais. No contexto do citado trabalho, esta definição abrange tanto sistemas relacionados a áudio quanto a manipulação de vídeo.

“Em geral, os SOFs podem ser vistos como linguagens de programação no sentido de que o usuário pode construir aplicações (programas de computador), definindo o fluxo de dados e de execução através da construção de fluxogramas em um ambiente gráfico utilizando uma linguagem visual.” (Jácome, 2007).

Um dos principais exemplos de SOFs é o software desenvolvido por Miller Puckette: Max (Puckette, 2002). Inspirado no trabalho RTSKED de Max Matthews (por causa disso, o nome), o software apresentava uma forma de combinar dinamicamente blocos pré-programados através da ligação entre eles, usando um computador para a geração interativa de saída sonora em tempo real. O projeto inicial deu origem a outros projetos como MAX/MSP, jMax e Pure Data, que fizeram com que, em vez de denominar apenas o programa, o nome Max começasse a ser associado a um paradigma de programação visual para a música. Na Figura 4.3, vê-se a interface do MAX/MSP e do Pure Data e percebe-se a semelhança do paradigma de programação.

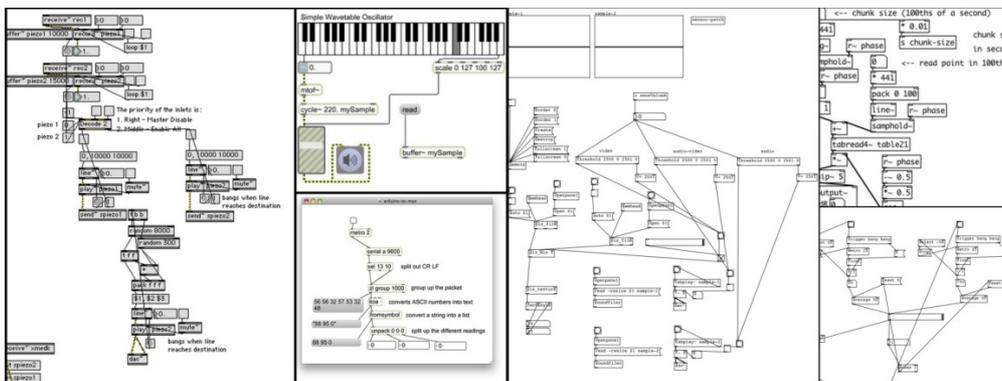


Figura 4.3: MAX/MSP e Pure Data

Citando outros SOFs, temos: EyesWeb, Isadora, vvvv, Quartz Composer, Visual Jockey, GePhex, Salvation, Reaktor, CPS, AudioMulch e Bidule. Referências mais detalhadas sobre esses sistemas podem ser encontradas em (Jácome, 2007).

Os SOFs podem ser considerados uma evolução das linguagens de programação, dando feedback visual e imediato às ações do usuário. Apesar deste ponto positivo, alguns SOFs podem ser bastante complexos, fazendo com que o usuário precise se aprofundar para gerar resultados não triviais.

4.1.4. Interfaces Físicas e Tangíveis

Este subconjunto dos sistemas modulares para a música leva em consideração projetos que permitem uma interação física e tangível com os módulos musicais, realizando as ligações através da manipulação espacial dos objetos. Neste sentido, são apresentados o Reactable (Figura 4.4) e o Molecule Synth (Figura 4.5), cujos principais pontos fracos são pouca diversidade e difícil disponibilidade.

Reactable é um instrumento colaborativo baseado numa superfície multitoque onde os usuários podem manipular objetos tangíveis que representam módulos musicais interconectáveis, como: oscilador, tocador de *loops*, *sampler*, entrada de microfone, filtro, *delay*, modulador, *wave shaper*, *Low Frequency Oscillator* (LFO), sequenciador, controle de parâmetros da música, da saída sonora e da tonalidade (Sergi Jordà, Geiger, Alonso, & Kaltenbrunner, 2007; Sergi Jordà, Kaltenbrunner, Geiger, & Bencina, 2005).



Figura 4.4: Reactable

Molecule Synth é um projeto que foi financiado em setembro de 2012, através do *Kickstarter*, e desenvolvido por Travis Feldman. Trata-se de um sintetizador modular e tangível, que tem como módulos de entrada: sensores de deformação, de luz, de pressão; como módulos de geração sonora: temporizadores, amplificadores operacionais e LFOs; e alto-falantes e *mixers* como módulos de áudio. Através do encaixe entre os objetos, o usuário pode experimentar quais as melhores formas de fazer seu próprio instrumento.

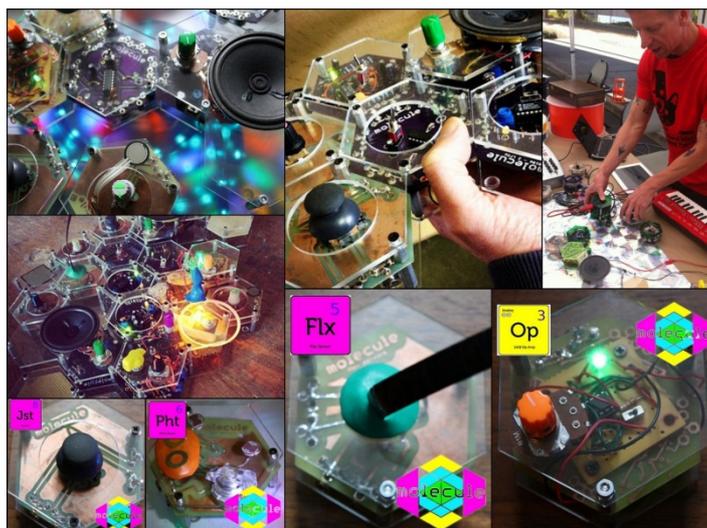


Figura 4.5: Molecule Synth

4.2. Sistemas de Mapeamento de Dados

Além dos Sistemas Modulares para Música, apresentados na Seção anterior, foram selecionados projetos diretamente relacionados com o mapeamento de controles em saídas musicais. Definem-se Sistemas de Mapeamento de Dados como sendo interfaces gráficas cuja principal função é fazer com que a tarefa de escolher mapeamentos seja realizada da forma mais eficiente possível. Para tanto, buscam formas visuais e intuitivas de apresentar os dados e as conexões entre eles.

Por tornarem o mapeamento rápido e eficiente, tais softwares têm um grande potencial para permitir a experimentação e a construção de protótipos e versões preliminares de DMIs, porém podem ser consideradas ferramentas intermediárias, pois sempre necessitam de sistemas adicionais que sejam responsáveis pelo controle de entrada e geração sonora. Além disso, alguns dos sistemas apresentados abaixo, apresentam problemas de usabilidade e feedback.



Figura 4.6: Ableton Live

Apesar de o Ableton Live (Figura 4.6) ser um software musical voltado para o disparo de loops, processamento de áudio e sequenciamento de padrões musicais, apresenta-se o exemplo da sua interface de mapeamento. Para realizar o mapeamento no Live, realizam-se os seguintes

passos: ativa-se o modo de mapeamento e uma camada transparente é sobreposta aos controles da interface gráfica; seleciona-se um controle do programa e envia-se um comando MIDI proveniente de algum controlador, software ou dispositivo conectado ao computador; desta forma, tal comando é associado ao controle selecionado, que responde deste momento em diante às alterações de valor do comando MIDI associado. Este esquema é prático, pois associa a ação no controlador físico diretamente à funcionalidade presente no programa.

Verifica-se que o Live apresenta um sistema embutido de mapeamento, dentro do próprio ambiente do software. Os projetos descritos a seguir funcionam de maneira autônoma como este sub-sistema do Live, porém com maior poder de integração com outros sistemas já que, na maioria das vezes, implementam protocolos de comunicação variados como o MIDI e o OSC (*Open Sound Control*).

4.2.1. OSCulator

O OSCulator (Figura 4.7) é um software desenvolvido por Camille Troillard para o sistema operacional Mac OS, que trabalha mapeando comandos de entrada para softwares musicais através de MIDI ou OSC. Apresenta uma interface tabular, na qual no lado esquerdo são apresentados os endereços das mensagens OSC advindas de uma determinada porta de rede do computador e do lado direito são mostradas as saídas associadas.

Nativamente, pode-se conectar um controle de Nintendo Wii ao aplicativo que já traduz os dados do dispositivo em mensagens facilmente associadas a saídas MIDI ou OSC. Com isso, de forma trivial, tem-se a possibilidade de experimentação com uma nova interface gestual.

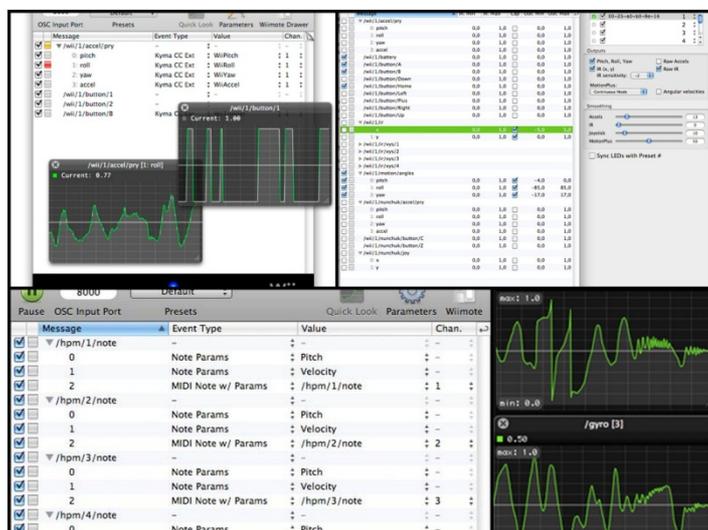


Figura 4.7: OSCulator

Uma funcionalidade interessante do OSCulator é a transcrição automática das mensagens que chegam para uma determinada porta de rede do computador. É uma verdadeira facilidade para se lidar com OSC, já que muitas vezes não se sabem quais mensagens fazem parte do protocolo de comunicação de determinado aplicativo.

Uma crítica ao OSCulator é a dificuldade de aplicação de operações aos valores de entrada; apesar de existir tal possibilidade, a forma como essas operações são manipuladas é bastante confusa. Além de uma crítica generalizada à interface tabular que enrijece o mapeamento e torna difícil a conexão de um-para-muitos e de muitos-para-um.

Como principais pontos positivos, destacam-se os feedbacks visuais presentes na interface, como o indicador de recebimento de mensagem, um pequeno item de interface que muda de cor quando chega uma nova mensagem e os gráficos de visualização rápida, que mostram os valores de entrada pelo tempo plotados em um gráfico, como pode-se ver também na Figura 4.7.

4.2.2. junXion v5

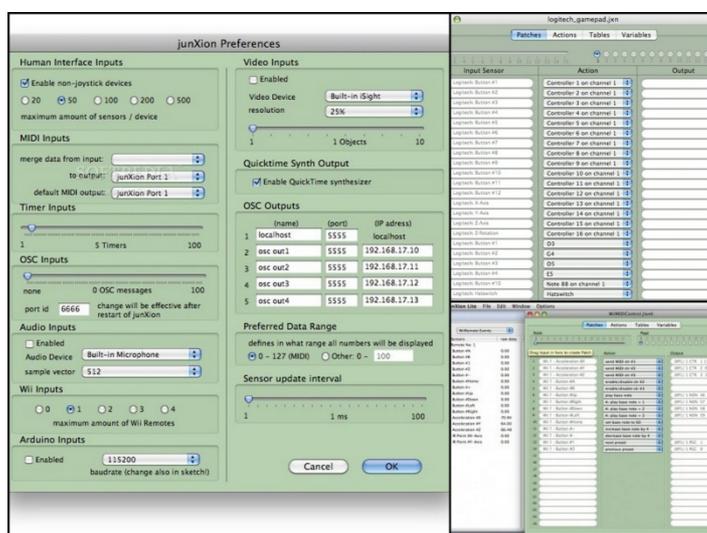


Figura 4.8: junXion v5

O junXion (Figura 4.8), desenvolvido pela STEIM, é um aplicativo para Mac OS X que funciona como um roteador de dados de entrada de sensores diversos (por exemplo: joystick, mouse, WiiMote, rastreamento de cor através da imagem de webcams, comunicação com Arduino etc.) a saídas MIDI ou OSC. Tem o conceito de Patch, que é o nome dado à conexão entre dados do sensor, ação aplicada a este dado e a saída em MIDI ou OSC. As ações podem ser manipulações numéricas, como mapeamentos e escalonamentos, e processamentos condicionais (por exemplo: se determinado valor passar de um limiar, dispara um evento MIDI).

Apesar de ter uma grande diversidade de sensores, sua interface, basicamente formada por objetos de formulários, traz pouca preocupação com o mapeamento visual, o que a torna um tanto complicada em termos de usabilidade. A modificação das conexões não é tão ágil, o que prejudica a experimentação. Além disso, esta estrutura tabular de dados, como dito anteriormente, é limitante no que diz respeito às conexões de um-para-muitos ou de muitos-para-um.

4.2.3. Libmapper

No intuito de criar um ambiente capaz de implementar estratégias de mapeamento de alto nível e possibilitar o teste rápido de ideias para DMIs, o *Input Devices and Music Interaction Laboratory* (IDMIL) da Universidade de McGill no Canadá, desenvolveu o *Digital Orchestra Toolbox* para ser usado na McGill Digital Orchestra, uma iniciativa de desenvolver uma orquestra contendo vários instrumentos musicais digitais com a participação ativa de músicos, compositores, engenheiros e pesquisadores em performance, composição e tecnologia musical (Malloch et al., 2008). Este sistema possui componentes que representam sub-rotinas comuns usadas em mapeamentos e permite que dispositivos de entrada (controladores gesturais, sensores etc.) e de saída (sintetizadores, softwares musicais etc.) de diferentes naturezas possam ter seus dados mapeados através da rede. A ideia é facilitar a comunicação entre pessoas técnicas e não técnicas na construção colaborativa de novos instrumentos musicais digitais para serem usados em performances.

Mais recentemente, o *Digital Orchestra Toolbox* foi reformulado e reimplementado como uma biblioteca da linguagem de programação C que recebeu o nome de *libmapper*. Como uma biblioteca, o *libmapper* permite que qualquer programa ou dispositivo que o implemente consiga interagir facilmente com outros programas e dispositivos que tenham o *libmapper* implementado, permitindo a comunicação através do protocolo *Mapper* de comunicação. Desta forma, “*libmapper* pode ser considerado como uma implementação de um protocolo de comunicação que foi criado para especificar mapeamentos através de uma rede distribuída” (Rudraraju, 2011).

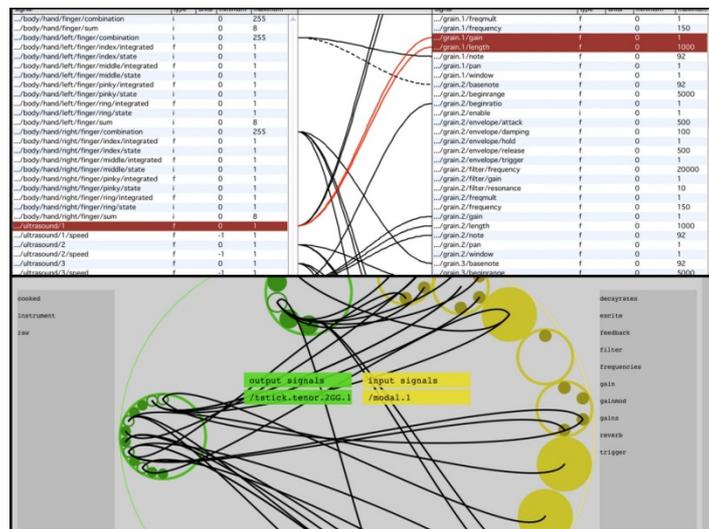


Figura 4.9: Interface Gráfica para o libmapper

Uma vantagem associada ao *libmapper* é que o “mapeamento entre dispositivos de controle e geração de som pode ser modificado sem que seja necessário compilar qualquer código, reiniciar qualquer sistema ou recarregar qualquer script” (Rudraraju, 2011). Além disso, o mapeamento é descentralizado, permitindo que qualquer membro da rede possa fazer

qualquer modificação, bastando que se inscreva na rede como um *monitor*. “Uma rede com dispositivos e monitores habilitados com *libmapper* é uma rede *Mapper*” (Rudraraju, 2011). Esta conectividade e descentralização fazem com que a colaboração seja mais ágil com resultados imediatos.

Na Figura 4.9, são vistas duas iniciativas de criação de interfaces gráficas para manipulação de mapeamentos disponíveis através do *libmapper*. Na parte superior, tem-se Maxmapper, que recebe esse nome por utilizar o ambiente de desenvolvimento do Max/MSP. Na parte inferior, há o Vizmapper, uma interface alternativa desenvolvida por (Rudraraju, 2011) que se baseia em conceitos de visualização de dados para permitir uma representação de mapeamentos num alto grau de escalabilidade.

Apesar de ser um projeto estruturado com bastante potencial de uso, tanto o *libmapper*, por ser uma biblioteca de programação, quanto as interfaces gráficas, que se baseiam nela, apresentam pouca disponibilidade para o público não técnico, necessitando a presença de algum técnico para que o sistema funcione adequadamente.

4.3. Novas Formas de Interação e Música

Nos últimos anos, tem-se visto uma grande popularização de novas formas de interação com dispositivos computacionais, muitas vezes, potencializada pela indústria de entretenimento. É fácil concluir que os novos controles de videogame e os mais novos *smartphones* e *tablets* são os vetores desta crescente popularidade.

Além disso, com o avanço de práticas de programação cada vez mais acessíveis, estes dispositivos estão sendo integrados de forma mais fácil aos projetos interativos desenvolvidos por programadores, designers, artistas e músicos. Esta integração se dá na maioria das vezes pelo *hackeamento*³ da interface de controle e a assimilação do seu funcionamento dentro desses sistemas. Como consequência, há o surgimento de vários projetos interativos e alguns deles voltados para a música. Abaixo, serão mencionadas algumas dessas novas interfaces e apresentadas algumas reflexões sobre os paradigmas de interação usados atualmente nessas interfaces e a incorporação de outros paradigmas.

4.3.1. Controles de Videogames

Uma dessas interfaces populares é o *Nintendo WiiMote*, lançado em novembro de 2006 como controle da plataforma de videogame *Nintendo Wii*. Trata-se de um controle sem fio com o qual o usuário pode interagir com os jogos através de movimentos, botões e apontando para tela.

³ Ato de se apropriar do funcionamento de um dispositivo, sem a permissão oficial do fabricante, para fins de programação, levando à modificação do seu uso a diferentes formas e contextos.

Para proporcionar estas interações, o *WiiMote* contém acelerômetros, botões e uma câmera que capta luz infravermelha, usada para rastrear a posição que o usuário está apontando na tela. Adicionalmente, um dispositivo chamado *Nunchuck*, composto de acelerômetros, dois botões e um pequeno joystick, pode ser acoplado ao *WiiMote*, ampliando a quantidade de movimentos e controles captáveis (vide Figura 4.10). Como formas de feedback presentes no controle, há um motor de vibração proporcionando feedback háptico; um pequeno alto-falante para o feedback sonoro e pequenos LEDs para o feedback físico (Lee, 2008).

A comunicação sem fio entre o controle e o console *Wii* se dá através do protocolo de comunicação *Bluetooth* e esta foi uma das suas características que mais influenciou a popularização no uso em projetos interativos. Por ser um protocolo padrão para comunicação em pequenas distâncias, o *Bluetooth* é implementado em diferentes plataformas, como smartphones e computadores e isto permite que tais dispositivos consigam se conectar com o *WiiMote*. Apesar das mensagens não estarem em formato aberto, ou seja, apesar do conteúdo enviado pelo *WiiMote* não ser publicamente divulgado pela Nintendo, a comunidade *hacker* conseguiu decifrar o conjunto de comandos enviados do *WiiMote* para o console *Wii* e desenvolveu drivers e programas para computador que conseguiram entender essas mensagens, tornando possível a sua integração em ambientes de desenvolvimento.



Figura 4.10: Nintendo WiiMote e Nunchuck

Apesar de apresentar certa latência para o disparo de eventos (Silva, 2012), o *WiiMote* vem sendo usado em projetos acadêmicos como interface gestual para música (Bencina & Wilde, 2008; Bott, Crowley, & LaViola, 2009; Kiefer, Collins, & Fitzpatrick, 2008). Em uma abordagem modular e parecida com o conceito do *WiiMote*, (Maestracci, Frechin, & Petrevski, 2011) apresentam diversos controles sem fio que podem ser usados na construção de novos instrumentos musicais digitais.

Outro dispositivo bastante popular em projetos interativos é o Microsoft Kinect Sensor. Lançado em novembro de 2010 como alternativa de controle gestual para o console Xbox 360 e em

fevereiro de 2012 para o sistema operacional Microsoft Windows, trata-se de uma câmera de profundidade capaz de “ver” em três dimensões. O Kinect é baseado em uma câmera infravermelho associada a um pequeno projetor de padrões infravermelhos. Os padrões são projetados no ambiente e pela deformação causada pelas diferentes profundidades do local, o sensor calcula uma nuvem de pontos aproximada da realidade.

A empresa Adafruit Industries⁴ incentivou o desenvolvimento do driver *open source* para o Kinect criando um concurso assim que o sensor foi lançado para ver quem seria o primeiro desenvolvedor a conseguir a façanha. Não demorou muito, para a comunidade hacker se apropriar deste novo sensor e em menos de duas semanas do seu lançamento, a interface já tinha um driver capaz de traduzir seus comandos para o computador através da USB.



Figura 4.11: Microsoft Kinect

Em junho de 2011, a Microsoft lançou o SDK do Kinect, permitindo que aplicações fossem criadas para Microsoft Windows 7 utilizando o sensor como interface de entrada.

Usando frameworks e bibliotecas de reconhecimento de padrões, como é o caso do OpenNI⁵, pode-se identificar a forma humana e rastrear vários pontos do corpo (vide Figura 4.12). Para maiores informações sobre o funcionamento do OpenNI, recomenda-se a leitura (Shotton et al., 2011).

Em experimentos com músicos, (Silva, 2012) apresenta uma comparação entre o WiiMote e o Kinect: apesar dos dois apresentarem perceptíveis latências, a resposta do Kinect se mostrou a pior entre os dois controles. Alguns músicos afirmaram que gostariam de aprofundar o uso com o WiiMote, mas o Kinect, em seu estado atual, ainda não se mostrou adequado para o cenário de disparo de eventos devido ao atraso inerente a interface.

⁴ <http://goo.gl/r6YbU>

⁵ <http://www.openni.org/>

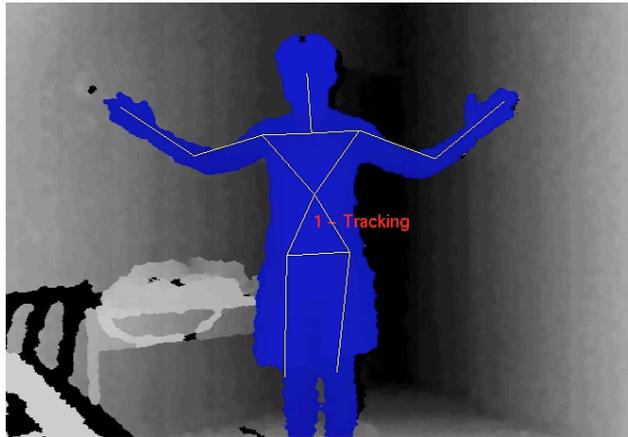


Figura 4.12: Rastreamento de pontos do corpo usando OpenNI e Kinect

Na Figura 4.13, pode ser visto o V Motion Project que usa o Kinect como interface de controle de um sistema musical baseado na manipulação de propriedades sonoras e processos musicais, aplicando efeitos e disparando loops pré-gravados. O usuário se posiciona em frente a uma parede onde sua silhueta ampliada está sendo projetada. A partir do movimento da sua silhueta, ele pode manipular o sistema através da ativação de controles presentes na interface. É um projeto com grande impacto visual e sonoro, porém, como será discutido a seguir, o paradigma usado na interação ainda é muito relacionado ao uso de elementos gráficos de interface. Salvo em alguns momentos em que o usuário movimenta as duas mãos e o ângulo de inclinação entre elas interfere no grau de efeito que está sendo aplicado ao som, o restante da interação se baseia no disparo de controles gráficos pelo “contato” com a silhueta do usuário.

Na página de *making of* do V Motion Project⁶, os desenvolvedores explicam que o sensor ainda apresenta muita latência quando é usado para o rastreamento de pontos do corpo. Para driblar este infortúnio, eles se basearam na interação em duas dimensões através da silhueta. Isto limita o usuário, porém potencializa a resposta da interação. De toda forma, a latência é perceptível e principalmente em ações que não são tolerantes ao atraso, como é o caso do disparo de eventos (no caso loops pré-gravados), o impacto no ritmo da música é audível.

⁶ <http://www.custom-logic.com/blog/v-motion-project-the-instrument/>



Figura 4.13: V Motion Project', projeto musical interativo feito com o sensor Kinect

Além dos controles de videogame já apresentados acima, outro bastante popular é o Playstation Move⁸ para o console Playstation da Sony, que se baseia no tipo de interação do WiiMote, através do movimento de controles sem fio. A principal diferença deste controle comparado ao Wii é que o rastreamento do ponteiro na tela é feito por uma câmera conectada ao console, chamada Playstation Eye, que identifica um bulbo luminoso posicionado na extremidade do controle. Para uma descrição mais detalhada dos controles de videogames, recomenda-se a leitura de (Madden, 2012).

4.3.2. Dispositivos Móveis e Telas Multitoque

As telas multitoque apresentam uma nova forma de interação e os dispositivos móveis como smartphones e tablets estão popularizando este novo tipo de interface. Abaixo, é apresentada uma breve descrição de como estes dispositivos vêm se popularizando.

Em fevereiro de 2006, Jeff Han, professor pesquisador da NYU, apresentou para a plateia do TED⁹ o seu protótipo de superfície multitoque criado em 2005. Não demorou muito para que projetos online, contendo descrições detalhadas de como construir superfícies multitoque de baixo custo, fossem lançados na internet. Um dos pioneiros e atualmente uma grande comunidade de compartilhamento de projetos para interfaces multitoque é NUI Group¹⁰ criado em 2006:

⁷ <http://www.v.co.nz/#the-motion-project>

⁸ <http://us.playstation.com/ps3/playstation-move/>

⁹ <http://www.ted.com>

¹⁰ <http://nuigroup.com/>

“NUI Group offers a collaborative environment for developers that are interested in learning and sharing new HCI (Human Computer Interaction) methods and concepts. This may include topics such as: augmented reality, voice/handwriting/ gesture recognition, touch computing, computer vision, and information visualization.” (NUI Group Authors, 2009)

Como exemplo de trabalho acadêmico que faz a ligação entre superfícies multitoque de baixo custo e música, pode-se citar (Montag, Sullivan, Dickey, & Leider, 2011). O Reactable (Sergi Jordà et al., 2005), descrito anteriormente, é um exemplo de instrumento musical digital que utiliza esta nova interface.

Porém, entre 2005 e 2007, as superfícies multitoque baseadas em mesa ou eram caras demais (Microsoft Surface custava em torno de 10 mil dólares), ou eram imprecisas e de baixa confiabilidade (quando construídas à maneira DIY - *Do It Yourself* ou Faça Você Mesmo). Foi quando em junho de 2007, o iPhone foi lançado. O smartphone da Apple apresentava superfície multitoque de alta qualidade, com pouca latência e boa definição de precisão, porém ainda era pequena (3,5 polegadas), cara e com dificuldades de compra devido a contratos com operadoras de celular. Em setembro de 2007, o iPod Touch foi lançado. Apesar de continuar com o mesmo tamanho de tela, este *player* era mais barato que o iPhone e não precisava de contrato com operadoras de celular.

Um fato importante para a popularização dos produtos da Apple foi o lançamento, em março de 2008, do SDK para iPhone OS e poucos meses depois, em julho de 2008, foi a vez da App Store ser lançada, permitindo que os desenvolvedores criassem aplicativos para os dispositivos móveis e vendessem de forma prática em uma plataforma de distribuição online de aplicativos.

O sistema operacional Android, principal concorrente da plataforma iOS, é lançado em setembro de 2008, que por ser *open source* fez com que uma grande quantidade de marcas (Samsung, Motorola, HTC etc.) o adotasse como plataforma. Seguindo a estratégia da App Store, em outubro de 2008, é lançada a loja de aplicativos para Android, o Android Market (que veio a se tornar em março de 2012 o Google Play).

Apesar do nome da plataforma ser único, existe uma grande fragmentação de versões do sistema operacional Android. Existem programas apenas compatíveis com algumas versões do SO, o que torna a plataforma heterogênea e pouco atrativa para o desenvolvimento de alguns aplicativos. A Figura 4.14 mostra a fragmentação das versões do Android no tempo.

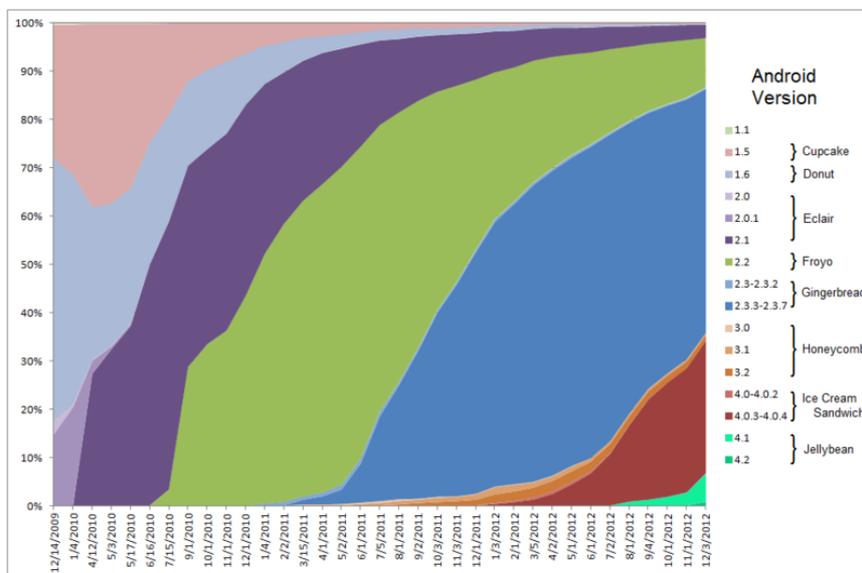


Figura 4.14: Fragmentação das versões do Android de dezembro de 2009 a dezembro de 2012¹¹

O iPad é lançado em abril de 2010, com superfície de 9,7 polegadas, ainda caro, porém com versões livres de contratos. Além de ser um dispositivo móvel, o iPad, por possuir uma tela maior que os antecessores e uma interação multitoque de boa qualidade (boa resolução gráfica e baixa latência), se tornou bastante popular como nova interface no meio musical.

Como exemplo do uso do iPad como instrumento musical, há o *Concerto for iPad and Orchestra*¹² por Ned McGowan (ver Figura 4.15), que apresenta uma orquestra tradicional com a solista tocando um iPad. Para dar feedback visual ao público, um telão apresentava um *close* na tela do iPad. O aplicativo usado implementa uma interface que é um misto entre a interação baseada em disparos do piano e o ajuste entre sonoridades baseado no violino.



Figura 4.15: Concerto for iPad and Orchestra

¹¹ Gráfico extraído de http://en.wikipedia.org/wiki/File:Android_historical_version_distribution.png

¹² <http://www.youtube.com/watch?v=eRYkC6fY190>

4.3.3. Reflexões sobre Paradigmas de Interação

Ilustrando algumas reflexões sobre os paradigmas de interação para novas interfaces, o projeto “*kinect studies*” de Johannes Kreidler, Figura 4.16, utiliza o Kinect como interface de entrada para mapeamentos um-para-um das coordenadas X e Y para definir, respectivamente, a altura e o volume da nota de cada instrumento. Para o cenário apresentado na figura, o timbre escolhido corresponde ao instrumento sendo movimentado, de modo que o movimento do violino resulta em um som sintetizado de violino e o movimento do teclado eletrônico resulta no som de um teclado eletrônico.

O resultado musical não é interessante, mas este não é o objetivo do projeto. Com este projeto, Kreidler reflete sobre a forma de como se está usando estas novas formas de interação na música. Muitos dos projetos musicais com essas interfaces apenas simulam o que já existe de instrumento acústico e não exploram os novos potenciais apresentados por elas. Esta também é uma crítica de (Sergi Jordà, 2005) para um contexto geral de novos instrumentos digitais e de (McGlynn et al., 2012) mais voltado ao contexto de superfícies multitoque.

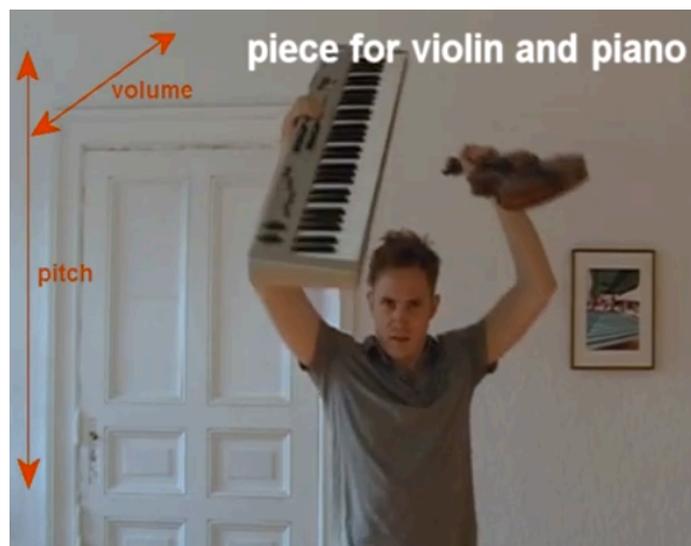


Figura 4.16: kinect studies por Johannes Kreidler¹³

(Sergi Jordà, 2005) argumenta que novos instrumentos digitais podem gerar novos gêneros musicais e não precisam necessariamente estar associados a gêneros já existentes ou formas tradicionais de tocar. Já (McGlynn et al., 2012) mostra que muitas vezes, atribui-se a novas interfaces antigos paradigmas, causando incompatibilidades e inadequações na interação com esses novos dispositivos.

No contexto de superfícies multitoque, isto fica evidenciado pelo intenso uso das interfaces gráficas baseadas em janelas, ícones, menus e ponteiros (paradigma WIMP), sem a reflexão

¹³ <http://www.youtube.com/watch?v=UA1cTnvBBS0>

necessária de adequação ao novo dispositivo, pois tal tipo de paradigma é mais adequado para interfaces como o mouse e o teclado. Em relação a instrumentos para iPad, como pode-se ver na Figura 4.17, que a interface de alguns aplicativos simulam instrumentos ou dispositivos existentes. A questão é saber quão adequada é esta simulação, baseada em outro paradigma de interação, aplicada a superfícies multitoque. Por exemplo, com a simulação de um piano, perde-se a noção tátil, tornando a interação complicada e muitas vezes imprecisa.



Figura 4.17: Exemplos de aplicativos para iPad que simulam instrumentos e dispositivos existentes

No caso do exemplo do aplicativo mostrado na Figura 4.18, vê-se o uso do paradigma baseado em interação com mesas de som formadas por *knobs*, *faders* e botões em um aplicativo musical para iPad. A interface apresenta uma grande carga cognitiva de informações para ser mostrada em uma tela de 9,7 polegadas. A dimensão dos botões é reduzida e pode causar erros de múltiplos toques e o uso dos *knobs* não é intuitivo, dado que são pequenos e não permitem a metáfora de rotação usada no mundo físico, forçando o uso de uma ação não correspondente a resposta dada pela interface, que é: arrastar o dedo para cima para girar o *knob* para direita e arrastar o dedo para baixo para girar o *knob* para esquerda.

Apesar de reforçar o uso de metáforas e referenciais culturais de interfaces com as quais já se está acostumado, (McGlynn et al., 2012; Saffer, 2008) apresentam os gestos como uma forma mais adequada de se pensar o paradigma de interação multitoque. A partir da combinação de toques e reconhecimento de gestos, a interface gestual associada à interface gráfica pode potencializar a usabilidade em superfícies multitoque.



Figura 4.18: Interface gráfica do aplicativo Rebirth for iPad

Indo além do contexto das interfaces multitoque, tal reflexão pode ser verdadeira para o contexto de interação através de movimentos captados por acelerômetro, como é o caso do WiiMote, e para interfaces baseadas em rastreamento do corpo, como é o caso do Kinect.

Robert McKee, famoso professor de roteiro em Hollywood, comentou em recente entrevista¹⁴ sobre o impacto de novas tecnologias na qualidade final do filme. Do cinema mudo para o cinema com áudio, por exemplo, os filmes perderam a narrativa por um tempo, preocupando-se em simplesmente usar a tecnologia e deixando de lado a qualidade do roteiro. O mesmo padrão se mostrou quando a película passou a ter cor, quando se lançou o *widescreen* e mais recentemente o 3D. Segundo McKee, há um desespero inicial em dominar a técnica e ao mesmo tempo uma sensação de novidade por parte do público, que pouco se importa se o filme é bom ou não, pois a tecnologia, por si só, já chama atenção.

Pode-se extrapolar esta reflexão de McKee para a situação que estamos vivendo em relação ao uso de novas interfaces para música. O surgimento e a popularização dessas novas interfaces ainda são recentes e a reflexão sobre a qualidade dos instrumentos é colocada de lado parte por conta da novidade tecnológica e parte para se ganhar experiência com a tecnologia.

Assim, é de se esperar que com o tempo para o domínio da técnica e para reflexões sobre os seus usos, as novas interfaces possam ser usadas de forma mais madura em futuros projetos musicais.

¹⁴ <http://goo.gl/Wp9aF>

5. Bases Conceituais

5.1. Movimento *Maker*

"More than mere consumers of technology, we are makers, adapting technology to our needs and integrating it into our lives. Some of us are born makers and others, like me, become makers almost without realizing it. [...] MAKE is a new magazine dedicated to showing how to make technology work for you." Dale Dougherty, co-fundador da revista MAKE, O'Reilly.

Assim começava a frase de boas-vindas da primeira edição da revista MAKE, publicada em janeiro de 2005, que desde então compila vários projetos e assuntos relacionados à cultura DIY, à apropriação e à customização de artefatos tecnológicos. O lançamento desta revista marca uma tendência que tem suas raízes no ciberativismo, no movimento punk e na cultura faça-você-mesmo e agora se potencializa através do massivo compartilhamento de informações pela internet. Muitos internautas estão percebendo que o conhecimento não está preso nas indústrias e nos laboratórios de pesquisa e sim que o mundo pode ser *hackeado*, ou seja, os sistemas e dispositivos que nos rodeiam podem ser modificados para um uso diferente do pensado inicialmente pelos seus produtores. É o apoderamento da tecnologia pelos usuários, clientes, consumidores.

Mark Frauenfelder, editor da revista MAKE, quando perguntado pelo site *Humans Invent*¹⁵ sobre o movimento Maker, deu a seguinte resposta:

"[The Maker movement] is the realisation by a lot of people that the physical world is a hackable platform in much the same way as computer systems. [...] A hacker is someone who creates an unintended use for something and makes something better than the original designer"

Outra vertente desta tendência é o surgimento de impressoras 3D mais baratas e voltadas para o uso pessoal. Em seu recente livro "*Maker: The New Industrial Revolution*" (Anderson, 2012), Chris Anderson define a popularização e uso das impressoras 3D como sendo uma nova revolução industrial. Os clientes poderão comprar modelos 3D de objetos que os interessem através da internet e imprimir objetos em casa utilizando uma impressora 3D, que no futuro

¹⁵ <http://www.humansinvent.com/#!/9093/the-rise-of-the-maker-movement/>

serão tão populares quanto nossas impressoras comuns de hoje em dia. Com isso, o poder da produção do artefato fica na mão do usuário.

Apesar de não se relacionar diretamente com o contexto deste projeto, já podem ser vistos alguns instrumentos musicais acústicos sendo construídos com impressoras 3D, como é o caso de uma flauta, cujas peças foram todas impressas em plástico por uma impressora 3D (Zoran, 2011).

Além disso, ferramentas de programação voltadas para artistas e designers vêm se tornando cada vez mais populares (Noble, 2012). Estas ferramentas integram tanto uma linguagem de programação, quanto um ambiente de desenvolvimento cujo objetivo é a obtenção de resultados gráficos, sonoros e interativos de forma simples e rápida, sem a preocupação com detalhes técnicos que muitas vezes são obstáculos para um público mais leigo – como configuração de bibliotecas, compiladores etc. Assim, há um crescente número de projetos interativos (Fry & Reas, n.d.) (Dougherty & Frauenfelder, 2005)(“Arduino.cc,” n.d.). Vê-se então uma tendência no desenvolvimento de ferramentas que facilitam a absorção da tecnologia existente por parte de usuários não técnicos como artistas e músicos. Este movimento também faz parte da tendência de dar poder ao usuário no processo de implementação de sistemas, possibilidade de codificar, de entender o que funciona por trás da tecnologia que ele consome.

Ampliando a definição, pode-se dizer que a tendência é que os usuários, clientes ou consumidores não sejam passivos no desenvolvimento de novas ferramentas e sistemas e sim que participem da construção e da modificação, sempre se apropriando desses sistemas e os adequando para suas próprias realidades e necessidades. O universal é inexistente, há sempre uma modificação, adequação ou customização que o usuário pode fazer para se apropriar de um artefato tecnológico - seja permitida e pensada a priori pelo produtor deste artefato ou não permitida e *hackeada* pelo usuário. É o caso mencionado anteriormente do controle de Wii usado no computador.

Esta tendência associada a um comportamento já existente há anos na música eletrônica, traz novas possibilidades para a construção de instrumentos, já que os músicos, designers e artistas estão cada vez mais perto da produção tecnológica. Como se percebe na frase abaixo, em tempos atrás, os músicos do Kraftwerk já sentiam a necessidade de novos dispositivos que gerassem o som que eles tinham em mente e para tanto, precisavam construir um instrumento customizado.

“Some of the components we have are standard things like echo machines, but more than half of the equipment is custom-built, built by us or our regular engineers because we couldn't play the music we wanted on regular instruments”. Resposta de Ralf Hütter, componente de Kraftwerk, à entrevista da revista Keyboard. (Kirn, 2011, p. 8)

Mais uma evidência desta tendência na música é chamada para artigos do Organised Sound, famoso periódico da área musical:

“A new community of ‘makers’ has emerged in electronic music. These makers are not content with using off-the-shelf ‘instruments’, but seek to create their own devices and systems to generate sound”. Chamada para submissões para *Organised Sound*¹⁶, volume 18/3, a ser publicado em dezembro de 2013.

O Projeto aqui apresentado tem por objetivo dar poder ao usuário para que ele mesmo consiga criar o seu próprio instrumento musical digital. Assim, a proposta deste trabalho se relaciona diretamente com esta tendência do usuário como agente ativo do processo de produção, desmistificação e apropriação da tecnologia.

5.2. Experimentação e Prototipação

O conceito de experimentação usado neste Projeto está relacionado ao ato de testar a adequação de uma ideia a um determinado contexto a partir da implementação incompleta de subconjuntos desta ideia em forma de protótipo. A partir da experimentação e da prototipação, pode-se antecipar inadequações, fazendo com que a ideia evolua gradativamente a partir do aprendizado.

“Experimentation matters because it fuels the discovery and creation of knowledge and thereby leads to the development and improvement of products, processes, systems, and organizations.” (Thomke, 2003)

Apesar de ser mais voltado para o mundo empresarial, (Thomke, 2003) faz uma reflexão em forma de princípios relacionados com o processo de experimentação. Dentre os apresentados nos livros destaca-se:

- a) Experimente frequentemente;
- b) Organize-se para experimentações rápidas e ágeis;
- c) Falhe cedo e regularmente.

Dentro do contexto de design de produtos e sistemas, há a fase de prototipação, onde as ideias concebidas como solução para um determinado problema são concretizadas e funcionam como ferramenta: de aprendizado sobre o que precisa ser refinado, de comunicação

¹⁶ <http://journals.cambridge.org/action/displaySpecialPage?pageId=3920>

entre os membros da equipe sobre as particularidades do projeto, de experimentação de diferentes abordagens para uma mesma solução.

*“Prototyping is a strategy for efficiently dealing with things
that are hard to predict.”
Scott Klemmer¹⁷*

Em (Warfel, 2009), o autor mostra alguns valores associados aos protótipos:

- a) Protótipos são gerativos, ou seja, o processo de prototipação gera várias ideias que podem ser péssimas ou brilhantes;
- b) Protótipos tem o poder de mostrar, contar e experimentar: apenas com o concreto é que conseguimos ter uma experiência da realidade;
- c) Protótipos reduzem problemas de interpretação;
- d) Protótipos salvam tempo, esforço e dinheiro;
- e) Protótipos reduzem perda;
- f) Protótipos proveem valores do mundo real: em vez de descrever a realidade em documentos, melhor apresentar um protótipo que já vem com valores reais agregados.

(Dow, 2011) justifica que segundo experimentos, focar na quantidade, ou seja, criar vários protótipos para um mesmo problema, faz com que as soluções sejam mais eficientes, pois “ao fazer, aprende-se com os próprios erros”.

*“The best way to have a good idea is to have lots of
ideas.”
Linus Pauling*

5.3. Metadesign

Um conceito intimamente relacionado com este Projeto é o Metadesign. O metadesigner pensa no metaproduto, ou meta-artefato, que será usado pelo usuário para produzir o artefato final. É uma formulação de abstração anterior ao produto, um ambiente que proporciona ao usuário o poder de criar o artefato levando em consideração seus gostos, intenções e contextos específicos. Apesar de pré-determinar parte da realidade, o ambiente do metadesign permite a combinação de variáveis que aumentam o poder de expressividade do usuário para sua construção própria e individual.

Quando usado pela primeira vez, em 1963, por Andriek Von Onck, o termo metadesign expressou o sentido de “processo de projeto do próprio processo de projeto” (Vassão, 2010).

¹⁷ <https://www.coursera.org/course/hci>

Neste caso, o prefixo “meta” assume a mesma função que no termo metalinguagem e metafísica.

A segunda conotação tem a ver com o design de um sistema cujas próprias características se modificam com o uso. Em outras palavras, se relacionam com o conceito de projeto de sistemas no qual o usuário tem o poder de modificar configurações e características para que melhor se adequem ao seu contexto. É o caso da definição proposta por (Fischer, 2003), que descreve sistemas que transformam o usuário em co-designers não apenas na fase de concepção e construção, mas em toda sua existência, inclusive durante o seu uso. Esta definição não se aplica a este Projeto.

No terceiro significado possível, o prefixo “meta” se relaciona com palavras como “além”, “após”, “depois de”, dando a ideia de movimento, de transposição. Assim como no caso de “metáfora”, “meta”, “metabolismo”. Neste caso, “metadesign” significa aplicar um conjunto de princípios de projeto em diferentes contextos de aplicação, através de operações genéricas, gerando diferentes resultados.

Neste trabalho, será utilizada a primeira aceção, segundo a qual Metadesign se relaciona em projetar não o objeto final, mas sim um objeto intermediário que tem como função projetar o objeto final. Este estágio intermediário deve prover diversos objetos abstratos, ou módulos, que, a partir de suas inter-relações, permitam construir o objeto final.

Como exemplo, tem-se a linguagem de programação MetaFont (Knuth, 1986), pela qual o usuário tem acesso a uma infinidade de combinações de formatações e características de fontes que, a partir de suas combinações, geram novas fontes.

6. Descrição do Processo

O processo de design usado para o Sketchument segue a mesma filosofia que ele se propõe a prover, ou seja, usar protótipos para aprender e construir sistemas com decisões de projeto bem mais maduras e ancoradas na opinião de potenciais usuários.

Neste Capítulo, é descrito de forma geral o processo de concepção e de construção do Sketchument. Tal processo de desenvolvimento do ambiente de experimentação foi formado a partir da escolha de métodos presentes nos processos de design apresentados a seguir.

O IDEO *Method Cards* é um conjunto de 51 métodos usados pela empresa de design IDEO para inspirar e manter as pessoas no centro do processo de design (IDEO, 2003). Para cada método, existe uma carta com uma imagem ilustrativa, uma descrição e uma breve história sobre como e quando usá-la. As cartas são divididas em quatro categorias: Aprender, Ver, Perguntar e Tentar. Manipulando as cartas, o designer consegue ter a visão total dos métodos e escolher os que melhor se adequam ao contexto do projeto.

O *Human Centered Design (HCD) Toolkit* é “um processo e um conjunto de técnicas usados para criar novas soluções para o mundo” (IDEO, 2011). Foi desenvolvido com uma abordagem mais humanitária do design, sendo usado para desenvolver projetos que já melhoraram a vida de milhões de pessoas, como o desfibrilador HeartStart, produtos como antibacterianos naturais CleanWell e o sistema de doadores de sangue da Cruz Vermelha. É um processo centrado no usuário cujo princípio é extrair a resposta para os principais problemas das comunidades através das pessoas que as compõem. Os métodos do HCD podem ser divididos em três categorias: Escutar, Criar e Entregar.

O *Running Lean* é um processo sistemático de desenvolvimento de projetos voltado para startups (Maurya, 2012). Sua abordagem baseia-se em ciclos curtos de iteração onde parte do produto é construída e validada por potenciais usuários. A partir do aprendizado de cada ciclo, o produto é modificado para melhor se adequar ao seu objetivo. A velocidade, o aprendizado, a participação dos usuários no desenvolvimento e a avaliação constante do produto são algumas máximas desse processo. O ciclo de atividades pode ser descrito a partir das seguintes etapas: Aprender (ideias), Construir (produtos) e Medir (dados).

O *eXtensible Design Methods (XDM)* (Neves et al., 2008) é um processo de design que utiliza o ambiente cibernético como apoio ao armazenamento e a troca de informações sobre o projeto entre a equipe de design. Esta equipe utiliza ferramentas, como blogs e wikis, para aplicar os métodos e registrar seus resultados, permitindo que todos os membros tenham acesso às informações e à evolução do processo de forma holística e assíncrona. Os métodos apresentados pelo XDM se dividem nas seguintes categorias: Exploração do Problema, Geração de Alternativas, Seleção de Alternativas, Avaliação de Alternativas e Descrição da Solução.

O processo utilizado neste Projeto, assim como os processos descritos anteriormente, baseiam-se no design centrado no usuário, agente fundamental na concepção e na validação das soluções propostas. Para a construção do processo, os métodos mais adequados ao Projeto foram selecionados e divididos em cinco categorias: inspiração, investigação, prototipação, ideação e avaliação, descritas a seguir:

- Para os **métodos de inspiração [INSP]**, o objetivo é entender uma determinada situação, encontrar problemas e oportunidades para desenvolvimento de soluções.
- Os **métodos de investigação [INVS]** buscam encontrar referências e fatos que ajudam a entender a situação e o problema.
- As **ferramentas para ideação [IDEA]** são listas ou artefatos visuais que ajudam a estruturar o processo de geração e seleção de alternativas, definindo um espaço de exploração e um conjunto de direções a serem tomadas.
- Os **métodos de prototipação [PROT]** tornam as alternativas concretas por meio de construção de um artefato que implementa parte das funções propostas pela solução.
- Os **métodos de avaliação [AVAL]** são usados para averiguar se as soluções propostas são adequadas ao problema e gerar modificações e melhoramentos para os protótipos.

Os resultados da aplicação de alguns métodos servem de entrada para outros métodos, emergindo uma inter-relação cíclica entre eles. Abaixo seguem as descrições dos métodos utilizados neste Projeto, compostas por pontos como objetivo, entradas, procedimento e saídas.

A Figura 6.1 apresenta, de forma geral, como se deu a evolução do processo de design do Sketchument. É importante deixar claro o caráter híbrido dos métodos relativos à Entrevista e ao Questionário, sendo ambos pertencentes tanto à categoria de Investigação quanto à de Avaliação, pois ao mesmo tempo em que servem de base para a coleta de informações do domínio da aplicação, também são usados como fonte de feedback dos protótipos. Outro ponto importante é que, tanto o método de Análise da Literatura, quanto o método de Definição das Diretrizes do Projeto não terão seus resultados descritos, pois já foram apresentados nos Capítulos 3 e 3. Trata-se de uma interseção entre o Método da Pesquisa apresentada neste documento e o Processo de Design do ambiente de experimentação.

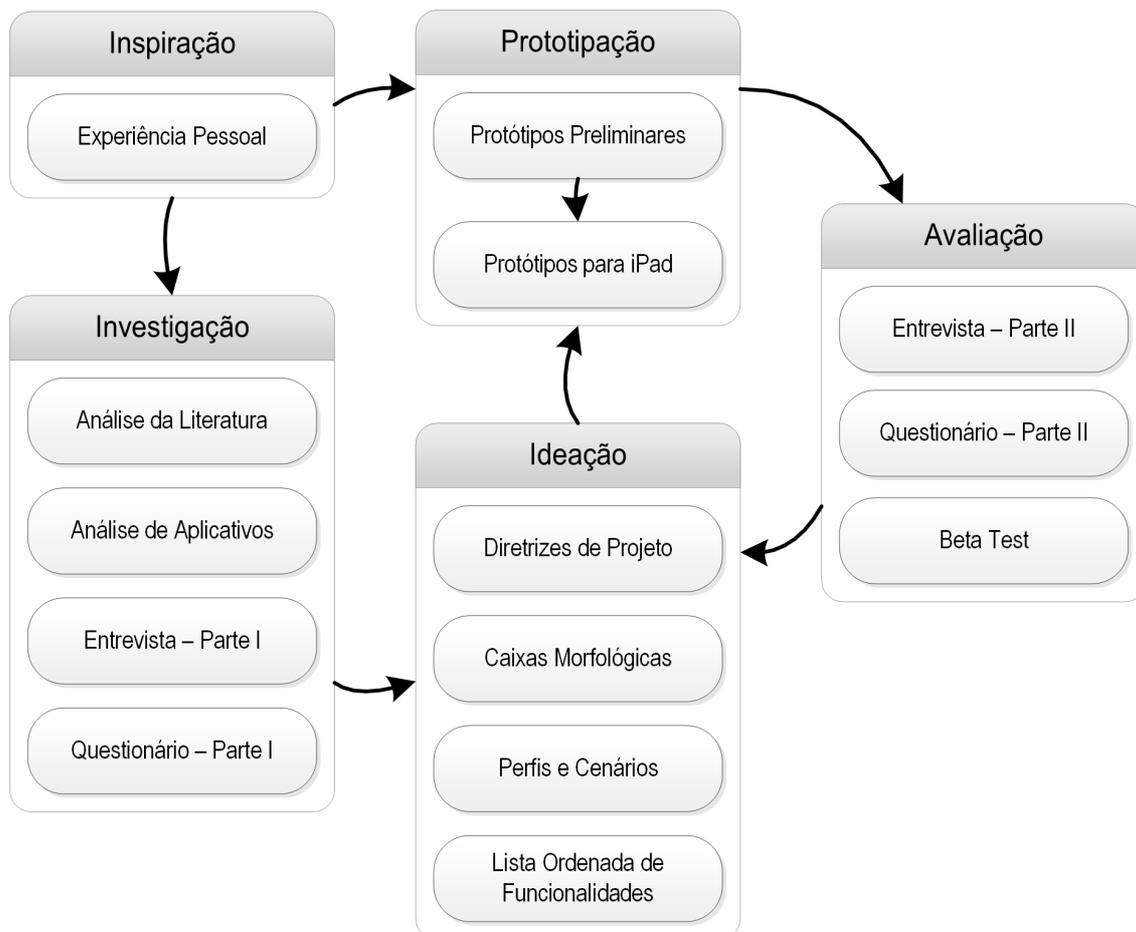


Figura 6.1: Métodos utilizados no processo de design do Sketchument

6.1. Métodos de Inspiração

6.1.1. Experiência Pessoal

Esta etapa do método tem como principal objetivo dar início ao ciclo do projeto. A partir de experiências anteriores na área de atuação – no caso deste Projeto, construção de instrumentos musicais digitais –, um problema abrangente é levantado, uma hipótese representa sua definição e soluções preliminares são propostas. Como saída desta etapa, existem os protótipos preliminares, geralmente de baixa-fidelidade, bastante rústicos e incompletos.

Donald Norman, em (D. Norman, 2006), defende a abordagem “primeiro faça, depois estude”, o que se relaciona bastante com a maneira como este Projeto foi desenvolvido. A necessidade surgiu de uma experiência pessoal, que inicialmente não foi constatada como necessidade de mais usuários. A partir daí uma ferramenta foi desenvolvida para ser então avaliada por um público mais amplo.

6.2. Métodos de Prototipação

6.2.1. Protótipos Preliminares

Esses protótipos têm como principal função concretizar ideias que poderão ser avaliadas ou servir de base de comunicação e comparação para a evolução do Projeto. Algumas definições e hipóteses funcionam como entrada desta etapa; as ideias são pensadas de forma concreta e um artefato torna-se saída desta etapa. Com este protótipo, as alternativas pensadas podem ser avaliadas com potenciais usuários e modificações são facilmente discutidas e repensadas.

6.2.2. Protótipos para iPad

Nesta etapa, a partir de uma lista de novas funcionalidades e de melhoramentos do protótipo levantados nas fases anteriores e alinhados com as diretrizes do Projeto, a arquitetura de uma versão de protótipo para iPad é concretizada. O diagrama esquemático da arquitetura apresenta as classes e suas relações. Este diagrama é utilizado para facilitar a fase de implementação.

A arquitetura do protótipo é então codificada para o iPad para a implementação do aplicativo. Como resultado, produz-se uma nova versão do protótipo que serve de base para futuras avaliações e modificações.

Nesta etapa, uma linha do tempo dos protótipos é descrita a fim de demonstrar a evolução das ideias e funcionalidades implementadas no aplicativo. Com isto, previne-se a recorrência de erros em decisões e facilita a priorização de novas decisões baseadas naquelas adotadas anteriormente.

6.3. Métodos de Investigação

6.3.1. Análise da Literatura

A análise da literatura compreende a leitura, interpretação e assimilação de conceitos a partir de materiais escritos como artigos científicos, artigos encontrados na internet, descrição de funcionamento de sistemas etc. O procedimento compreende a seleção de trechos importantes, a associação desses trechos a categorias ou nós de informação que depois podem ser associados ou separados em grupos distintos, gerando um modelo de entendimento do domínio do Projeto. Como saídas desta etapa, além de um melhor entendimento do contexto, há a formação de uma base conceitual e a obtenção do estado da arte.

6.3.2. Análise de Aplicativos de iPad

A Figura 6.2 apresenta a divisão de aplicativos para iPad selecionados para a análise.

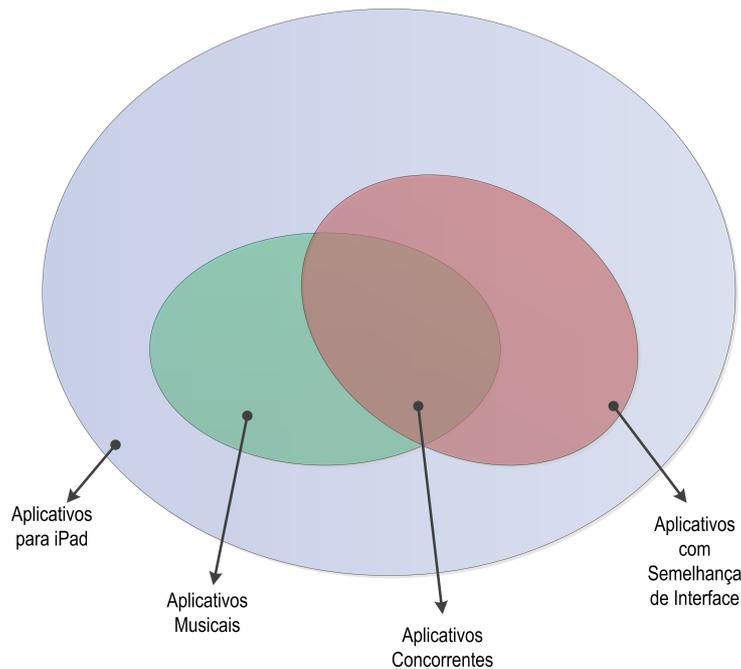


Figura 6.2: Diagrama de Análise de Aplicativos

6.3.2.1 Análise de Aplicativos com Semelhança de Interface

Os aplicativos com semelhança de interface são aqueles que compartilham a mesma plataforma de desenvolvimento do aplicativo proposto – no caso deste Projeto, o iPad – e não necessariamente fazem parte do mesmo domínio de uso deste aplicativo. Estes projetos fornecem inspirações para a construção da interface do aplicativo proposto, pela combinação de ideias extraídas da análise de suas interfaces. Esta etapa é necessária para ver soluções já existentes e sua adequação no contexto do projeto.

Como procedimento, pode-se dizer que, a partir de uma lista de aplicativos com interfaces semelhantes, todos os aplicativos são primeiramente experimentados. Todas as funcionalidades que possam se relacionar com o aplicativo proposto são levantadas e listadas, em um primeiro momento, sem nenhuma ordem ou agrupamento.

As funcionalidades levantadas na análise de aplicativos com semelhança de interface são separadas por afinidade e categorias são definidas. Com estas categorias em mãos, os aplicativos são revisitados a fim de confirmar ou povoar tais categorias com mais valores possíveis. Depois disso, uma caixa morfológica é construída com categorias e os possíveis valores assumidos. Esta caixa morfológica, “uma matriz onde cada linha representa variações possíveis de uma determinada característica” (Neves et al., 2008), é uma estrutura de visualização que dá um panorama geral de quais são os valores que certas funcionalidades da interface podem assumir, dando base para a escolha de caminhos possíveis para a implementação da interface do aplicativo proposto. Se alguma funcionalidade não for bem avaliada pelo usuário, já se encontram disponíveis algumas alternativas àquela implementação.

6.3.2.2 *Análise de Aplicativos Musicais*

A definição de aplicativos musicais no contexto deste Projeto abrange todos os aplicativos que não têm a pretensão de ser um ambiente de experimentação de Instrumentos Musicais Digitais, mas já apresentam algum tipo de instrumento ou interação bem definida. Ou seja, tais aplicativos seriam um resultado de um ambiente de experimentação para DMIs. Assim, suas funcionalidades são importantes para serem analisadas, pois podem ser assimiladas por parte do objeto resultante do processo de experimentação: o instrumento em si.

Como entrada desta etapa, tem-se uma lista de aplicativos musicais dos mais abrangentes tipos. Esta grande lista é analisada e a partir de parâmetros de popularidade (aplicativos mencionados nas etapas de pesquisa) e qualidade de implementação (aplicativos que funcionam sem erros ou inconsistências), alguns são descartados de uma análise mais profunda.

Os aplicativos selecionados são então visitados e, seguindo o mesmo procedimento da análise de aplicativos de interface semelhante, as funcionalidades são extraídas, agrupadas e apresentadas em uma caixa morfológica. Esta caixa agora apresenta um mapa de possibilidade de funcionalidades específicas dos aplicativos musicais selecionados e dão base para a escolha de alternativas que possam vir a ser interessantes para serem implementadas no ambiente de experimentação.

6.3.2.3 *Análise dos Concorrentes*

São considerados concorrentes do aplicativo proposto, todos aqueles que compartilham da mesma plataforma que o ambiente proposto (no caso, o iPad), e que se baseiam no princípio da modularidade, que permitem a modificação de suas estruturas internas para proporcionar a experimentação de novas formas de interação musical na plataforma. Para fazer a análise destes aplicativos concorrentes, critérios de comparação que levam em conta diretamente as diretrizes do Projeto são colocados para a análise de uma lista de aplicativos que se enquadrem na definição supracitada. Os aplicativos são analisados seguindo tais critérios e um quadro comparativo é apresentado.

6.4. Métodos de Avaliação

6.4.1. Entrevistas

Nesta etapa do método, potenciais usuários do aplicativo são entrevistados com a finalidade de colher informações sobre possíveis contextos de uso e oportunidades de atuação do Projeto. Usando uma abordagem de entrevista semiestruturada, são feitas questões referentes às práticas musicais dos participantes e como se dá o uso de tecnologias digitais nessas práticas. Todas as entrevistas são gravadas e depois processadas. Os trechos de maior importância são selecionados e associados a um tema. Depois esses tais temas são agrupados e descritos

como resultados das entrevistas. As descrições possuem frases ilustrativas que situam o pesquisador nas opiniões dos entrevistados. Como resultado desta etapa, têm-se as descrições de cada tema e algumas oportunidades e ideias para posterior processamento e geração de alternativas de funcionalidades para o aplicativo.

6.4.2. Questionário Online

Considera-se o questionário aplicado neste Projeto como método tanto de avaliação quanto de investigação, pois ao mesmo tempo em que serve de base para colher feedbacks sobre o protótipo, também apresenta informações relevantes sobre os perfis de potenciais usuários e cenários de uso.

O questionário online é uma boa alternativa para conseguir impressões e colher opiniões de um grande número de participantes em relação a um projeto. Esta etapa foi incluída no método com o intuito de testar se o aplicativo, ainda em estado de protótipo, está passando uma ideia correta e concreta para o público de potenciais usuários, além de colher informações, referências e sugestões de novas funcionalidades. A entrada para essa etapa é o protótipo funcional do aplicativo que é apresentado ao público através de um vídeo com suas principais funcionalidades sendo demonstradas. O cabeçalho do questionário apresenta a descrição da pesquisa, o pedido para assistir ao vídeo antes de responder às perguntas e o link do vídeo.

A primeira parte do questionário é destinada à fase de levantamento de perfil do público, como faixa etária, relações com a música (profissional ou não), com o uso de tecnologias digitais e se toca algum instrumento musical.

A segunda parte é constituída por questões referentes ao vídeo do Projeto. Todas as questões da segunda parte são abertas, aumentando assim a liberdade das respostas e potencializando a emergência de conceitos, temas ou impressões não pensados *a priori* pelo pesquisador. Este material não estruturado é de grande importância para a avaliação do protótipo e a descoberta de novas linhas de atuação durante o Projeto.

Os temas abordados pelas questões iniciais foram: opiniões sobre a relevância do conceito do aplicativo (um ambiente de experimentação para criação de interações musicais no iPad), indicações de projetos semelhantes, impressões sobre a interface do aplicativo, indicações de potenciais perfis de usuários que utilizariam o aplicativo, indicações de possíveis contextos de uso para o aplicativo.

Depois destas perguntas mais exploratórias, que têm como objetivo o levantamento de informações que podem servir como futuras referências, as perguntas seguintes tratam de colher pontos positivos, negativos e sugestões de melhoramentos para o protótipo apresentado.

Com o material respondido, as questões são processadas e os temas emergentes são transformados em categorias. Tais categorias são quantificadas e servem como base para a

priorização dos temas, ou seja, temas mais comentados têm uma maior importância e devem ser levados mais em conta que temas menos abordados.

Gráficos com essa quantificação são então elaborados para uma melhor visualização das respostas. É importante ressaltar a importância das perguntas abertas no processo de coleta de dados, pois isto permite a emergência de temas não pensados a priori pelo pesquisador. Apesar de um resultado quantitativo, o processo se baseia na análise textual das respostas, o que é uma forma qualitativa de avaliar os dados obtidos.

As saídas desta etapa são os gráficos com os temas abordados nas questões e uma lista de sugestões de funcionalidades e melhoramentos para o protótipo.

6.4.3. Beta Test

A partir da distribuição online, a versão do protótipo é enviada para potenciais usuários que têm a possibilidade de instalar e usar indefinidamente o aplicativo. Além de prover dados de uso do aplicativo (como tempo, ações realizadas dentro do software etc), a plataforma de distribuição online permite que feedbacks textuais sejam enviados de dentro do aplicativo. Desta forma, os usuários testam as funcionalidades fora de um ambiente controlado de experimento, gerando impressões mais profundas, analisando de forma mais espontânea os problemas de interface e mau funcionamento do sistema. A partir dos dados de uso, gráficos são plotados e a partir dos feedbacks textuais, os problemas são descritos e as modificações propostas.

6.5. Métodos de Ideação

6.5.1. Definição das Diretrizes do Projeto

As diretrizes são ideias que servem como linha guia para a tomada de decisão durante as demais etapas do Projeto. São conceitos que norteiam os caminhos que o Projeto irá percorrer. A entrada desta etapa é compreendida pela análise da base conceitual (descrita no Capítulo 5) . A destilação de alguns conjuntos de tópicos importantes é feita e estes tópicos são agrupados por afinidades de conceito. Depois disto, é pensada uma descrição quanto a esses grupos que assim se transformam em diretrizes de projeto. A finalização desta etapa é a lista das diretrizes seguidas de sua descrição .

6.5.2. Caixas Morfológicas

Este é o resultado do Método de Investigação [INVS.2], Análise de Aplicativos, com o qual pode-se ter uma visão geral e mais ampla das opções e assim determinar um caminho a se seguir na construção da interface, além de listar as funcionalidades presentes nos aplicativos musicais e assim escolher de forma sistemática quais funcionalidades serão disponibilizadas pelo ambiente de experimentação.

6.5.3. Descrição de Perfis de Usuários e Cenários de Uso

A partir das análises do questionário e das entrevistas, perfis de potenciais usuários e cenários de uso para o aplicativo são descritos. Tais descrições são resultado do agrupamento das informações em categorias que servirão para a construção de um modelo de usuários e cenários de uso. Como resultado desta etapa, surgem as categorias de usuários e as categorias de cenários de uso, seguidas de uma descrição detalhada e de um modelo. Este modelo facilita a visualização do público-alvo e serve de guia para a geração de alternativas para o aplicativo.

6.5.4. Lista Ordenada de Funcionalidades

Depois de todas as etapas de coleta de dados e avaliação dos protótipos, nesta fase do método, as sugestões, críticas, melhoramentos e ideias para novas funcionalidades são processados e ordenados seguindo as diretrizes do Projeto e a quantidade de menções a estas funcionalidades. Como entrada deste processo, há todas as análises anteriores que dão base para a geração de alternativas. Como resultado, é apresentada uma lista de ações para a próxima versão do protótipo, ordenada seguindo os seguintes critérios: 1) correção de bugs e de mau funcionamento do aplicativo; 2) modificação de funcionalidades na versão corrente do protótipo que de alguma forma feriram as diretrizes do Projeto; 3) melhoramento de funcionalidades da versão corrente do protótipo; 4) escolha de novas funcionalidades por potencializar as diretrizes do Projeto.

7. Inspiração e Protótipos Preliminares

Neste Capítulo, será apresentada a inspiração para a construção do ambiente de experimentação e o conjunto de protótipos que ajudou a concretizar as ideias e o conceito do Projeto.

Neste Projeto, entende-se por protótipo preliminar todo protótipo de baixa-fidelidade (papel ou apresentação de slides) ou em plataforma diferente do protótipo implementado para iPad que, de alguma forma, ajudou a construir a ideia do aplicativo proposto. Abaixo, seguem suas descrições e algumas ideias obtidas com esses protótipos.

Os protótipos desta fase basearam-se fortemente no exemplo do Kinect como nova interface de interação, por isso pode-se perceber o uso constante de uma imagem representativa dos pontos rastreáveis pela câmera de profundidade do dispositivo. A dinâmica de funcionamento independe da forma de entrada, já que neste estágio tinha-se como objetivo entender qual a melhor forma de apresentar as informações de entrada e saída e como seria mais adequada a realização dos mapeamentos. Desta forma, os resultados obtidos por feedbacks de potenciais usuários e especialistas foram úteis e essenciais para a construção dos protótipos do Sketchument para iPad.

7.1. Experiência Pessoal

A experiência pessoal do autor funcionou como estímulo inicial para o Projeto, definindo um problema e uma hipótese de solução, que foi testada nas seguintes etapas do processo de desenvolvimento do aplicativo.

Há alguns anos, o autor deste trabalho vem construindo aplicações interativas voltadas para música e, durante o desenvolvimento, pôde perceber a carência de uma ferramenta preliminar ao projeto em si, na qual o projetista pudesse experimentar uma determinada forma de interação ou mapeamento, podendo mudá-la rápida e agilmente, combinar com outra, sempre obtendo resultados imediatos. Foi visando resolver um problema real do seu processo de criação que o autor resolveu fazer o primeiro protótipo.

Inicialmente, foi pensado um aplicativo no qual o usuário poderia escolher qualquer tipo de dispositivo de entrada (WiiMote, Kinect, Telas Multitoque, Câmera, Arduino etc.), qualquer tipo de saída sonora ou musical e, através de uma interface de mapeamento simples, ele experimentaria quais as melhores estratégias para um protótipo de instrumento. Não seria necessariamente o instrumento musical final, porém um protótipo rústico e preliminar com o qual o projetista pudesse entender melhor suas vontades musicais e maneiras de se expressar através de uma concretização simples e rápida.

A partir desta inspiração, foram construídos os protótipos iniciais de baixa fidelidade baseados no uso do Kinect como interface de controle. Os resultados podem ser vistos na seção 7.

7.2. Protótipo em Papel

O primeiro protótipo preliminar foi feito utilizando papel e o seu funcionamento foi simulado através de um pequeno filme em *stop motion*. Como se encontra na Figura 7.1, tratava-se de uma interface baseada em um guia sequencial de ajuda, na qual o usuário passava pelos estágios de configuração e chegava a um ambiente onde poderia conectar, desconectar e editar os elementos previamente escolhidos.

Apesar de bastante rústico, este protótipo serviu como um estímulo inicial para a construção do sistema atual, além de permitir uma melhor comunicação com potenciais usuários e especialistas. Foi com este protótipo que se percebeu a necessidade de divisão entre variáveis contínuas e variáveis discretas e a importância dessa divisão para um melhor entendimento da interface por parte do usuário.

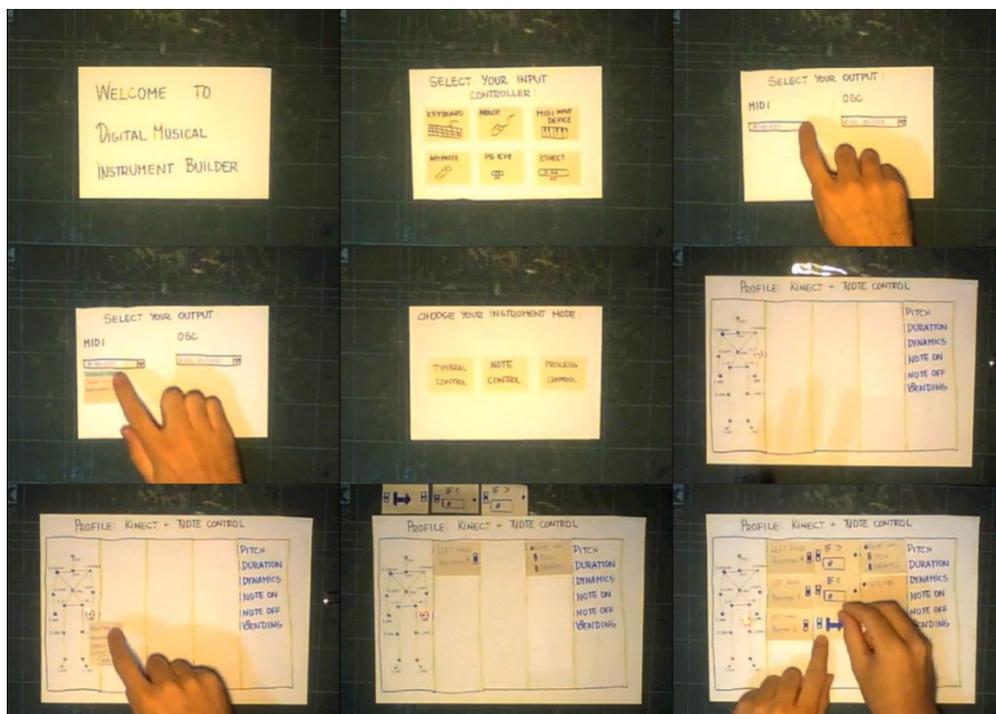


Figura 7.1: Primeiro protótipo feito em papel e animado através de um vídeo em *stop motion*¹⁸

¹⁸ <http://youtu.be/IUA5dDNZmdM>

7.3. Protótipo em Keynote

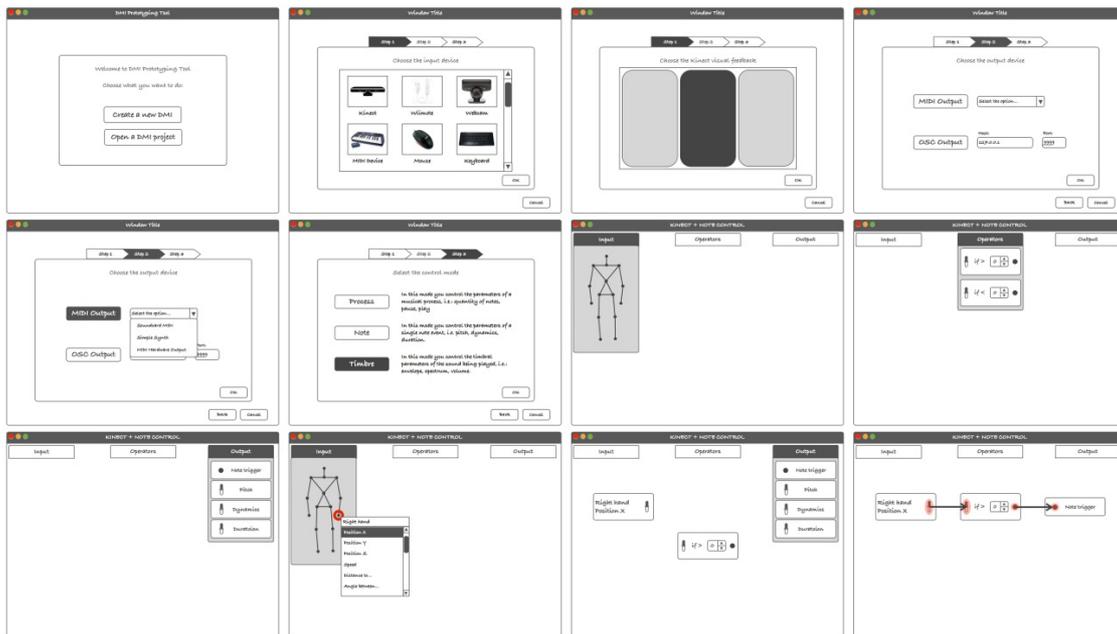


Figura 7.2: Protótipo feito em Keynote¹⁹

O protótipo apresentado na Figura 7.2 foi desenvolvido no software Keynote (voltado para apresentação de slides) e contou com recursos interativos simulados baseados em links entre os slides da apresentação. Com ele, a interface foi experimentada de forma interativa e se percebeu um maior entendimento por parte de alguns potenciais usuários e especialistas. Dos feedbacks obtidos, constatou-se certa confusão em relação aos conectores dos objetos. Muitos afirmaram que não entendiam o que eles significavam ou como poderiam ser conectados. Também se queixaram da classificação do instrumento em Processo, Nota ou Timbral – forma de classificação de DMIs – alegando que um instrumento poderia assumir mais de uma dessas categorias ao mesmo tempo mas, do jeito que estava organizado no protótipo, não permitia tal tipo de construção múltipla.

7.4. Protótipo em Processing

Com o intuito de coletar resultados mais completos e concretos sobre a interação em tempo real, um protótipo funcional foi desenvolvido utilizando a linguagem de programação Processing, com a tecnologia Kinect (Microsoft Corporation, 2012) como interface de entrada (ver Figura 7.3).

¹⁹ http://youtu.be/OQ8Gr8n0_7s

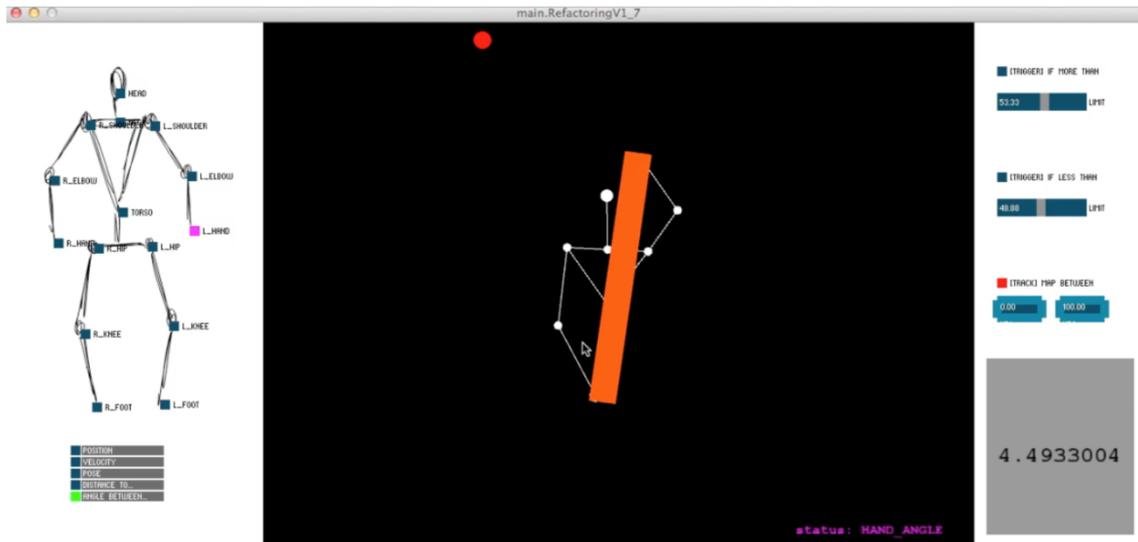


Figura 7.3: Protótipo implementado em Processing usando Kinect como interface de entrada²⁰

Durante avaliações informais, os usuários reclamaram da sensação de atraso devido a uma latência excessiva presente na interface. Como latência e atraso contradizem a diretriz de feedback, como uma decisão de projeto, a interface de entrada foi alterada para o iPad (Apple Inc., n.d.), escolhido por apresentar uma tela multitoque com dimensão razoável, boa precisão, baixa latência e um ambiente de desenvolvimento produtivo, além de ser bastante popular no meio musical.

²⁰ <http://youtu.be/JghnqJSF-9c>

8. Análise de Aplicativos

Tendo em vista a mudança de interface de entrada, realizou-se uma análise detalhada de aplicativos voltados para o iPad. Neste Capítulo, serão listados os aplicativos musicais analisados e cujas funcionalidades são importantes como inspiração para funcionalidades que os protótipos de DMIs, desenvolvidos no Sketchument, devam ter.

8.1. Aplicativos Musicais

A lista de aplicativos analisados²¹ foi a seguinte: GarageBand, KORG iMS-20, KORG iKaossilator, KORG iElectribe, Reactable mobile, Figure, Polychord, Orphion, MorphWiz, ThumbJam, SoundPrism, AmpliTube, TC-11, CamBox, MadPad, Beatsurfing, TouchOSC, Everyday Looper, DM1, TNR-i, Rj Voyager, SingingFingers e AirVox. O resultado da análise de funcionalidades será apresentado na Seção 9, no formato de uma caixa morfológica.

8.2. Aplicativos Concorrentes

São considerados concorrentes do Sketchument aplicativos desenvolvidos para plataforma iOS (sistema operacional do iPad) que funcionem baseados em módulos musicais editáveis e interconectáveis voltados para experimentação e construção de instrumentos ou interações musicais na tela multitoque.

Analisando a lista de aplicativos, destacam-se quatro principais concorrentes para análise, utilizando como critério as diretrizes do Projeto listadas anteriormente. Seguem abaixo uma descrição de cada concorrente e um quadro comparativo, resumindo os resultados e relacionando-os com a proposta do Sketchument.

8.2.1. Tabletop

Este aplicativo exclusivo para iPads é um ambiente modular que simula um estúdio de gravação (ver Figura 8.1). Possui 28 módulos (11 objetos já vem com o aplicativo e mais 17 podem ser comprados de dentro do aplicativo) que simulam dispositivos presentes em estúdios de gravações e setup de DJs, como *mixers*, *samplers*, sequenciadores, efeitos, gravadores de áudio, teclados MIDI etc. As conexões de entrada e saída de áudio são simuladas por fios que conectam os módulos e podem ser feitas ativando-se o modo de edição de conexões.

Tomando as diretrizes de feedback e usabilidade, o Tabletop permite que os atributos de cada módulo sejam alterados diretamente na representação do módulo na tela, não precisando

²¹ Maiores informações sobre os aplicativos podem ser encontradas em <https://itunes.apple.com/br/app/>

entrar em nenhum modo de edição, o que torna a edição bastante ágil. Os resultados podem ser obtidos de forma imediata a partir da conexão e disparo de algum controle. Os efeitos são facilmente ativados, modificados ou desativados, obtendo-se feedback sonoro em tempo real.

Sobre a integração, o Tabletop implementa a tecnologia WIST²² (*Wireless Sync-Start Technology*), que permite enviar para outros aplicativos ou dispositivos o momento que determinado controle é ativado ou desativado. É uma conexão limitada, porém torna possível o seu uso em um setup já existente. Além desta forma de integração, pode-se citar a possibilidade de compartilhamento com o SoundCloud, a implementação do AudioCopy e a exportação de arquivos WAV. Na versão analisada, o protocolo MIDI não foi implementado.

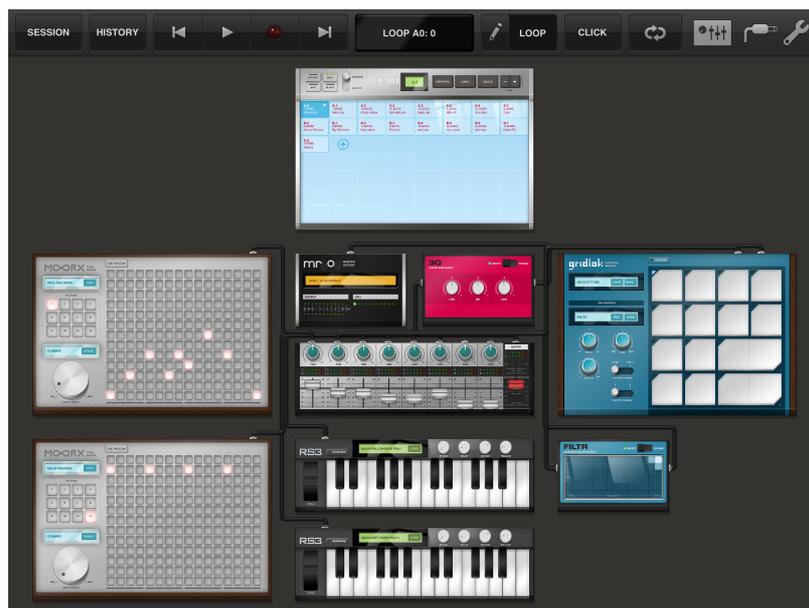


Figura 8.1: Aplicativo Tabletop²³

Como possibilidades de customização, cita-se o módulo de gravação utilizando a entrada do microfone do iPad, além do aplicativo permitir importar músicas da biblioteca do dispositivo. Apesar de presentes, podem-se considerar limitadas tais opções de customização.

Percebe-se ausência do uso de gestos nas interações com o Tabletop. O aplicativo se baseia completamente no paradigma WIMP, sendo, inclusive para funções simples como zoom in e zoom ou, necessário tocar em um botão na tela.

Apesar de julgar a sua forma de ativação inadequada para a realidade de uma superfície multitoque, o zoom da área de edição potencializa bastante o uso da tela, pois alterna entre a visão do todo e os detalhes de cada módulo. Por outro lado, as conexões se mostram um

²² <http://www.korguser.net/wist/>

²³ <https://itunes.apple.com/us/app/tabletop/id436080882?mt=8>

pouco confusas, pois fios ficam sobrepostos dificultando a identificação. Só é possível identificar as conexões ativando o modo de conexão, o que se configura um passo a mais.

Para quem já está acostumado com os dispositivos existentes, as simulações do aplicativo são triviais de se entender. Porém, quem não tem este conhecimento, pode encontrar dificuldades para obter resultados interessantes.

8.2.2. Reactable mobile

Esta é a versão simplificada do Reactable, descrito na Seção 4.1.4. Para a versão do iPad (ver Figura 8.2), os objetos tangíveis foram substituídos por imagens na tela multitoque e adicionados alguns módulos de entrada referentes à plataforma do iPad, como o acelerômetro.

Os possíveis objetos estão posicionados em um menu retrátil no canto inferior da tela. Para acondicionar todos os objetos neste menu, existe a possibilidade de rolá-lo.

Um dos pontos negativos da tradução do Reactable para a versão iPad foi perder o feedback háptico dos objetos tangíveis.

Existem 14 categorias de módulos disponíveis no Reactable mobile. São elas: oscilador, *loop player*, *sampler*, entrada de microfone, filtro, *delay*, modulador, *wave shaper*, LFO, sequenciador, acelerômetro, controle de parâmetros da música, da saída sonora e da tonalidade.

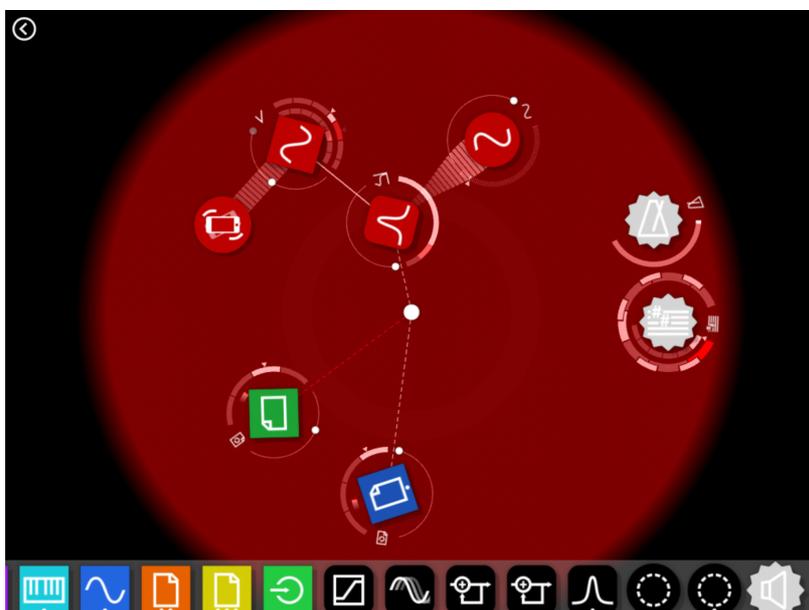


Figura 8.2: Aplicativo Reactable mobile

A edição de cada módulo pode ser ativada com dois toques no objeto, seus parâmetros modificados e o feedback sonoro obtido em tempo real. A única forma de integração possível com outros sistemas é através do áudio exportado, o que torna o aplicativo bastante limitado para se integrar a setups pré-existentes. Do ponto de vista de customização, os loops próprios

do usuário podem ser importados através de um servidor web disponível no aplicativo ou através do envio pelo iTunes.

Sobre a usabilidade, foram considerados os feedbacks relativos às funcionalidades dos módulos reduzidos, dificultando o entendimento em um primeiro momento. O método de tentativa-e-erro leva a alguns resultados sonoros, porém as conexões não são tão intuitivas. Não se sabe em um primeiro momento quais conexões são possíveis e quais fazem sentido de serem feitas, tornando demorado o domínio do aplicativo por parte do usuário.

8.2.3. Beatsurfing

Beatsurfing (Figura 8.3) é um controlador MIDI voltado para superfície multitoque cuja principal inovação é permitir que o usuário organize sua própria interface, posicionando os objetos e definindo valores e ações para eles. O nome Beatsurfing está relacionado com a funcionalidade de o usuário poder “surfear” entre os objetos, ou seja, disparar as ações associadas a cada objeto apenas arrastando o dedo sobre eles.

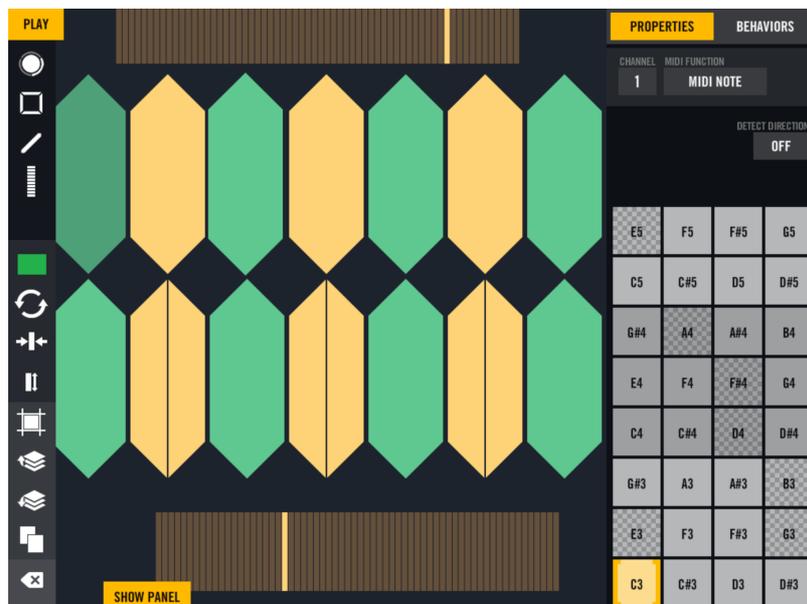


Figura 8.3: Aplicativo Beatsurfing

A conexão entre o Beatsurfing e um sintetizador MIDI é feita através da biblioteca CoreMIDI, que permite conexões MIDI com o computador através de uma conexão WiFi, além de conseguir enviar mensagem para outros aplicativos do iPad que tenham a função de receber mensagens MIDI mesmo estando em background.

O Beatsurfing possui quatro tipos de objetos de entrada, que nada mais são do que botões e *knobs* cujas características, como quantidade de divisões, tamanho, cor e posição, podem ser editados. A edição é facilitada pela presença constante de um menu lateral contendo todas as opções de modificação.

Por sua natureza de ser um controlador, o aplicativo não possui uma forma de geração de som, que faz com que seus resultados sonoros sejam dependentes de outro aplicativo. No aspecto de resultado imediato, o Beatsurfing é considerado como limitado, pois precisa fazer uma conexão com outro aplicativo para se obter o resultado sonoro. A única forma de integração com outros sistemas é através da saída MIDI implementada utilizando-se o CoreMIDI.

Do ponto de vista de customização, o aplicativo permite a mudança de cores seguindo uma paleta de cores pré-definida e a alteração do tamanho dos objetos, o que pode ser restritivo para o usuário.

Sob o ponto de vista de adequação à interface multitoque do iPad, o Beatsurfing inova ao implementar o conceito de “*surfing*” por sobre os objetos, fazendo com que eles disparem suas ações assim que o toque é identificado. Além disso, alguns botões podem identificar se o toque veio da direita para esquerda ou vice-versa, disparando ações distintas para cada ação. Porém, as possibilidades de gestos são restritas a estas interações, que, apesar de inovadoras, ainda são bastante limitadas.

Apesar de usar ganchos culturais como o conceito de botão e knobs, a interface do Beatsurfing é um pouco complicada quando usada pela primeira vez. O fato do objeto estar associado sequencialmente a notas MIDI se torna bastante restritivo. Além da ausência de feedback sonoro intrínseco ao aplicativo, limita a agilidade para a experimentação.

8.2.4. Audulus for iPad

O Audulus (Figura 8.4) é um processador musical modular voltado para estudantes, *hobbystas* e profissionais de música, que permite construir sintetizadores e processar áudio em tempo real. Ele conta com uma versão para plataforma Mac e outra para iPad e foi pensado para que os projetos sejam independentes de plataforma. Ou seja, um projeto pode ser criado no iPad e aberto na versão para Mac e vice-versa. É um programa com grande apelo visual, possibilitando que os mais de trinta módulos sejam facilmente ligados a outros através de conexões representadas por linhas de fluxo e de controle. Tem uma grande integração com MIDI, permitindo o uso de controladores cujos disparos podem estar relacionados a eventos no programa.

O Audulus é voltado para o baixo-nível da manipulação sonora, trabalhando com síntese e processamento de áudio. Dentre os seus módulos podem ser encontrados filtros passa-baixa, passa-alta e efeitos como delay e distorção, além de osciladores com os quatro formatos clássicos de onda: triangular, senoidal, quadrada e dente-de-serra. A sua versão para iPad infelizmente não utiliza o potencial da tela multitoque para disparar eventos dentro do aplicativo, ou seja, não utiliza nenhum tipo de manipulação gestual para disparos, contando apenas com um teclado na tela. Isso é uma das principais críticas feitas ao aplicativo, que tem um grande potencial para a experimentação, por ser modular e permitir a conexão de forma fácil e intuitiva.

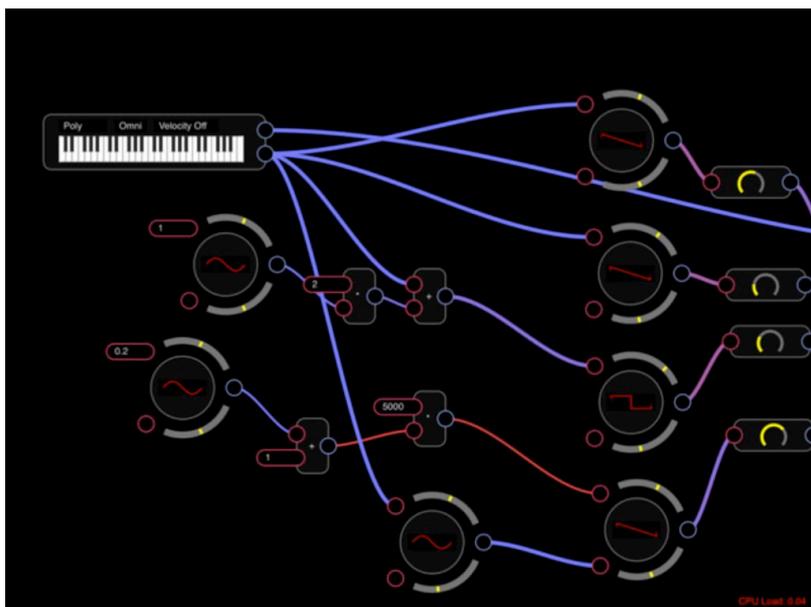


Figura 8.4: Aplicativo Audulus for iPad

Quanto à usabilidade, a interface mostra-se bastante intuitiva desde o primeiro momento de uso, com diversos feedbacks visuais que facilitam a sua utilização.

8.2.5. Quadro Comparativo

Este grupo de concorrentes foi analisado seguindo as diretrizes do projeto, com a adição de uma diretriz, que é intrínseca à interface multitoque do iPad, e que é apresentada na literatura como um recurso importante para a aplicações musicais nesta plataforma (McGlynn et al., 2012): o uso de gestos para controlar variáveis musicais.

Em relação à usabilidade, em geral, os aplicativos apresentaram interfaces organizadas. Aqueles cujas conexões entre os módulos são explícitas são mais fáceis de entender. Os módulos, geralmente, estão associados a ícones autoexplicativos, que ajudam o usuário a memorizar a sua função. Todos os concorrentes têm exemplos introdutórios, a fim de mostrar aos usuários como utilizar o sistema e a maioria tem guias de ajuda rápida dentro do aplicativo ou um link para documentação externa.

Sobre o feedback, pode-se ver que as animações são comumente usados para mostrar ao usuário o funcionamento das conexões em tempo real. Em todos os testes, a latência é baixa ou imperceptível. Além disso, os aplicativos que produzem seu próprio áudio permitem uma experimentação mais ágil, basicamente, porque o usuário não é dependente de outro aplicativo ou dispositivo para ter o feedback sonoro.

A diversidade foi objetivamente medida por meio da quantidade dos módulos fornecidos pelo sistema.

Quanto à diretriz de integração, foram destacados os recursos dentro do aplicativo que permitem integrá-lo com outros sistemas.

A diretriz de configuração foi analisada a partir das propriedades dos módulos que podem ser modificadas pelo usuário, por exemplo, cor, posição e tamanho.

Devido à distribuição online da App Store e a maneira fácil de acessar e instalar todos os aplicativos desta plataforma, a disponibilidade de tais aplicativos já é, em sua essência, alta. Assim, o preço foi escolhido como critério de comparação da disponibilidade de cada aplicativo analisado.

O Quadro 8.1 resume a comparação objetiva entre os concorrentes. Analisando isso, vemos que a maioria dos aplicativos tem um número expressivo de módulos (dúzia ou mais) e todos os concorrentes têm pelo menos uma forma de integração com outros sistemas, sendo MIDI a mais comum (nenhum permite conexões OSC). Todos os aplicativos permitem o acesso às configurações de módulos, mas poucos lidam com a personalização aparência (cor, tamanho etc.).

Sobre disponibilidade, todos os preços analisados são altos em comparação com os preços mais comuns da App Store, que são de US\$ 0,99 a US\$ 1,99.

Quanto ao controle gestual, apenas três concorrentes usam gestos, que são usados de forma simples ou limitada. Basicamente, os gestos são utilizados no modo de edição, e não durante o modo de execução para controlar as variáveis musicais.

Quadro 8.1: Quadro comparativo de concorrentes

Aplicativos Concorrentes	Usabilidade	Feedback	Diversidade	Integração	Configuração	Disponibilidade	Gestos
Tabletop	- Exemplos; - Guia para usuário; - Conexões gráficas; - Ícones representativos;	- Som próprio; - Widgets reativos;	28	- MIDI in; - Music Library access; - SoundCloud; - WAV export; - AudioCopy; - WIST;	- Propriedades específicas dos Módulos	Grátis + US\$ 13 módulos opcionais	- Nenhum gesto, apenas interação com a GUI;
Audulus	- Exemplos; - Guia para usuário; - Conexões gráficas; - Ícones representativos;	- Som próprio; - Conexões com animações;	29	- Audiobus; - JACK; - MIDI in; - Samples import;	- Propriedades específicas dos Módulos	US\$ 14	- Arrastar para mover; - Arrastar para conectar;
Reactable mobile	- Exemplos; - Guia para usuário; - Conexões gráficas; - Ícones representativos;	- Som próprio; - Conexões com animações;	14	- Samples import; - WAV export; - Download from web;	- Propriedades específicas dos Módulos	US\$ 9.99	- Arrastar para mover; - Arrastar para ativar.
Beatsurfing	- Exemplos; - Sem conexões gráficas; - Ícones abstratos;	- Sem som; - Widgets reativos;	4	- MIDI out;	- Tamanho e cor; - Propriedades específicas dos Módulos	US\$ 11.99	- Arrastar para tocar;

8.3. Aplicativos com Semelhança de Interface

Foram analisados alguns aplicativos para iPad cuja interface se assemelha com a proposta do Sketchment, ou seja, aplicativos que manipulam objetos em um canvas e realizam operações nestes objetos.

O conjunto de aplicativos com interface semelhante é composto pelos aplicativos concorrentes, já descritos anteriormente, com a adição do Adobe Proto, que será descrito abaixo.

8.3.1. Adobe Proto

Adobe Proto (Figura 8.5) é um aplicativo cuja funcionalidade é fazer *wireframes* para websites diretamente do iPad. Os *wireframes* são desenhados pelo usuário seguindo uma série de gestos pré-definidos. Cada gesto corresponde a um objeto a ser posicionado na tela, por exemplo, a representação de uma imagem, de uma caixa de texto, de um menu etc. Além de diagramar a interface, o aplicativo ainda permite a construção de interações simples, como clicar em um botão e levar para outro *wireframe*.

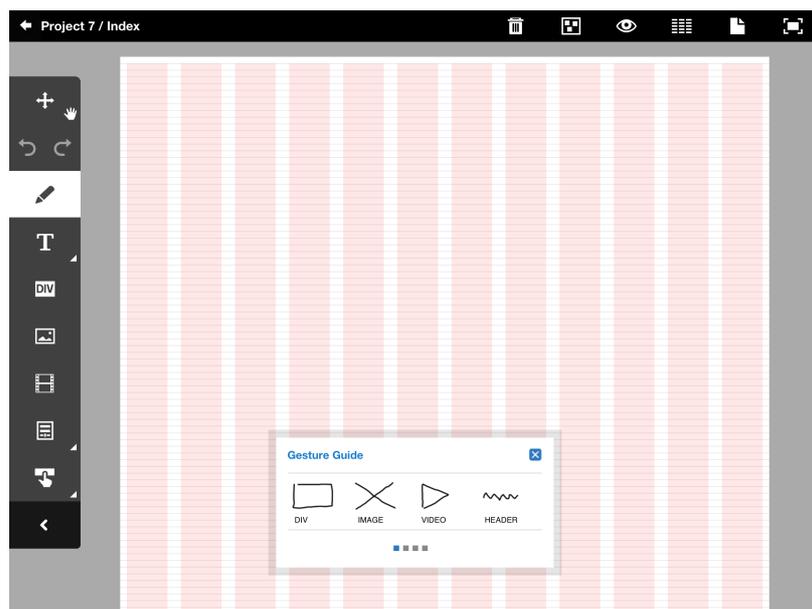


Figura 8.5: Aplicativo Adobe Proto

O aplicativo utiliza fortemente o paradigma gestual, tornando a diagramação do *wireframe* bastante ágil e natural. O usuário, porém, precisa memorizar quais os gestos para cada objeto que quer inserir ou criar. Para auxiliá-lo nesta tarefa, uma pequena janela na área inferior indica quais os gestos possíveis para cada objeto.

Sobre a análise da interface do Adobe Proto, pode-se levantar os seguintes pontos: os objetos são adicionados principalmente por gesto, ou seja, de uma lista de gestos pré-definidos o usuário executa-os na tela principal e o objeto é posicionado. Alternativamente, existe um

menu retrátil do lado esquerdo, onde os objetos podem ser selecionados e adicionados à tela principal através de um toque.

Os objetos estão sempre alinhados às linhas de grade. Na adição ou no reposicionamento, caso os objetos estejam posicionados fora das linhas de grade, o aplicativo automaticamente os reposiciona de forma correta.

A ajuda do aplicativo se baseia em balões informativos apontando para cada funcionalidade apresentada, além de um guia de gestos que se posiciona próximo à margem inferior da tela principal e alguns avisos em pop-ups quando algum gesto não é reconhecido ou alguma ação errada é disparada.

O aplicativo segue o paradigma de edição-execução, possuindo dois modos separados para cada ação. Depois de editar o *wireframe*, o usuário pode dirigir-se ao modo de execução através de um botão posicionado no canto superior direito da tela.

A análise de aplicativos com semelhança de interface levou em conta os aplicativos concorrentes, apresentados na Seção anterior, mais o Adobe Proto, apresentado aqui, e os resultados foram sintetizados em forma de uma caixa morfológica, apresentada na Seção 9.2.

9. Caixas Morfológicas

9.1. Caixa Morfológica de Funcionalidades

O Quadro 9.1 apresenta o resultado do método de Análise de Aplicativos Musicais no formato de uma caixa morfológica, na qual as características e funcionalidades são relacionadas a categorias, permitindo uma visão mais geral das opções para concepção de novas funcionalidades que os instrumentos criados no Sketchument podem implementar. Para maiores detalhes sobre as categorias desta caixa morfológica, ver o Anexo II.

Quadro 9.1: Caixa Morfológica de Funcionalidades

Definição	Looper	Sequenciador	Sampler	Sintetizador	Controlador	Máquina Generativa	Game	Drum Machine	DAW	Instrumento simulado
Plataforma	iPhone		iPad				Universal			
Níveis de Controle	Processo (Control Level)		Evento (Note Level)				Propriedades (Timbral Level)			
Funcionalidade dos Musicais	Define tempo	Define escala	Controla Loops	Transposição	Define acordes	Efeitos de Som	Metrônomo			
Interface Musical	Keyboard	Drum Pad		Step Sequencer		Smart Instrument		Interface Alternativa		
Opções de gravação	Record		Mix				Multitrack			

Edição	Editor de Bateria	Editor de Loops de Sample	Editor de Notas	Piano Roll	
Tipos de Arquivos Exportáveis	WAV	AAC	MP3	AIFF	
Tipos de Projetos Exportáveis	GarageBand		Logic Pro		
Opções de compartilhamento	Presets	Video da Performance	Audio	Projeto	
Formas de compartilhamento	SoundCloud	iCloud	Dropbox		
Conexão com outros aplicativos	WIST	CoreMIDI	AudioBus	AudioCopy/AudioPaste	OSC
Conexões com o computador	MIDI		OSC	WIST	
Conexões com outros dispositivos	Jam Session	MIDI	OSC	WIST	
Integração com hardware	Media Kit	iRig MIDI	iRig	Line 6 MIDI	AKAI Synthstation
Loops pré-carregados?	Sim		Não		

Formas de entrada	Mic Input/Plug Instrument		MIDI Input		OSC Input	
Formas de importação	iTunes	Music Library	internet	Dropbox	SoundCloud	
Produz som?	Sim			Não		
Download de presets	Sim			Não		
Customização	Gestos	Variáveis Geométricas	Objetos	Tamanho dos objetos	Disposição dos objetos	
In App-Purchase	Loops		Presets		Instrumentos	
Colaboração	No mesmo dispositivo			Em diferentes dispositivos		
Modo de Edição/Execução	Sim			Não		
Sensores como entrada	Tela Multitoque	GPS	Sensor Magnético	Acelerômetro	Câmera	

9.2. Caixa Morfológica da Interface

Da análise das interfaces de cada aplicativo, chegou-se à caixa morfológica apresentada no Quadro 9.2. Para maiores detalhes sobre as categorias desta caixa morfológica, ver o Anexo III.

Quadro 9.2: Caixa Morfológica da Interface

Categoria	Opções				
Adicionar objetos	Gestual Forma Pré-definida	Arrastar de um menu		Selecionar em um menu e tocar a tela	
Mover objetos	Arrastar				
Excluir objetos	Apertar botão que sempre segue o objeto	Arrastar para cima de uma área	Arrastar para fora da tela	Selecionar e apertar botão no menu	Ativar modo de exclusão
Editar objetos	Selecionar o objeto e tela sobreposta		Selecionar e escolher opções no menu lateral		
Editar tamanho do objeto	Pontos-guia				
Conectar objetos	Aproximar um objeto do outro	Ativar modo de conexão		Editar o objeto	
Desconectar objetos	Distanciar o objeto do outro	Arrastar uma das extremidades para um ponto vazio		Aplicar gesto na conexão	
Ajuda	Balões informativos apontando para objetos na tela	Guia apresentado na primeira execução do aplicativo	Camada de imagem posicionada por todo o aplicativo	Roteiro interativo	Exemplos
Ante-canvas	Painel de Projetos Salvos		Painel de Projetos da Comunidade		Painel de Exemplos
Modo de Execução	Ativado por botão		Sempre ligado		Misto
Detalhes de Interface	Permitir Zoom		Organizar objetos em linhas de grade		Possibilitar a escolha de cores
Menus	Contexto com barra de rolagem	Retrátil Temporizado	Retrátil por gesto	Retrátil por botão	Retrátil com barra de rolagem

10. Protótipos Iniciais no iPad

Nesta Seção, inicialmente, serão descritas as tecnologias utilizadas para o desenvolvimento dos protótipos no iPad e, posteriormente, são apresentados os protótipos iniciais implementados para esta plataforma.

10.1. Tecnologias Utilizadas

Os protótipos construídos no iPad foram desenvolvidos em Objective-C usando principalmente o framework UIKit, disponibilizado pela Apple, que provê classes responsáveis por implementar e gerenciar a interface com o usuário de uma aplicação para iOS. Com este framework, o desenvolvedor tem acesso a objetos que representam a aplicação, gerenciamento de eventos, modelos para desenho, janelas, visualizações e controles específicos para a interface multitoque (Apple Inc., n.d.).

Dentre estes controles, destaca-se o `UIGestureRecognizer`, que facilita o reconhecimento de gestos e foi muito importante para a implementação das funcionalidades deste Projeto.

Nesta Seção, serão apresentadas as tecnologias utilizadas para o desenvolvimento do sistema e a descrição das funcionalidades implementadas até a versão 3 do protótipo do Sketchument. Para maiores informações sobre a arquitetura do sistema, ver Anexo I.

10.1.1. `UIGestureRecognizer`s

O `UIGestureRecognizer` é uma classe abstrata pertencente ao framework UIKit. Ela fornece a base para a instanciação de subclasses concretas que especificam o tipo de gesto desejado, por exemplo, o gesto de tocar na tela é gerenciado pela subclasse concreta `UITapGestureRecognizer`.

Uma grande vantagem de usar tais classes de reconhecimento de gestos é que elas separam a lógica do reconhecimento de um gesto com a ação associada a este reconhecimento, pois elas encapsulam a parte de tratamento de dados brutos provenientes do sensor da tela multitoque, ao abstrair tais detalhes no momento que apenas lançam um evento referente ao reconhecimento de um padrão nestes dados. Assim, os reconhecedores de gestos permitem que o desenvolvedor se preocupe mais na adequação da ação para a melhor experiência do usuário.

Por padrão, o UIKit provê um número limitado de subclasses concretas de reconhecimento de gesto, porém o desenvolvedor pode criar seu próprio reconhecedor ao implementar uma subclasse de `UIGestureRecognizer`, especificando os detalhes de funcionamento do seu gesto.

As subclasses básicas já providas pelo UIKit são:

- **UITapGestureRecognizer**: responsável por reconhecer o gesto de toque na tela. Em seus atributos, podem ser especificadas as quantidades de toques sequenciais e toques ao mesmo tempo. Por exemplo, dois toques seguidos feitos com dois dedos ao mesmo tempo;
- **UISwipeGestureRecognizer**: responsável por reconhecer o gesto de movimentação rápida de um toque para uma determinada direção da tela. Podem ser configurados o número de toques simultâneos e a direção (acima, abaixo, à direita ou à esquerda);
- **UILongPressGestureRecognizer**: é o reconhecedor do gesto de toque longo na tela, onde uma ação é disparada assim que há um toque, permanecendo a tela sem alteração por determinado período de tempo, que é disponibilizado como atributo configurável desta subclasse;
- **UIPinchGestureRecognizer**: associado ao reconhecimento do gesto de pinça, pelo qual os dois dedos se afastam ou se aproximam. Como informação disponível para a ação, está a distância entre os dois dedos e a velocidade com que a ação é realizada.
- **UIRotationGestureRecognizer**: reconhece o gesto pelo qual dois dedos giram em torno do ponto central entre eles. Esta subclasse disponibiliza o ângulo de rotação e a velocidade com que a rotação foi executada.
- **UIPanGestureRecognizer**: é a subclasse que manipula o reconhecimento do gesto de movimentação contínua de um toque na tela. Disponibiliza a configuração dos números de toques simultâneos e provê dados como translação do toque (ou seja, a diferença em relação à última posição reconhecida), a posição absoluta do toque em relação à tela, além da velocidade de movimentação do toque.

O *UIGestureRecognizer* baseia-se na ideia que existem duas categorias de gestos possíveis: gestos discretos e gestos contínuos. Os gestos discretos estão associados a movimentos que desencadeiam uma ação específica ao término do reconhecimento. Por exemplo, quando um gesto de toque é reconhecido pelo *UITapGestureRecognizer*, uma ação é disparada após este reconhecimento. Porém, quando se trata de gestos contínuos, a ação de reconhecimento é da mesma forma disparada, mas tais reconhecedores enviam continuamente dados relativos ao movimento, como é o caso do *UIRotationGestureRecognizer*, que, enquanto o gesto continuar sendo reconhecido, ele continuamente envia o ângulo da rotação que está sendo realizada. São exemplos de reconhecedores de gestos discretos: *UITapGestureRecognizer*, *UISwipeGestureRecognizer* e *UILongPressGestureRecognizer*. São exemplos de gestos contínuos: *UIPinchGestureRecognizer*, *UIRotationGestureRecognizer* e *UIPanGestureRecognizer*.

Os *UIGestureRecognizer*s (Figura 10.1) funcionam através de uma máquina de estados. Todo reconhecimento começa com o estado “Possível”, que se refere ao reconhecimento de um toque na tela. Sendo um gesto discreto, a única possibilidade de modificação do estado possível é o reconhecimento do gesto e a passagem para o estado “Reconhecido”, que imediatamente dispara a ação associada ao reconhecimento. Para o caso de um gesto

contínuo, assim que reconhecido, o estado é definido para “Iniciado”. Assim que há uma movimentação, o estado é definido para “Modificado”, que é responsável por enviar os dados contínuos do gesto. Quando o gesto não é mais reconhecido pelo aplicativo, o estado é transportado para “Terminado”. Em gestos contínuos, o estado “Cancelado” é apenas alcançado quando o sistema operacional coloca uma ação prioritária à execução do aplicativo que está fazendo o reconhecimento de gestos, por exemplo, uma chamada telefônica no iPhone. Quando o toque inicial é reconhecido, porém não existe o reconhecimento de nenhum gesto, a máquina vai para o estado “Falho”.

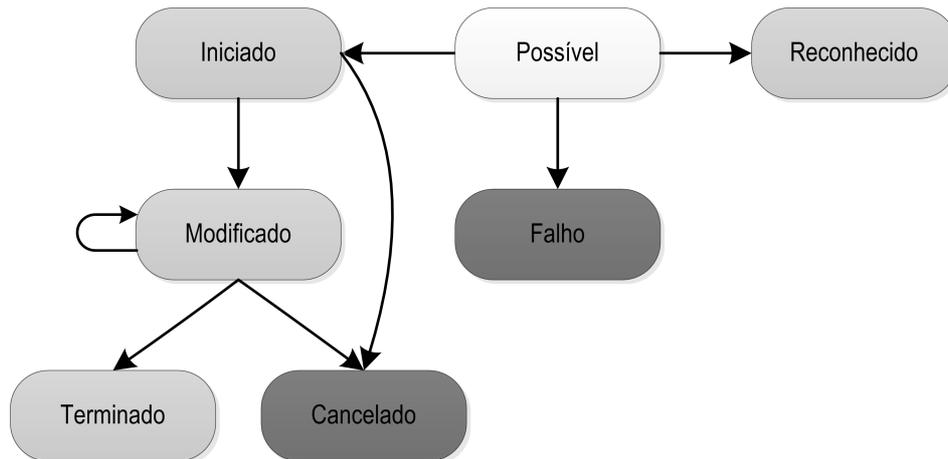


Figura 10.1: Máquina de Estados do UIGestureRecognizer

Cada UIGestureRecognizer está relacionado com uma UIView, classe que define uma área retangular na tela e gerencia as ações de desenho e animação, a diagramação e as subvisualizações e a manipulação de eventos. Para esta última ação, a UIView delega funções para o UIGestureRecognizer, registrando-as através do método addGestureRecognizer:.

10.1.2. libPD

Como foi mencionado anteriormente, o Pure Data é uma poderosa ferramenta interativa de experimentação sonora. No intuito de trazer todo o poder da prototipação sonora para o mundo do desenvolvimento e produção de softwares, foi criado o libPD. Esta biblioteca incorpora todas as funcionalidades do Pure Data, trazendo seu funcionamento para o âmbito do código de projetos de software em outras linguagens de programação além do Pure Data. O sound designer pode criar seu patch no Pure Data e encapsular todas as suas funcionalidades em projetos escritos em Java, Processing, Objective-C e Python além de poder integrá-lo a plataformas móveis como Android e iOS. Os desenvolvedores do libPD deixam muito claro que o projeto não se trata de um fork do Pure Data e sim o próprio Pure Data completamente implementado em formato de biblioteca.

Seguindo a diretriz de integração, torna-se de essencial importância a incorporação do libPD no Sketchument. Pela comunidade de Pure Data ser forte e consolidada, o usuário não

precisaria desenvolver do início um projeto no Sketchument, mas sim integrá-lo como forma de interface para um projeto já desenvolvido em PD.

Isoladamente, tal funcionalidade não é original, dado que existem já algumas iniciativas, como o RjDj²⁴, que já permite tal integração entre iOS/multitoque e patches do PD. Porém, vislumbra-se uma potencial contribuição na ferramenta Sketchument como um todo, onde o usuário pode definir sua própria interface para um projeto já consolidado em Pure Data.

A integração do código escrito em Objective-C e o patch do Pure Data se dá de forma simples através de uma classe baseada no padrão *Singleton*, intitulada *PdBase*. Com esta classe, pode-se carregar um arquivo “.pd” para o projeto em Objective-C e enviar mensagens de diversos tipos para o Pure Data, por exemplo, inteiros, pontos flutuantes, texto e especificidades como o bang. A resposta sonora é imediata, apesar de se perceber um atraso considerável quando a quantidade de mensagens é grande.

10.1.3. Core MIDI e OSC

Para implementar os objetos que lidam com protocolos de comunicação, foram utilizadas as bibliotecas CoreMIDI (framework já disponível no iOS SDK) e VVOSC²⁵.

CoreMIDI permite a comunicação do iPad e o computador através da rede sem fio, enviando mensagens em formato MIDI. Além disso, tal biblioteca apresenta a possibilidade de enviar mensagens MIDI para outros aplicativos no mesmo dispositivo. Para o primeiro cenário, foi identificado certo atraso, provavelmente devido à comunicação por rede sem fio. No segundo cenário, não houve atrasos perceptíveis.

A principal comunicação através de OSC se dá com o computador, pois poucos são os aplicativos para iPad que recebem OSC como entrada.

A escolha de implementação de objetos MIDI e OSC para o Sketchument segue as diretrizes de integração e diversidade. Além disso, foi percebido durante a análise das entrevistas e do questionário que existe uma real necessidade de integração entre novos equipamentos e interfaces com os dispositivos e sistemas já existentes no setup do músico. Sem esta característica, os novos sistemas podem se tornar isolados, já que não existe comunicação entre o novo e o existente.

²⁴ <http://rjdj.me/>

²⁵ <http://code.google.com/p/vvopensource/>

10.2. Versão 0

Esta primeira versão, apresentada na Figura 10.2, foi resultado de experimentações de como poderiam ser posicionados os elementos da interface, por exemplo, como seriam apresentados os objetos, os conectores e as linhas de conexão.

Além disso, dois botões, posicionados nos cantos inferiores, ativavam os menus de objetos de entrada e de saída, sem funções ainda. Neste protótipo, apesar de não funcionar, já foi pensado o botão de ativação do modo de execução, como se vê no canto superior direito da Figura.

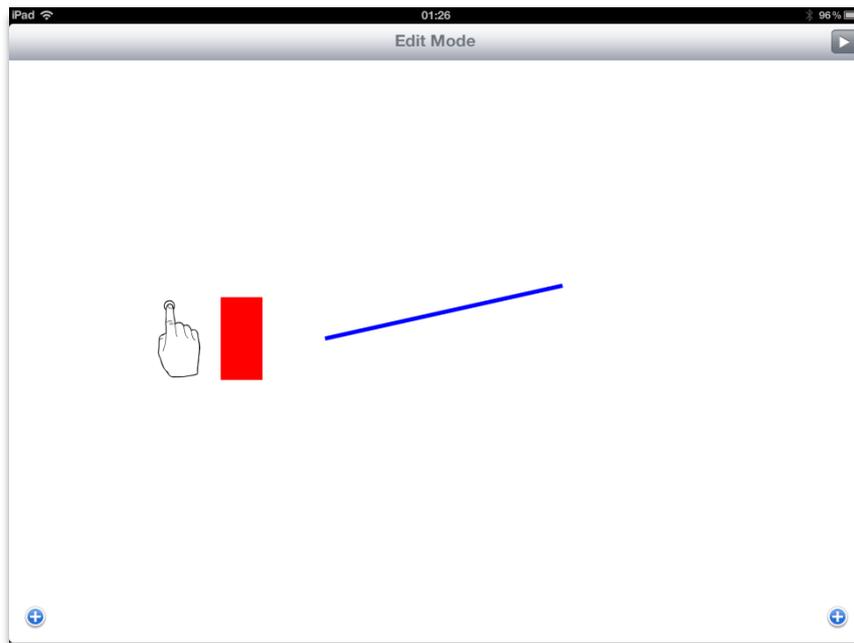


Figura 10.2: Protótipo para iPad v0

10.3. Versão 1

Nesta versão, o protótipo focou em melhoramentos da interface e o teste de conexão entre objetos de entrada e de saída. No início da execução do aplicativo, os objetos já estavam posicionados, apenas as conexões ainda não estavam feitas. Mais detalhes podem ser vistos na Figura 10.3.

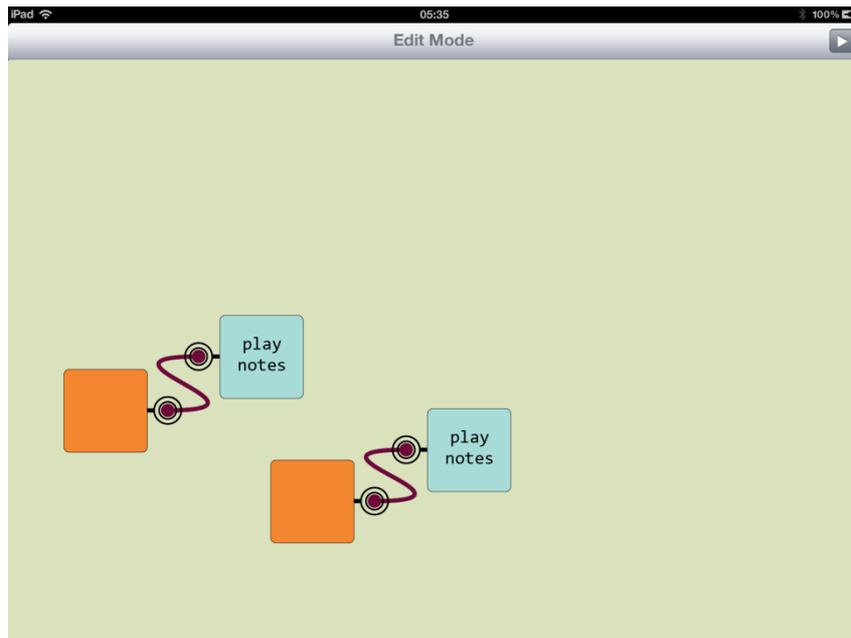


Figura 10.3: Protótipo para iPad v1

11. Protótipo Versão 2

Segue abaixo a descrição do funcionamento das diversas partes da versão 2 do protótipo.

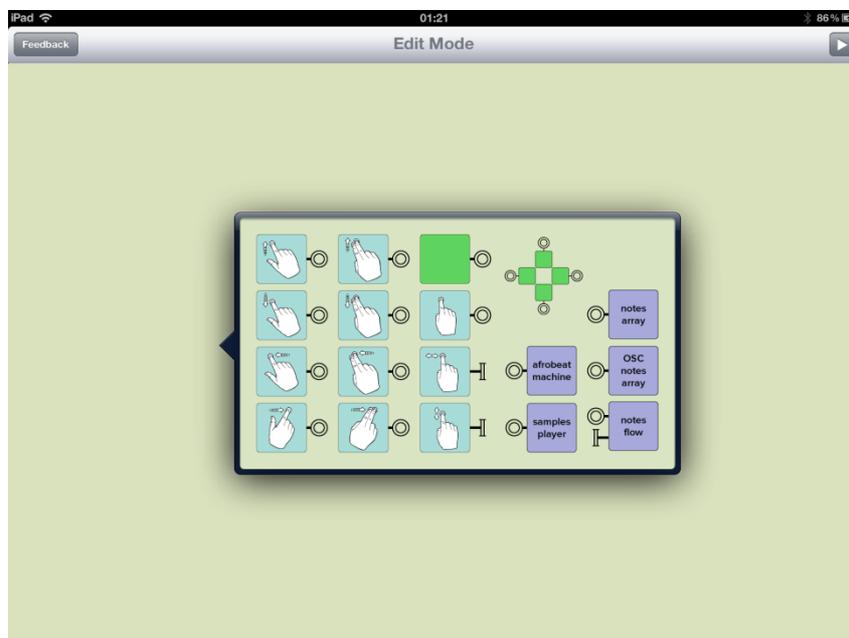


Figura 11.1: Tela de edição da versão 2 do protótipo²⁶

Na Figura 11.1, pode-se ver a tela inicial do aplicativo, que já se inicia em Modo de Edição. Vê-se na barra superior o botão com o ícone triangular (Play), com o qual o usuário pode ativar o Modo de Execução do instrumento. Neste modo, os objetos de entrada tornam-se ativos às ações do usuário.

O balão central aparece quando o usuário toca na tela e apresenta todos os possíveis objetos de entrada e saída. Objetos azuis são objetos de entrada baseados em gestos, objetos verdes são objetos de entrada baseados em toques na tela (como botões) e os objetos roxos são objetos de saída, ou seja, produção sonora ou de comunicação externa ao aplicativo (por mensagens OSC). Abaixo, cada objeto será descrito com detalhes.

11.1. Objetos de Entrada

Os objetos de entrada representam as opções de controle disponíveis para o usuário construir a interface do seu instrumento.

²⁶ <https://vimeo.com/49199339>

11.1.1. Objetos Gestuais

Os objetos gestuais representam os gestos que poderão ser selecionados para serem reconhecidos no Modo de Execução do instrumento. Para a concepção destes módulos, foi utilizada a classificação para os gestos presente na Seção 2.1.1: gestos discretos, que disparam uma ação assim que são reconhecidos e gestos contínuos, que continuamente enviam valores sobre o seu estado atual.

Na Figura 11.2, estão os módulos representando os gestos de deslizamento de um e de dois dedos na tela. Por diversidade, são apresentadas quatro direções para este tipo de gesto, permitindo que o usuário associe uma ação a cada direção.

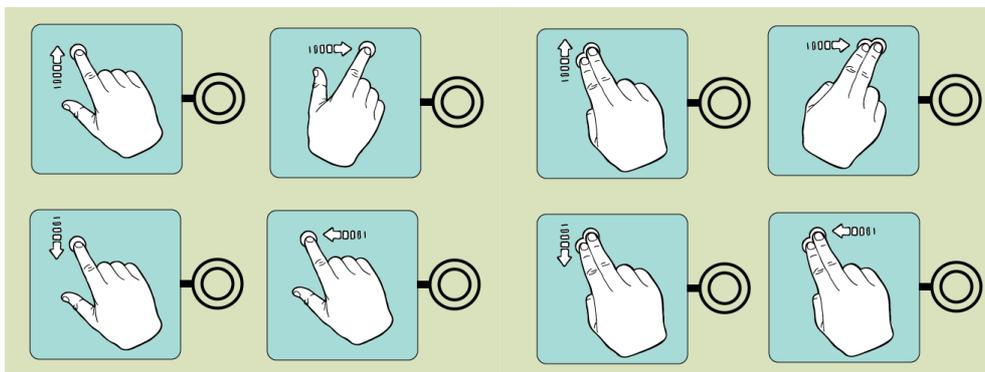


Figura 11.2: Gestos discretos de deslizamento com um dedo

O deslizamento é um gesto discreto, diferente na movimentação do dedo na tela, como é mostrado na Figura 11.3, que apresenta gestos contínuos. Para o primeiro módulo da figura, o estado de toque na tela é avaliado continuamente. Se o dedo é posicionado na tela, a ação é iniciada. Se o dedo é retirado da tela, a ação é interrompida. Os módulos restantes são responsáveis por enviar a posição do dedo no eixo X e no eixo Y, respectivamente.

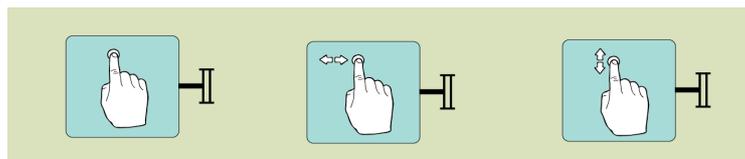


Figura 11.3: Gestos contínuos de toque e movimento pela tela

11.1.2. Objetos Tocáveis

Os objetos tocáveis (ver Figura 11.4) funcionam como botões na interface que, quando tocados, disparam ações. Um detalhe sobre estes objetos é que, além do toque em cima do botão, ele também pode ser acionado através da passagem do dedo por sobre sua área, ou seja, se o usuário estiver passando o dedo em alguma área da tela e continuar o movimento por sobre o botão, ele disparará a ação associada. Este tipo de disparo oferece mais diversidade de interação com o botão.

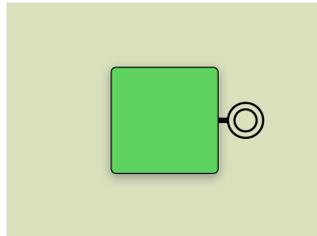


Figura 11.4: Objeto tocável

Seguindo a diretriz de customização, a funcionalidade de redimensionamento do botão foi implementada. Para isto, basta que o usuário posicione o dedo no objeto e espere até que o modo de redimensionamento seja ativado. Um ícone representando o redimensionamento aparece e movendo-o, o tamanho do botão é alterado. Na Figura 11.5, encontra-se a sequência de passos de um redimensionamento.

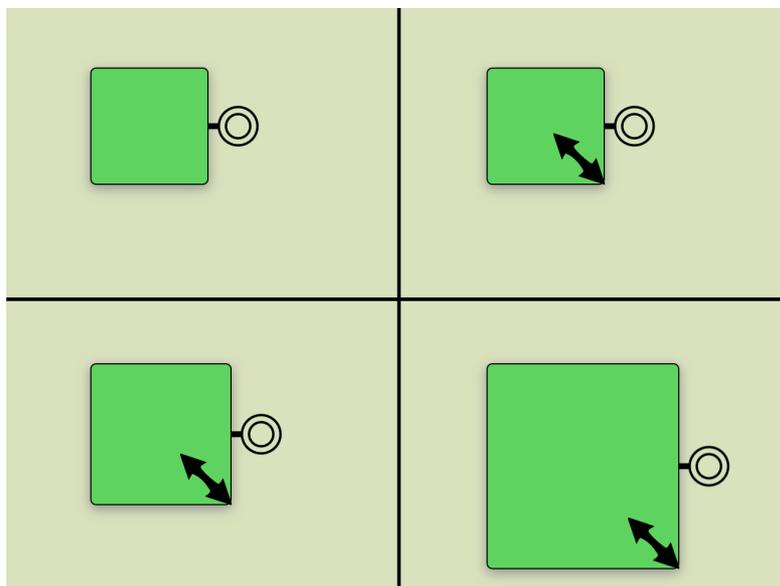


Figura 11.5: Redimensionamento do objeto tocável

No Protótipo v3, é apresentado apenas o botão, ou objeto tocável, associado a um conector discreto, podendo disparar ações com o toque ou com a passagem do dedo por sobre sua área. Porém, pode-se pensar, para futuras versões, em um botão com um conector contínuo cujo comportamento seria o seguinte: se o usuário está com o dedo em cima do botão, a ação é iniciada; se o usuário retira o dedo, a ação é interrompida, semelhante ao funcionamento do objeto gestual contínuo apresentado na Figura 11.3.

11.1.3. Objetos Tangíveis

Preocupando-se com a ausência de feedback tátil nos instrumentos digitais, descrito pelos músicos como um dos pontos fracos destes instrumentos (Magnusson & Mendieta, 2007), joysticks para tela capacitiva foram escolhidos como forma de fornecer tal feedback para o usuário.

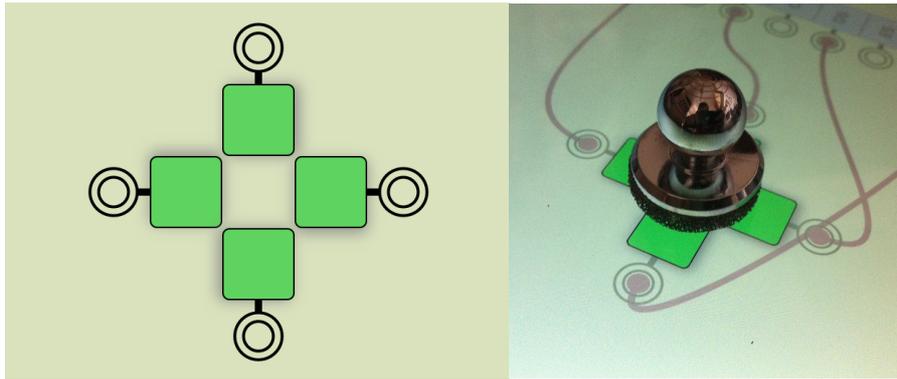


Figura 11.6: Agrupamento de objetos tocáveis para joystick

Para integrar de forma rápida à interface do Sketchument, alguns objetos tocáveis foram agrupados no tamanho adequado para o posicionamento dos joysticks, conforme Figura 11.6. Após alguns testes realizados com o protótipo v2, que indicaram falhas de funcionamento destes joysticks e com isso uma não aceitação pelos usuários, houve a decisão de retirá-los da versão 3 do protótipo para se aprofundar os estudos quanto aos problemas levantados.

11.2. Objetos de Saída

Os objetos de saída representam os módulos de geração sonora do ambiente de experimentação. Possuem conectores que permitem a associação de suas ações com disparo ou ajuda dos objetos de entrada. Abaixo, estão descritos os objetos de saída já implementados até agora.

11.2.1. Conjunto de Notas

Este objeto de saída, conforme Figura 11.7, apresenta um conjunto de notas que podem ser associadas a disparos provenientes dos objetos de entrada.

Na versão 3 do protótipo, o timbre dos sons gerados se assemelha ao timbre do piano e, por enquanto, não é customizável. Para esta versão, tal requisito foi deixado em segundo plano, pois a prioridade residiu em entender melhor a interação para o mapeamento entre entradas e saídas antes de lidar diretamente com a qualidade do som.

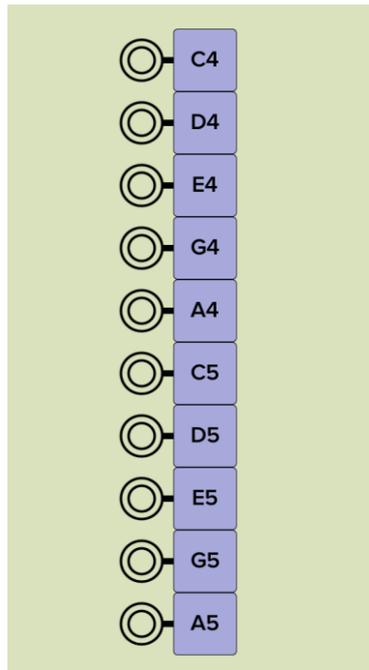


Figura 11.7: Conjunto de notas

11.2.2. Disparador de *Samples*

Entende-se por *sample*, uma porção de gravação sonora armazenada em um arquivo individual. Por exemplo, se for gravada uma bateria e no final cortarmos apenas alguns compassos e salvarmos em um arquivo individual, tem-se um *sample* de bateria.

O objeto de saída apresentado nesta Seção tem como principal função disparar *samples*. Na versão atual (conforme Figura 11.8), o objeto é formado por seis *samples* curtos de gravações de pandeiro, cada um relacionado a algum tipo de batida (polegar, base da mão, ponta dos dedos etc.).

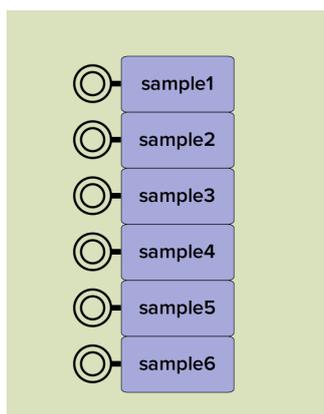


Figura 11.8: Conjunto de Samples

11.2.3. Fluxo de Notas

Este objeto (ver Figura 11.9) permite que, através de conexões contínuas, um fluxo de notas seja disparado ou interrompido, podendo tais notas sofrer alterações de duração e altura

durante a execução. Trata-se de um objeto bem simples, implementado para demonstrar as conexões contínuas.

Os sons das notas são os mesmos do conjunto de notas apresentados anteriormente, ou seja, com timbre semelhante ao piano. Um exemplo de ligação para o protótipo v2 é conectar o movimento vertical do dedo na tela à altura (*pitch*) e o movimento horizontal à duração (*duration*). Para iniciar o fluxo de notas, deve-se associar o toque contínuo na tela, ou seja, se o usuário estiver tocando, as notas são executadas. Caso contrário, as notas são interrompidas. Neste exemplo, quanto mais à direita, maior é a duração das notas e, quanto mais à esquerda, menor a duração. Quanto mais para cima, mais agudas as notas e vice-versa.

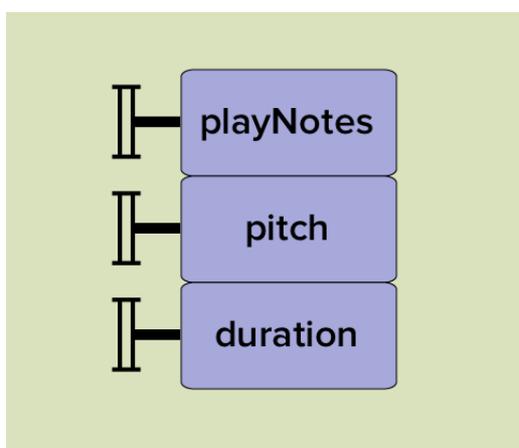


Figura 11.9: Objeto musical responsável por liberar um fluxo de notas

Por resultados não satisfatórios nas avaliações do protótipo v2 (o objeto não estava sendo compreendido facilmente pelos usuários) e pelo tempo reduzido para uma análise mais detalhada das causas dos problemas detectados, tal objeto foi retirado da versão 3.

11.2.4. Afrobeat Machine

O objeto Afrobeat Machine é uma implementação no âmbito do Sketchument do patch de Pure Data (vide Figura 11.10) desenvolvido por Jarbas Jácome²⁷. Trata-se de uma máquina de acompanhamento baseada em quatro músicas do multi-instrumentista e compositor nigeriano Fela Kuti²⁸. Além de outras funcionalidades, esta máquina permite a escolha da música a ser executada, a alteração do andamento e a ativação ou desativação dos instrumentos que estão sendo tocados.

²⁷ <http://jarbasjacome.wordpress.com/>

²⁸ http://en.wikipedia.org/wiki/Fela_Kuti

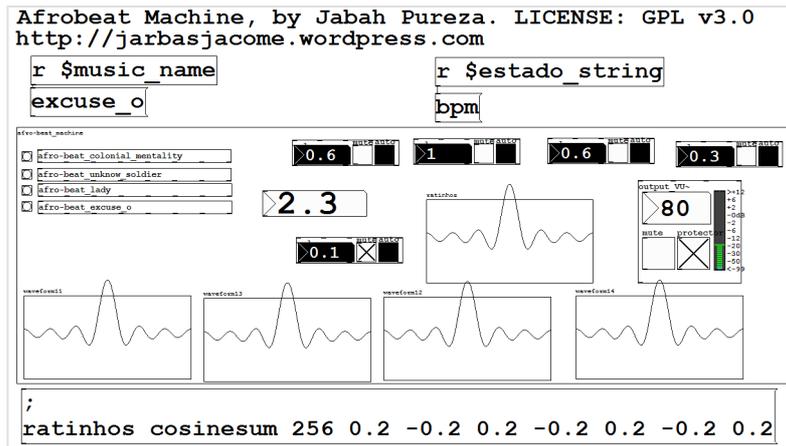


Figura 11.10: Afrobeat Machine no Pure Data desenvolvido por Jarbas Jácome

O objeto implementado no Sketchument teve como principal objetivo demonstrar a capacidade de integração do ambiente com sistemas já existentes. A única alteração feita no patch do Pure Data foi a definição de mensagens de controle em alguns objetos do Afrobeat Machine, porém nenhuma lógica teve de ser alterada para o seu funcionamento na plataforma iOS. O mérito de tal transparência e facilidade é da biblioteca libPd que permite esta integração de maneira simples.

A Figura 11.11 apresenta o objeto de saída de seus sete submódulos. Os quatro primeiros estão relacionados com a escolha da música, cada uma podendo ser disparada por uma ação distinta. Os submódulos *bpmUp* e *bpmDown* permitem a alteração do andamento da música sendo tocada. *TrackSel* faz com que, através de um disparo, os instrumentos sejam ativados sequencialmente.

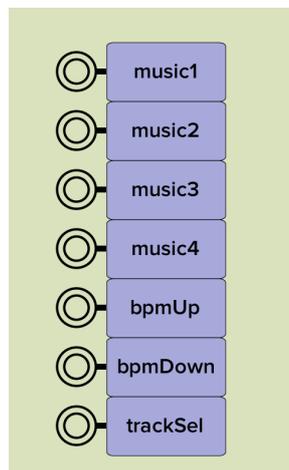


Figura 11.11: Afrobeat Machine

11.2.5. Objeto OSC

O objeto OSC (*Open Sound Control*) realiza a conexão do Sketchument com outros softwares ou dispositivos, que implementam o protocolo OSC em uma mesma rede. Podemos entender este protocolo pode ser entendido como uma alternativa ao MIDI, permitindo que softwares e

dispositivos troquem mensagens através de uma rede de computadores. Tais mensagens são transmitidas usando caracteres de texto, o que facilita a leitura e a compreensão por seres humanos.

Este objeto está relacionado com a diretriz de integração e proporciona a comunicação do Sketchument com sistemas pré-existentes que implementam o protocolo OSC. O objeto (ver Figura 11.12) ainda está em desenvolvimento e não foi oficialmente colocado na versão 3, porém já funciona com mensagens fixas: “/note1”, “/note2”, “/note3”, “/note4”, “/note5” e “/note6”, sendo enviadas para um endereço fixo de rede “192.168.1.2” e porta “8000”. A ideia é tornar estas mensagens customizáveis e com a descoberta automática na rede de dispositivos disponíveis para comunicação.

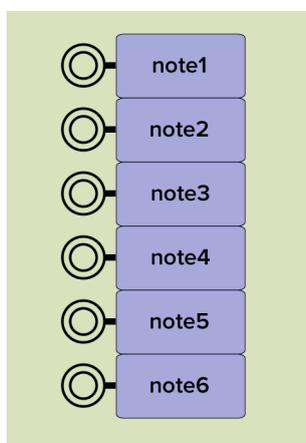


Figura 11.12: Objeto OSC responsável por enviar mensagens pela rede

11.3. Mapeamento

O Sketchument permite dois tipos principais de conexões entre objetos de entrada e objetos de saída: conexões discretas (Figura 11.13), que nada mais são que disparos provenientes dos objetos gestuais ou tocáveis; e conexões contínuas (Figura 11.14), entendidas como ajustes contínuos advindos dos controles de entrada.

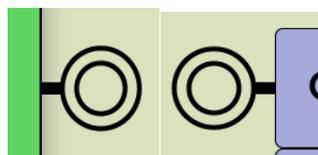


Figura 11.13: Conexões discretas

A concepção desta forma de conexão se baseou na extrapolação de classificação de gestos apresentada na Seção 2.1.1.1 e na sua interpretação como protocolo de conexão baseado no disparo (ação discreta) e no ajuste (ação contínua).

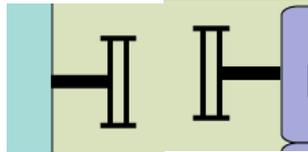


Figura 11.14: Conexões contínuas

Como forma de potencializar o entendimento do usuário em relação às conexões possíveis, o destaque visual foi implementado apresentando as restrições de conexão. Conforme Figura 11.15, no momento em que o usuário está tentando fazer uma conexão discreta, apenas as possibilidades discretas são destacadas.

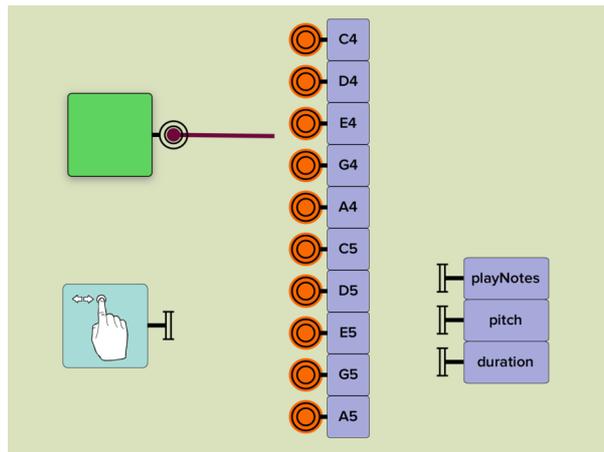


Figura 11.15: Destaque para conexões discretas

Da mesma forma, na Figura 11.16, verifica-se que, quando o usuário tenta fazer uma conexão contínua, apenas as opções contínuas são destacadas. Isto facilita o entendimento das restrições de conexões e condiz com a diretriz de usabilidade do projeto.

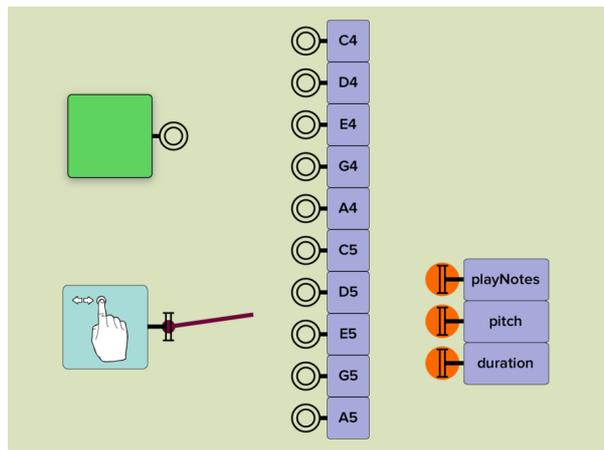


Figura 11.16: Destaque para conexões contínuas

Um objeto de entrada pode se fazer uma ou mais conexões de saída. Com isso, o disparo de um objeto de entrada pode gerar ações em paralelo. Por exemplo, para o caso de uma conexão com o conjunto de notas, um objeto pode estar associado ao disparo de um acorde, como na Figura 11.17.

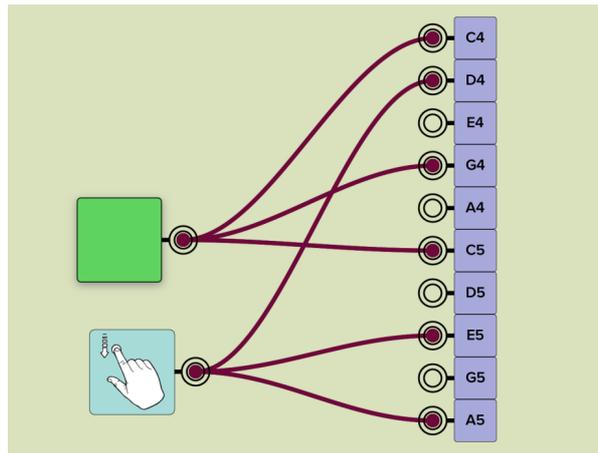


Figura 11.17: Múltiplas conexões discretas com um objeto de geração sonora

Além disso, o ambiente permite ligações do mesmo objeto de entrada com objetos de saída de diferentes naturezas. Na Figura 11.18, um mesmo objeto tocável está associado ao disparo de ações sonoras tanto do conjunto de notas, quanto do disparador de *samples*.

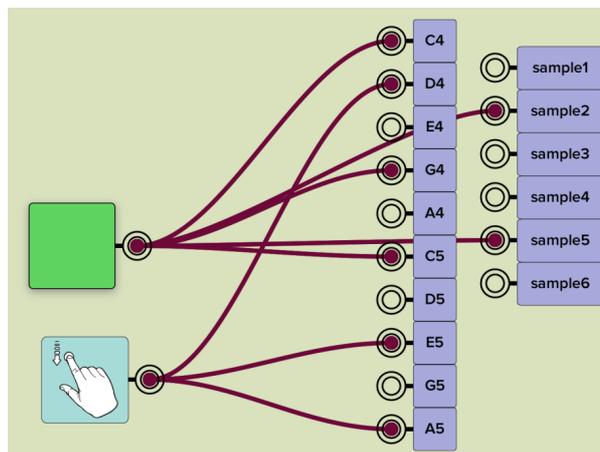


Figura 11.18: Múltiplas conexões discretas com múltiplos objetos de geração sonora

12. Questionário Aberto

Nesta etapa, um questionário online, contendo questões abertas relativas ao projeto, foi aplicado a um público de potenciais usuários do aplicativo. No cabeçalho do questionário, um link para o vídeo de demonstração da versão 2 do protótipo foi apresentado e o usuário orientado a responder às questões baseadas no vídeo.

A primeira parte do questionário tratava-se da etapa de *profiling*, onde os participantes preencheram informações sobre faixa etária, envolvimento com tecnologia digital e relação com música, como por exemplo, se tocam algum instrumento e se possuem uma relação profissional com a música. Além disso, foi perguntado se o participante já teria utilizado algum tablet com superfície multitoque.

Na segunda parte, um conjunto de questões abertas foram feitas. A sequência das perguntas segue abaixo:

- O que você achou do conceito do aplicativo: um ambiente para experimentação e criação de interações musicais com o iPad?
- Você poderia citar alguns projetos que acha que se relacionam com a ideia deste aplicativo?
- O que você achou da interface? Por exemplo: cores, formas, conexões, maneira como se edita e como se toca?
- Que perfis de usuário você acha que mais utilizariam este aplicativo?
- Em quais contextos você imagina que este aplicativo poderia ser usado?
- Você usaria este aplicativo? Em quais contextos?
- Para você, quais os pontos POSITIVOS deste aplicativo?
- E os NEGATIVOS?
- Você teria alguma SUGESTÃO de melhoramento? Alguma crítica, comentário?

As perguntas foram abertas com o intuito de não limitar a opinião do participante a um conjunto pré-definido de temas, além de permitir a emergência de pontos não pensados a priori pelo pesquisador. Com as respostas destas perguntas, pode-se compreender melhor o entendimento do usuário para com o protótipo e pensar em modificações futuras em questões que não foram bem entendidas ou que diferem do objetivo do projeto.

Em pouco mais de duas semanas de ativação do questionário online, foram recebidas respostas de um total de 87 participantes. Dentre estes, foram selecionados 31, cujo perfil se enquadrou na seguinte descrição: relação com música entre 4 e 5, numa escala de 0 a 5; toca ou já tocou algum instrumento musical; já experimentou algum tablet com superfície multitoque.

A partir das respostas selecionadas, uma análise temática foi aplicada e destacados os temas mais mencionados nas respostas abertas. Abaixo, apresentamos os gráficos relativos a cada pergunta do questionário, com os temas e a taxa que foram mencionados.

12.1. Opiniões sobre o Conceito

Sobre o conceito, pode-se verificar que um pouco mais da metade (51% ou 16 participantes) das respostas foi positiva (ver Figura 12.1), sem nenhum comentário adicional. Para o tema “Positivo” foram associadas respostas como: “Bastante interessante”, “bacana”, “Genial”, “interessante”. Este tema se caracterizou por respostas simples sem comentários adicionais. Por não ter uma profundidade, muitas destas respostas podem ser consideradas com menos impacto que as respostas positivas com ressalvas e comentários.

Quatro participantes, ou 13%, tiveram suas respostas associadas ao tema “Positivo com ressalvas à interface”, que se relaciona com a opinião de a interface não estar rebuscada o suficiente. Respostas como: “Conceitualmente é interessante... Mas me parece que precisa de refinamentos” ou “Experimental, por enquanto. Com uma interface mais amigável pode vir a instigar o usuário a criar bastante” são algumas ilustrações deste tema.

Também com quatro respostas, há o tema “Positivo com ressalvas à originalidade”, relativamente ao qual os participantes disseram que a ideia do aplicativo não é original quando comparada a outras aplicações existentes. Como se constata na resposta a seguir: “Interessante, mas não especialmente inovador (já vi outras interfaces parecidas anos atrás). Seria mais uma implementação dessas interfaces numa plataforma diferente”. Outra resposta associada a este tema foi: “Acho muito válida a ideia, mas ao mesmo tempo, acho que o foco tem que ser em fatores que diferenciam este aplicativo de outros similares”.

Para os demais temas, houve uma ou no máximo duas respostas. A seguir, tem-se o tema e um trecho da resposta associada.

Como exemplo para o tema “Positivo com ressalvas ao som”: “Quem sabe com uma boa base de áudios, um DJ não poderia criar diversos experimentos musicais simplesmente com um iPad”.

Para o tema “Positivo com ressalvas à simplicidade”: “O conceito eu achei legal, mas o funcionamento me deixou um pouco confuso”.

Quanto ao tema “Positivo com ressalvas à facilidade”: “... há algum nível de dificuldade que certamente impediria [de usar o aplicativo]”.

A gravação se relacionou com a impossibilidade de gravar sons através do aplicativo.

O tema integração teve a ver com a possibilidade de integração com *patches* já existentes do Pure Data: “Se for possível integrar com várias bibliotecas (além da afrobeat) ou fosse possível inserir a sua própria, seria maravilhoso”.

Já o tema interação se relacionou com a seguinte resposta: “Interessante, me lembrou MAX e Pure Data, mas o modo de interagir me pareceu meio travado”.

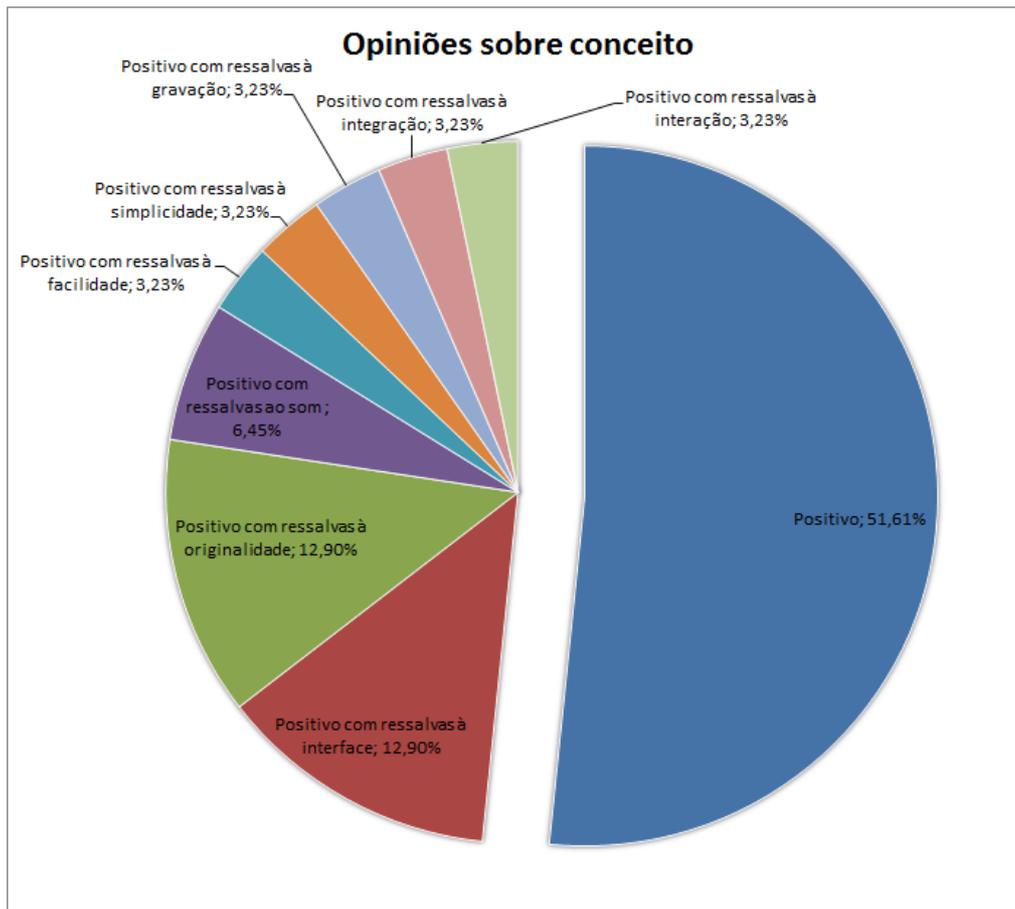


Figura 12.1: Opiniões sobre o conceito agrupadas por categorias

12.2. Opiniões sobre Projetos Relacionados

Pode-se ver que a maioria dos projetos citados (ver Figura 12.2) se relaciona de alguma maneira com o contexto do aplicativo. Mais de 23% das menções se relacionaram com sistemas modulares para música: Reactable, MAX/MSP e Pure Data. Isto significa que o conceito do aplicativo, apesar de ainda precário, de alguma forma está sendo compreendido corretamente pelos potenciais usuários.

Além de avaliar esta compreensão, a lista de projetos relacionados serve de referência para futuras pesquisas sobre estes sistemas.

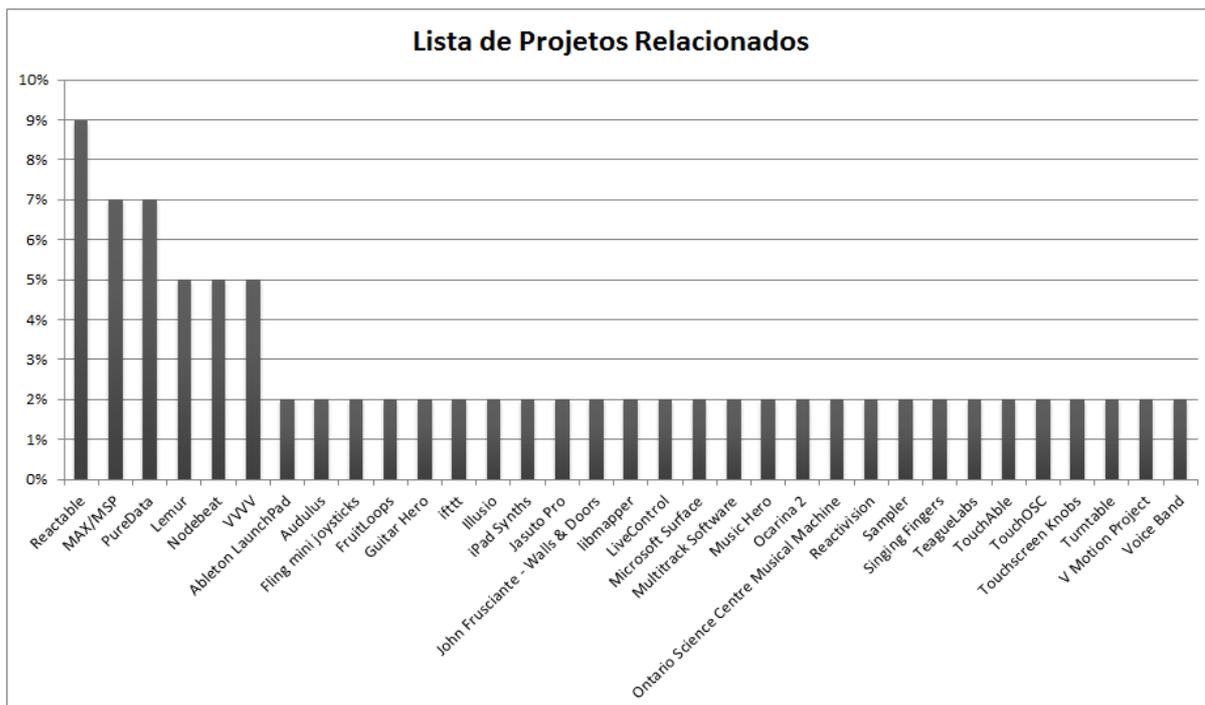


Figura 12.2: Projetos relacionados ao Sketchument mencionados no questionário

12.3. Impressões sobre a Interface

Sobre a interface, houve mais de 24% (9) de menções positivas classificando-a como intuitiva, porém 21% (8) disseram que a estética é um ponto a ser melhorado (ver Figura 12.3). Respostas como: “Graficamente acho que poderia ser mais elaborado, as cores e os detalhes não aproveitam as possibilidades do hardware” ou até mais enfáticas: “Achei que é um experimento e, como tal, o que conta é funcionar e validar a ideia, mas a interface está horrível - jamais lançaria isso num Apple Store, por exemplo” ilustram o tema do melhoramento estético da interface.

Com 13% (5) das menções, tem-se os temas: “Criar metáforas de instrumentos musicais” e “Melhorar feedback”. O primeiro tema pode ser entendido como o uso de elementos gráficos que remetam a instrumentos já existentes para facilitar a compreensão do usuário em relação ao funcionamento dos módulos do ambiente. Como ilustração a este tema, segue a seguinte frase: “Acho que o formato dos componentes deveria ser um pouco mais intuitivo. Por exemplo: se o componente é uma batida de Bateria, então ele teria um formato que lembrasse um prato”.

Sobre o tema de feedback, alguns usuários afirmaram que o sistema não está provendo a quantidade de feedbacks visuais necessários para um bom entendimento do seu funcionamento. É o que se apreende desta frase: “Acho que em alguns casos a informação visual não foi suficiente pra entender o que o gesto faz ou em que controle ele está mexendo”.

Os demais temas foram mencionados uma ou duas vezes e as ideias serão aproveitadas para a listagem final de funcionalidades do sistema, gerando material para uma posterior fase de concepção.

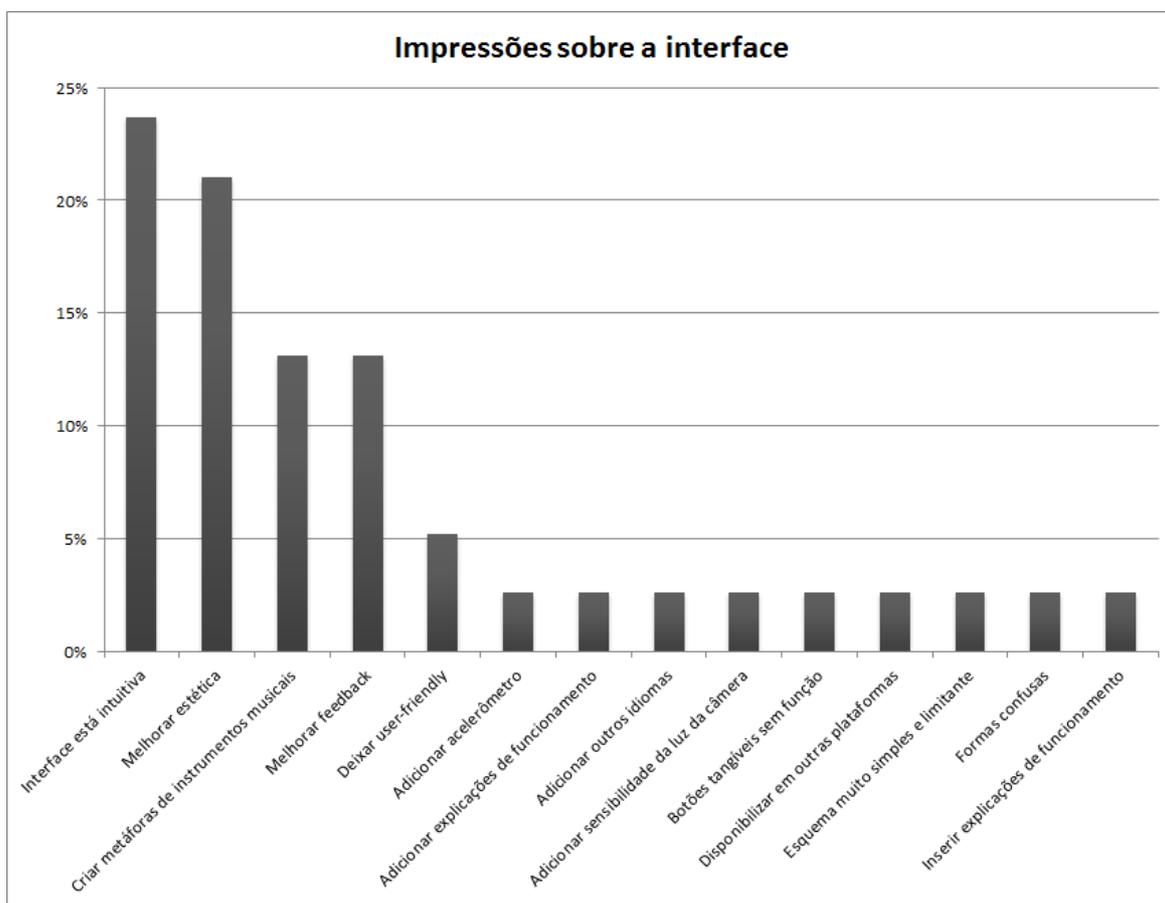


Figura 12.3: Impressões sobre a interface mencionadas no questionário

12.4. Opiniões sobre Perfis de Potenciais Usuários

Quanto aos perfis de potenciais usuários (Figura 12.4), 23% das menções foram para músicos. De certa forma, é um resultado de confirmação que o aplicativo está passando a mensagem correta, porém sem muita expressividade dado que o termo músico pode ter várias atribuições como instrumentista, intérprete, compositor etc.

Com 16% das menções, há "Curiosos", que é um tema também um tanto vago e sem expressividade semântica. Com 13% dos termos mencionados, aparece o "DJ" em terceiro lugar.

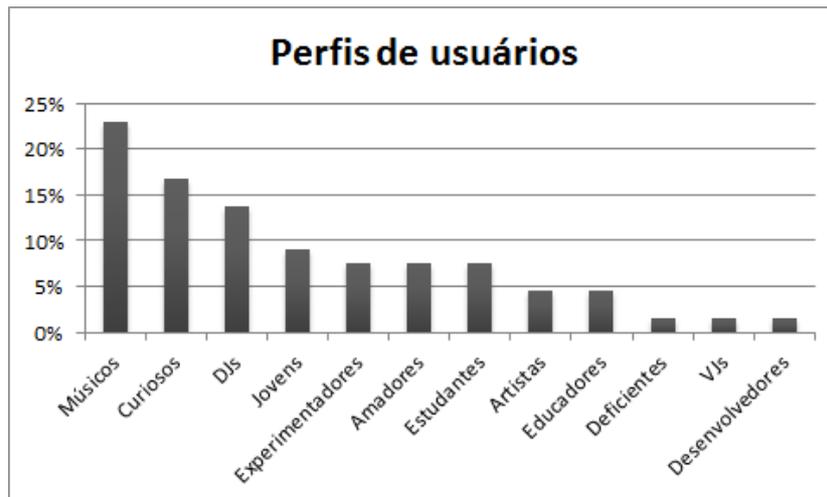


Figura 12.4: Perfis de potenciais usuários mencionados no questionário

Os dados obtidos serviram de guia para captar o entendimento dos participantes sobre os potenciais usuários e se chegou à conclusão que por um lado, a mensagem passada está certa, porém por outro, ainda há certa indefinição sobre o público-alvo do aplicativo. O resultado talvez tenha sido vago pelo caráter inovador da plataforma multitoque, ou pela interface ainda sem um tema gráfico e estético bem definido, ou pelo conceito ainda não consolidado do ambiente de experimentação.

12.5. Opiniões sobre Contextos de Uso

Sobre em que contexto de uso o Sketchument melhor se aplicaria (Figura 12.5), 23% das menções foram relacionadas ao tema “Educação”. Este tema está relacionado com o fato de se usar o aplicativo para o ensino de Música para crianças, a partir dos seguintes trechos: “no ensino de música para crianças”; “aulas de iniciação musical”; “aulas de músicas para crianças”.

Também com 23% dos itens mencionados, tem-se o contexto de uso em performances musicais: “Artistas procurando uma forma fácil e simples de soltar *samples* e gerenciar alguns elementos digitais em performances ao vivo”.

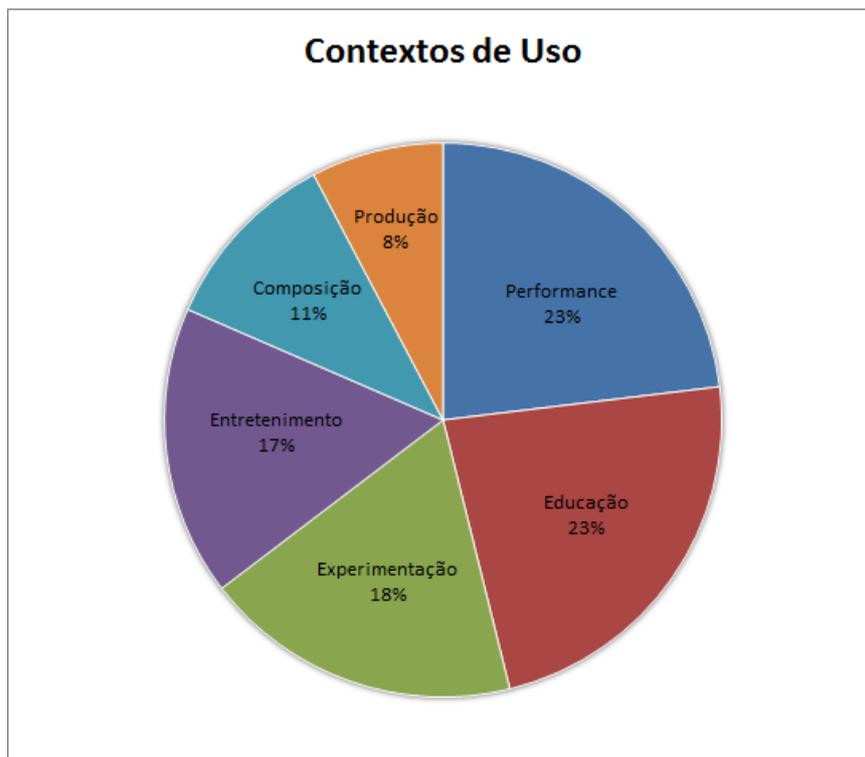


Figura 12.5: Contextos de uso agrupados por categorias

Quando perguntados se usariam este aplicativo em algum contexto, 84% dos participantes responderam que sim (vide Figura 12.6). E sobre o contexto em que mais usariam, os temas mais mencionados foram “Experimentação” e “Entretenimento” (Figura 12.7), um resultado satisfatório para a avaliação do conceito do aplicativo. Isto quer dizer que, de alguma forma, o ambiente está sendo visto como oportunidade de ser usado para a experimentação musical.

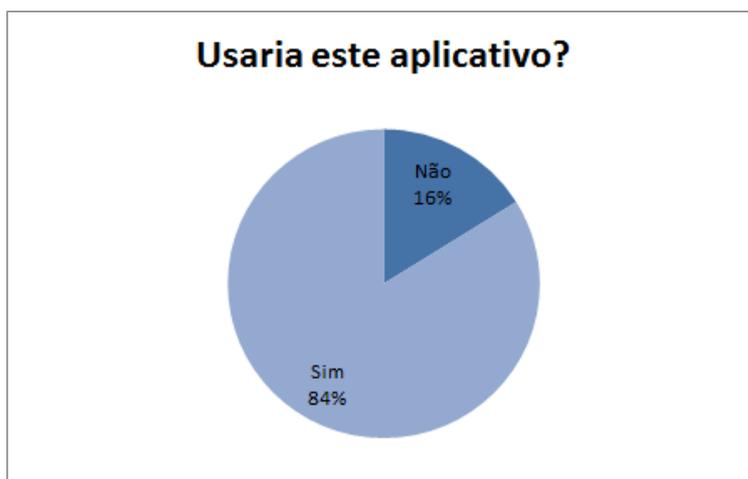


Figura 12.6: Número de participantes que utilizariam ou não o Sketchument

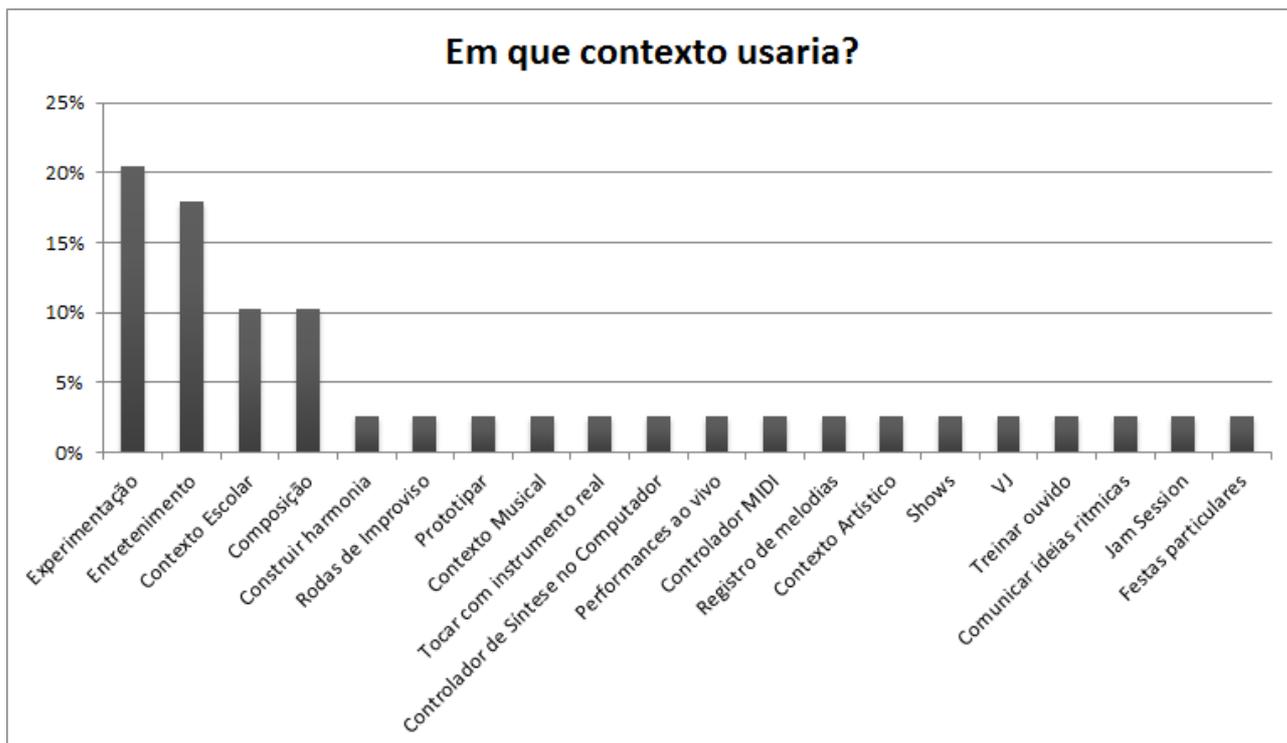


Figura 12.7: Contexto de utilização mencionados pelos participantes

12.6. Pontos Positivos, Negativos e Sugestões

Como se vê na Figura 12.8, os principais pontos positivos mencionados estão relacionados com simplicidade, facilidade de usar e adaptabilidade.

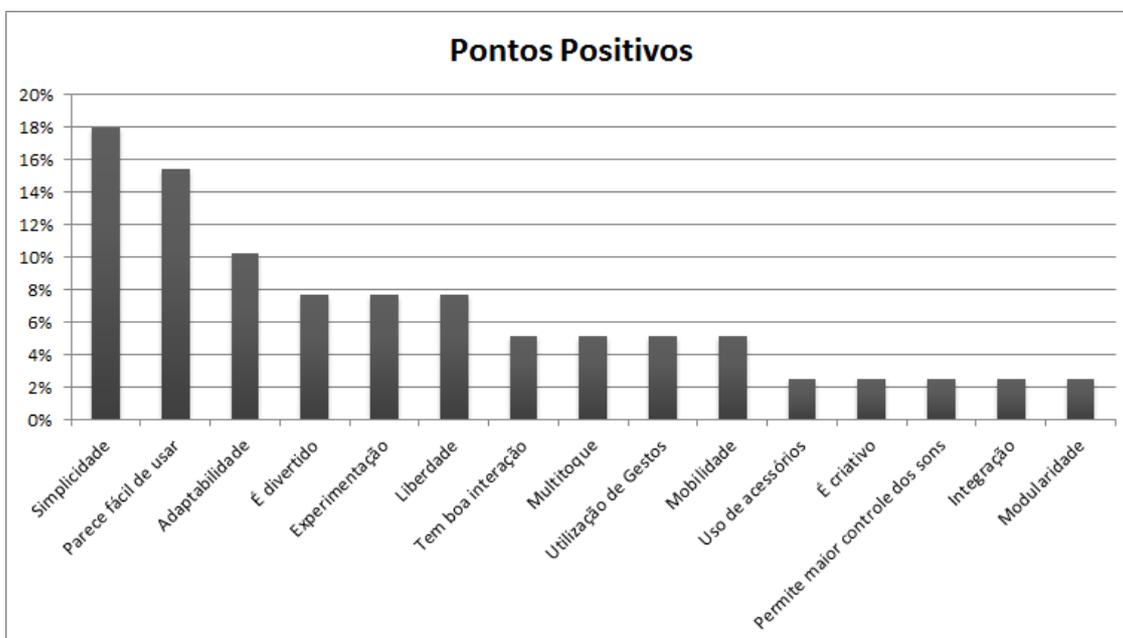


Figura 12.8: Pontos positivos mencionados no questionário agrupados por categorias

Sobre os pontos negativos, como se vê na Figura 12.9, o refino da interface dispara com 25% dos itens mencionados e mostra-se como ponto de alta relevância para se pensarem as próximas versões do Sketchument.

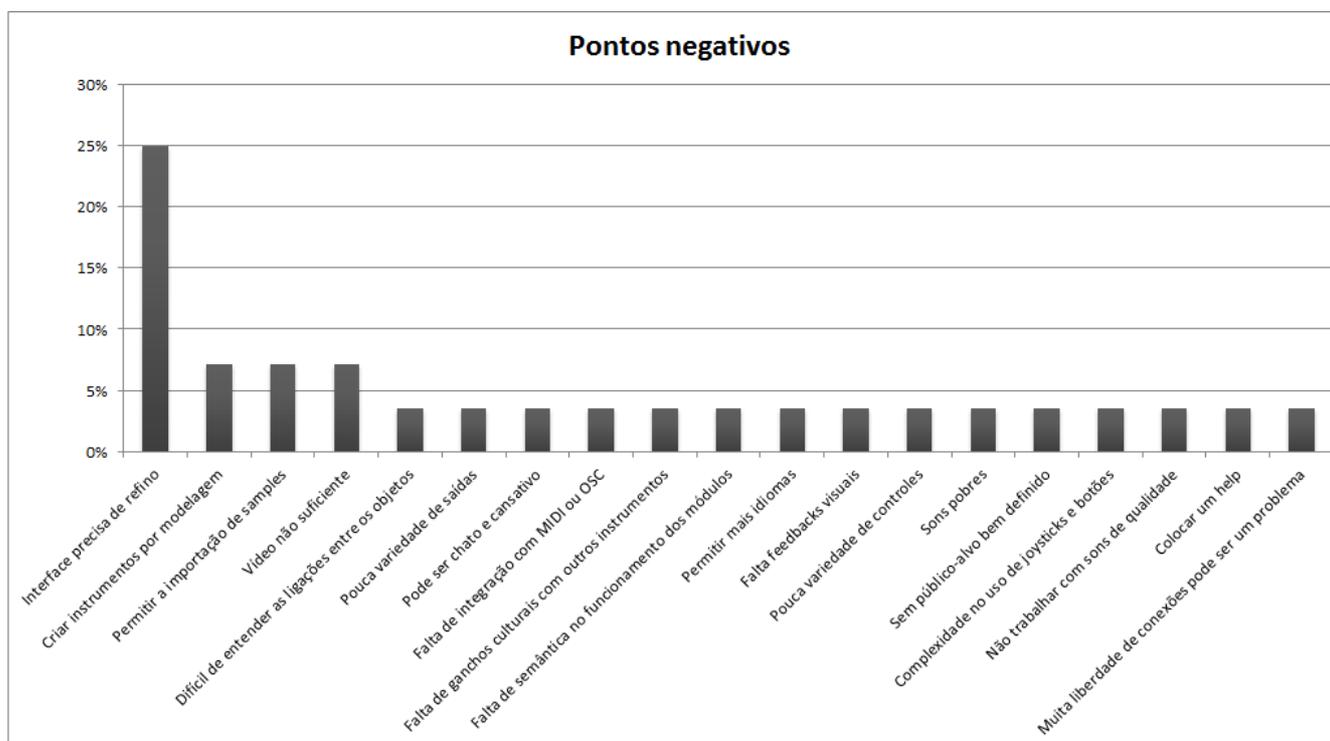


Figura 12.9: Pontos negativos mencionados no questionário agrupados por categorias

No Quadro 12.1, estão algumas sugestões de modificação ou implementação extraídas das respostas da última questão do questionário. A porcentagem de respostas associadas ao tema pode servir de base para um ordenamento que auxilie nas próximas etapas de desenvolvimento. Porém, no caso desta última questão, não há dados suficientes para gerar uma ordenação válida para a implementação. De toda forma, as sugestões serão computadas nas etapas posteriores de desenvolvimento.

Quadro 12.1: Sugestões mencionadas no questionário

Importar samples externos	28,60%
Permitir a gravação do som	7,14%
Fazer um tutorial para o início	7,14%
Permitir integração com outros programas	7,14%
Fazer uma maior ligação com instrumentos já existentes	7,14%
Mais outputs	7,14%
Permitir conexão por WIST com vários iPads	7,14%
Uso do acelerômetro	7,14%
Uso do sensor magnético	7,14%
Criar uma performance com o Sketchument	7,14%
Mais inputs	7,14%

13. Entrevistas

As entrevistas foram realizadas com sete potenciais usuários do aplicativo, isto é, músicos com perfil de experimentação que usam ou já usaram tecnologias digitais em suas práticas musicais.

Esta fase teve como objetivos: 1) coletar informações sobre o background musical dos participantes e suas experiências com tecnologias digitais na música; 2) entender o uso de tablets com telas multitoque, mais especificamente o iPad, em suas práticas musicais; 3) colher impressões e feedbacks sobre a versão 2 do protótipo do aplicativo.

A abordagem utilizada foi de entrevista semi-estruturada (Preece, Rogers, & Sharp, 2002), pela qual um roteiro é seguido, podendo as perguntas e respostas fugirem um pouco do tema central. Isto é importante nesta fase de coleta, pois é preciso ter uma visão mais ampla dos contextos, atividades e visões dos potenciais usuários para, dando suporte para fases posteriores de geração de alternativas baseadas nesses dados.

As entrevistas duraram em média uma hora e trinta minutos. Depois de selecionadas, as principais citações foram interpretadas e associadas a uma breve descrição, que serviu de base para um posterior agrupamento por afinidade de tema. A seguir, os temas principais serão descritos.

13.1. Aspectos Gerais

A maioria dos entrevistados destacou que uma das principais vantagens de usar o iPad no contexto musical é o tamanho da tela e a baixa latência que a plataforma oferece. Além de ser um dispositivo móvel que facilita bastante o transporte e a manipulação, visto que o usuário não tem que se esforçar muito para ter acesso às suas funcionalidades. Esse sentimento pode ser visto nesta frase: “Quando às vezes eu quero ensaiar com a galera, eu uso o AmpKit. Em vez de levar as coisas, eu levo o iPad”. Também pela facilidade de locomoção, alguns usuários afirmaram que usam o iPad como um registro de ideias para futuras composições: “Hoje, ele parece pra mim como um caderninho pra fazer música. É um cara que tá do meu lado o tempo inteiro”.

Outro aspecto interessante do iPad levantado pelos entrevistados é o uso desfocado e desprezioso em contraposição à forma de interação com computadores de estúdio, onde o usuário sempre está sentado, extremamente focado e de forma rígida: “você tá cheio de plugin... um monte de track... é tenso! O iPad tirou essa tensão”. Como ilustração, destaca-se um rápido cenário descrito por um dos entrevistados: “gosto de ficar deitado no sofá experimentando novos sons nos aplicativos, acho tranquilo para ter mais ideias”.

Muito associado ao cenário anterior, está o fato de se poder usar o tablet como estímulo inicial para uma composição, ou seja, começar o trabalho no iPad e depois transportá-lo para algum ambiente de edição no computador: "Defino os parâmetros, crio um som trash, e depois evoluo a ideia". Este tipo de conexão entre o iPad e o computador é destacado como ponto importante na escolha de aplicativos: "Transferir as minhas ideias para a minha plataforma de trabalho no computador, ia ser um fator decisivo no uso de alguma coisa". A agilidade é um ponto ressaltado: "Eu comprei o iPad para isso: economizar tempo fazendo música."

Seja por conta da interface, do ar lúdico e desprezioso ou da mobilidade, a maioria dos entrevistados vê o iPad como um artefato voltado para experimentação musical - o que está bem ilustrado por esta frase: "Eu gosto de usar o iPad para prototipar ideias".

Alguns entrevistados falaram da possibilidade de já se produzir uma música usando completamente o iPad em todas as suas etapas. Desde gravação, edição, aplicação de efeitos, mixagem e exportação em áudios de qualidade. Um dos entrevistados inclusive citou que já fez trilhas sonoras completas para filmes curtas-metragens usando apenas o iPad: "Fiz 8 trilhas pra 8 curtas tudo no iPad". Em contrapartida, este processo ainda não é adotado por todos, seja por conta da interface limitada, ou a ligação cultural com o paradigma teclado-mouse: "Eu ainda não consigo finalizar uma música aqui. Eu acabo levando pro Logic".

Uma das desvantagens para o uso do iPad é que a comunicação entre ele e o computador ainda não é tão intuitiva, tendo de se gastar um certo tempo para fazer as conexões ou exportações necessárias: "A vinculação disso aqui [iPad] com o Live ainda não é tão trivial. Se existisse uma maneira mais rápida de fazer essa ligação, eu usaria mais". Poucos aplicativos tornam automático, rápido e fácil o processo de levar os experimentos do iPad para o computador.

Um dos entrevistados, citando o iPad como alternativa aos controladores MIDI atuais, destacou que uma desvantagem é a ausência de sensibilidade ao toque por parte de alguns aplicativos. Por sua experiência, as formas de captar a sensibilidade do toque ainda estão muito inexpressivas, tendo de ter uma grande variação de intensidade para se perceber algum resultado satisfatório.

No mesmo contexto de uso como controlador, a falta de feedback háptico foi apresentado como outra desvantagem. Sem este tipo de feedback, segundo os entrevistados, os músicos perdem a noção de espaço, forçando-os a sempre olharem para a tela, o que para alguns, é prejudicial para a performance.

13.2. Possíveis Usos Musicais para o iPad

Da análise das entrevistas e do agrupamento das citações, foi possível identificar alguns cenários de uso do iPad como ferramenta. A maioria dos entrevistados afirma não utilizar os instrumentos simulados do iPad por acharem bastante limitados, salvo em contextos de

geração inicial de ideias para composição. "Eu tenho um programinha muito massa, DM-1. Mas é muito fácil você ficar com o som muito igual a qualquer pessoa que use o DM-1".

O iPad como controlador, já mencionado anteriormente, é o cenário onde ele é usado para mandar mensagens para outro dispositivo ou aplicativo que sintetiza o som. A tela multitoque, neste caso, se resume a uma interface de controle de outro aplicativo. Esta situação é vista na seguinte frase: "Uma coisa que eu usei ultimamente e gostei, foi usar o iPad para controlar o Live. Uso o Lemur. Eu já pego os templates prontos". O aplicativo Lemur funciona, neste caso, como uma interface de controle para o Ableton Live. Esse uso pode estar relacionado com o fato do paradigma WIMP ser um limitante de interação, por não permitir controles paralelos, ou seja, com o mouse, o usuário só consegue ter acesso a um controle por vez. Podemos ver esta dificuldade na seguinte frase: "não tem essa possibilidade de ao vivo eu ter que me preocupar de ir com o mouse em algum lugar específico". Neste caso, existe uma necessidade por agilidade durante a performance, o que em alguns casos é resolvido com o uso de controladores MIDI. Destaca-se aqui uma interessante oportunidade de usar o iPad para ampliar a expressividade com maior agilidade durante a performance e acredita-se que o uso customizável dos gestos para controlar ações específicas seja um forte ponto atrelado a esta oportunidade.

Outro cenário mencionado durante as entrevistas foi o iPad como processador de sinal. Neste caso, ele funciona como um intermediário entre o som produzido por um instrumento ou um microfone e a saída de som: uma caixa de som ou um amplificador, o que se assemelha muito com o papel dos pedais e pedaleiras de guitarra. Os aplicativos citados que entram nesta definição são o AmpKit e o AmpliTube. Foi discutido que, com o avanço do poder do processamento do iPad, muitos efeitos estão sendo capazes de ser simulados ou reproduzidos com grande qualidade. Como já foi mencionado acima, isto impacta diretamente na facilidade de transporte de equipamentos, tornando o iPad uma potencial ferramenta para este cenário.

O iPad como sintetizador de sons também foi comentado durante as entrevistas. Neste cenário, o iPad permite a modificação de propriedades do som a fim de se definir um timbre. Geralmente, nesses casos, controladores MIDI são conectados ao iPad através de interfaces MIDI compatíveis com a plataforma. Estes controladores têm a função de apenas disparar as notas tocadas, deixando para o iPad a função de sintetizar o som.

Com menor frequência, foi comentado que existem hardwares no mercado que foram feitos especificamente para serem controlados por iPad. É importante ressaltar a diferença de atribuição do iPad neste caso: o iPad controla os atributos de configuração do hardware específico, como por exemplo, mesas de som, alguns controladores MIDI etc., diferenciando-se dos cenários descritos anteriormente.

Com relação ao contexto de uso, foi mencionado o uso do iPad no processo de composição, na experimentação timbrística, durante uma performance. Num contexto mais voltado para

leigos, foi também citado como primeiro contato com um instrumento musical, podendo dar ao usuário leigo a sensação de tocar um determinado instrumento, mesmo não sabendo tocar.

O processo de composição abre oportunidades para várias práticas, como a experimentação desfocada para geração inicial de ideias, mencionada anteriormente; como registro de ideias musicais e como estímulo inicial para um projeto, começando a desenvolver no iPad e depois levando-o para uma estação de trabalho no computador, isto é, um software de gravação e edição. Além disto, existe a viabilidade de se fazer todo o trabalho de produção diretamente no iPad.

Para o processo descrito como *sound design*, o iPad é utilizado como plataforma de experimentação para geração de novos timbres. Podemos considerar uma etapa relacionada à composição, apesar de ser independente desta.

O contexto mais comentado foi o do uso do iPad durante performances musicais, seja através do músico que vai integrar o iPad no seu setup de equipamentos já existentes, seja como controlador para softwares de DJ, podendo dar mais expressividade a este perfil de profissionais, funcionando como uma ferramenta agregadora.

13.3. Impressões sobre o Sketchument

Sobre as impressões e feedbacks acerca do Sketchument, um dos entrevistados resumiu que o público-alvo do aplicativo, mais do que apenas músicos que usam tecnologias digitais em suas práticas musicais, são “pessoas que trabalham com arte e música e experimentam novas interfaces nos dispositivos que utilizam”. Esta afirmação condiz com o objetivo do projeto de criação de um ambiente de experimentação e mostra que o protótipo está sendo bem entendido pelos potenciais usuários. Além disso, foram mencionados cenários onde pessoas com pouca ou nenhuma formação musical pudessem ter uma primeira experiência com musicalização.

O uso do joystick para o iPad como uma alternativa de feedback háptico não foi bem recebida pelos participantes. Um dos entrevistados afirmou que ficou com medo de usar o joystick por apresentar riscos à tela do iPad. Outro criticou a baixa qualidade de rastreamento que apresenta muitas falhas de resposta.

Uma sugestão de melhoramento foi de permitir o carregamento dos próprios samples do usuário no objeto Samples Player, o que se relaciona fortemente com as diretrizes de customização e de diversidade, apresentados nas Seções anteriores.

Dois dos entrevistados citaram o paradigma de alta especialização dos aplicativos para a plataforma iOS, explicando que, para um aplicativo ter sucesso, sua função precisa estar muito bem definida. A sugestão dada para a versão testada do protótipo foi de, caso o objetivo seja lançar o produto na App Store, começar com uma versão “*stripped*”, ou seja, com apenas o núcleo das funcionalidades e com o tempo adicionar mais funcionalidades através das

atualizações. Este procedimento ao mesmo tempo mostraria que o projeto está sempre em evolução e daria o foco funcional que o aplicativo precisa. Aprofundando-se mais, um dos entrevistados constatou que a missão do Sketchument é ser um controlador gestual, modular e customizável, ou seja, na primeira versão, apenas o conjunto de gestos de entrada e objetos de saída MIDI e OSC deveriam ser implementados. Segundo ele, já seria um começo inovador, visto que não existem controladores assim no mercado. Neste ponto de vista, a possibilidade de escolha de gestos foi vista como potencial inovação voltada para construção de controles gestuais customizáveis.

Além dos resultados apresentados acima, uma lista de oportunidades para a geração de ideias de novas funcionalidades foram selecionadas. O processamento dessas informações será discutido na Seção 17.

13.4. Testes de Interface

Na etapa final da entrevista, foi usada uma abordagem baseada em tarefa, na qual o entrevistado foi requisitado a completar uma determinada atividade no sistema.

O foco das tarefas propostas foi testar se o requisito de fácil entendimento da atividade de adicionar uma entrada, adicionar uma saída e fazer uma conexão entre as duas estava sendo bem sucedido. A lista de tarefas requisitadas segue abaixo:

- 1) fazer com que, através de um toque na tela, uma nota musical fosse tocada;
- 2) fazer com que, através de um gesto na tela, uma nota musical fosse tocada;
- 3) fazer com que, através de qualquer entrada (gestos ou objetos tocáveis) escolhida pelo participante, um acorde fosse tocado.

Além disso, para ampliar a compreensão do entendimento do sistema por parte do usuário, foi requisitado que para cada ação o participante falasse em voz alta o que estava pensando para tomar as duas decisões – o que é chamado de *think-aloud protocol*.

Os resultados seguiram uma abordagem qualitativa, sem medição de tempo de realização da tarefa ou dados semelhantes, visto que nesta etapa de avaliação é mais importante entender se o modelo mental do programa é compreendido pelo usuário do que apenas aperfeiçoar ou otimizar as etapas de execução das tarefas.

A partir da observação dos participantes, pôde ser constatado que o menu flutuante (Figura 13.1) não estava intuitivo o suficiente. Assim que se abria o aplicativo, pela falta de informações na tela, nenhum usuário entendeu no primeiro momento que teria de tocar a tela para abrir o menu. Muitos afirmaram que não sabiam o que fazer, até tocar a tela e o menu aparecer – o que muitas vezes foi encarado com surpresa.

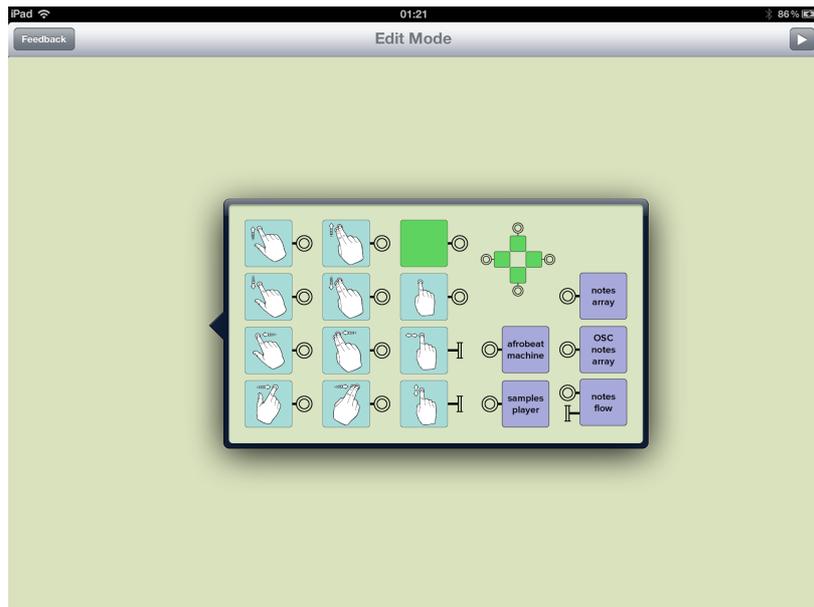


Figura 13.1: Menu flutuante do protótipo versão 2

O menu único, que agrega objetos de entrada e saída no mesmo local, gerou confusão na hora da escolha de qual objeto deveria ser colocado para completar a tarefa requisitada. Dois dos seis participantes disseram que não entenderam o código de cores, cuja intenção era dividir entre objetos tocáveis, gestuais e saídas. Um usuário afirmou não ter entendido os ícones dos conectores: “É um ‘I’ e um ‘O’, quer dizer ‘input’ e ‘output’?”

13.5. Exemplos de Instrumentos Construídos

Abaixo, são apresentados alguns exemplos de instrumentos construídos pelos usuários no ambiente de experimentação Sketchument.

13.5.1. Pandeiro

Este instrumento é uma simulação de um pandeiro para tela multitoque (ver Figura 13.2). É formado por um conjunto de três objetos tocáveis de tamanhos customizados para permitir uma interação semelhante à forma de se tocar um pandeiro real. Como objetos de saída, há o disparador de *samples*, que possui sons pré-carregados de vários tipos de toque de um pandeiro: polegar, base da mão, ponta dos dedos etc. Cada área tocável foi associada ao *sample* correspondente ao toque do pandeiro real.

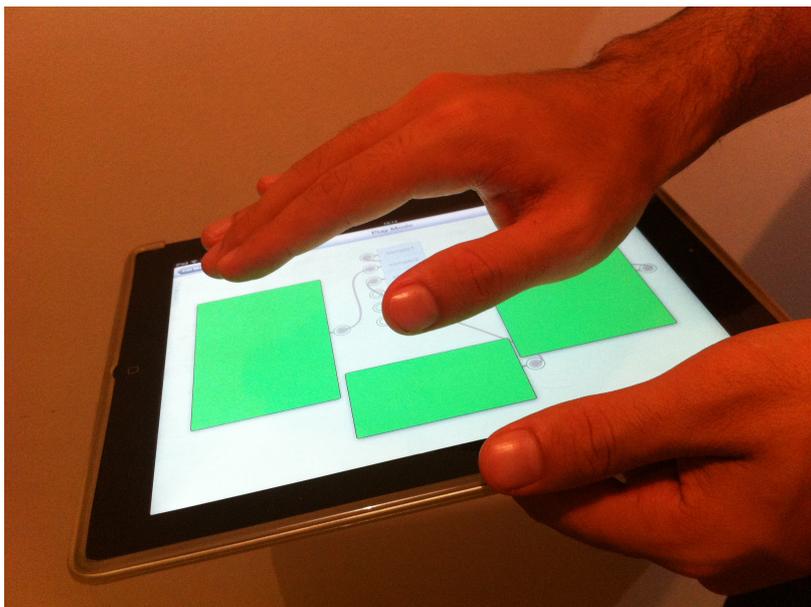


Figura 13.2: Pandeiro digital

Com esse exemplo, procura-se demonstrar a capacidade de criação de instrumentos simulados com o Sketchument. Fazendo este gancho cultural com as metáforas de interação de instrumentos já existentes, pode-se extrapolar com a experimentação e, a partir de referências conhecidas, criar outros tipos de interação e instrumentos.

Em avaliações informais com potenciais usuários, o pandeiro se mostrou interessante do ponto de vista de interface, já que para um pandeirista o repertório de gestos é praticamente o mesmo, porém os resultados sonoros não foram satisfatórios. A principal queixa se relaciona com um atraso no disparo de um segundo sample depois de um primeiro ser disparado. Isto pode ser um problema de arquitetura que já está registrado para melhoramento na próxima versão.

13.5.2. Acompanhamento Afrobeat

O instrumento apresentado a seguir (ver Figura 13.3) baseia-se no funcionamento do *patch* Afrobeat Machine desenvolvido por Jarbas Jácome e já descrito anteriormente. Podemos dizer que o Sketchument, neste caso, funciona apenas como uma interface de controle para um *patch* de Pure Data, onde o usuário pode definir de forma customizada o arranjo dos elementos da interface para facilitar a interação.

Para o disparo de cada música, existe um objeto tocável associado e para a mudança de andamento, conecta-se um gesto de deslizamento para cima, aumentando os batimentos por minutos (BPMs) e um gesto de deslizamento para baixo, diminuindo os BPMs.

O objetivo deste exemplo é apresentar o Sketchument como uma ferramenta de construção customizada de interface para sistemas já existentes, além de demonstrar a integração com tais sistemas. No futuro, este perfil de uso será potencializado com a adição de objetos de

comunicação que implementam o protocolo MIDI e a definição do objeto OSC como módulo oficial da próxima versão do protótipo.

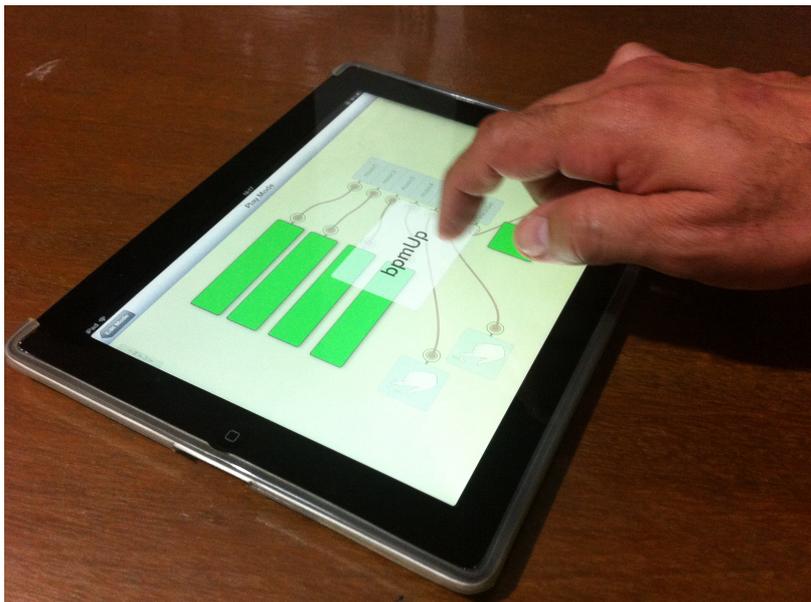


Figura 13.3: Acompanhamento Afrobeat

Este instrumento também foi testado por alguns usuários e três problemas foram levantados. Primeiro em relação ao andamento, quando uma música é trocada, o andamento é definido para o padrão da música. Ou seja, mesmo que o usuário tenha definido um batimento próprio, esta informação é perdida quando a música é trocada. Segundo, a falta de algum feedback visual informando o andamento atual. E terceiro, o fato de não haver controle de pausa nas músicas, isto é, assim que o Modo de Execução é ativado, a música já começa a tocar.

Todos os problemas apresentados pelo usuário se relacionam com a forma como o patch do Pure Data foi pensado. Neste ponto, como o patch foi importado diretamente para o Sketchument, constata-se a necessidade de fazer algumas alterações que o adequem ao contexto do ambiente.

13.5.3. Disparador de Acordes

O instrumento ora apresentado facilita o disparo de acordes definidos nota a nota pelo usuário (ver Figura 13.4). Além de permitir o disparo paralelo de notas, o usuário também pode tocar de forma sequencial as notas do acorde.

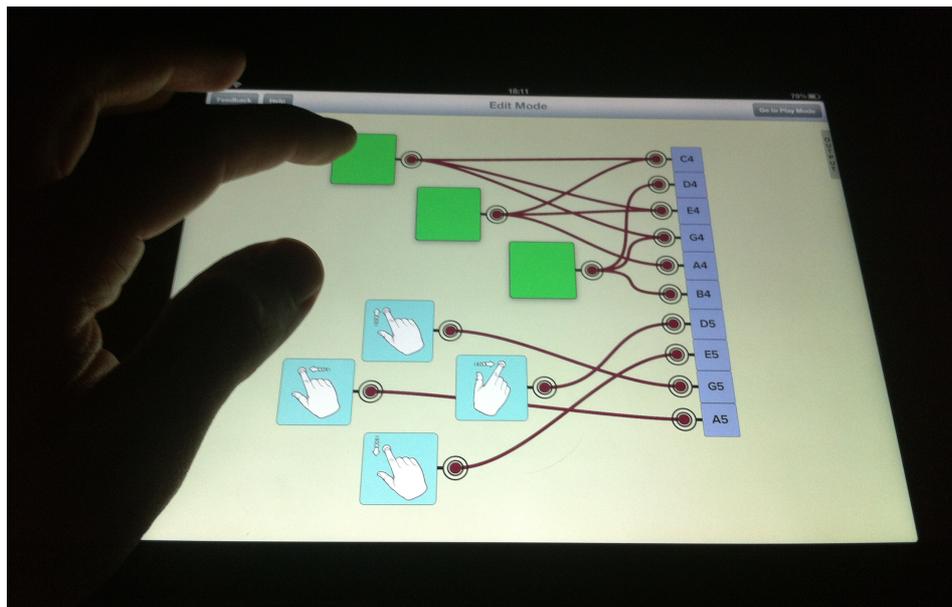


Figura 13.4: Disparador de acordes

Como componentes do instrumento, há objetos tocáveis fazendo o papel de cordas e os objetos gestuais disparando notas em paralelo. Como objeto de geração de som, tem-se o conjunto de notas.

14. Perfis de Usuários e Cenários de Uso

Como resultado do processamento das informações obtidas nas análises das entrevistas e do questionário, juntamente com a avaliação da literatura, alguns cenários de uso serão descritos abaixo.

O principal conjunto para descrever os perfis de potenciais usuários é o de pessoas que trabalham com arte e música e experimentam novas formas de interação para identificar possíveis integrações a práticas artísticas.

Partindo deste grande grupo, foram identificados cenários de uso que, associados às diretrizes do projeto, servirão de base para as tomadas de decisões do projeto.

Em primeiro lugar, é importante ressaltar a tendência de alguns músicos de fazer combinações e construir seus próprios dispositivos que garantem sons únicos e formas exclusivas de se tocar.

O **perfil luthier** é o usuário que irá usar o sistema com o intuito de construir instrumentos/interações musicais com o aplicativo para uso próprio ou para compartilhá-los com outros usuários. Este perfil está associado a dois cenários de uso: **experimentação interativa**, que busca escolher os melhores mapeamentos dos gestos e controles com as formas de geração de som relacionadas com sua intenção e contexto; e **experimentação sonora**, ou *sound design*, que busca a configuração de variáveis sonoras que melhor se adequam à sua intenção e contexto.

Para o **perfil compositor**, o resultado do instrumento/interação em si não é importante, mas sim os resultados musicais e sonoros do processo de construção. Este perfil está associado com a mobilidade do iPad, que permite que ideias musicais sejam trabalhadas e registradas nos mais diversos lugares e contextos. Percebem-se dois cenários de uso para este perfil. O primeiro é o de **experimentação musical**, no qual manipulando objetos e brincando com as possibilidades oferecidas pelo aplicativo, surgem ideias que podem ser trabalhadas futuramente como material para uma nova obra. O segundo é o **registro** destas sementes de ideias na forma de gravação, que é importante para dar continuidade ao trabalho em outras plataformas.

O **perfil produtor** utiliza o iPad no cenário de **estação de trabalho de produção musical** como um todo: gravando, editando, aplicando efeitos, mixando e exportando o material. É um perfil ainda não muito comum, dadas as limitações do iPad e o costume já consolidado de trabalhar em outras plataformas.

O **perfil performer** usa um instrumento já definido durante sua performance. Pode-se definir isto como **cenário instrumental**. Entram neste perfil os DJs que preparam seu set antes de efetivamente tocar e todos os músicos que de alguma forma executam suas ações numa interface previamente definida. Com relação à obra executada, pode-se subdividir o cenário

instrumental em **interpretação**, na qual uma obra pré-existente à performance é interpretada e a **improvisação**, na qual o músico durante a performance improvisa algo que previamente não existia.

O **perfil educador** se assemelha com o luthier no fato de preparar o ambiente previamente. Este o faz com o intuito de apresentá-lo para seus alunos associando ao conteúdo a ser trabalhado. Associa-se a este perfil o **cenário educativo**.

O **perfil leigo** utiliza o aplicativo em um **cenário exploratório**, buscando entender os funcionamentos e conexões dos elementos existentes. Não está preocupado nem com o resultado instrumental, nem com o resultado sonoro ou musical; busca apenas explorar o sistema como forma de entretenimento. Entram neste perfil pessoas com pouca ou nenhuma formação musical e alunos iniciantes em música. Pode ser uma boa opção para pessoas que estejam querendo entender um pouco mais sobre instrumentos musicais.

É importante ressaltar que um mesmo usuário pode assumir diferentes perfis dependendo do contexto de utilização. Abaixo, seguem alguns exemplos de processos e utilizações possíveis do Sketchument, assumindo papéis de **ferramenta** de construção do instrumento e do próprio **instrumento** a ser utilizado na música.

*Luthier + **Ferramenta** + Processo de Construção = Instrumento*

*Compositor + **Ferramenta** = Insights*

*Compositor + **Ferramenta** + Insights = Registro de Ideias*

Compositor + Registro de Ideias + Processo Criação Musical = Registro da Obra

*Performer + **Instrumento** + Processo de Interpretação + Registro da Obra = Obra Interpretada*

*Performer + **Instrumento** + Processo de Improvisação = Obra Improvisada*

Produtor + Estação de Trabalho + Processo de Gravação + Registro da Obra = Obra Gravada

*DJ + **Ferramenta** + Processo de Mistura + Obra Gravada = Remix*

*Leigo + **Ferramenta** + Processo Exploratório = Entendimento + Entretenimento*

*Professor + **Ferramenta** + Processo Educativo = Instrumento Educativo*

*Estudante + **Instrumento Educativo** + Processo Exploratório = Aprendizado*

A Figura 14.1 sintetiza os diferentes perfis de usuário e os cenários de uso relacionados.

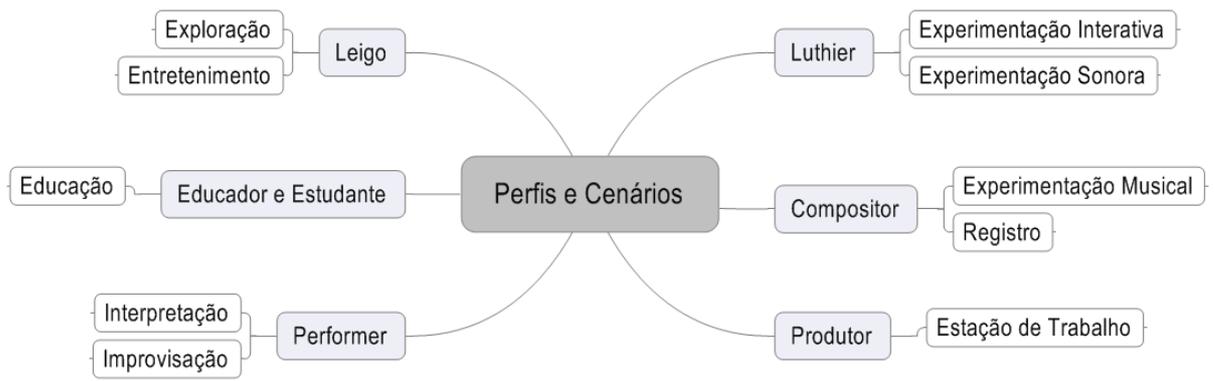


Figura 14.1: Relação entre Perfis e Cenários

15. Protótipo Versão 3

Baseando-se nos resultados das avaliações com potenciais usuários, alguns melhoramentos foram implementados na versão 3 do protótipo do Sketchument (ver Figura 15.1).

Por conta da confusão reportada durante a adição de novos objetos, a forma de apresentação dos objetos de entrada e saída foi reorganizada. O menu flutuante foi substituído por duas abas laterais: na aba esquerda, foram posicionados os objetos de entrada e na aba direita, os objetos de saída. Adicionalmente, o ícone do botão de ativação do modo de execução foi alterado para um texto: “Go to Play Mode”.

Devido à rejeição por parte dos usuários, os objetos tangíveis foram removidos do conjunto de objetos de entrada. Além disso, os objetos com conectores contínuos foram inativados nesta versão por estarem causando confusão e precisarem de melhoramentos da interface. Para as versões posteriores, um trabalho mais amplo de entendimento do problema e geração de alternativas deverá ser feito.

Além disso, devido a problemas de entendimento de como um objeto poderia ser excluído da tela, uma lixeira foi implementada e posicionada na parte inferior da tela. Arrastando o objeto para esta área, a lixeira é ativada e posicionando o objeto na lixeira, o objeto é deletado da tela.

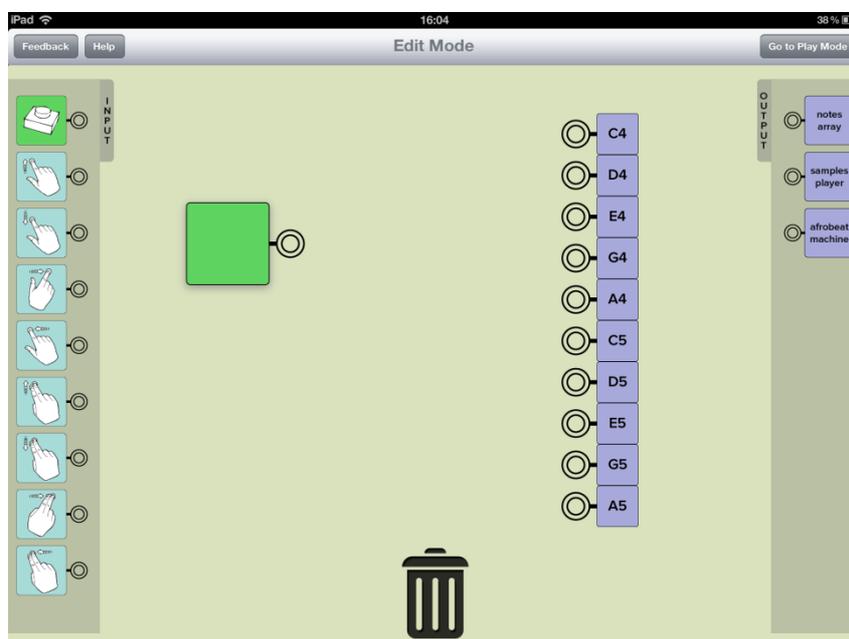


Figura 15.1: Protótipo para iPad v3

Com o intuito de ajudar o usuário a entender melhor o sistema, um guia rápido (ver Figura 15.2), com as principais funcionalidades do ambiente, foi adicionado.

Preparando o aplicativo para a fase de beta test, um formulário de feedback foi adicionado na interface.

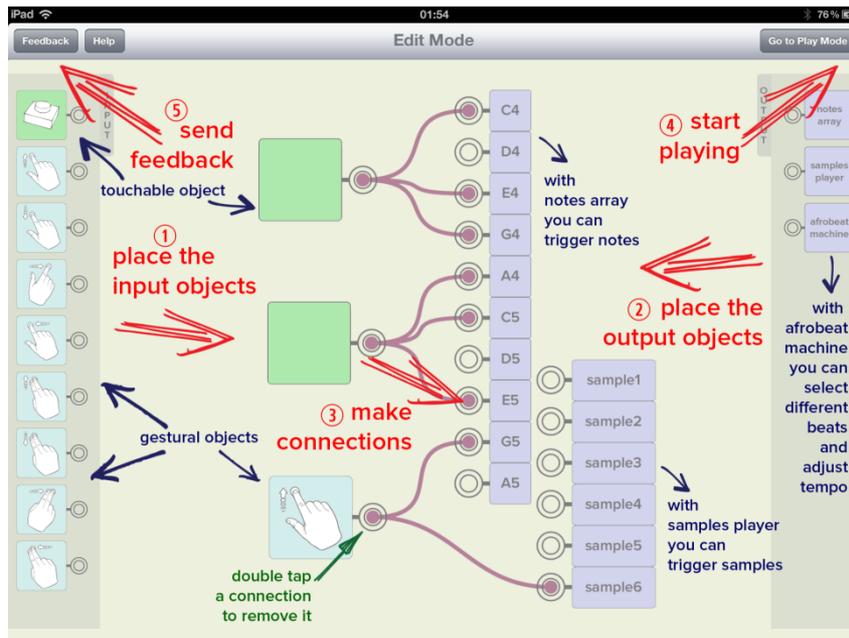


Figura 15.2: Tela de ajuda adicionada a versão 3 do protótipo

15.1. Considerações

Os objetos de entrada e de saída apresentados no protótipo versão 3 são apenas ilustrações de funcionalidades que o ambiente pode implementar. O potencial do Sketchument não se limita a estas possibilidades e, dada a sua estrutura modular, nele podem ser desenvolvidos diferentes tipos de funcionalidades. A ideia é que no futuro, a comunidade crie novos objetos através de uma API e integre-os facilmente ao ambiente.

16. Beta Test

Nesta Seção, são apresentados os resultados obtidos no programa de *beta test* do protótipo v3 do Sketchument. Serão apresentadas avaliações quantitativas relativas aos dados de uso do aplicativo por 09 usuários e as avaliações qualitativas provenientes do formulário aberto, presente no interior do aplicativo, onde o usuário pode enviar seus feedbacks em forma de texto.

Antes de tudo, é preciso entender como funciona a plataforma de distribuição online que facilitou a execução destes testes.

16.1. Testflight



TestFlight é uma plataforma de distribuição online de aplicativos para iOS que ainda estejam em estado beta. Ela facilita a criação de grupos de distribuição e a coleta de *feedback*, através de formulários embutidos dentro do próprio aplicativo. Esta plataforma facilita o serviço de distribuição *ad-hoc* da Apple chamado de *Over-the-air*, pelo qual o desenvolvedor pode preparar seu aplicativo para ser instalado através de um link de internet pelo usuário beta tester.

Além de fornecer *feedback* textual, que possibilita ao usuário responder perguntas abertas ou questionários, esta plataforma permite a avaliação quantitativa de alguns parâmetros do aplicativo, como por exemplo, quanto tempo o usuário passou testando-o, quantas vezes o abriu, quantas vezes o aplicativo travou etc. Se o desenvolvedor optar, também pode colocar marcadores no código e saber se o usuário passou por determinado ponto do seu programa, os chamados checkpoints.

Com todo esse ferramental, o uso do *TestFlight* permite extrair informações importantes sobre o funcionamento e a recepção do aplicativo nas mãos do usuário final. Também facilita bastante a distribuição e permite que o feedback seja mais profundo, pois o usuário pode testá-lo de acordo com o seu contexto de uso, com tempo suficiente para explorá-lo como bem entender – o que não acontece em um ambiente controlado de testes de usabilidade. Com esse tipo de avaliação, as impressões dos usuários são mais profundas e conseqüentemente mais valiosas para o desenvolvedor.

16.2. Dados de Uso (Quantitativo)

Através dos serviços disponibilizados pelo *TestFlight*, pôde-se levantar alguns dados quantitativos do uso do protótipo v3 do Sketchument por parte dos *beta testers* que o instalaram em seus dispositivos. Os dados disponíveis foram os seguintes: quantidade de vezes que o aplicativo interrompeu subitamente o seu funcionamento (*crash*); a quantidade de vezes que o usuário abriu o aplicativo (número de sessões); a duração de cada sessão; e a quantidade de checkpoints passados pelo usuário.

Os checkpoints são linhas de código adicionadas nas classes de controle do Sketchument e representam ações realizadas pelo usuário, como por exemplo, a adição de um objeto gestual de deslizamento com um dedo para cima, a remoção de um objeto de saída do tipo conjunto de notas etc. O número de checkpoints passados depende de quantas ações o usuário realizou no aplicativo. Por decisão de projeto, os checkpoints tiveram o nível de detalhe bastante granular para permitir uma análise remota do tempo dispendido para realizar as ações na interface, dando assim possíveis dicas da usabilidade do programa. Esses dados não foram utilizados no decorrer deste Projeto, porém poderão servir de base para versões futuras do protótipo.

Pelos resultados, não foi constatada nenhuma falha crítica do funcionamento do Sketchument. Atribui-se este fato à estabilidade proporcionada pela plataforma de desenvolvimento iOS. Apesar de ser utilizado um conjunto de biblioteca de terceiros, o conjunto de classes disponibilizadas pelo SDK do iOS forneceu uma boa base para evitar o surgimento de erros críticos.

O Quadro 16.1 mostra os dados obtidos pela interação de nove usuários com o aplicativo.

Quadro 16.1: Dados de uso dos beta testers

Usuário	Duração (minutos)	Sessões	Média de Duração por sessão (minutos)	Checkpoints
Usuário 1	32:56	5	06:35	1014
Usuário 2	05:15	2	02:38	225
Usuário 3	01:45	1	01:45	52
Usuário 4	02:13	1	02:13	118
Usuário 5	07:48	1	07:48	447
Usuário 6	00:56	1	00:56	56
Usuário 7	10:37	1	10:37	297
Usuário 8	18:44	16	01:10	2240
Usuário 9	08:36	4	02:09	322

Constata-se um grande engajamento por parte do Usuário 1 e o Usuário 8, porém a maioria dos usuários apresentou um tempo de utilização menor que 10 minutos. Uma hipótese para este resultado é a falta de diversidade dos módulos presentes neste ambiente de experimentação, causando pouco interesse do usuário em experimentar um maior número de combinações possíveis.

O engajamento dos usuários pode ser visualizado na Figura 16.1, que apresenta a combinação de número de sessões, checkpoints passados e o total de tempo gasto no programa. Apenas dois usuários se destacam com um somatório de engajamento alto, não tendo o restante se aprofundando no uso do aplicativo.

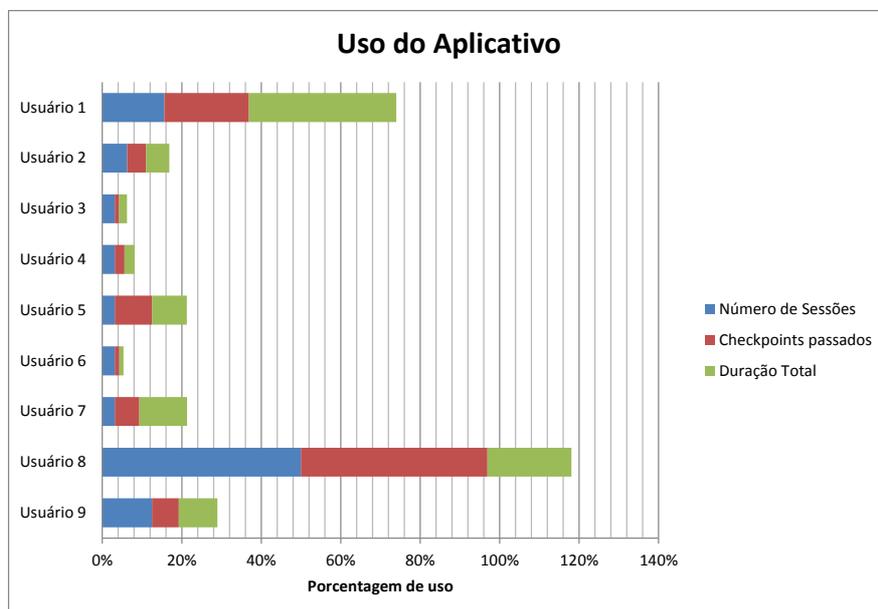


Figura 16.1: Uso do aplicativo pelos beta testers

16.3. Feedbacks (Qualitativo)

A partir do formulário de feedback presente no aplicativo, algumas impressões, críticas e sugestões foram colhidas e serão apresentadas a seguir. É importante destacar que o usuário, através da plataforma de distribuição online, teve a oportunidade de testar o aplicativo à sua maneira, sem nenhuma interferência por estar participando de algum experimento em laboratório. Acredita-se que, desta forma, houve um maior aprofundamento na interação, gerando respostas bastante significativas para a avaliação do protótipo.

Uma sugestão importante dada por um dos usuários foi que os objetos de saída deveriam apresentar algum tipo de feedback sonoro ainda no modo de edição para que se possa escolher de forma efetiva a saída sonora para determinada entrada. Falando especificamente do Sample Player, o usuário descreve: “Senão, terei de colocar um botão em todas [as saídas], ir ao Play Mode, ouvir cada um, e voltar ao Edit Mode e tirar os botões dos samples que não quero”. Realmente, da forma como está implementada, tal funcionalidade fere a diretriz de feedback. Esta sugestão será listada como melhoramento a se fazer na versão 4 do protótipo.

Também foram relatados alguns conflitos com o modo de “Multitasking Gestures” do iOS 5 e posteriores, já que alguns gestos do aplicativo são interpretados pelo sistema como gestos gerais de controle do iOS, gerando respostas inconsistentes ao aplicativo. Uma sugestão dada é: “o app poderia aconselhar o usuário a desativar os gestos do sistema para se ter uma melhor experiência”.

Foi percebido que, durante os comentários, o modo de edição dos botões não ficou tão claro, já que alguns usuários comentaram que gostariam de redimensionar o botão. Apesar desta funcionalidade já estar implementada, poucos conseguiram perceber como ativá-la. Assim, pode-se entender como melhoramento, o fato de definir uma forma mais intuitiva de ativar o modo de redimensionamento do botão.

Também foi comentado por alguns usuários que visualmente está faltando uma melhor forma de diferenciar o modo de edição do modo de execução. “O background poderia mudar de cor, ou haver uma forma mais clara de dizer que se está no play ou edit mode”. Esta percepção se deu pela confusão do usuário achar que está no modo de execução e quando foi tocar, acabou modificando as configurações dos objetos no modo de edição. Este tipo de confusão foi comum, reforçando assim a importância de modificação do feedback visual para cada tipo de modo.

Além disso, uma crítica foi feita ao modo de transição entre o Modo de Edição (*Edit*) e o Modo de Execução (*Play*): “Quando vou do play mode para o edit mode, vejo as opções de output correndo da esquerda para a direita”. Aqui percebe-se que a animação que “corre” da esquerda para direita não tem nenhuma ligação semântica com a mudança de modos no aplicativo e esta modificação será listada como melhoramento da próxima versão.

Foi notada uma falha de simultaneidade quando mais de três botões são tocados ao mesmo tempo: “[...] é perceptível que os botões não são tocados simultaneamente”. Vê-se que há um atraso que prejudica a execução das notas. Na lista de funcionalidades e modificações ao protótipo, tal falha será categorizada como erro e terá uma prioridade maior que os melhoramentos.

Um conflito de resposta entre gestos e botões também foi mencionado pelos usuários. Quando o usuário realiza um gesto e ao mesmo tempo mantém um botão pressionado, ações inconsistentes acontecem, o que notadamente é um bug.

Uma limitação do protótipo mencionada nos feedbacks foi a impossibilidade de manter uma nota sendo tocada ininterruptamente, ou seja, ativar um botão que ficará sempre ligado enviando um sinal para o objeto de saída associado. Deste comentário pode-se pensar em um objeto de entrada que funcionaria como um toggle button do Pure Data ou do Max/MSP, que com um toque permanece ligado até que outro toque o desligue.

Sobre o objeto Afrobeat Machine, foi detectado um erro: sempre quando ele é adicionado, mesmo que não exista nenhum objeto de entrada associado, ele começa tocando a música 4.

“Eu sei que pode ser a parte chique do app, usar comandos multitoque para interação, ou deslizar o dedo, mas não senti a mínima vontade. Só queria usar os botões mesmo”. Esta impressão é de extrema importância para o planejamento da próxima versão, pois apesar de ser uma nova forma de usar a tela multitoque, o conjunto de gestos não foram bem recebidos por parte dos usuários. Alguns possíveis problemas levantados a partir desta afirmação é a

forma como os gestos são dispostos e a maneira como são associados às ações, sendo bastante limitados e independentes de regiões da tela. Uma possível funcionalidade que pode ser testada na próxima versão do aplicativo é o zoneamento da tela e a associação de gestos por zona, o que permite maior possibilidade de combinações e conseqüentemente mais diversidade (e expressividade) para o usuário.

Como feedback positivo, ressalta-se: “Eu usaria sim o app, mesmo tendo outros aqui como o GarageBand. Só o fato de eu poder organizar a posição dos botões como eu quiser, e colocar o instrumento que eu quiser na mesma tela, já vale a pena”.

17. Lista Ordenada de Funcionalidades

Nesta seção, serão apresentadas a lista de funcionalidades já implementadas até a versão 3 do protótipo e a lista de futuras funcionalidades levantadas a partir das informações vindas da avaliação e do uso das ferramentas de ideação.

No Quadro 17.1, é apresentada uma lista de funcionalidades já implementadas até a versão 3 do protótipo do Sketchument. O Quadro está ordenado pelo nome da funcionalidade e através dele pode-se identificar quais métodos do processo de design influenciaram seu desenvolvimento.

Quadro 17.1: Lista de Funcionalidades Implementadas

Nome	Descrição	Método Relacionado
Abas	Abas retráteis com objetos para serem adicionados na tela	Entrevistas: Testes de Interface Caixa Morfológica de Interface
Ajuda por tela sobreposta	Quando ativada a ajuda, uma tela sobreposta é apresentada com informações sobre o sistema	Entrevistas: Teste de Interface Caixa Morfológica de Interface
Apresentar conexões no Modo de Execução	Apresentar as conexões dos objetos durante o Modo de Execução	Diretriz: usabilidade
Arrastar objeto	Arrastar objeto da aba para o canvas	Entrevistas: Teste de Interface Caixa Morfológica de Interface
Dois toques para excluir	Dois toques no objeto para excluí-lo	Diretriz: edição ágil
Editar tamanho do objeto tocável	Tornar possível a edição do tamanho do objeto tocável	Diretriz: configuração Caixa Morfológica de Interface
Feedback tátil	Usar objetos tangíveis que se acoplam à tela do iPad	Análise da Literatura (Silva, 2012) (Reactable)
Lixeira	Arrastar o objeto para a lixeira para excluí-lo	Entrevistas: Teste de Interface Caixa Morfológica de Interface
Modos de Edição e Execução	Ter dois modos de funcionamento: edição e execução	Análise da Literatura (PD, Max/MSP) Caixa Morfológica de Interface
Objeto de Saída: Notes array	Objeto de produção sonora baseado na disposição de notas musicais disparáveis	Diretriz: diversidade Diretriz: edição ágil
Objeto de Saída: afrobeat machine	Objeto de produção sonora baseado no patch do PD: Afrobeat Machine	Diretriz: diversidade Diretriz: integração
Objeto de Saída: samples player	Objeto de produção sonora baseado na disposição de samples disparáveis	Diretriz: diversidade Diretriz: edição ágil
Objetos de entrada gestuais	Permitir que o usuário controle o instrumento através de gestos	Análise da Literatura (McGlynn et al., 2012)
Objetos de entrada tangíveis	Permitir que o usuário controle o instrumento através de objetos tangíveis acoplados à tela	Análise da Literatura (Silva, 2012)

Objetos de entrada tocáveis	Permitir que o usuário controle o instrumento através de toques na tela	Análise de Concorrentes
Objetos de Entrada: conexões contínuas	Adicionar objetos com conexões contínuas	Análise da Literatura (Seção X) Diretriz: diversidade
Pop-up com gesto reconhecido no Modo de Execução	Quando o gesto é reconhecido no Modo de Execução, um pop-up aparece com a descrição da ação associada	Diretriz: usabilidade
Surfing em objetos de entrada	Disparar a ação associada ao objeto quando o dedo for arrastado em cima do objeto tocável	Diretriz: diversidade Análise dos Concorrentes (BeatSurfing)
Toque longo para editar	Toque longo para ativar edição do objeto	Diretriz: configuração

As futuras funcionalidades, apresentadas no Quadro 17.2, foram ordenadas pelos seguintes critérios abaixo:

- 1) correções de bugs e mau funcionamento do aplicativo;
- 2) modificação de funcionalidades no protótipo v3 que de alguma forma feriram as diretrizes do projeto;
- 3) melhoramentos de funcionalidades do protótipo v3;
- 4) novas funcionalidades escolhidas para potencializar as diretrizes do projeto.

Este Quadro sintetiza o resultado do processo de design e serve de guia para futuros ciclos de implementação do protótipo.

Quadro 17.2: Lista Ordenada de Funcionalidades Futuras

Nome	Descrição	Método Relacionado
Estética da Interface	Mudar a estética da interface para atender melhor ao público-alvo	Entrevistas Questionário
Feedback Sonoro no Modo de Edição	Durante o Modo de Edição, ao se clicar em um objeto, a ação associada é disparada	Diretriz: usabilidade Diretriz: feedback Beta Test
Importação de samples	Permitir que o usuário importe seus próprios samples	Entrevistas Diretriz: diversidade Diretriz: configuração
Exportar áudio	Permitir que o áudio gerado no Sketchument seja gravado e exportado	Entrevistas Diretriz: Integração
Objeto de Saída: OSC Out	Adicionar um objeto que permite o envio de mensagens no protocolo OSC	Entrevistas Diretriz: Integração
Objeto de Saída: MIDI Out	Adicionar um objeto que permite o envio de mensagens no protocolo MIDI	Entrevistas Diretriz: Integração
Objeto de Entrada: OSC In	Adicionar um objeto que permite a recepção de mensagens no protocolo OSC	Entrevistas Diretriz: Integração

Objeto de Entrada: MIDI In	Adicionar um objeto que permite a recepção de mensagens no protocolo MIDI	Entrevistas Diretriz: Integração
Importação de patches do PD	Permitir que o usuário importe seu próprio patch do PD	Diretriz: integração Diretriz: configuração
AudioBus	Permitir que o Sketchument se comunique com outros aplicativos através do AudioBus	Diretriz: integração Caixa Morfológica de Funcionalidades
Zoneamento da tela	Definir zonas na tela que determina o funcionamento dos objetos de entrada posicionados sobre ela	Análise da Literatura (McGlynn et al., 2012)
WIST	Permitir que o Sketchument se comunique com outros aplicativos através do protocolo WIST	Diretriz: integração Caixa Morfológica de Funcionalidades
Gravação de loops	Permitir que o usuário possa gravar loops no Sketchument	Entrevistas Caixa Morfológica de Funcionalidades
Objetos tocáveis por desenho do usuário	Permitir o usuário desenhar a forma do seu objeto tocável	Diretriz: diversidade Diretriz: configuração
Objeto sequenciador	Adicionar um objeto que dispara automática e sequencialmente	Análise da Literatura (Reactable, PD) Entrevistas
Objeto Entrada: Gestos Customizados	Um objeto no qual o usuário grava um determinado gesto e permite ter uma ação associada	Análise da Literatura (Bevilacqua et al., 2005) Diretriz: configuração
Objeto de Saída: Biblioteca de Acordes	Objeto que permite escolher o acorde a ser disparado	Diretriz: diversidade
Objeto de Entrada: Velocidade do Dedo	Adicionar um objeto de entrada que detecta a velocidade de movimentação de um ponto que passe sobre ele	Diretriz: diversidade Diretriz: configuração Análise dos Aplicativos Musicais (GarageBand)
Objeto de Entrada: slide	Objeto inspirado nos faders ou sliders presentes em equipamentos de som	Entrevistas
Objeto de Entrada: Microfone	Adicionar o microfone como objeto de entrada (ex: variáveis volume e frequência)	Diretriz: diversidade Diretriz: configuração
Objeto de Entrada: Intensidade do Toque	Inserir objeto que mede a intensidade do toque do usuário na tela	Análise de Aplicativos Musicais (GarageBand, Orphion)
Objeto de Entrada: Câmera	Adicionar a câmera como objeto de entrada (ex: detectar cor ou quantidade de luz)	Diretriz: diversidade Diretriz: configuração
Objeto de Entrada: Acelerômetro	Adicionar o acelerômetro como objeto de entrada	Diretriz: diversidade Diretriz: configuração

Objeto de Entrada: Sensor Magnético	Adicionar o sensor magnético como objeto de entrada	Diretriz: diversidade Diretriz: configuração
Modelos físicos de instrumentos	Colocar cordas como objetos de entrada, braços de violão	Questionário
Criar um modo de detecção automática de áreas	Quando o usuário ativar o modo de detecção de áreas, onde o usuário tocar será definido como objeto tocável	Diretriz: configuração
Adicionar operadores	Permitir que os dados de entrada sejam manipulados através de operadores	Análise da Literatura Diretriz: configuração

18. Conclusão

Neste trabalho, foi apresentado o conceito de ambiente de experimentação para a construção de instrumentos musicais digitais como uma ferramenta para auxílio da etapa de prototipação do processo de desenvolvimento de tais instrumentos. Como prova de conceito, foi desenvolvido um aplicativo para iPad chamado Sketchument, cujas diferentes versões serviram de base para vários tipos de avaliação e posteriores modificações e melhoramentos, trabalhando-se em um processo cíclico.

O princípio fundamental do Sketchument é lidar com a problemática de concepção e de construção de um DMI, dando a um público de músicos e artistas um ambiente de prototipação adequado e atrativo, tornando-os os *luthiers* dos seus próprios instrumentos. Apesar de preliminares, os resultados são encorajadores, pois apontam que a combinação da cultura *do it yourself* e o movimento *maker* com alguns princípios de design, como a prototipação, é uma trilha interessante a ser explorada no desenvolvimento de DMIs, tocando particularmente no seu ponto central que é o mapeamento.

As informações contidas neste Projeto podem ser de grande valia para pesquisadores que queriam desenvolver novos instrumentos musicais digitais, seja especificamente para superfícies multitoque ou para outras formas de interação.

18.1. Críticas ao Processo

Abaixo, serão feitos alguns comentários e críticas à metodologia utilizada e ao estágio atual do ambiente de experimentação baseado nos métodos de avaliação.

Dado que a fase de inspiração foi baseada na experiência pessoal, a fase de prototipação se mostrou adequada como sequência imediata, já que o protótipo funcionou como artefato concreto de comunicação com potenciais usuários sobre as ideias iniciais do aplicativo, gerando críticas, melhoramentos e modificações para a versão do protótipo sendo avaliada.

Devido a restrições de tempo, o protótipo não pôde passar por mais ciclos de implementação, sendo boa parte dos resultados das etapas de avaliação aproveitada para a formação da lista ordenada de funcionalidades.

A análise de aplicativos musicais para iPad e a resultante caixa morfológica de funcionalidades fornecem uma ferramenta visual de possíveis escolhas a serem implementadas em projetos musicais em geral. Desta forma, a escolha pode acontecer de forma estruturada, percorrendo as categorias apresentadas e dando mais clareza das opções. Este espaço concreto serviu de base para a geração de alternativas a alguns problemas expostos nas etapas de avaliação, por exemplo, frente ao problema de integração do iPad a programas voltados para programação musical para plataforma desktop, pode-se associar alternativas como exportação de arquivos de áudio em formatos padrão (WAV, AIFF e MP3) e a importação desses arquivos nos

programas existentes ou a conexão do aplicativo de iPad ao programa de computador através de protocolos MIDI ou OSC. Essas opções de escolha estão apresentadas nas categorias de “Tipos de Arquivos Exportáveis” e “Conexões com o computador” na caixa morfológica de funcionalidades.

A análise de aplicativos concorrentes foi importante para perceber a oportunidade de desenvolver uma interface modular para gestos em superfícies multitoque, já que os concorrentes, apesar de oferecerem a possibilidade de customização da interface, apresentam apenas opções de interação através de elementos gráficos. Tal análise resultou na caixa morfológica da interface, que apresenta os mesmos benefícios de escolha estruturada da caixa morfológica de funcionalidades.

O questionário online contendo questões abertas serviu para dois propósitos: primeiro, avaliar o protótipo versão 2 a partir de críticas e sugestões de modificação; segundo, levantar referências sobre o contexto do aplicativo, como aplicativos relacionados, cenários de uso e potenciais usuários. Através da distribuição do questionário online, obteve-se uma grande variedade de respostas, gerando boa quantidade de resultados aproveitáveis. Ao mesmo tempo, esta quantidade de respostas demandou muito tempo de processamento e extração de resultados, sendo preciso filtrar os participantes para adequar a quantidade de respostas ao tempo disponível. Apesar de ter diminuído o espaço de respostas, os dados obtidos foram satisfatórios e várias ideias puderam ser aproveitadas.

Apesar de a entrevista ter fornecido relevantes feedbacks para o projeto, tendo sido realizada em apenas um momento com os entrevistados, as respostas tiveram um foco mais abrangente, perdendo, por conta do tempo, alguns detalhes. Para trabalhos futuros, mostra-se mais adequada a divisão da entrevista em dois momentos distintos: uma primeira entrevista investigativa com o objetivo de levantar informações para construir um espaço de oportunidades de projeto e uma segunda entrevista voltada para a avaliação do protótipo, coletando informações profundas e relevantes a partir das impressões dos participantes.

Na entrevista de avaliação, além de coletar as impressões e comentários como base de resultados qualitativos, pode ser feito um teste de usabilidade baseado em tarefas no final da entrevista. Esta abordagem foi experimentada informalmente em parte dos entrevistados e gerou resultados interessantes de como o usuário pensa o modelo da interface com a qual está interagindo. Como um dos resultados, destaca-se a forma de adição de objetos na versão 2 do protótipo, parte dos usuários não conseguiram entender rapidamente que precisam tocar na tela para o menu de contexto aparecer com as opções de objetos. Este problema de usabilidade foi detectado em um teste informal e sanado na versão 3 pela adição das abas laterais retráteis, apresentando objetos de entrada à esquerda e objetos de saída à direita.

A partir do processamento dos resultados das entrevistas e dos questionários, puderam ser descritos alguns perfis de potenciais usuários para o ambiente de experimentação e cenários

de uso. Juntamente com as caixas morfológicas e as diretrizes de projeto, os perfis e cenários servem de ferramentas para geração de novas funcionalidades para futuros protótipos.

A fase de avaliação baseada no beta test, apesar da participação de poucos usuários, forneceu resultados aprofundados sobre a versão 3 do protótipo. Esta profundidade se dá pelo nível de detalhes dos feedbacks, já que os usuários estavam testando o aplicativo em seu próprio ambiente, com o seu próprio dispositivo, sem a interferência de um espaço de testes ou da presença de um avaliador. Além disso, com os dados quantitativos, tem-se uma visão geral do nível de engajamento dos usuários com o aplicativo e constatou-se que a maioria dos usuários passou pouco tempo experimentando o ambiente. Esse resultado pode ser associado ao fato do protótipo ter pouca diversidade de objetos, não despertando interesse para uma maior utilização. Os checkpoints registrados podem servir de base para futuros processamentos no intuito de melhor entender o comportamento do usuário a partir da reconstrução dos passos percorridos. Com isso, dada uma tarefa, pode-se estimar que solução melhor se aplica para tornar a realização desta tarefa mais eficiente.

Finalmente, a execução dos métodos gerou uma grande quantidade de informações para o projeto do ambiente de experimentação. O próximo passo é processar essas informações em forma de funcionalidades para tornar o protótipo cada vez mais próximo às diretrizes do Projeto.

18.2. Considerações sobre o Protótipo Atual

O atual protótipo do Sketchument apresenta ainda poucas alternativas para uma experimentação mais expressiva. Desta forma, o processo de construção de DMIs neste ambiente de experimentação ainda não tem a capacidade de gerar instrumentos interessantes e envolventes para o usuário. Apesar disso, o processo de concepção e desenvolvimento do Sketchument trilhou caminhos maduros para a sua evolução, agregando informação e validando inspirações provenientes da experiência pessoal do autor.

Não se cria nada do zero, sempre é preciso um referencial cultural que associado a novas oportunidades gera algo inovador. O problema é quando não é feita a reflexão sobre paradigmas existentes e na sua adequação a novas situações. É fundamental entender os potenciais das novas interfaces e assim extrair o máximo de suas peculiaridades para gerar novas formas de expressão musical que abram novos horizontes para a música.

A tecnologia frequentemente facilita alguns ofícios, fornecendo ao público leigo a possibilidade experimentar por diversas áreas. Um exemplo disso é a máquina fotográfica, que permitiu a pessoas, que não dominavam a técnica da pintura, registrassem a realidade nas fotos. Com o Sketchument, espera-se que a luteria seja experimentada por usuários que não dominem as técnicas de construção de instrumentos acústicos ou digitais, gerando novos instrumentos, novas formas de expressão e novos gêneros musicais. Os resultados obtidos neste Projeto são

encorajadores, indicando que há muito caminho ainda a ser percorrido, porém em uma direção promissora.

18.3. Trabalhos Futuros

Como trabalho futuro, pensou-se em incorporar a etapa de avaliação diretamente no ambiente de experimentação através da distribuição online dos protótipos e a avaliação baseada na comunidade. Além disso, com a lista ordenada de funcionalidades para implementação, a versão atual do protótipo será melhorada com o intuito de lançar o aplicativo o mais breve possível na App Store.

Com uma ferramenta de experimentação e avaliação, pode-se pensar em um conjunto de métodos estruturados para o desenvolvimento de DMIs, ou seja, uma metodologia de design para tais instrumentos baseada fortemente no conceito modular e combinatório.

Para aumentar a diversidade, novos módulos precisam ser implementados e em relação à plataforma iPad boa parte das novas funcionalidades já estão listadas. Porém, pensando no projeto de forma mais ampla, pode-se permitir a integração de formas de interação gestual baseada em outros dispositivos, como sensores genéricos através do Arduino, Kinect, WiiMote, PS Move etc. A interface poderá ser descentralizada, no computador ou no iPad e para facilitar a integração entre as entradas, seriam implementados objetos baseado em OSC ou MIDI e um middleware responsável por traduzir os dados da interface para um protocolo conhecido pelo Sketchument. Os benefícios de um sistema de mapeamento destes resultam principalmente em melhor diversidade e integração.

Uma alternativa para se alcançar este objetivo é pensar em uma forma de integrar o Sketchument com sistemas de mapeamento já existentes como é o caso do *libmapper*. O Sketchument poderia ser implementado como um monitor de uma rede Mapper e funcionar como controle de entrada além de interface para fazer e editar mapeamentos.

Porém, antes de aumentar a complexidade do aplicativo, uma abordagem mais simples pode ser seguida para o lançamento na App Store. Sabe-se da tendência de especialização, na qual cada aplicativo tem sua função bem definida e a tendência de crescimento de escopo através de atualizações nas funcionalidades do aplicativo. Seguindo essas tendências, para o lançamento Sketchument, pensa-se em fazer uma versão “recortada”, ou seja, apenas com as funcionalidades mais representativas do sistema. No caso, pode-se implementar apenas um controlador gestual multitoque com características modulares e customizáveis: será apresentado um conjunto de gestos e um conjunto de mensagens de saída configuráveis baseados em OSC ou MIDI. Assim, o usuário irá usar o aplicativo inicialmente como controlador e, com o tempo, novas funcionalidades serão adicionadas por atualização baseadas na avaliação de recepção do aplicativo, como as revisões do usuário, quantidade de downloads etc.

Como parte dos trabalhos futuros, a implementação para o sistema operacional Android é uma das possibilidades. Além disso, parcerias com músicos, trazendo-os para o desenvolvimento de novos instrumentos com o ambiente e levantando mais requisitos propostos por eles, é uma abordagem poderosa no avanço do projeto.

Enfim, espera-se que no futuro, o uso do Sketchument permita a construção de instrumentos musicais digitais da mesma forma de hoje se constroem estruturas e robôs com peças de Lego, deixando o processo lúdico e experimental. Assim, abrir novas possibilidades de interação musical e possivelmente ser base para o desenvolvimento de novos gêneros e novas músicas.

Referências

- Anderson, C. (2012). *Makers: The New Industrial Revolution* (p. 272). Random House Business.
- Apple Inc. (n.d.). Apple Developer. Retrieved from <http://developer.apple.com>
- Arduino.cc. (n.d.). Retrieved October 31, 2012, from <http://arduino.cc/>
- Baldwin, C., & Hippel, E. Von. (2010). Modeling a paradigm shift: From producer innovation to user and open collaborative innovation. *Harvard Business School Finance ...*, (November 2009), 1–37.
- Barbosa, J., Calegario, F., Magalhães, F., Teichrieb, V., Cabral, G., & Ramalho, G. (2011). Towards an evaluation methodology for digital music instruments considering performer's view: a case study. In *Proceedings of 13th Brazilian Symposium on Computer Music*. Retrieved from <http://www.cin.ufpe.br/~fcac/Vers?o-Final-SBCM.pdf>
- Barbosa, J., Calegario, F., Teichrieb, V., & Ramalho, G. (2012). Considering Audience's View Towards an Evaluation Methodology for Digital Musical Instruments. In *Proceedings of the International Conference on New Interfaces for Musical Expression*. Retrieved from http://www.eecs.umich.edu/nime2012/Proceedings/papers/174_Final_Manuscript.pdf
- Bencina, R., & Wilde, D. (2008). Gesture-Sound Experiments : Process and Mappings. *Technology*.
- Bevilacqua, F., Müller, R., & Schnell, N. (2005). MnM: a Max/MSP mapping toolbox. In *Proceedings of the International Conference on New Interfaces for Musical Expression*.
- Birnbaum, D., Fiebrink, R., Malloch, J., & Wanderley, M. M. (2005). Towards a dimension space for musical devices. In *Proceedings of the International Conference on New Interfaces for Musical Expression* (pp. 192–195).
- Bogers, M., Afuah, A., & Bastian, B. (2010). Users as Innovators: A Review, Critique, and Future Research Directions. *Journal of Management*, 36(4), 857–875.
doi:10.1177/0149206309353944
- Bott, J. N., Crowley, J. G., & LaViola, J. J. (2009). One Man Band: A 3D Gestural Interface for Collaborative Music Creation. *2009 IEEE Virtual Reality Conference*, 273–274.
doi:10.1109/VR.2009.4811051

- Boulanger, R. (2000). *The Csound Book: Perspectives in Software Synthesis, Sound Design, Signal Processing, and Programming*. Cambridge, Massachusetts: MIT press.
- Cadoz, C. (1994). Le geste canal de communication homme-machine. *Sciences Informatiques, numéro spécial: Interface homme-machine*, 13(1), 31–61.
- Carse, J. (1986). *Finite and Infinite Games* (p. 160). Free Press.
- Chadabe, J. (1975). The Voltage-controlled synthesizer. In J. Appleton (Ed.), *The Development and Practice of Electronic Music*. Prentice-Hall.
- Chagas, P. (1992). Le MIDI et la musique électronique Quelques remarques esthétiques et techniques. *Revue Informatique Et Statistique Dans Les Sciences Humaines*, 28, 15.
- Cont, A., Coduys, T., & Henry, C. (2004). Real-time gesture mapping in pd environment using neural networks. In *Proceedings of the International Conference on New Interfaces for Musical Expression*.
- Cook, P. R. (2002). *Real Sound Synthesis for Interactive Applications* (p. 263). A K Peters/CRC Press.
- Csikszentmihalyi, M., & Csikszentmihalyi. (2002). *Flow: The Psychology of Happiness: The Classic Work on How to Achieve Happiness* (p. 320). Rider.
- Dougherty, D., & Frauenfelder, M. (2005). MAKE Magazine. Retrieved October 31, 2012, from <http://makezine.com/>
- Dow, S. (2011). How prototyping practices affect design results. *Interactions*, 18(3), 54. doi:10.1145/1962438.1962451
- Fels, S., Gadd, A., & Mulder, A. (2002). Mapping transparency through metaphor: towards more expressive musical instruments. *Organised Sound*, 7(02), 109–126. doi:10.1017/S1355771802002042
- Fischer, G. (2003). Meta-design: beyond user-centered and participatory design. In J. A. Jacko & C. Stephanidis (Eds.), *Human-Computer Interaction: Theory and Practice (Part 1)* (pp. 88–92). Taylor & Francis Group. Retrieved from <http://books.google.com.br/books?id=G43pknjDsAgC>
- Fry, B., & Reas, C. (n.d.). Processing.org. Retrieved October 31, 2012, from <http://processing.org/>

- Gerber, E., & Carroll, M. (2011). The psychological experience of prototyping. *Design Studies*, 33(1), 64–84. doi:10.1016/j.destud.2011.06.005
- Goldstein, M. (1998). Gestural coherence and musical interaction design. In *IEEE International Conference on Systems, Man, and Cybernetics* (pp. 1076–1079). Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=727834
- Gray, D., Brown, S., & Macanuso, J. (2010). *Gamestorming: A Playbook for Innovators, Rulebreakers, and Changemakers* (p. 290). O'Reilly Media. Retrieved from <http://www.amazon.co.uk/Gamestorming-Playbook-Innovators-Rulebreakers-Changemakers/dp/0596804172>
- Gurevich, M., & Cavan Fyans, a. (2011). Digital Musical Interactions: Performer–system relationships and their perception by spectators. *Organised Sound*, 16(02), 166–175. doi:10.1017/S1355771811000112
- Hartmann, B. (2009). Gaining design insight through interaction prototyping tools, (September). Retrieved from <http://bjoern.org/dissertation/hartmann-diss.pdf>
- Hunt, A., Wanderley, M. M., & Paradis, M. (2003). The Importance of Parameter Mapping in Electronic Instrument Design. *Journal of New Music Research*, 32(4), 429–440. doi:10.1076/jnmr.32.4.429.18853
- IDEO. (2003). *IDEO Method Cards: 51 Ways to Inspire Design* (p. 102). William Stout. Retrieved from <http://www.amazon.co.uk/IDEO-Method-Cards-Inspire-Design/dp/0954413210>
- IDEO. (2011). *Human-Centered Design Toolkit: An Open-Source Toolkit to Inspire New Solutions in the Developing World* (p. 200). Authorhouse. Retrieved from <http://www.amazon.co.uk/Human-Centered-Design-Toolkit-Open-Source-Developing/dp/0984645705>
- Isbister, K., & Schaffer, N. (2008). *Game Usability: Advice from the experts for advancing the player experience*. CRC Press LLC.
- Jácome, J. (2007). *Sistemas Interativos de Tempo Real para Processamento Audiovisual Integrado*. Universidade Federal de Pernambuco.
- Jordà, S. (2003). Interactive music systems for everyone: exploring visual feedback as a way for creating more intuitive, efficient and learnable instruments. In *Proceedings of the Stockholm Music Acoustics* (Vol. 2003, pp. 1–6). Retrieved from [http://infodate.nctu.edu.tw/teaching/techart/assi/94/9342804%E9%9F%B3%E6%A8%82%](http://infodate.nctu.edu.tw/teaching/techart/assi/94/9342804%E9%9F%B3%E6%A8%82%141)

E6%96%87%E7%8D%BB(%E9%83%AD%E6%80%A1%E5%A9%B7)/Interactive Music Systems for Everyone.pdf

- Jordà, Sergi. (2004a). Digital instruments and players: part I---efficiency and apprenticeship. *Proceedings of the International Conference on New Interfaces for Musical Expression*, 59–63. Retrieved from <http://dl.acm.org/citation.cfm?id=1085898>
- Jordà, Sergi. (2004b). Digital Instruments and Players : Part II – Diversity , Freedom and Control. In *Proceedings of the 2004 International Computer Music Conference*. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.75.1249&rep=rep1&type=pdf>
- Jordà, Sergi. (2005). *Digital Lutherie*. Universitat Pompeu Fabra. Retrieved from <http://dialnet.unirioja.es/servlet/tesis?codigo=19509>
- Jordà, Sergi, Geiger, G., Alonso, M., & Kaltenbrunner, M. (2007). The reactTable: exploring the synergy between live music performance and tabletop tangible interfaces. In *Proceedings of the 1st International Conference on Tangible and Embedded Interaction*. Retrieved from <http://dl.acm.org/citation.cfm?id=1226998>
- Jordà, Sergi, Kaltenbrunner, M., Geiger, G., & Bencina, R. (2005). The reactable*. *Proceedings of the International Computer Music Conference*, 2–5. Retrieved from <http://www.mtg.upf.es/system/files/publications/reactTableKickOff2003.pdf>
- Kiefer, C., Collins, N., & Fitzpatrick, G. (2008). HCI methodology for evaluating musical controllers: A case study. In *Proceedings of the International Conference on New Interfaces for Musical Expression*. Retrieved from <http://nime2008.casapaganini.org/documents/Proceedings/Papers/193.pdf>
- Kirn, P. (2011). *Keyboard Presents the Evolution of Electronic Dance Music* (p. 256). Backbeat Books. Retrieved from <http://www.amazon.co.uk/Keyboard-Presents-Evolution-Electronic-Dance/dp/1617130192>
- Knuth, D. E. (1986). *The Metafont Book*. Addison Wesley. Retrieved from <http://www.amazon.co.uk/The-Metafont-Book-Computers-Typesetting/dp/0201134446>
- Koster, R. (2005). *Theory of Fun for Game Design* (p. 240). PARAGLYPH PRESS. Retrieved from <http://www.amazon.co.uk/Theory-Game-Design-Raph-Koster/dp/1932111972>
- Lee, J. C. (2008). Hacking the Nintendo Wii Remote. *IEEE Pervasive Computing*, 7(3), 39–45. doi:10.1109/MPRV.2008.53

- Madden, R. (2012). *Analysis and Comparison of Kinect, Wii-Mote, and PlayStation Move*. Robertjamesmadden.com. University of Stafford. Retrieved from <http://robertjamesmadden.com/wp-content/uploads/2012/10/RJM-Masters-Paper-Final.doc>
- Maestracci, B., Frechin, J., & Petrevski, U. (2011). Modular Musical Objects Towards Embodied Control Of Digital Music. In *TEI* (pp. 1–4). Retrieved from <http://articles.ircam.fr/textes/Rasamimanana11a/index.pdf>
- Magnusson, T., & Mendieta, E. (2007). The acoustic, the digital and the body: A survey on musical instruments. *Proceedings of the International Conference on New Interfaces for Musical Expression*, 94–99. Retrieved from <http://dl.acm.org/citation.cfm?id=1279757>
- Malloch, J. (2008). *A Consort of Gestual Musical Controllers: Design, Construction and Performance*. McGill University.
- Malloch, J., Birnbaum, D., Sinyor, E., & Wanderley, M. M. (2006). Towards a New Conceptual Framework for Digital Musical Instruments. In *Proceedings of the 9th International Conference on Digital Audio Effects* (pp. 49–52). Retrieved from http://www.dafx.ca/proceedings/papers/p_049.pdf
- Malloch, J., Sinclair, S., & Wanderley, M. M. (2008). A network-based framework for collaborative development and performance of digital musical instruments. *Proceedings of the International Conference On Sound and Music Computing*, 401–425. Retrieved from <http://www.springerlink.com/index/34n516476m177329.pdf>
- Marshall, M. T. (2010). *Physical interface design for digital musical instruments*. MCGILL UNIVERSITY. Retrieved from <http://gradworks.umi.com/NR/56/NR56932.html>
- Maurya, A. (2012). *Running Lean: Iterate from Plan A to a Plan That Works* (p. 240). O'Reilly Media. Retrieved from <http://www.amazon.co.uk/Running-Lean-Iterate-Works-OReilly/dp/1449305172>
- McCartney, J. (2002). Rethinking the computer music language: SuperCollider. *Computer Music Journal*, 26(4), 61–68.
- McGlynn, P. (2011). Towards more effective mapping strategies for digital musical instruments. In *Proceedings of the 9th Annual Linux Audio Conference*. Retrieved from <http://www.cin.ufpe.br/~fcac/pesquisasCln/41.pdf>
- McGlynn, P., Lazzarini, V., Delap, G., & Chen, X. (2012). Recontextualizing the Multi-touch Surface. *Proceedings of the International Conference on New Interfaces for Musical*

- Expression*. Retrieved from http://www.eecs.umich.edu/nime2012/Proceedings/papers/132_Final_Manuscript.pdf
- Microsoft Corporation. (2012). *Human Interface Guidelines: Kinect for Windows v1.5.0*. Retrieved from <http://www.microsoft.com/en-us/kinectforwindows/develop/learn.aspx>
- Miranda, E. R. (1998). *Computer Sound Synthesis for the Electronic Musician* (p. 240). Butterworth-Heinemann.
- Miranda, E. R., & Wanderley, M. M. (2006). *New Digital Musical Instruments: Control and Interaction Beyond the Keyboard*. Retrieved from http://books.google.com/books?hl=en&lr=&id=CGEwXZ7hcloC&oi=fnd&pg=PR11&dq=New+Digital+Musical+Instruments:+Control+and+Interaction+Beyond+the+Keyboard&ots=rZgv3GcE-p&sig=MpLrXUKPFVgxb_RrqCojYS8Sins
- Montag, M., Sullivan, S., Dickey, S., & Leider, C. (2011). A Low-Cost , Low-Latency Multi-Touch Table with Haptic Feedback for Musical Applications. *Proceedings of the International Conference on New Interfaces for Musical Expression*, (June), 8–13.
- Mulder, A. (2000). Towards a choice of gestural constraints for instrumental performers. In M. Wanderley & M. Battier (Eds.), *Trends in Gestural Control of Music* (pp. 315–335). IRCAM—Centre Pompidou.
- Neves, A., Campos, F., Campello, S., Castillo, L., Barros, S., & Aragão, I. (2008). XDM - Métodos Extensíveis de Design. In *Anais Do 8º Congresso Brasileiro De Pesquisa e Desenvolvimento Em Design* (pp. 249–259).
- Nielsen, J. (1993). *Usability Engineering* (p. 362). Morgan Kaufmann Publishers In. Retrieved from <http://www.amazon.co.uk/Usability-Engineering-Interactive-Technologies-Nielsen/dp/0125184069>
- Noble, J. (2012). *Programming Interactivity* (p. 728). O'Reilly Media. Retrieved from <http://www.amazon.co.uk/Programming-Interactivity-Joshua-Noble/dp/144931144X>
- Norman, D. (2006). Why doing user observations first is wrong. *interactions*, 50–51. Retrieved from <http://dl.acm.org/citation.cfm?id=1142199>
- Norman, D. A. (2004). *Emotional Design*. (B. Books, Ed.) *Ubiquity* (Vol. 4, p. 1). Basic Books. doi:10.1145/966012.966013

- Nort, D. Van, & Wanderley, M. (2006). The LoM Mapping Toolbox for Max/MSP/Jitter. In *Proceedings of the International Computer Music Conference*. Retrieved from http://idmil.org/_media/wiki/icmc2006_vannortwand.pdf?id=publications&cache=cache
- NUI Group Authors. (2009). *Multitouch Technologies*. NUI Group. Retrieved from http://bramvandeputte.files.wordpress.com/2011/04/multi-touch_technologies_v1-01.pdf
- O'Modhain, S. (2011). A framework for the evaluation of digital musical instruments. *Computer Music Journal*, 35(1), 28–42. Retrieved from http://muse.jhu.edu/journals/computer_music_journal/v035/35.1.o-modhain.html
- O'Modhain, S., & Chafe, C. (2000). Incorporating haptic feedback into interfaces for music applications. In *Proceedings of the International Symposium on Robotics with Applications, World Automation Conference*.
- Orio, N., Schnell, N., & Wanderley, M. M. (2001). Input devices for musical expression: borrowing tools from HCI. In *Proceedings of the International Conference on New Interfaces for Musical Expression* (pp. 1–4). National University of Singapore. Retrieved from <http://portal.acm.org/citation.cfm?id=1085157>
- Overholt, D. (2009). The Musical Interface Technology Design Space. *Organised Sound*, 14(02), 217. doi:10.1017/S1355771809000326
- Preece, J., Rogers, Y., & Sharp, H. (2002). *Interaction Design: Beyond Human-Computer Interaction*. (Wiley Sons, Ed.) *Design* (Vol. 18, pp. 68–68). Wiley. doi:10.1016/S0010-4485(86)80021-5
- Puckette, M. (2002). Max at seventeen. *Computer Music Journal*. Retrieved from <http://www.mitpressjournals.org/doi/abs/10.1162/014892602320991356>
- Rovan, J. B., Wanderley, M. M., Dubnov, S., & Depalle, P. (1997). Instrumental gestural mapping strategies as expressivity determinants in computer music performance. In *Proceedings of Kansei-The Technology of Emotion Workshop* (pp. 3–4). Citeseer. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.52.4788&rep=rep1&type=pdf>
- Rudraraju, V. (2011). *A Tool for Configuring Mappings for Musical Systems using Wireless Sensor Networks*.
- Ryan, J. (1991). Some remarks on musical instrument design at STEIM. *Contemporary music review*. Retrieved from <http://www.tandfonline.com/doi/abs/10.1080/07494469100640021>

- Saffer, D. (2008). *Designing Gestural Interfaces* (p. 247). O'Reilly Media, Inc. Retrieved from <http://books.google.com/books?hl=en&lr=&id=iTsZ5cg7gegC&pgis=1>
- Schloss, W. (1990). Recent advances in the coupling of the language Max with the Mathews/Boie Radio Drum. In *Proceedings of the International Computer Music Conference*. Retrieved from http://hapticity.net/pdf/nime2005_192-works_cited/ICMC1990_Schloss.pdf
- Schmeder, A., & Freed, A. (2008). uosc: The open sound control reference platform for embedded devices. In *Proc of the 8th International Conference on New Interfaces for Musical Expression (NIME)*. Citeseer.
- Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., ... Blake, A. (2011). Real-time human pose recognition in parts from single depth images. In *CVPR*. Retrieved from <http://www.stat.osu.edu/~dmsl/BodyPartRecognition.pdf>
- Silva, J. (2012). *Avaliando Interfaces Gestuais para Prática de Instrumentos Virtuais de Percussão*. Universidade Federal de Pernambuco.
- Sinyor, E. (2006). *Digital Musical Instruments: A Design Approach Based on Moving Mechanical Systems*.
- Steiner, H. (2005). Building your own instrument with Pd. In *Proceedings of the 1st International Pd Conference*. Retrieved from https://gem.iem.at/pd/pd/community/conventions/convention04/lectures/steiner-pd_instruments/BuildingYourOwnInstrumentWithPd.pdf
- Stowell, D., Robertson, A., Bryan-Kinns, N., & Plumbley, M. D. (2009). Evaluation of live human-computer music-making: Quantitative and qualitative approaches. *International Journal of Human-Computer Studies*, 67(11), 960–975. doi:10.1016/j.ijhcs.2009.05.007
- Thomke, S. (2003). *Experimentation Matters: Unlocking the Potential of New Technologies for Innovation*. Harvard Business School Press Books. Harvard Business School Press. Retrieved from <http://search.ebscohost.com/login.aspx?direct=true&db=buh&AN=11212920>
- Vassão, C. A. (2010). *Metadesing: Ferramentas, Estratégias e Ética para a Complexidade*.
- Wanderley, M. M. (2006). Instrumentos Musicais Digitais: Gestos, Sensores e Interfaces. In *Em Busca Da Mente Musical* (Vol. 60). Editora da Universidade Federal do Paraná.

- Wanderley, M. M., & Depalle, P. (2004). Gestural Control of Sound Synthesis. *Proceedings of the IEEE*, 92(4), 632–644. doi:10.1109/JPROC.2004.825882
- Wanderley, M. M., & Orio, N. (2002). Evaluation of Input Devices for Musical Expression: Borrowing Tools from HCI. *Computer Music Journal*, 26(3), 62–76.
- Wang, G. (2008). *The Chuck Audio Programming Language: A Strongly-timed and On-the-fly Environmentality*. Princeton University.
- Warfel, T. Z. (2009). *Prototyping: A Practitioner's Guide* (p. 197). Rosenfeld Media. Retrieved from <http://www.amazon.co.uk/Prototyping-Practitioners-Todd-Zaki-Warfel/dp/1933820217>
- Wessel, D., & Wright, M. (2002). Problems and prospects for intimate musical control of computers. *Computer Music Journal*, 26(3), 11–22. doi:10.1162/014892602320582945
- Wessel, D., Wright, M., & Schott, J. (2002). Intimate musical control of computers with a variety of controllers and gesture mapping metaphors. In *Proceedings of the International Conference on New Interfaces for Musical Expression* (pp. 1–3). Retrieved from <http://portal.acm.org/citation.cfm?id=1085213>
- Wobbrock, J., Morris, M., & Wilson, A. (2009). User-defined gestures for surface computing. In *CHI*. New York, New York, USA: ACM Press. doi:10.1145/1518701.1518866
- Zoran, A. (2011). The 3D Printed Flute: Digital Fabrication and Design of Musical Instruments. *Journal of New Music Research*, 40(4), 379–387.

19. Anexo I: Arquitetura do Sketchument

O desenvolvimento de aplicativo para a plataforma iOS baseia-se no padrão de arquitetura de projetos chamado de Modelo-Visualização-Controlador (MVC). Este padrão separa de modo bem definido os dados, ou modelos, da forma como eles são apresentados para o usuário na interface, ou visualização. Isto permite a criação de aplicativos mais complexos e a proteção da manipulação dos dados sem a interferência na interface e da modificação da interface sem alterar a estrutura dos dados. Para garantir esta proteção, existe o controlador, que é uma entidade que se posiciona como intermediário nas trocas de dados entre o modelo e a interface. Além disso, o padrão possibilita o reuso de várias partes do código, por exemplo, para um mesmo modelo, podem existir diferentes tipos de visualização. Este padrão de arquitetura define não apenas o papel de cada entidade, mas também as formas de comunicação entre elas, aplicando restrições quando necessário para proteção da integridade dos dados. (Apple Inc., n.d.)

Em uma aplicação iOS, seguindo o padrão MVC, para cada tela que é desenhada, existe um controlador que faz a ligação das informações provenientes do modelo com a exibição na interface. Dada a grande quantidade de classes já pré-definidas no SDK do iOS, muitos dos controladores já são apresentados em forma de esqueleto, tendo o programador o papel de preencher os detalhes de trocas de dados e tratamento de interação. Para citar os controladores mais usados: *View Controller*, *Navigation Controller*, *Tab Bar Controller*, *TableView Controller*, dentre outros.

Neste Projeto, foram utilizados dois tipos de controladores: *View Controller* que é responsável por gerenciar uma tela genérica, onde os elementos são especificados pelo programador; e um *Navigation Controller* que se responsabiliza pela navegação hierárquica entre telas.

Na Figura 19.1, é apresentado o diagrama de fluxo entre as telas do Sketchument. Na primeira tela à esquerda, há apenas a definição do uso de um *Navigation Controller*. Esta tela não aparece para o usuário e serve apenas para o programador definir a estrutura de navegação do aplicativo.

Na tela do meio, há o “*Edit Mode*” ou Modo de Edição, tela inicial do protótipo v3, onde os objetos de entrada e saída podem ser escolhidos e interligados. Existe uma barra na parte superior com o título da tela “*Edit Mode*” e os botões de *Feedback*, *Ajuda* e “*Go to Play Mode*”. Quem se responsabiliza pelo gerenciamento desta barra é o *Navigation Controller*. A parte inferior a esta barra é de responsabilidade do *View Controller* do Modo de Edição, o que sendo nomeado de *P1EditViewController* (ver Figura 19.2).

A tela à direita é a interface de execução do instrumento, chamada de “*Play Mode*” ou Modo de Execução e gerenciada pelo *P1PlayViewController* (ver Figura 19.3). Pode-se chegar a este modo apertando o botão “*Go to Play Mode*” localizado na interface do Modo de Edição. Nesta tela, os objetos posicionados na interface têm suas ações ativadas para a interação com o

usuário. Quando um objeto é então tocado ou um gesto reconhecido, as ações relacionadas a estes objetos são disparadas.

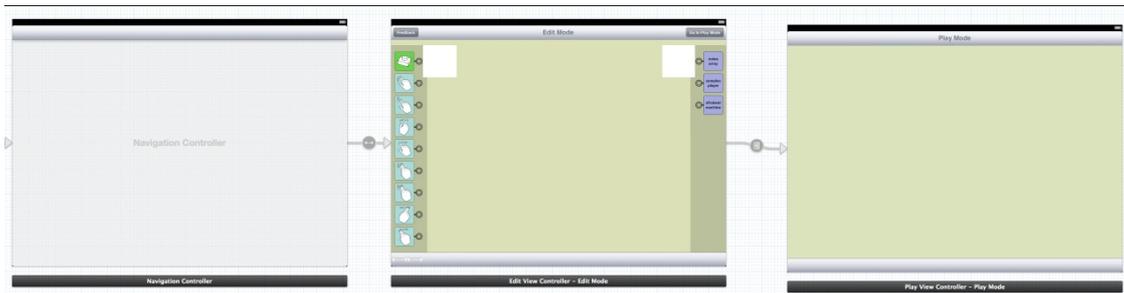


Figura 19.1: Diagrama do Fluxo de Telas do Sketchument

A Figura 19.2 mostra o diagrama de classes do Modo de Edição. Neste primeiro pacote de classes, há o *P1EditViewController* como controlador, o *P1EditView* como visualização e o *P1InputObjectView* e o *P1OutputObjectView* como modelo. Gerenciando a instanciação destes modelos, encontra-se o *P1ObjectFactory*, implementado seguindo o padrão de arquitetura *Factory*, que centraliza a responsabilidade de configuração e instanciação de objetos.

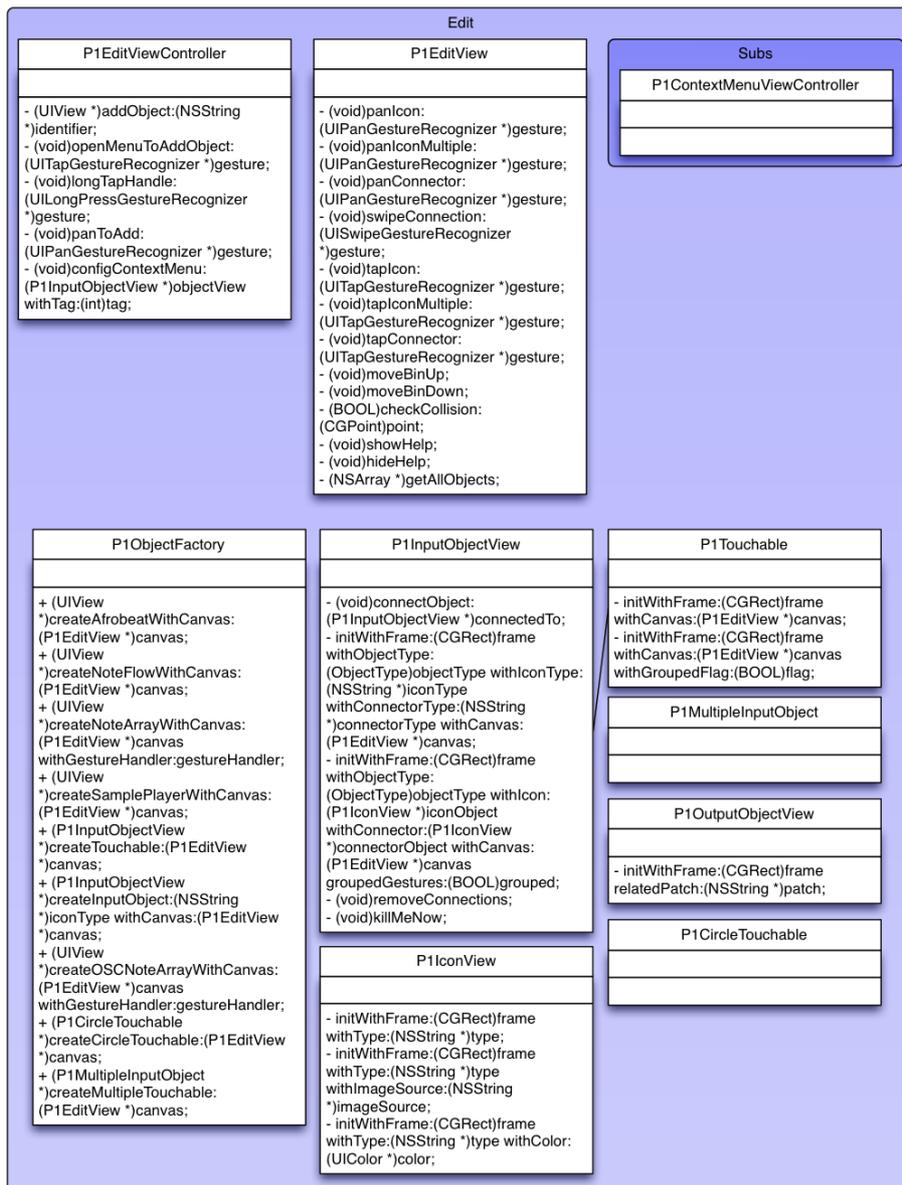


Figura 19.2: Pacote de classes relativo ao modo de edição

A Figura 19.3, mostra o pacote de classes do Modo de Execução com o *P1PlayViewController* como controlador, *P1PlayView* como visualização e o *P1PlayTouchable* e *P1PlayAction* como modelo. Este pacote se apresenta bem mais simples por já importar vários dados relativos aos modelos do Modo de Edição, tendo apenas que encaminhar os dados, ativar as ações e os reconhecedores de gestos para a interface.

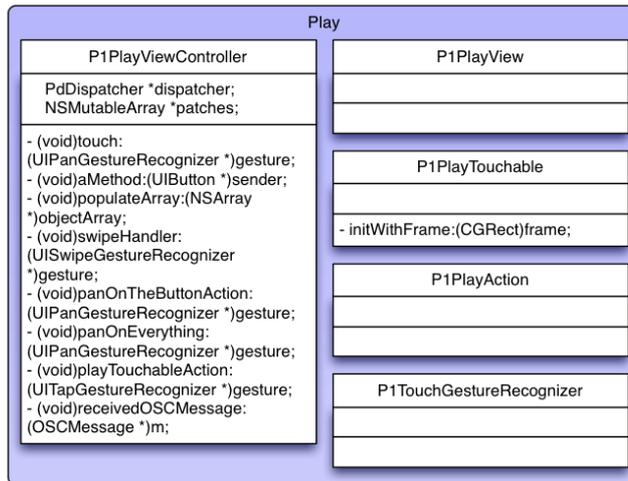


Figura 19.3: Pacote de classes relativo ao modo de execução

A Figura 19.4 apresenta o pacote de classes auxiliares. Neste pacote, visualiza-se a classe TestFlight responsável por gerenciar os serviços de dados de uso providos pela plataforma de distribuição online TestFlight (descrita na seção 16.1).



Figura 19.4: Pacote de classes auxiliares

20. Anexo II: Descrição da Caixa Morfológica de Funcionalidades

20.1. Definição

Nesta Seção, está descrita de forma geral a classe à qual o aplicativo pertence. Seguem abaixo as possíveis classes resultantes da lista de aplicativos analisados.

- **Looper:** são aplicativos que apresentam uma interface de controle de loops, permitindo que o usuário grave um áudio, coloque-o para tocar e grave outro áudio por cima;
- **Sequenciador:** são interfaces que permitem o sequenciamento de determinados sons, podendo seguir uma estrutura de notas musicais pertencentes a alguma escala ou áudios pré-definidos. Tem a característica de serem cíclicos;
- **Sampler:** uma simulação de pads que disparam áudios pré-gravados. Apresentados normalmente em forma de botões na tela do iPad;
- **Sintetizador:** um aplicativo no qual as propriedades do som podem ser modificadas para se produzir novos timbres e sonoridades. Apresenta uma forma simples de execução das notas, apenas para feedback do que está sendo produzido. Geralmente, permite conexões de dados com dispositivos e outros aplicativos baseadas em algum protocolo de comunicação (MIDI, OSC etc.);
- **Controlador:** um aplicativo que serve apenas de interface de controle e que não produz som. Sua função é enviar mensagens para um aplicativo ou dispositivo que sintetize o som seguindo algum protocolo de comunicação (MIDI, OSC etc.);
- **Máquina Generativa:** aplicativos que permitem a geração de melodias, sequências harmônicas e ritmos apenas manipulando algumas variáveis, geralmente apresentadas de forma interativa. É interessante ressaltar que o usuário não tem controle da nota que está sendo tocada, apenas do fluxo dos eventos musicais;
- **Jogo musical:** um aplicativo que apresenta desafios musicais e tem como principal objetivo a diversão do usuário. É importante dizer que tais aplicativos não são voltados para fazer música e sim completar uma série de missões ou estágios;
- **DAW:** sigla para *Digital Audio Workstation*, tais aplicativos trazem para o universo do iPad um ambiente simulado de produção musical com funcionalidades diversas, como gravação, edição e exportação de arquivos de áudio de qualidade;
- **Instrumento Simulado:** aplicativos que têm como objetivo simular instrumentos já existentes na tela multitoque do iPad.

20.2. Plataforma

Este campo descreve as possibilidades de execução do aplicativo, podendo ser apenas no iPhone, apenas no iPad ou em ambos os casos, ou seja, universal.

20.3. Níveis de Controle

Seguindo a definição de (Birnbaum et al., 2005), pode-se classificar os aplicativos em três níveis de controle:

- **Processo:** o mais alto nível dos controles, pelo qual o usuário controla apenas o fluxo de eventos disparados automaticamente pelo aplicativo, por exemplo, quantidade de notas tocadas por período de tempo, altura e duração das notas etc;
- **Evento:** nível de controle pelo qual o usuário dispara eventos musicais, por exemplo, numa interface simulada de um piano na tela multitoque, quando o usuário toca numa tecla, uma nota é executada;
- **Propriedade:** neste nível de controle o usuário lida com as propriedades do som, podendo aplicar efeitos e modificar seus parâmetros na execução dos eventos.

20.4. Funcionalidades Musicais

Este é um conjunto de funcionalidades apresentadas pelo aplicativo que lidam com algumas variáveis musicais de alto nível, como:

- **Definição de tempo:** funcionalidade que permite controlar o andamento da música, geralmente apresentada em unidades de batimentos por minuto (BPM);
- **Definição de escala:** permite a mudança de escala da música executada;
- **Transposição:** permite transpor a música para uma escala diferente;
- **Define acordes:** permite definir quais acordes serão apresentados na interface de controle;
- **Efeitos de som:** apresenta uma lista de efeitos de som que podem ser aplicados na música como um todo ou em partes selecionadas;
- **Metrônomo:** permite ativar ou desativar o metrônomo.

20.5. Interface Musical

Apresenta quais interfaces musicais estão disponíveis no aplicativo.

- **Keyboard:** simulação do teclado de um piano;
- **Drum Pad:** disposição de botões na interface com áudios associados a cada um;
- **Step sequencer:** um sequenciador cíclico que permite a ativação ou desativação dos passos que serão tocados;

- **Smart instruments:** instrumentos musicais simulados que possuem formas de execução pré-gravadas, cabendo ao usuário apenas escolher qual tonalidade quer tocar;
- **Interface Alternativa:** interfaces experimentais que possuem pouca ou nenhuma relação com instrumentos pré-existentes.

20.6. Opções de gravação

Conjunto de funcionalidades relacionadas com a gravação do áudio produzido no aplicativo.

- **Record:** opção de gravar determinado áudio produzido durante a interação com o aplicativo;
- **Mix:** opção de definir o volume áudio gravados;
- **Multitrack:** opção de editar os áudios gravados separadamente por trilhas.

20.7. Edição

Formas de editar as notas dentro do aplicativo.

- **Editor de Notas:** estrutura onde as notas podem ser adicionadas no formato de pentagrama e figuras musicais;
- **Sequenciador de Samples:** trata-se de uma estrutura tabular, na qual as trilhas são representadas nas linhas, o tempo representado nas colunas e as células são as possíveis posições dos samples;
- **Piano Roll:** trata-se de uma estrutura tabular na qual as teclas de pianos são apresentadas em um dos lados e a sequência das linhas representa a execução no tempo. As células da estrutura podem ser ativadas para que a nota correspondente a determinada tecla seja tocada em determinado momento.

20.8. Tipos de Arquivos Exportáveis

Lista os tipos de arquivos que podem ser exportados: WAV, AAC, MP3 e AIFF.

20.9. Tipos de Projetos Exportáveis

Lista os tipos de projetos que podem ser exportados para serem abertos em programas de outras plataformas. Foram identificados projetos para Logic Pro, GarageBand para Mac e Audulus.

20.10. Opções de Compartilhamento

Possibilidades de compartilhamento de elementos do aplicativo.

- **Projeto ou Presets:** são representações de como os elementos do aplicativo estão dispostos e configurados em uma determinada sessão. Estes arquivos podem ser exportados, abertos em programas compatíveis e podem ser compartilhados com a comunidade;
- **Vídeo da Performance:** o vídeo da movimentação dos objetos na tela e da execução de suas ações é registrado e compartilhado com a comunidade;
- **Áudio:** o áudio resultado da interação com o aplicativo é gravado e pode ser compartilhado com a comunidade.

20.11. Formas de Compartilhamento

É a forma como o áudio ou o projeto são compartilhados. Pode ser através de alguma plataforma de armazenamento da nuvem, como é o caso o **iCloud** e do **Dropbox**. Pode ser online e diretamente enviado a uma plataforma de distribuição de música, como é o caso do **SoundCloud**. Pode ser localmente, através do envio do arquivo para ser aberto no **iTunes**.

20.12. Conexão com outros Aplicativos

Forma de conexão com outros aplicativos dentro de um mesmo dispositivo.

- **WIST:** sigla para Wireless Sync-Start Technology. É uma tecnologia desenvolvida pela empresa Korg que permite a sincronização entre dois aplicativos que implementem esta tecnologia, tornando possível que os processos musicais dos dois aplicativos comecem no mesmo instante;
- **MIDI:** aplicativos compatíveis com entradas MIDI são identificados por aplicativos com saídas MIDI. Desta forma, uma conexão pode ser realizada e ocorrer a troca de mensagens;
- **AudioBus:** é uma estratégia recente de transmissão de fluxos puramente sonoros em tempo real entre os aplicativos musicais de um mesmo dispositivo. O ponto positivo é que os aplicativos funcionam como verdadeiros módulos de produção sonora como acontece na realidade de estúdios musicais. Um ponto negativo é que, por ser muito recente, poucos aplicativos incorporaram a compatibilidade com tal sistema, o que, entretanto, não será problema com a afirmação desta comunicação daqui para frente;
- **AudioCopy/AudioPaste:** desenvolvido pela empresa Sonoma Wire Works, funciona como um verdadeiro Copiar/Colar para samples e arquivos de áudio. É uma ótima comunicação entre aplicativos do tipo DAW, que manipulam e sequenciam gravações e samples. Porém, o cenário de uso de performance não é contemplado com tal tecnologia, por sua natureza não permitir o funcionamento em tempo real. Assim, aplica-se melhor a cenários de produção musical usando o iPad;
- **OSC:** forma de comunicação baseada no envio de mensagens de texto entre os aplicativos através da rede, geralmente usando mensagens UDP. Por não apresentar

um padrão bem definido, ainda não é amplamente incorporado à maioria dos aplicativos. Um ponto positivo é a liberdade de criação de protocolos baseados nesta forma de comunicação.

20.13. Conexões com o Computador

Forma de conexão com aplicativos da plataforma desktop. Podem assumir valores como MIDI e OSC que já foram descritos anteriormente. A principal forma de comunicação é através da rede sem fio.

20.14. Conexões com Outros iPhone/iPads

Forma de conexão com outros dispositivos como iPhone ou iPad. Além de formas descritas anteriormente, como **MIDI** e **OSC**, há o **JamSession**, que é uma funcionalidade exclusiva do GarageBand for iPad, e que permite que haja uma comunicação entre dois dispositivos que estejam rodando o citado GarageBand, criando uma Jam Session virtual entre eles.

20.15. Integração com Hardware

Formas de integração com hardwares.

- **Media Kit**: kit disponibilizado pela Apple que converte o conector padrão de iPads e iPhones para entradas USB e de cartão de memória SD. Através da entrada USB, controladores MIDI, que trabalham com um baixo consumo de energia e que possuem saída USB, podem ser usados nos aplicativos que adotam o protocolo MIDI;
- **iRig MIDI**: trata-se de um hardware específico para conexões da plataforma iOS com controladores MIDI. Fornece entrada e saída MIDI simultâneas e MIDI THRU através de cabos DIN de 5 pinos. Além disso, permite que o dispositivo continue conectado à energia USB através de um conector lateral. Compatível com todos os aplicativos que implementam Core MIDI;
- **Line 6 MIDI Mobilizer II**: funciona da mesma forma que o iRig MIDI, porém não permite conexão simultânea do dispositivo iOS à energia. Compatível com todos os aplicativos que implementam Core MIDI;
- **Yamaha IMX1 MIDI Interface**: concorrente direto do Line 6 MIDI Mobilizer II. Compatível com todos os aplicativos que implementam Core MIDI;
- **AKAI Synthstation**: controlador MIDI preparado para acoplagem de iPad na sua parte superior. O iPad funciona como o gerador sonoro do conjunto;
- **iRig**: hardware específico para conexão de instrumentos musicais ao iPad através da entrada de fone de ouvido/microfone. O iRig permite que o usuário tenha o retorno sonoro ao mesmo tempo que o instrumento está conectado.

20.16. Loops Pré-carregados

Define se o aplicativo apresenta como padrão alguns loops vindos de fábrica.

20.17. Formas de Entrada

Formas de entrada de sinais ou de dados no aplicativo, podendo assumir os possíveis valores:

Entrada de microfone/Plugar instrumento, Entrada MIDI e Entrada OSC.

20.18. Formas de Importação

Formas de importação de áudio ou de projetos.

- **iTunes:** através da transferência do iTunes para o aplicativo;
- **Music Library:** através da importação de músicas da biblioteca musical do dispositivo;
- **Internet:** online a partir de um link avulso;
- **Dropbox:** online, através do Dropbox;
- **SoundCloud:** online, através do SoundCloud.

20.19. Produz Som

Indica se o aplicativo produz som ou necessita de produção sonora externa.

20.20. Download de *Presets* da Internet

Indica se é possível fazer download de *presets* ou projetos através da internet.

20.21. Customização

Apresenta as formas de customização da interface disponível para o aplicativo.

- **Gestos:** o aplicativo permite a escolha dos gestos que serão utilizados durante a interação musical;
- **Variáveis geométricas:** o aplicativo permite que variáveis geométricas advindas da posição dos dedos do usuário sejam usadas como formas de controle de parâmetros musicais;
- **Objetos:** o aplicativo permite a escolha de objetos que podem ser configurados de forma customizada durante a execução;
- **Tamanho dos Objetos:** o aplicativo permite que o objeto seja redimensionado;
- **Disposição dos Objetos:** o aplicativo permite que a posição do objeto seja alterada.

20.22. In App-Purchase

Indica quais as possibilidades de compra dentro do aplicativo, podendo assumir valores como **Loops, Presets** ou **Instrumentos**.

20.23. Colaboração

Indica as formas de colaboração que o aplicativo permite.

- **No mesmo dispositivo:** o aplicativo tem a interface preparada para a interação para dois ou mais indivíduos;
- **Em dispositivos diferentes:** o aplicativo provê formas de conexões entre dispositivos e permite que haja uma interação para dois ou mais indivíduos com a interface.

20.24. Modo de Edição/Modo de Execução

Indica se o aplicativo segue o paradigma de edição e execução, segundo o qual primeiro o usuário edita sua interface através da manipulação de objetos no modo de edição, depois ativa o modo de execução e interage com a interface com o intuito de produzir som.

21. Anexo III: Descrição da Caixa Morfológica de Interface

21.1. Adicionar Objetos

- **Gestual Forma Pré-definida:** o usuário faz um gesto de forma pré-definida na tela e o objeto associado ao gesto é adicionado na área desejada;
- **Arrastar de um menu:** o usuário arrasta o objeto, que está posicionado em um menu, e posiciona-o na área;
- **Selecionar em um menu e tocar a tela:** o usuário seleciona o objeto no menu, ficando este objeto em destaque e o após usuário tocar em algum lugar da tela, esse objeto é adicionado àquela posição.

21.2. Mover Objetos

- **Arrastar:** o usuário posiciona o dedo em cima do objeto e o arrasta pela tela.

21.3. Excluir Objetos

- **Apertar botão que sempre segue o objeto:** o usuário toca o botão de exclusão que está posicionado próximo ao objeto;
- **Arrastar para cima de uma área:** o usuário arrasta e posiciona o objeto em cima de uma área pré-determinada como área de exclusão;
- **Arrastar para fora da tela:** o usuário arrasta e solta o objeto para fora da tela. O aplicativo facilita esta ação adicionando movimento ao objeto;
- **Selecionar e apertar botão no menu:** o usuário seleciona o objeto, que fica destacado, e depois toca em um botão de exclusão posicionado em um menu;
- **Ativar modo de exclusão:** o usuário ativa o modo de exclusão, tocando em algum botão de ativação e a partir daí pode tocar nos objetos que deseja deletar.

21.4. Editar Objetos

- **Selecionar o objeto e tela sobreposta:** o usuário seleciona o objeto, que fica destacado, e uma tela sobreposta ao objeto permite editar suas configurações;
- **Selecionar e escolher opções no menu lateral:** o usuário seleciona o objeto, que fica destacado, e suas configurações podem ser editadas em um menu lateral.

21.5. Redimensionar Objetos

- **Pontos-guia:** o usuário move os pontos-guia posicionados em torno do objeto e assim modifica o seu tamanho.

21.6. Conectar Objetos

- **Aproximar um objeto do outro:** o usuário aproxima um objeto do outro e uma conexão automaticamente aparece;
- **Ativar modo de conexão:** o usuário toca em um botão que ativa um modo de conexão e, assim, o usuário pode arrastar o dedo de um objeto para outro;
- **Editar o objeto:** a conexão é feita no modo de edição do objeto, arrastando algum ponto-guia em direção a outro objeto.

21.7. Desconectar Objetos

- **Distanciar o objeto do outro:** o usuário distancia um objeto do outro e a conexão é automaticamente desfeita;
- **Arrastar uma das extremidades para um ponto vazio:** o usuário toca em cima de uma das extremidades da conexão e arrasta-a para um ponto vazio na tela;
- **Aplicar gesto na conexão:** um gesto é realizado em cima da conexão.

21.8. Ajuda

- **Balões informativos apontando para objetos na tela:** são apresentados alguns balões contendo informações sobre o funcionamento dos objetos e botões na tela;
- **Guia apresentado na primeira execução do aplicativo:** assim que o aplicativo é executado pela primeira vez, um guia em forma de vídeo ou sequência de imagens é apresentado ao usuário;
- **Camada de imagem posicionada por todo o aplicativo:** uma imagem com fundo transparente é posicionada por sobre toda a tela, apresentando informações sobre os elementos do aplicativo;
- **Roteiro interativo:** uma série de instruções são dadas ao usuário e assim que ele completa uma tarefa, novas informações são apresentadas;
- **Exemplos:** o aplicativo apresenta, como padrão, vários exemplos que podem ajudar o usuário a entender o aplicativo.

21.9. Ante-canvas

Entende-se por ante-canvas a tela que antecede a tela de edição.

- **Painel de Projetos Salvos:** os projetos salvos anteriormente pelo usuário são mostrados nesta tela;
- **Painel de Projetos da Comunidade:** os projetos salvos pela comunidade e submetidos a uma plataforma de distribuição são apresentados nesta tela;
- **Painel de Exemplos:** os projetos de exemplos, produzidos pelos próprios desenvolvedores do aplicativo, são apresentados nesta tela.

21.10. Modo de Execução

Modo de execução é o modo por meio do qual os objetos desempenham as ações com as quais estão associados. Abaixo, são descritas as formas como este modo pode ser ativado.

- **Ativado por botão:** o usuário toca em um botão e é levado ao modo de execução;
- **Sempre ligado:** todas as ações associadas a um determinado objeto podem ser disparadas a qualquer momento;
- **Misto:** o aplicativo possui um modo de execução que pode ser ativado pelo usuário, porém suas ações podem ser disparadas de forma restrita no modo de edição.

21.11. Detalhes de Interface

- **Permitir Zoom:** o aplicativo permite que o usuário distancie ou aproxime sua visão dos objetos na tela;
- **Organizar objetos em linhas de grade:** o aplicativo organiza os objetos de forma a estarem sempre alinhados a uma grade pré-determinada de linhas e colunas;
- **Possibilitar a escolha de cores:** o aplicativo permite que o usuário customize seus objetos e a interface como um todo, modificando as cores usadas.

21.12. Menus

- **Contexto com barra de rolagem:** menu flutuante (ou menu de contexto) com barra de rolagem;
- **Retrátil Temporizado:** menu retrátil que é mostrado pela ativação do usuário, porém se esconde depois de passado um certo período de tempo;
- **Retrátil por gesto:** menu retrátil que é ativado ou desativado a partir de um gesto realizado pelo usuário;
- **Retrátil por botão:** menu retrátil que ativado ou desativado tocando-se em um botão;
- **Retrátil com barra de rolagem:** menu retrátil que possui barra de rolagem.